



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 141501

***QUALITY MEASUREMENT OF OBJECT ORIENTED
CODE USING CHIDAMBER AND KEMERER METRIC
IN THE PERSPECTIVE OF MAINTAINABILITY,
EFFICIENCY, UNDERSTANDABILITY AND
REPLACEABILITY (CASE STUDIES SOFTWARE
ACCOUNTING XYZ)***

Aula Ayubi
NRP 5211 100 111

Supervisor
Feby Artwodini Muqtadiroh, S.Kom, M.T.
Amna Shifia Nisafani, S.Kom, M.Sc.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Information and Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

**PENGUKURAN KUALITAS KODE PROGRAM
BERORIENTASI OBJEK DENGAN MENGGUNAKAN
METRIK CHIDAMBER DAN KEMERER DILIHAT
DARI ASPEK MAINTAINABILITY, EFFICIENCY,
UNDERSTANDABILITY DAN REPLACEABILITY
(STUDI KASUS SOFTWARE ACCOUNTING XYZ)**

Aula Ayubi
NRP 5211 100 111

Dosen Pembimbing
Feby Artwodini Muqtadiroh, S.Kom, M.T.
Amna Shifia Nisafani, S.Kom, M.Sc.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015

LEMBAR PENGESAHAN
PENGUKURAN KUALITAS KODE PROGRAM
BERORIENTASI OBJEK DENGAN
MENGGUNAKAN METRIK CHIDAMBER DAN
KEMERER DILIHAT DARI ASPEK
MAINTAINABILITY, EFFICIENCY,
UNDERSTANDABILITY DAN REPLACEABILITY
(STUDI KASUS SOFTWARE ACCOUNTING XYZ)

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

AULA AYUBI
5211 100 111

Surabaya, Juli 2015

KETUA
JURUSAN SISTEM INFORMASI

Dr. Eng. Febrilivana Samopa S.Kom, M.Kom
NIP. 19730219 199802 1 001



LEMBAR PERSETUJUAN
PENGUKURAN KUALITAS KODE PROGRAM
BERORIENTASI OBJEK DENGAN
MENGGUNAKAN METRIK CHIDAMBER DAN
KEMERER DILIHAT DARI ASPEK
MAINTAINABILITY, EFFICIENCY,
UNDERSTANDABILITY DAN REPLACEABILITY
(STUDI KASUS SOFTWARE ACCOUNTING XYZ)

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

AULA AYUBI
5211 100 111

Disetujui Tim Penguji : Tanggal Ujian : 3 Juli 2015
Periode Wisuda : September 2015

Feby Artwodini Muqtadiroh, S.Kom, M.T. (Pembimbing 1)

Amna Shifia Nisafani, S.Kom, M.Sc.

(Pembimbing 2)

Tony Dwi Susanto, S.T, M.T, Ph.D

(Penguji 1)

Sholiq, S.T, M.Kom, M.SA

(Penguji 2)

**PENGUKURAN KUALITAS KODE PROGRAM
BERORIENTASI OBJEK DENGAN
MENGUNAKAN METRIK CHIDAMBER DAN
KEMERER DILIHAT DARI ASPEK
MAINTAINABILITY, EFFICIENCY,
UNDERSTANDABILITY DAN REPLACEABILITY
(STUDI KASUS SOFTWARE ACCOUNTING XYZ)**

Nama Mahasiswa : AULA AYUBI
NRP : 5211 100 111
Jurusan : Sistem Informasi FTIF-ITS
Dosen Pembimbing 1 : Feby Artwodini Muqtadiroh, S.Kom, M.T.
Dosen Pembimbing 2 : Amna Shifia Nisafani, S.Kom, M.Sc.

ABSTRAK

Peningkatan kualitas sebuah perangkat lunak mempunyai peran yang penting dalam praktek rekayasa perangkat lunak. Pendekatan engineering menginginkan bahwa kualitas perangkat lunak dapat diukur secara kuantitatif dalam bentuk angka yang dapat dipahami oleh orang lain. Object Oriented Programming (OOP) seringkali menjadi pilihan pengembang dalam membangun perangkat lunak.

Pengukuran kualitas software berdasarkan aspek object oriented dengan menggunakan Metrik Chidamber dan Kemerer. Metrik tersebut mempunyai enam metrik di dalamnya yaitu Weighted Method per Class (WMC), Depth of Inheritance Tree (DIT), Number OF Children (NOC), Response For a Class (RFC), Coupling Between Object Classes (CBO) dan Lack of Cohesion Method (LCOM).

Karakteristik kualitas yang dapat diukur menggunakan metrik ini berdasarkan ISO/IEC 9126-1 adalah efficiency,

understandability, reusability, dan maintainability/testability. Sehingga pengembang software dapat mengetahui sejauh mana kualitas konsep object oriented yang ada pada kode perangkat lunak tersebut.

Hasil akhir yang diharapkan dari tugas akhir ini adalah perbaikan kode program berupa class diagram yang lebih baik berdasarkan kualitas yang telah dipilih.

Kata kunci: Pengukuran Kualitas, Object Oriented Programming (OOP), Metrik Chidamber dan Kemerer, ISO/IEC 9126-1.

***QUALITY MEASUREMENT OF OBJECT ORIENTED
CODE USING CHIDAMBER AND KEMERER METRIC
IN THE PERSPECTIVE OF MAINTAINABILITY,
EFFICIENCY, UNDERSTANDABILITY AND
REPLACEABILITY (CASE STUDIES SOFTWARE
ACCOUNTING XYZ)***

Name : AULA AYUBI
NRP : 5211 100 111
Department : Information Systems FTIF -ITS
Supervisor 1 : Feby Artwodini Muqtadiroh, S.Kom, M.T.
Supervisor 2 : Amna Shifia Nisafani, S.Kom, M.Sc.

ABSTRACT

Improving the quality of a software has an important role in the practice of software engineering. Engineering approach wants that the quality software can be measured quantitatively in the form of numbers that can be understood by others. Object Oriented Programming (OOP) is often the choice of the developer in building software.

Quality measurement based on aspects of object-oriented software using Chidamber Metrics and Kemerer. Metrics Chidamber and Kemerer has six metric. The metric are weighted Method per Class (WMC), Depth of Inheritance Tree (DIT), Number OF Children (NOC), Response For a Class (RFC), Coupling Between Object Classes (CBO), and Lack of Cohesion Method (LCOM).

Quality characteristics that can be measured using this metric is based on ISO / IEC 9126-1 are efficiency, understandability, reusability, and maintainability / testability. So the software developers can determine the quality of object-oriented concepts that exist in the software code.

Final results are expected from this thesis are improvement of program code in the form of a better class diagram based on the quality that has been selected.

Keywords: Quality Measurement, Object Oriented Programming (OOP), Chidamber and Kemerer Metric, ISO / IEC 9126-1.

KATA PENGANTAR

Alhamdulillah waa syukurillah. Tiada dzat yang Maha Perkasa selain Allah Swt., yang selalu melimpahkan rahmat dan ridhoNya, sehingga penulis mampu menyelesaikan tugas akhir dengan judul:

**“PENGUKURAN KUALITAS KODE PROGRAM
BERORIENTASI OBJEK DENGAN
MENGUNAKAN METRIK CHIDAMBER DAN
KEMERER DILIHAT DARI ASPEK
MAINTAINABILITY, EFFICIENCY,
UNDERSTANDABILITY DAN REPLACEABILITY
(STUDI KASUS *SOFTWARE ACCOUNTING XYZ*)”**

sebagai salah syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini merupakan hasil perjuangan dan usaha berbagai pihak yang telah memberi bimbingan, bantuan, fasilitas dan motivasi selama proses pengerjaan tugas akhir, sehingga penulis dapat dengan lancar menyelesaikan penelitian dan menerima banyak ilmu dan pengalaman. Oleh karena itu, penulis ucapkan terimakasih yang mendalam kepada:

1. Bapak Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom, selaku Ketua Jurusan Sistem Informasi ITS
2. Ibu Feby Artwodini Muqtadiroh, S.Kom, M.T. dan Ibu Amna Shifia Nisafani, S.Kom, M.Sc., selaku dosen pembimbing yang telah memberikan bimbingan dan inspirasi kepada penulis.
3. Bapak Tony Dwi Susanto, S.T., M.T., Ph.D dan Bapak Radityo Prasetyanto W., S.Kom., M.Kom. selaku dosen penguji yang telah memberi saran kepada penulis demi kesempurnaan penelitian

4. Bapak Radityo Prasetyanto W., S.Kom., M.Kom. selaku dosen wali yang selalu memberi penulis motivasi dan inspirasi, serta bimbingan selama 4 tahun terakhir
5. Bapak & Ibu tercinta, Bapak Ahmad Muchibbin dan Ibu Azizah Ulwiyah, sumber kebahagiaan penulis. Tidak lupa, Kakak Nur Amalia Choironi, Kakak Aniq Atiqi Rohmawati, Kakak Achamd arif Alfin dan keluarga besar yang selalu memberikan doa, dukungan dan kekuatan
6. Sahabat terbaik, Sella, Hendra, Brantas yang selalu memberi keceriaan dan kebersamaan, serta motivasi untuk bersama-sama berada di panggung wisuda 112
7. Teman-teman seperjuangan, keluarga BASILISK dan anggota Laboratorium PPSI, yang selalu bersemangat hingga detik-detik terakhir di masa perkuliahan
8. Seluruh pihak yang secara langsung dan tidak langsung telah menyumbangkan pikiran dan tenaganya sehingga penulis dapat memperoleh kemudahan dalam penyusunan buku tugas akhir ini.

Penelitian ini diharapkan dapat memberi manfaat bagi pembaca dan menambah referensi penelitian selanjutnya khususnya terkait dengan pengukuran kualitas perangkat lunak.

Penelitian ini jauh dari sempurna, Penulis berharap adanya kritik dan saran dari pembaca, sehingga kualitas dari penelitian terkait topik ini dapat ditingkatkan dan dapat lebih berkontribusi terhadap dunia penelitian perangkat lunak.

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xix
BAB I PENDAHULUAN	1
1.1.Latar Belakang	1
1.2.Perumusan Permasalahan.....	4
1.3.Batasan Masalah.....	4
1.4.Tujuan Penulisan.....	4
1.5.Manfaat Penulisan.....	5
1.6.Relevansi	5
BAB II TINJAUAN PUSTAKA	7
2.1.Studi Sebelumnya.....	7
2.2.Dasar Teori.....	15
2.2.1. Object Oriented Programming.....	15
2.2.2. Metrik Chidamber dan Kemerer	21
2.2.3. ISO/IEC 9126.....	28
2.2.4. <i>Design Patterns</i>	36
2.2.5. <i>Class Diagram</i>	42
BAB III METODOLOGI PENULISAN	47
3.1.Tahap Perancangan	49
3.1.1. Analisa Kode Program.....	49
3.1.2. Pembuatan <i>Class Diagram</i>	49
3.2.Tahap Implementasi	50
3.3.Tahap Pembahasan Hasil	50
3.3.1. Analisa Perhitungan Metrik Chidamber dan Kemerer	50
3.3.2. Membuat <i>Class Diagram</i> Baru	51

3.3.3.	Perhitungan Nilai Metrik Chidamber dan Kemerer	51
3.3.4.	Pembuatan Kesimpulan dan Saran.....	51
BAB IV	PERANCANGAN KONSEPTUAL.....	53
4.1.	Gambaran Umum <i>Software Accounting XYZ</i>	53
4.2.	<i>Code Dependencies</i>	58
4.2.1.	<i>Extends Dependencies</i>	59
4.2.2.	<i>Contains Dependency</i>	59
4.2.3.	<i>Returns Dependency</i>	59
4.2.4.	<i>Has Param Dependency</i>	60
4.2.5.	<i>Throws Dependency</i>	61
4.2.6.	<i>Calls Dependency</i>	61
4.2.7.	<i>Accesses Dependency</i>	62
4.2.8.	<i>Is of Type Dependency</i>	62
4.2.9.	<i>References Dependency</i>	63
4.3.	<i>Package dataLayer</i>	63
4.3.1.	<i>Code Dependency Antar Class dataLayer</i>	64
4.3.2.	<i>Code Dependency Package dataLayer ke Package appLayer</i>	68
4.3.3.	<i>Code Dependency Package dataLayer ke Package appLayer.transactionRelated</i>	71
4.3.4.	<i>Code Dependency Package dataLayer ke Package GUILayer</i>	73
4.4.	<i>Package appLayer</i>	74
4.4.1.	<i>Code Dependency Antar Class appLayer</i>	74
4.4.2.	<i>Code Dependency Package appLayer ke Package dataLayer</i>	99
4.4.3.	<i>Code Dependency Package appLayer ke Package appLayer.taxRelated</i>	103
4.4.4.	<i>Code Dependency Package appLayer ke Package appLayer.transactionRelated</i>	107
4.4.5.	<i>Code Dependency Package appLayer ke Package GUILayer</i>	110
4.5.	<i>Package appLayer.taxRelated</i>	112

4.5.1.	<i>Code Dependency Antar Class</i>	
	appLayer.taxRelated	112
4.5.2.	<i>Code Dependency Package appLayer.taxRelated ke Package dataLayer</i>	115
4.5.3.	<i>Code Dependency Package appLayer.taxRelated ke Package appLayer</i>	116
4.6.	<i>Package appLayer.transactionRelated</i>	118
4.6.1	<i>Code Dependency Antar Class</i>	
	appLayer.transactionRelated	118
4.6.2	<i>Code Dependency Package</i>	
	appLayer.transactionRelated ke <i>Package dataLayer</i>	120
4.6.3	<i>Code Dependency Package</i>	
	appLayer.transactionRelated ke <i>Package appLayer</i>	122
4.6.4	<i>Code Dependency Package</i>	
	appLayer.transactionRelated ke <i>Package</i>	
	appLayer.taxRelated	133
4.6.5	<i>Code Dependency Package</i>	
	appLayer.transactionRelated ke <i>Package GUILayer</i>	134
4.7.	<i>Package GUILayer</i>	135
4.7.1	<i>Code Dependency Antar Class GUILayer</i>	135
4.7.2	<i>Code Dependency Package GUILayer ke Package dataLayer</i>	136
4.7.3	<i>Code Dependency Package GUILayer ke Package appLayer</i>	140
4.7.4	<i>Code Dependency Package GUILayer ke Package appLayer.taxRelated</i>	199
4.7.5	<i>Code Dependency Package GUILayer ke Package appLayer.transactionRelated</i>	206
4.8.	<i>Pembuatan Class Diagram</i>	215
BAB V IMPLEMENTASI		287
5.1.	<i>Weighted Method per Class (WMC)</i>	287
5.2.	<i>Depth of Inheritance Tree (DIT)</i>	296
5.3.	<i>Number OF Children (NOC)</i>	305
5.4.	<i>Coupling Between Object Classes (CBO)</i>	312
5.5.	<i>Response For a Class (RFC)</i>	320

5.6.Lack of Cohesion Method (LCOM)	328
BAB VI HASIL DAN PEMBAHASAN	337
6.1. Analisa Hasil Perhitungan Metrik Chidamber dan Kemerer	337
6.2.Membuat <i>Class Diagram</i> Baru	340
6.3.Perhitungan Metrik Chidamber dan Kemerer pada <i>Class Diagram</i> Baru	343
BAB VII PENUTUP	345
7.1. Kesimpulan	345
2.2. Saran	346
DAFTAR PUSTAKA	347
BIODATA PENULIS	351
LAMPIRAN A – <i>Method</i> dan Atribut	A-1
LAMPIRAN B – <i>Generaization (Inheritance) Relationship</i>	B-1

DAFTAR TABEL

Tabel 2. 1 Studi atau Penulisan Sebelumnya.....	10
Tabel 2. 2 Pemetaan Metrik Chidamber dan Kemerer dengan Kode Program.....	22
Tabel 2. 3 Pemetaan Metrik Chidamber dan Kemerer dengan <i>Object Oriented Design Element</i>	22
Tabel 2. 4 Nilai Metrik Chidamber dan Kemerer.....	23
Tabel 2. 5 Batasan Metrik Chidamber dan Kemerer.....	23
Tabel 2. 6 Pembagian ISO/IEC 9126.....	29
Tabel 2. 7 Deskripsi Kriteria dan Subkriteria ISO/IEC 9126-1...30	
Tabel 2. 8 Pemetaan Quality Model ISO/IEC 9126-1 dengan Metrik Chidamber dan Kemerer.....	34
Tabel 2. 9 Kriteria dan Subkriteria ISO/IEC 9126-1 yang akan diukur.....	34
Tabel 2. 10 Deskrip Kriteria dan Subkriteria ISO/IEC 9126-1 ...35	
Tabel 2. 11 Korelasi Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-1.....	36
Tabel 2. 12 Pemetaan <i>Design Pattern</i>	40
Tabel 2. 13 <i>Design Pattern</i> yang Sering Digunakan.....	41
Tabel 2. 14 <i>Code dependency</i>	42
Tabel 2. 15 Multiplisitas.....	45
Tabel 2. 16 Pemetaan <i>Dependency Kind</i> dan <i>Relationship Class Diagram</i>	45
Tabel 3. 1 Metodologi Pengerjaan Tugas akhir.....	48
Tabel 4. 1 <i>Code Dependency</i> Antar <i>Class</i> dataLayer.....	64
Tabel 4. 2 <i>Code Dependency Package</i> dataLayer ke <i>Package</i> applayer.....	68
Tabel 4. 3 <i>Code Dependency Package</i> dataLayer ke <i>Package</i> appLayer.transactionRelated.....	71
Tabel 4. 4 <i>Code Dependency Package</i> dataLayer ke <i>Package</i> GUILayer.....	73
Tabel 4. 5 <i>Code Dependency</i> Antar <i>Class</i> appLayer.....	74
Tabel 4. 6 <i>Code Dependency Package</i> appLayer ke <i>Package</i> dataLayer.....	99

Tabel 4. 7 <i>Code Dependency Package</i> appLayer ke <i>Package</i> appLayer.taxRelated	103
Tabel 4. 8 <i>Code Dependency Package</i> appLayer ke <i>Package</i> appLayer.transactionRelated.....	107
Tabel 4. 9 <i>Code Dependency Package</i> appLayer ke <i>Package</i> GUILayer	110
Tabel 4. 10 <i>code dependency</i> antar <i>class</i> appLayer.taxRelated	112
Tabel 4. 11 <i>Code Dependency Package</i> appLayer.taxRelated ke <i>Package</i> dataLayer.....	115
Tabel 4. 12 <i>Code Dependency Package</i> appLayer.taxRelated ke <i>Package</i> appLayer.....	116
Tabel 4. 13 <i>Code Dependency</i> antar <i>class</i> appLayer. transaction Related	118
Tabel 4. 14 <i>Code Dependency Package</i> appLayer. transaction Related ke <i>Package</i> dataLayer	120
Tabel 4. 15 <i>Code Dependency Package</i> appLayer. transaction Related ke <i>Package</i> appLayer.....	122
Tabel 4. 16 <i>Code Dependency Package</i> appLayer. transaction Related ke <i>Package</i> appLayer.taxRelated.....	133
Tabel 4. 17 <i>Code Dependency Package</i> appLayer.transaction Related ke <i>Package</i> GUILayer	134
Tabel 4. 18 <i>Code Dependency Antar Class</i> GUILayer	135
Tabel 4. 19 <i>Code Dependency Package</i> GUILayer ke <i>Package</i> dataLayer.....	136
Tabel 4. 20 <i>Code Dependency Package</i> GUILayer ke <i>Package</i> appLayer	141
Tabel 4. 21 <i>Code Dependency Package</i> GUILayer ke <i>Package</i> appLayer.taxRelated	199
Tabel 4. 22 <i>Code Dependency Package</i> GUILayer ke <i>Package</i> appLayer.transactionRelated.....	206
Tabel 4. 23 Pemetaan <i>Code Dependency</i> ke <i>Relationship Class Diagram</i>	216
Tabel 5. 1 Nilai Metrik WMC <i>Package</i> dataLayer	289
Tabel 5. 2 Nilai Metrik WMC <i>Package</i> appLayer.....	289
Tabel 5. 3 Nilai Metrik WMC <i>Package</i> appLayer.taxRelated..	291

Tabel 5. 4 Nilai Metrik WMC <i>Package</i> appLayer.transaction Related.....	292
Tabel 5. 5 Nilai Metrik WMC <i>Package</i> GUILayer	292
Tabel 5. 6 Nilai <i>Average</i> Metrik WMC	295
Tabel 5. 7 Nilai Metrik DIT <i>Package</i> dataLayer	298
Tabel 5. 8 Nilai Metrik DIT <i>Package</i> appLayer.....	299
Tabel 5. 9 Nilai Metrik DIT <i>Package</i> appLayer.....	300
Tabel 5. 10 Nilai Metrik DIT <i>Package</i> appLayer.....	301
Tabel 5. 11 Nilai Metrik DIT <i>Package</i> GUILayer	302
Tabel 5. 12 Nilai Metrik NOC <i>Package</i> dataLayer	306
Tabel 5. 13 Nilai Metrik NOC <i>Package</i> appLayer	306
Tabel 5. 14 Nilai Metrik NOC <i>Package</i> appLayer.taxRelated..	308
Tabel 5. 15 Nilai Metrik NOC <i>Package</i> appLayer . transaction Related.....	309
Tabel 5. 16 Nilai Metrik NOC <i>Package</i> GUILayer.....	309
Tabel 5. 17 Nilai <i>Average</i> Metrik NOC	312
Tabel 5. 18 Nilai Metrik CBO <i>Package</i> dataLayer	313
Tabel 5. 19 Nilai Metrik CBO <i>Package</i> appLayer	314
Tabel 5. 20 Nilai Metrik CBO <i>Package</i> appLayer .taxRelated.	316
Tabel 5. 21 Nilai Metrik CBO <i>Package</i> appLayer . transaction Related.....	316
Tabel 5. 22 Nilai Metrik CBO <i>Package</i> GUILayer	317
Tabel 5. 23 Nilai <i>Average</i> Metrik CBO.....	320
Tabel 5. 24 Nilai Metrik RFC <i>Package</i> dataLayer	321
Tabel 5. 25 Nilai Metrik RFC <i>Package</i> appLayer.....	322
Tabel 5. 26 Nilai Metrik RFC <i>Package</i> appLayer .taxRelated..	324
Tabel 5. 27 Nilai Metrik RFC <i>Package</i> appLayer .transaction Related.....	324
Tabel 5. 28 Nilai Metrik RFC <i>Package</i> GUILayer	325
Tabel 5. 29 Nilai <i>Average</i> Metrik RFC	327
Tabel 5. 30 Nilai Metrik LCOM <i>Package</i> dataLayer	329
Tabel 5. 31 Nilai Metrik LCOM <i>Package</i> appLayer.....	330
Tabel 5. 32 Nilai Metrik LCOM <i>Package</i> appLayer .taxRelated	331
Tabel 5. 33 Nilai Metrik LCOM <i>Package</i> appLayer .transaction Related.....	332

Tabel 5. 34 Nilai Metrik LCOM <i>Package</i> GUILayer	333
Tabel 5. 35 Nilai <i>Average</i> Metrik LCOM.....	335
Tabel 6. 1 <i>Class</i> yang Memerlukan Perbaikan.....	339
Tabel 6. 2 <i>Refactoring</i> yang Digunakan	340
Tabel 6. 3 Pengukuran Matrik Chidamber dan Kemerer terhadap <i>Class</i> yang Diperbaiki.....	343
Tabel 6. 4 Nilai Metrik Chidamber dan Kemerer Baru	344

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi DIT.....	26
Gambar 3. 1 Metodologi Pengerjaan Tugas akhir.....	47
Gambar 4. 1 Kode Program <i>Extends Dependencies</i> (Class cancelation).....	59
Gambar 4. 2 Kode Program <i>Contains Dependencies</i> (Class money Transfer).....	59
Gambar 4. 3 Kode Program <i>Retruns Dependency</i> (Class money Transfer).....	60
Gambar 4. 4 Kode Program <i>Has Param Dependency</i> (Class item TableList).....	60
Gambar 4. 5 Kode Program <i>Has Param Dependency</i> (Class prepareDB).....	61
Gambar 4. 6 Kode Program <i>Calls Dependency</i> (Class page Activator).....	62
Gambar 4. 7 Kode Program <i>Access Dependency</i> (Class money Transfer).....	62
Gambar 4. 8 Kode Program <i>Access Dependency</i> (Class money Transfer).....	63
Gambar 4. 9 Kode Program <i>References Dependency</i> (Class fileImporters).....	63
Gambar 4. 10 <i>Class Diagram Package</i> dataLayer.....	277
Gambar 4. 11 <i>Class Diagram Package</i> appLayer part 1.....	278
Gambar 4. 12 <i>Class Diagram Package</i> appLayer part 2.....	279
Gambar 4. 13 <i>Dependencies Relationship</i> pada package appLayer.....	280
Gambar 4. 14 <i>Class Diagram Package</i> appLayer.taxRelated.....	281
Gambar 4. 15 <i>Class Diagram Package</i> appLayer . transaction Related.....	282
Gambar 4. 16 <i>Class Diagram Package</i> GUILayer part 1.....	283
Gambar 4. 17 <i>Class Diagram Package</i> GUILayer part 2.....	284
Gambar 4. 18 <i>Class Diagram Package</i> GUILayer part 3.....	285
Gambar 5. 1 <i>Source Code</i> Contoh Metrik WMC (Class contentBookingProvider).....	288

Gambar 5. 2 <i>Source Code</i> Metrik WMC (<i>Class</i> HBCI callback Dialog)	288
Gambar 5. 3 <i>Source Code</i> Contoh Metrik DIT (<i>Class</i> items)...	296
Gambar 5. 4 <i>Class Diagram</i> contoh Metrik DIT (<i>Class</i> items)	296
Gambar 5. 5 <i>Class Diagram</i> contoh Metrik DIT	297
Gambar 5. 6 Contoh Perhitungan <i>Average</i> Metrik DIT	298
Gambar 5. 7 <i>Average</i> Metrik DIT	304
Gambar 5. 8 <i>Class Diagram</i> contoh Metrik NOC (<i>Class</i> transaction).....	305
Gambar 5. 10 Rumus LCOM.....	328
Gambar 5. 11 <i>Source Code</i> contoh Metrik LCOM (<i>Class</i> booking CellModifier)	329
Gambar 6. 1 <i>Class Diagram</i> baru <i>Package</i> dataLayer.....	336
Gambar 6. 2 <i>Class Diagram</i> baru <i>Package</i> appLayer part 1	337
Gambar 6. 3 <i>Class Diagram</i> baru <i>Package</i> appLayer part 2....	338
Gambar 6. 4 <i>Class Diagram</i> baru <i>Package</i> appLayer.taxRelated	339
Gambar 6. 5 <i>Class Diagram</i> baru <i>Package</i> appLayer. Transaction Related	340
Gambar 6. 6 <i>Class Diagram</i> baru <i>Package</i> GUILayer part 1..	341
Gambar 6. 7 <i>Class Diagram</i> baru <i>Package</i> GUILayer part 2...	342
Gambar B. 1 <i>Method</i> dan Atribut <i>Package</i> dataLayer part 1	1
Gambar B. 2 <i>Method</i> dan Atribut <i>Package</i> dataLayer part 2	2
Gambar B. 3 <i>Method</i> dan Atribut <i>Package</i> appLayer part 1	3
Gambar B. 4 <i>Method</i> dan Atribut <i>Package</i> appLayer part 2	4
Gambar B. 5 <i>Method</i> dan Atribut <i>Package</i> appLayer part 3	5
Gambar B. 6 <i>Method</i> dan Atribut <i>Package</i> appLayer part 4	6
Gambar B. 7 <i>Method</i> dan Atribut <i>Package</i> appLayer part 5	7
Gambar B. 8 <i>Method</i> dan Atribut <i>Package</i> appLayer part 6	8
Gambar B. 9 <i>Method</i> dan Atribut <i>Package</i> appLayer.taxRelated.	9
Gambar B. 10 <i>Method</i> dan Atribut <i>Package</i> appLayer . transactionRelated part 1	10
Gambar B. 11 <i>Method</i> dan Atribut <i>Package</i> appLayer . transactionRelated part 2	11
Gambar B. 12 <i>Method</i> dan Atribut <i>Package</i> appLayer . transactionRelated part 3	12

Gambar B. 13	<i>Method dan Atribut Package GUILayer part 1</i>	...13
Gambar B. 14	<i>Method dan Atribut Package GUILayer part 2</i>	...14
Gambar B. 15	<i>Method dan Atribut Package GUILayer part 3</i>	...15
Gambar B. 16	<i>Method dan Atribut Package GUILayer part 4</i>	...16
Gambar B. 17	<i>Method dan Atribut Package GUILayer part 5</i>	...17
Gambar B. 18	<i>Method dan Atribut Package GUILayer part 6</i>	...18
Gambar B. 19	<i>Method dan Atribut Package GUILayer part 7</i>	...19
Gambar B. 20	<i>Method dan Atribut Package GUILayer part 8</i>	...20
Gambar B. 21	<i>Method dan Atribut Package GUILayer part 9</i>	...21
Gambar C. 1	<i>Inheritance Hierarchy Package dataLayer</i>1
Gambar C. 2	<i>Inheritance Hierarchy Package appLayer . Tax Related</i>2
Gambar C. 3	<i>Inheritance Hierarchy Package appLayer . transaction Related</i>3
Gambar C. 4	<i>Inheritance hierarchy package appLayer</i>4
Gambar C. 5	<i>Inheritance hierarchy package GUILayer</i>5

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

Bab Pendahuluan akan dijelaskan mengenai latar belakang masalah yang menyebabkan studi kasus ini diangkat menjadi penelitian, perumusan masalah, batasan masalah, tujuan, manfaat yang dapat diambil dari hasil penelitian, serta relevansi hasil penelitian dalam tugas akhir terhadap bidang keilmuan jurusan sistem informasi dan laboratorium.

1.1. Latar Belakang

Saat Salah satu faktor penting dari suatu perangkat lunak adalah kualitas [1]. Kualitas perangkat lunak akan mempengaruhi baik buruknya suatu kinerja dari perusahaan yang bersangkutan dalam menjalankan bisnisnya. Perangkat lunak yang berkualitas akan memudahkan perusahaan dalam menjalankan proses bisnisnya sehingga sumber daya yang dikeluarkan oleh perusahaan akan semakin efisien dan efektif.

Peningkatan kualitas sebuah perangkat lunak mempunyai peran yang penting dalam praktek rekayasa perangkat lunak [2]. Oleh karena itu pengukuran kualitas perangkat lunak sangat dibutuhkan agar Perangkat lunak dapat diperbaiki secara terus menerus. Pengukuran kualitas akan memberikan pandangan bagi organisasi atau perusahaan dalam mengalokasikan sumber dayanya sehingga biaya yang dikeluarkan sesuai dengan perencanaan.

Beberapa metode untuk melakukan pengukuran kualitas perangkat lunak adalah inspeksi kode, penelusuran desain, simulasi prototipe, dan pengukuran berbasis analisis [3]. ISO/IEC juga telah membuat standar kualitas perangkat lunak dengan mengkombinasikan model dan menghubungkan karakteristik dan sub karakteristik kualitas model. Beberapa penelitian juga menghasilkan metrik perangkat lunak sebagai alat untuk

mengukur kode program, arsitektur dan kinerja dari perangkat lunak. Sampai saat ini proses penilaian kualitas perangkat lunak tetap menjadi topik pembahasan yang terus dikembangkan [4].

Salah satu pengukuran kualitas perangkat lunak adalah pengukuran kualitas kode program pada perangkat lunak tersebut [5]. Pendekatan *engineering* menginginkan kualitas perangkat lunak dapat diukur secara kuantitatif dalam bentuk angka yang dapat dipahami oleh orang lain. Faktor efisiensi pada kode program akan mempengaruhi manajemen perusahaan terkait perangkat lunak tersebut. Banyak metode pengukuran kualitas kode program seperti metode LOC, *Function Point*, dan Cocomo [6]. Pengembang perangkat lunak saat ini lebih cenderung menggunakan konsep *Object Oriented Programming* (OOP) yang memudahkan untuk membangun perangkat lunak. Beragamnya pola implementasi berorientasi *object* menimbulkan perbedaan pendapat dalam melihat kualitas sebuah perangkat lunak [7].

Pengukuran kualitas pada desain berorientasi *object* (OOP) akan memudahkan perusahaan menilai tingkat kualitas perangkat lunak untuk dipelihara (maintenance) oleh perusahaan tersebut [8]. Beberapa riset terdahulu telah menghasilkan alat ukur kualitas perangkat lunak yang diwujudkan dalam beberapa parameter berupa *Metrics for Object Oriented Software Engineering* [9]. Parameter tersebut mempunyai nama lain *Object Oriented (OO) Metrics*. *OO metrics* mempunyai banyak jenis [10]. Tujuan dari metrik yang berbasis *object oriented* adalah agar dapat memahami perangkat lunak sehingga dapat memperkirakan keefektifan proses dan meningkatkan kualitas kerja sesuai dengan pemahaman perangkat lunak yang didapat sebelumnya.

Metrik Chidamber dan Kemerer telah dipilih dalam pengerjaan tugas akhir ini karena pada penulisan sebelumnya telah menghasilkan pemetaan antara Metrik Chidamber dan Kemerer dengan properti kualitas [11] [12] [13]. Metrik Chidamber dan

Kemerer merupakan salah satu kumpulan metrik yang digunakan untuk mengukur kualitas perangkat lunak melalui kode program pada perangkat lunak tersebut. Metrik Chidamber dan Kemerer mengukur kualitas perangkat lunak berdasarkan enam metrik dengan melihat pada perspektif desain berorientasi *object* [14].

Perangkat lunak yang akan dijadikan studi kasus dalam tugas akhir ini adalah *Software Accounting* yang digunakan untuk mengelola dan menyajikan laporan dari berbagai aktivitas bisnis yang meliputi penjualan, pembelian stok, keuangan, proyek dan aset. *Software accounting XYZ* adalah produk dari sebuah *software house* yang akan terus menerus diperbaiki untuk memenuhi kebutuhan pelanggannya. Terdapat beberapa modul dari *software* ini yang saling terintegrasi, *software* ini memiliki kode program yang mempunyai konsep desain berorientasi *object* (OOP) dengan Bahasa Pemrograman Java.

Pengukuran perangkat lunak dengan menggunakan studi kasus *Software Accounting XYZ* akan menghasilkan nilai kuantitatif yang akan merepresentasikan tingkat *efficiency*, *understandability*, *reusability/replaceability*, dan *maintainability/testability software accounting XYZ*. Sehingga tugas akhir ini akan dapat menghasilkan nilai kuantitatif yang mempresentasikan kualitas *Software Accounting XYZ*. Tugas akhir ini juga menghasilkan rekomendasi perbaikan struktur kode program berupa *class diagram* berdasarkan *design pattern*. Hasil tugas akhir ini diharapkan dapat menjadi referensi bagi pengembang *Software Accounting XYZ* untuk memperbaiki desain kode program perangkat lunak tersebut.

1.2. Perumusan Permasalahan

Rumusan permasalahan dalam penulisan ini adalah:

1. Bagaimana bentuk struktur kode program pada *Software Accounting XYZ*
2. Bagaimana hasil pengukuran kualitas *Software Accounting XYZ* berdasarkan kriteria kualitas *efficiency* dan *maintainability* serta sub kriteritia *understandability* dan *reusability/replaceability* dengan menggunakan Metrik Chidamber dan Kemerer
3. Bagaimana bentuk rekomendasi perbaikan yang diberikan kepada pengembang *Software Accounting XYZ*.

1.3. Batasan Masalah

Batasan masalah dalam penulisan ini adalah:

1. Kriteria dan subkriteria yang diambil dari ISO/IEC 9126-1 adalah *Efficiency, Understandability, Reusability/Replaceability dan Maintainability* (Tidak semua kriteria dan subkriteria pada ISO/IEC 9126-1 digunakan)
2. Rekomendasi perbaikan berupa *class diagram* baru yang dibuat berdasarkan *Design Pattern*
3. Analisa kode program *Software Accounting XYZ* menggunakan *tool* Eclipse IDE for Java Developers Luna Service Release 2 (4.4.2)
4. Pembuatan *class diagram* menggunakan *tool* Enterprise Architect 7.5.848.

1.4. Tujuan Penulisan

Tujuan dari penulisan ini adalah:

1. Menemukan bentuk struktur kode program pada *Software Accounting XYZ*
2. Mengetahui nilai Metrik Chidamber dan Kemerer yang merepresentasikan kualitas *Software Accounting XYZ* berdasarkan kriteria kualitas *efficiency* dan *maintainability*

serta sub kriteria *understandability* dan *reusability/replaceability*

3. Membuat rekomendasi berupa *class diagram* yang lebih baik berdasarkan *Design Pattern*.

1.5. Manfaat Penulisan

Manfaat yang didapat dari penulisan ini adalah:

Bagi penulis

1. Mengaplikasikan ilmu yang diperoleh selama kuliah
2. Membandingkan teori dan hasil penulisan sebelumnya dengan keadaan sebenarnya.
3. Tugas akhir ini dapat menjadi referensi dalam pembelajaran untuk mengukur kualitas desain perangkat lunak yang lainnya, baik itu aplikasi desktop maupun aplikasi web.

Bagi Perusahaan

1. Perusahaan khususnya pengembang dapat mengetahui kualitas *Software Accounting* secara kuantitatif
2. Melalui tugas akhir ini perusahaan khususnya pengembang dapat memperbaiki kode program *Software Accounting* ini sesuai rekomendasi penulis yang didapat berdasarkan *Design Pattern*.

1.6. Relevansi

Tugas akhir ini disusun untuk memenuhi syarat kelulusan Sarjana. Penulisan mengangkat topik Manajemen Kualitas karena masih sedikit topik terkait pengukuran kualitas yang dapat menghasilkan angka kuantitatif, sedangkan dalam penulisan diluar jurusan sistem informasi ITS topik ini terus menerus mengalami perkembangan untuk membantu pengembang atau

perusahaan untuk mengembangkan perangkat lunaknya. Topik ini sangat relevan dengan peta penulisan Laboratorium Perencanaan dan Pengembangan Sistem Informasi (PPSI).

Keterkaitan penulisan ini dengan perkuliahan yang telah dipelajari oleh penulis adalah Manajemen Kualitas Teknologi Informasi (MKTI). Penulisan ini termasuk dalam topik *Quality Management* pada peta jalan penulisan salah satu laboratorium yang ada di Jurusan Sistem Informasi yaitu Perencanaan dan Pengembangan Sistem Informasi (PPSI), dengan sub topik pengukuran kualitas perangkat lunak. Pengukuran kualitas yang dilakukan pada penulisan ini menggunakan metrik sehingga akan menghasilkan pengukuran kualitas yang dapat diukur secara kuantitatif, metrik yang digunakan pada penulisan ini adalah Metrik Chidamber dan Kemerer yang sesuai dengan studi kasus perangkat lunak yang menggunakan bahasa pemrograman java berbasis *object oriented programming*(OOP). Adapun penulisan diakhiri dengan pemberian rekomendasi *class diagram* berdasarkan *Design Pattern* akan dapat menjadi masukan bagi pengembang untuk melakukan perbaikan pada *software* tersebut.

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan pustaka atau literatur yang digunakan selama penulisan ini.

2.1. Studi Sebelumnya

Pada bagian ini memaparkan acuan yang digunakan oleh penulis dalam melakukan penulisannya, acuan yang berupa teori maupun penulisan yang sejenis dengan penulisan yang dilakukan.

Dalam menyusun penulisan ini, penulis mempunyai beberapa referensi dari penulisan sebelumnya. Terdapat delapan penulisan sebelumnya yang diambil oleh penulis sebagai referensi penulisan ini.

Penulisan pertama yang menjadi referensi adalah penulisan yang dilakukan oleh Yeresime Suresh, Jayadeep Pati, dan Santanu Ku Rath dari Departemen *Computer Science and Engineering, National Institute of Technology, Rourkela, India*. Penulisan ini mencoba melakukan validasi terhadap Metrik Tradisional, Metrik Chidamber dan Kemerer dan Metrik R.C. Martins. Penulisan mengilustrasikan banyaknya manfaat penggunaan metrik perangkat lunak untuk mengukur kualitas perangkat lunak dan bagaimana melakukan evaluasi secara efektif dengan metrik tersebut.

Penulisan kedua yang menjadi referensi adalah penulisan yang dilakukan oleh Arti Chhikara and R.S.Chhillar dari Universitas Maharaja Agrasen, Delhi, India, Departemen *Computer Science And Applications, Rohtak, India*. Penulisan ini mencoba membuat rangking dari 20 program java berdasarkan hasil perhitungan 11 metrik perangkat lunak,

pemeringkatan disusun berdasarkan tingkat kompleksitas program.

Penulisan ketiga yang menjadi referensi adalah penulisan yang dilakukan oleh M. Rizwan Jameel Qureshi dari Fakultas *Computing & Information Technology* dari Universitas King Abdul Aziz, Jeddah, Saudi Arabia. Penulisan ini berusaha mengevaluasi Metrik Chidamber dan Kemerer agar dapat mengurangi kecacatan perangkat lunak pada tahap desain perangkat lunak. Penulisan ini juga menjabarkan pengaruh setiap Metrik Chidamber dan Kemerer pada jumlah kecacatan pada perangkat lunak.

Penulisan keempat yang menjadi referensi adalah penulisan yang dilakukan oleh Wahyu Rifa'i Dwi septian dari Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif, Hidayatullah Jakarta, Indonesia. Penulisan ini bertujuan untuk melakukan pemeringkatan *Framework* Java dengan menggunakan metode baru dalam pengukuran kualitas perangkat lunak.

Penulisan kelima yang menjadi referensi adalah penulisan yang dilakukan oleh Roni Yunis dan Arwin Halim dari STMIK Mikroskil, Indonesia. Penulisan ini menguji pengaruh antara salah satu metrik kualitas internal yaitu kohesi dengan kecenderungan kesalahan pada perangkat lunak berorientasi *object*. Pengukuran metrik kohesi dalam penulisan ini dilakukan pada tahap desain dan implementasi.

Penulisan keenam yang menjadi referensi adalah penulisan yang dilakukan oleh Efano Hermawan dan Petrus Mursanto dari Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Indonesia. Penulisan ini mengusulkan pengukuran kualitas desain berdasarkan hasil implementasinya dalam *java source code*. Metode yang diusulkan merupakan gabungan dari *Metrics for Object Oriented Software Engineering* (MOOSE), properti kualitas desain perangkat lunak dan konsep *Analytic Hierarchy Process* (AHP).

Penulisan ketujuh yang menjadi referensi adalah penulisan yang dilakukan oleh Achmad Arwan dan Siti Rochimah dari Teknik Informatika, Universitas Brawijaya. Penulisan ini bertujuan untuk mengetahui adanya hubungan antara Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-2.

Penulisan kedelapan yang menjadi referensi adalah penulisan yang dilakukan oleh Wayan Gede Suka Parwita dan luh Arida Ayu Rahning Putri dari Fakultas MIPA, Universitas Gadjah Mada, Yogyakarta, Indonesia. Penulisan ini menjabarkan berbagai jenis *quality model* dari perangkat lunak yang ada beserta komponen-komponen yang digunakan dalam melakukan penilaian dari masing-masing model.

Pada Tabel 2. 1 terdapat penjabaran penelitian sebelumnya yang menjadi referensi penulis dalam pengerjaan tugas akhir ini.

Tabel 2. 1 Studi atau Penulisan Sebelumnya

Nama Pengarang	Nama Jurnal Nomor Publikasi	Tahun	Judul Penulisan	Hasil/Kesimpulan
Yeresime Suresh Jayadeep Pati Santanu Ku Rath	SciVerse ScienceDirect. Procedia Technologia 6(2012) 420-427	2012	<i>Effectiveness of software metrics for object oriented system</i> (Keefektivan metrik perangkat lunak untuk sistem berorientasi <i>object</i>).	Pengujian validitas pada penulisan ini dengan melakukan evaluasi pada metrik perangkat lunak membuktikan banyak metrik tradisional dan berorientasi <i>object</i> yang ada termasuk Metrik Chidamber dan Kemerer dapat memberikan informasi berharga terkait penilaian kualitas perangkat lunak kepada para pengembang, coders dan penguji.
Arti Chhikara R.S.Chhillar	IJCSI International Journal of Computer Science. Issues, Vol. 9, Issue 1, No 3. ISSN (Online):1694-	2012	<i>Analyzing The Complexity of Java Programs Using Object Oriented Software Metrics</i> (Analisa kompleksitas dari Program Java menggunakan Metrik	Penulisan ini telah mempelajari lebih dalam penerapan metrik berorientasi <i>object</i> pada perangkat lunak untuk mengukur kompleksitas dari sebuah software aplikasi java. Penulisan ini telah mempresentasikan sebelas metrik yang digunakan untuk membuat rangking beberapa program Java berdasarkan nilai kompleksitas.

Nama Pengarang	Nama Jurnal Nomor Publikasi	Tahun	Judul Penulisan	Hasil/Kesimpulan
	0814.		Berorientasi <i>Object</i>).	
M. Rizwan Jameel Qureshi			<i>Evaluation of the Design Metric to Reduce the Number of Defects in Software Development</i> (Evaluasi Desain Metrik untuk Mengurangi Jumlah Kecacatan pada Pembangunan Perangkat Lunak).	Analisis regresi menunjukkan bahwa semua <i>variabel</i> independen [CBO, DIT, NOC, WMC, RFC] memiliki efek signifikan pada jumlah cacat kecuali LCOM. Perusahaan pengembangan perangkat lunak harus berkonsentrasi pada LCOM untuk mengontrol cacat desain. Waktu yang digunakan untuk memperbaiki <i>bug</i> juga harus diperhitungkan. Setelah memperkirakan jumlah cacat, kita dapat dengan mudah menghitung waktu yang dibutuhkan untuk memperbaiki <i>bug</i> .
Wahyu Rifa'i Dwi septian		2010	Analisis Perbandingan Framework PHP Berdasarkan MOOSE CK dan Properti Kualitas Desain Menggunakan	Analisis Perbandingan <i>Framework</i> PHP Berdasarkan Metrik Chidamber dan Kemerer dan Properti Kualitas Desain Menggunakan Metode <i>Analytic Hierarchy Process</i> (AHP).

Nama Pengarang	Nama Jurnal Nomor Publikasi	Tahun	Judul Penulisan	Hasil/Kesimpulan
			Metode Analytic Hierarchy.	
Roni Yunis Arwin Halim		2014	Studi Empiris Hubungan Metrik Kohesi Dengan Kecenderungan Kesalahan Pada Aplikasi Berorientasi <i>Object</i> .	Nilai metrik kohesi yang diukur pada kode program mampu mendeteksi kecenderungan kesalahan perangkat lunak berorientasi <i>object</i> . Nilai metrik yang diukur pada kode program khususnya metrik kohesi LCOM memiliki korelasi positif yang signifikan terhadap kecenderungan kesalahan perangkat lunak. Nilai metrik kualitas yang diukur menggunakan desain UML memiliki hasil yang beragam dalam mendeteksi kecenderungan kesalahan perangkat lunak. Pada Apache Xalan 2.5.0 terlihat adanya korelasi positif yang signifikan tetapi hal ini tidak didukung oleh hasil korelasi pada versi 2.4.0.
Efano Hermawan Petrus			Pemeringkatan Software Aplikasi Berdasarkan Properti	Penulisan ini telah mengusulkan sebuah Metode baru untuk menentukan peringkat kualitas desain software berdasarkan

Nama Pengarang	Nama Jurnal Nomor Publikasi	Tahun	Judul Penulisan	Hasil/Kesimpulan
Mursanto			Kualitas Desain Dan <i>Metrics For Object Oriented Software</i> Menggunakan <i>Analytic Hierarchy Process (AHP)</i> .	implementasi code-nya dalam Java. Metode ini merupakan kombinasi dari Metrik Chidamber dan Kemerer dan Properti Kualitas Desain Menggunakan Metode <i>Analytic Hierarchy Process (AHP)</i>
Achmad Arwan			Membangun Hubungan Antara Metrik Chidamber & Kemerer Dengan Karakteristik Efisiensi Pada ISO/IEC 9126-2 untuk Aplikasi Sumber Terbuka.	Perhitungan pada penulisan ini menyatakan bahwa metrik yang berkaitan erat dan berbanding lurus adalah sebagai berikut, 1. CBO dengan waktu tanggap 2. NOC dan DIT berkaitan erat dengan rasio pemenuhan rata-rata waktu throughput 3. WMC, RFC dan LCOM berkaitan erat dengan waktu turnaround Penulisan ini juga telah memetakan metrik yang berkaitan erat namun berbanding terbalik sebagai berikut, 1. Waktu tanggap dengan DIT dan NOC,
Siti Rochimah				

Nama Pengarang	Nama Jurnal Nomor Publikasi	Tahun	Judul Penulisan	Hasil/Kesimpulan
				rasio pemenuhan rata-rata. 2. Rasio waktu tanggap kasus terburuk dengan DIT dan NOC 3. Waktu throughput dengan WMC, CBO, Rfc, LCOM. 4. Rasio pemenuhan rata-rata waktu throughput dengan CBO. 5. rasio pemenuhan rata-rata waktu turnaround dengan DIT dan NOC.
Wayan Gede Suka Parwita luh Arida Ayu Rahning Putri	ISBN 979-26-0255-0.		Komponen Penilaian Kualitas Perangkat Lunak Berdasarkan <i>Software Quality Models</i> .	Penulisan ini menjabarkan berbagai jenis software quality model yang ada beserta komponen-komponen yang digunakan dalam melakukan penilaian dari masing-masing model yang ada beserta komponen-komponen yang digunakan dalam melakukan penilaian dari masing-masing model.

2.2. Dasar Teori

Pada bagian ini dipaparkan beberapa teori yang digunakan dalam pengerjaan tugas akhir ini.

2.2.1. Object Oriented Programming

Pemrograman berorientasi *object* adalah teknik membuat suatu program berdasarkan *object* [15]. Pendekatan berorientasi *object* mulai berkembang karena adanya kesulitan dalam pengembangan sistem pada skala besar untuk menghasilkan sistem yang berkualitas sesuai dengan biaya dan waktu yang ada. Pendekatan ini menggabungkan data dan proses secara bersamaan dalam bentuk *object*, hal tersebut menjadi kelebihan dari pendekatan ini karena kode program menjadi lebih mudah digunakan kembali. Pendekatan ini memperkenalkan istilah *class*, *object*, atribut dan *method*.

Setiap *object* mempunyai *field*/atribut dan *method*. *Field*/atribut adalah segala sesuatu yang berhubungan dengan karakteristik *object*. *Method* merupakan fungsi atau segala sesuatu yang dapat dilakukan oleh *object*. *Class* adalah tempat *object* tersebut berada [15].

Object mempunyai dua karakteristik yaitu *variabel* atau *field* sebagai status dan *method* sebagai perilaku dari sebuah *object*. *Object* menyimpan statusnya pada *variabel* dan mendefinisikan perilakunya melalui sebuah *method*. *Method* akan mengakses nilai dari *field object* dan sebagai mekanisme utama komunikasi antar *object*, sehingga dunia luar tidak perlu mengetahui bagaimana *object* dapat saling berkomunikasi melalui sebuah *method*. *Method* adalah sebuah operasi pada sebuah *object* dan didefinisikan dalam deklarasi *class*. *Message* adalah permintaan dari *object* lain untuk melakukan sebuah operasi/fungsi. *Object* tersebut menjawab *message* dengan sebuah *method*. *Cohesion* adalah tingkat *method* dalam *class* yang direlasikan ke *class* lain. Desain berorientasi objek

yang efektif memaksimumkan *cohesion* karena meningkatkan *encapsulation*. *Cohesion* yang tinggi mengindikasikan subdivisi *class* yang baik. *Coupling* adalah *method* dalam sebuah *class* memanggil *method* pada *class* lain. *Inheritance* adalah hirarki *class*, terdapat *superclass* dan *subclass* yang mewarisi atribut atau *method* dari *superclass*. *Package* merupakan sekelompok *class* dan *interface* yang saling terkait [16].

Pemrograman *object* mempunyai empat ciri-ciri yaitu abstraksi (*abstraction*), pembungkusan (*encapsulation*), pewarisan (*inheritance*) dan polimorfisme (*polymorphism*) [17].

2.2.1.1 Abstraksi (*abstraction*)

Abstraksi adalah pengabstrakan atau penyembunyian kerumitan dari suatu proses. Abstraksi yaitu menyembunyikan detail detail ketika mereka tidak diperlukan. Konsep berorientasi *object* dapat menyembunyikan kerumitan proses. Abstraksi menjelaskan definisi dari *object* pada *class* dalam suatu *method* atau konstruktor. Tujuan dari *class* abstrak adalah agar *class* lain dapat melakukan *extend* pada *class* tersebut dengan menjadi *subclass* dari *class* tersebut [8].

2.2.1.2 Pembungkusan (Encapsulation)

Encapsulation merupakan teknik yang membuat *variabel* atau *field class* menjadi bersifat *private/protected*. Jika *field* di deklarasikan sebagai *private*, maka *field* ini tidak bisa diakses oleh siapapun diluar *class*, dengan demikian *field* disembunyikan di dalam *class*. Dengan kata lain *encapsulation* juga berfungsi untuk menyembunyikan data.

Encapsulation juga merupakan tembok penghalang yang mencegah kode atau data diakses oleh data lain di luar *class*. Akses ke kode dan data di kontrol melalui *interface*. Manfaat utama teknik *encapsulation* adalah kita mampu memodifikasi

kode tanpa merusak kode yang telah digunakan pada *class* lain.

Encapsulation berarti suatu *object* menyembunyikan apa yang dilakukan dari *object-object* lain. *Encapsulation* disebut juga dengan penyembunyian informasi. Dalam melakukan pembungkusan kode dan data didalam java, terdapat tiga tingkat akses yang perlu diketahui, yaitu *private*, *protected*, dan *public*.

Pemrograman berorientasi *object* memperlakukan prosedur dan data tidak seperti pemrograman konvensional yang dipisahkan satu sama lain, salah satu konsep berorientasi *object* yang paling penting adalah membungkus prosedur dan data menjadi satu *object*. Konsep tersebut disebut sebagai *encapsulation* [18].

Dalam Java, dasar *encapsulation* adalah *class*. Suatu *class* yang menyatakan bahwa atribut atau *method* sebuah *class* tidak dapat diakses oleh *class* lain dengan menjadikan *class* tersebut *private*, atau menjadikan *class* tersebut *protected* – yaitu hanya bisa diakses oleh turunannya, atau menjadikan *class* tersebut *public* – yaitu bisa diakses oleh sembarang *class*. Pembungkusan dalam menyembunyikan *method* pada *super class*, dapat juga digunakan untuk membungkus suatu fungsi [8].

2.2.1.3 Pewarisan (Inheritance)

Model pemrograman berorientasi *object* juga menerapkan konsep pewarisan, sama halnya seperti yang terjadi di dunia atau kehidupan nyata (alam sekitar). Pemrograman berorientasi *object* memodelkan *object* yang ada di dunia nyata ke dalam *object* pemrograman, dalam berorientasi *object* juga terdapat pewarisan (*inheritance*) dimana suatu *object* dapat mewariskan sifat-sifat yang dimilikinya kepada *object* turunannya. Dalam sebuah program, suatu *class* dapat diturunkan menjadi *class-class* baru lainnya yang akan mewarisi beberapa sifat atau

prilaku dari *class* induknya. Dalam terminologi java, *class* induk dinamakan dengan *superclass* dan *class* turunan disebut dengan *subclass* [19].

2.2.1.4 Polimorfisme (Polymorphism)

Polymorphism merupakan kemampuan objek untuk menjadi berbagai bentuk. Paling umum terjadi adalah ketika referensi *superclass* digunakan untuk *subclass*. *Polimorfisme* adalah kemampuan suatu *object* untuk mengungkap banyak hal melalui satu cara yang sama. *Polymorphism* membuat *object-object* yang berasal dari *subclass* yang berbeda, diperlakukan sebagai *object-object* dari satu *superclass*. Sebagai contoh terdapat *class* A yang diturunkan menjadi *class* B, C, dan D. Dengan konsep *polimorfisme*, kita dapat menjalankan *method-method* yang terdapat pada *class* B, C, dan D hanya dari *object* yang diinstansiasi dengan *class* A. *Polomorfisme* sering dinamakan dengan *dynamic binding*, maupun *runtime binding* [19].

2.2.1.5 Pemograman Java

Java adalah salah satu pemograman berorientasi *object* (OOP) [19]. Java memiliki keutamaan dibanding bahasa pemograman lain, Java memiliki *Cross Platform* dengan adanya *Java Virtual Machine* (JVM), pengembangannya didukung dengan *programmer* secara luas dan *Automatic Garbage Collection* yang membebaskan *programmer* dari tugas manajemen memori.

- Java mempunyai beberapa keunggulan yaitu sederhana, berorientasi *object*, terdistribusi, kuat, aman, netral arsitektur, *portabel*, interpreter, kinerja yang tinggi, dan *multithreaded* [20].
 - *Syntax* untuk Java seperti *syntax* pada C++ tetapi *syntax* Java tidak memerlukan *header file*, *pointer arithmetic* (atau bahkan *pointer syntax*), struktur

union, operator *overloading*, *class virtual base*, dan yang lainnya.

- Fasilitas pemrograman berorientasi *object* pada Java pada dasarnya adalah sama dengan C++. Fitur pemrograman berorientasi *object* pada Java sebanding dengan C++, perbedaan utama antara Java dengan C++ terletak pada inhirarki berganda (*multiple inheritance*), untuk ini Java memiliki cara penyelesaian yang lebih baik.
- Java memiliki *library* rutin yang luas untuk dirangkai pada protokol TCP/IP seperti HTTP dan FTP dengan mudah.
- Aplikasi Java dapat membuka dan mengakses *object* untuk segala macam NET lewat URL sama mudahnya seperti yang biasa dilakukan seorang programmer ketika mengakses file sistem secara lokal.
- Java dimaksudkan untuk membuat suatu program yang dapat dipercaya dalam berbagai hal. Java banyak menekankan pada pengecekan awal untuk kemungkinan terjadinya masalah, pengecekan pada saat *runtime* dan mengurangi kemungkinan timbulnya kesalahan (*error*). Perbedaan utama antara Java dan C++ adalah Java memiliki sebuah model pointer yang mengurangi kemungkinan terjadinya penimpaan (*overwriting*) pada memory dan kerusakan data (*data corrupt*).
- Java dimaksudkan untuk digunakan pada jaringan terdistribusi. Sebelum sampai pada bagian tersebut, penekanan terutama ditujukan pada masalah keamanan. Java memungkinkan penyusunan program yang bebas virus, sistem yang bebas dari kerusakan.
- Kompiler membangkitkan sebuah format file dengan *object* arsitektur sistem syaraf, program yang di kompilasi dapat dijalankan pada banyak prosesor, diberikan sistem *run time* dari Java. Kompiler Java melakukannya dengan membangkitkan instruksi-instruksi kode *byte* yang tidak dapat dilakukan oleh

arsitektur komputer tertentu. Java dirancang untuk mempermudah penterjemahan pada banyak komputer dengan dan diterjemahkan pada komputer asal pada saat *run-time*. Tidak seperti pada C dan C++, di Java terdapat ketergantungan pada saat implementasi (implement dependent). ukuran dari tipe data primitif ditentukan, sebagaimana kelakuan aritmatik padanya. *Library* atau pustaka merupakan bagian dari sistem yang mendefinisikan interface yang portabel.

- *Interpreter* Java dapat mengeksekusi kode byte Java secara langsung pada komputer-komputer yang memiliki *interpreter*. Dan karena proses linking dalam Java merupakan proses yang kenaikannya tahap demi tahap dan berbobot ringan, maka proses pengembangan dapat menjadi lebih cepat.
 - Meskipun kinerja kode *byte* yang di interpretasi biasanya lebih dari memadai, tetapi masih terdapat situasi yang memerlukan kinerja yang lebih tinggi. Kode *byte* dapat diterjemahkan (pada saat *run-time*) di dalam kode mesin untuk CPU tertentu dimana aplikasi sedang berjalan.
 - *Multithreading* adalah kemampuan sebuah program untuk melakukan lebih dari satu pekerjaan sekaligus. Keuntungan dari *multithreading* adalah sifat respon yang interaktif dan *real time*.
 - Dalam sejumlah hal, Java merupakan bahasa pemrograman yang lebih dinamis dibandingkan dengan C atau C++. Java dirancang untuk beradaptasi dengan lingkungan yang terus berkembang. *Library* Java dapat dengan mudah menambah *method* dan *variabel* contoh yang baru tanpa banyak mempengaruhi klien.
- Kelebihan Java dibandingkan dengan C++
 - Program Java telah merancang java untuk menghilangkan pengalokasian dan dealokasi memori

secara manual, karena java memiliki *Garbage Collection*.

- Java mempunyai *array* yang sebenarnya dan menghilangkan aritmatika pointer. Hal ini yang sering menyebabkan memori *overwrite*.
- Java menghilangkan *multiple inheritance*, diganti dengan interface [21].

2.2.2. Metrik Chidamber dan Kemerer

Metrik merupakan suatu prosedur yang memasang karakteristik tertentu pada entitas yang diamati menjadi sebuah nilai numerik [1]. Nilai numerik pada metrik akan memberikan pengetahuan pengamat mengenai nilai yang terlalu tinggi atau terlalu rendah, terlalu banyak atau terlalu sedikit. Manfaat metrik sangat tergantung pada apa yang akan dicapai dari hasil pengukuran yang telah dilakukan.

Metrik Chidamber dan Kemerer adalah salah satu metrik yang digunakan untuk mengukur kualitas disain sebuah perangkat lunak berdasarkan enam metrik dengan melihat pada perspektif desain berorientasi object [22].

Metrik Chidamber dan Kemerer mempunyai enam metrik yaitu Weighted Method per Class (WMC), Depth of Inheritance Tree (DIT), Number OF Children (NOC), Response For a Class (RFC), Coupling Between Object Classes (CBO) dan Lack of Cohesion Method (LCOM).

SATC (Software Assurance Technology Center) mendiskusikan pendekatannya untuk mengidentifikasi sekumpulan *object oriented metric* [16]. SATC telah menghasilkan pemetaan Metrik Chidamber dan Kemerer pada kode program perangkat lunak, pemetaan tersebut ada pada Tabel 2. 2.

Tabel 2. 2 Pemetaan Metrik Chidamber dan Kemerer dengan Kode Program

Source	Metrik	Object Oriented Construct
<i>Object Oriented (New)</i>	<i>Weighted Method per Class (WMC)</i>	<i>Class/Method</i>
<i>Object Oriented (New)</i>	<i>Depth of Inheritance Tree (DIT)</i>	<i>Inheritance</i>
<i>Object Oriented (New)</i>	<i>Number OF Children (NOC)</i>	<i>Inheritance</i>
<i>Object Oriented (New)</i>	<i>Response For a Class (RFC)</i>	<i>Class/Message</i>
<i>Object Oriented (New)</i>	<i>Coupling Between Object Classes (CBO)</i>	<i>Coupling</i>
<i>Object Oriented (New)</i>	<i>Lack of Cohesion Method (LCOM)</i>	<i>Class/Cohesion</i>

Parameter-parameter Metrik Chidamber dan Kemerer mempunyai objektifitas dari pengukuran kualitas perangkat lunak. Terdapat orientasi dari masing-masing Metrik Chidamber dan Kemerer [23]. Metrik Chidamber dan Kemerer dapat dipetakan dengan *Object Oriented Design Element* seperti pada Tabel 2. 3.

Tabel 2. 3 Pemetaan Metrik Chidamber dan Kemerer dengan *Object Oriented Design Element*

<i>Metrics</i>	Object definition	Objet attributes	Object communication
WMC	V	V	
DIT	V		
NOC	V		
RFC		V	V
CBO			V
LCOM		V	

Object Oriented Design Element yang telah dipetakan dengan parameter-parameter Metrik Chidamber dan Kemerer dapat menjadi *tools* yang dapat digunakan dalam membandingkan parameter-parameter Metrik Chidamber dan Kemerer.

Penelitian sebelumnya telah menghasilkan kategori dari Metrik Chidamber dan Kemerer [24] [25] yang telah disajikan pada Tabel 2. 4, hijau mewakili kategori *good*, kuning mewakili kategori *medium*, merah mewakili kategori *bad*.

Tabel 2. 4 Nilai Metrik Chidamber dan Kemerer

Metrik Chidamber dan Kemerer	Hijau	Kuning	Merah
WMC	$1 \leq x < 50$	$50 \leq x \leq 150$	$150 < x$
DIT	$x < 5$	$5 \leq x < 8$	$8 \leq x$
NOC	$x < 4$	$4 \leq x < 8$	$8 \leq x$
CBO	$x < 14$	$14 \leq x \leq 150$	$150 < x$
RFC	$x < 100$	$100 \leq x \leq 1000$	$1000 < x$

Metrik Chidamber dan Kemerer digunakan sebagai tolak ukur dalam pengukuran *Software Accounting XYZ*. Metrik Chidamber dan Kemerer dapat dideskripsikan secara singkat pada Tabel 2. 5.

Tabel 2. 5 Batasan Metrik Chidamber dan Kemerer

Metrik	Hasil (Semakin Tinggi Nilai Berarti)
<i>Weighted Method per Class</i>	Membatasi potensi <i>reuse</i>
	Semakin kompleks
	Membutuhkan waktu dan usaha yang tinggi pada tahapan <i>developing and maintenance</i>
	Kemungkinan banyaknya <i>bug</i>
<i>Depth of Inheritance Tree</i>	Kemungkinan <i>reuse</i> semakin tinggi
	Semakin kompleks

Metrik	Hasil (Semakin Tinggi Nilai Berarti)
<i>Number Of Children</i>	Kemungkinan <i>reuse</i> semakin tinggi
	Mebutuhkan usaha <i>testing</i> yang lebih tinggi
	Semakin besar kemungkinan ketidakcocokan <i>subclass</i>
<i>Coupling Between Object Classes</i>	Kemungkinan banyaknya <i>bug</i> dalam perangkat lunak tersebut
	Membatasi potensi <i>reuse</i>
<i>Response For a Class</i>	<i>Testing</i> dan <i>debugging</i> akan semakin kompleks
	Kemungkinan banyaknya <i>bug</i> dalam perangkat lunak tersebut
<i>Lack of Cohesion Method</i>	Membatasi potensi <i>reuse</i>
	Mengindikasikan <i>class</i> tersebut sederhana (tidak kompleks)

2.2.2.1 Weighted Methods (Per) Class

Metrik *Weighted Method per Class* adalah pengukuran jumlah *method* dalam setiap *class*-nya [26]. Kompleksitas suatu *class* dapat dinilai langsung dengan WMC. *Method* merupakan properti dari *object*, Kompleksitas sebuah *object* ditentukan dengan menghitung propertinya.

Metrik WMC memprediksikan waktu dan usaha yang diperlukan untuk membangun dan *maintenance* suatu *class* [27]. Nilai WMC yang tinggi dapat mengindikasikan kesalahan yang lebih banyak. *Class* dengan WMC rendah seringkali mengindikasikan *polimorfisme* yang lebih besar. *Class* dengan *method* yang banyak akan cenderung menjadi aplikasi atau perangkat lunak yang spesifik sehingga akan membatasi kemungkinan penggunaan kembali. Semakin besar nilai WMC maka akan membatasi potensi penggunaan kembali, oleh karena itu WMC harus dijaga serendah mungkin[16].

Dalam 20 program atau perangkat lunak yang diuji pada penulisan ini menunjukkan bahwa peningkatan WMC akan meningkatkan kompleksitas dan mengurangi kualitas program. WMC yang tinggi telah mengakibatkan program rentan terhadap *bug* dan terlalu kompleks untuk dipahami [28]. Nilai WMC yang baik adalah 20 sampai 50, semakin rendah akan semakin bagus. Nilai WMC yang terlalu tinggi mempunyai kecenderungan kemungkinan kegagalan software [29]. Sebuah studi dari 30 proyek dengan bahasa C++ menunjukkan bahwa peningkatan WMC akan meningkatkan kepadatan bug dan menurunkan kualitas [25].

Apabila *class* C_1 dengan *method* c_1, \dots, c_n maka, WMC dirumuskan pada persamaan 2.1 [26].

$$WMC = \sum_{i=1}^n c_i + \sum_{i=1}^n \sum_{j=i}^m d_{ij}$$

$c_1, c_2, \dots, c_n = \text{method in class } C$
 $i = \text{method ke } 1 \text{ sampai } n$

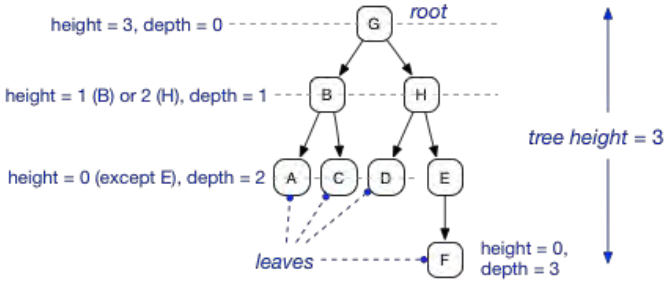
(2.1)

2.2.2.2 Depth of Inheritance Tree (DIT)

Metrik *Depth of Inheritance Tree* adalah panjang maksimum dari *root* ke *leaves* dalam sebuah pohon hirarki pada sebuah *class* [27]. Kedalaman suatu *class* disebut hirarki, DIT akan mengukur kedalaman sebuah *class*, dimulai dari *class leaves* dan menurun pada *class root* [30]. Hirarki yang dalam akan menyebabkan tingginya kompleksitas dari disain karena semakin banyak *method* dan *class* yang terlibat, sehingga akan sulit untuk membangun dan *maintenancenya*. Namun, Hirarki yang dalam meningkatkan kemungkinan penggunaan kembali [25].

Nilai DIT yang direkomendasikan adalah maksimal 5. Sebuah penulisan dari 30 proyek C++ menunjukkan bahwa peningkatan DIT dapat meningkatkan bug dan menurunkan

kualitas perangkat lunak [25]. Pada Gambar 2. 2 dilustrasikan penggambaran inheritance dan cara penghitungan DIT.



Gambar 2. 1 Ilustrasi DIT

2.2.2.3 Number of Children (NOC)

Metrik *Number Of Children* menghitung jumlah *subclass* dari *superclass* dalam sebuah hirarki *class* [27]. Metrik NOC dapat menjadi indikator besarnya pengaruh sebuah *class* terhadap disain sistem secara keseluruhan. Semakin besar nilai NOC, semakin besar kemungkinan ketidakcocokan *subclass* dengan abstraksi yang ada pada *superclass* [28]. Hal ini akan berpengaruh pada kesalahan penggunaan *subclass*. Sebuah *class* dengan nilai NOC dan WMC yang tinggi menunjukkan kompleksitas pada *superclass*. Hal ini menunjukkan perlunya desain ulang pada sistem tersebut [18].

Metrik NOC menghitung jumlah anak (*subclass*) dari keturunan langsung dari *superclass* pada sebuah *class*. Nilai NOC akan menunjukkan tingkat *reusability* dan usaha *testing* yang diperlukan. Semakin banyak jumlah *subclass* maka akan semakin besar tingkat penggunaan kembali karena *inheritance* merupakan bentuk dari *reuse*, semakin banyak *testing* yang harus dilakukan karena apabila terjadi perubahan di *superclass* dapat mempengaruhi *subclass* dari *class* tersebut [30].

2.2.2.4 Coupling Between Object Classes (CBO)

Metrik *Coupling Between Object Classes* digunakan untuk menghitung *class* yang berhubungan/tergantung dengan *class* lainnya [27]. Semakin banyak *class* yang tidak tergantung satu sama lain (semakin rendah nilai metrik CBO) akan meningkatkan kemungkinan *modularity* dan penggunaan kembali (*reuse*). Hal ini berhubungan dengan tingkat *independency* antar modul [31].

Nilai CBO yang tinggi akan mengindikasikan *coupling* yang berlebihan sehingga merugikan desain modular dan membatasi *reuse*, menyulitkan *testing* dan modifikasi. Semakin independen suatu *class* maka akan semakin mudah untuk menggunakan kembali pada aplikasi lain. *Coupling* yang berlebihan juga akan mengidentifikasi kelemahan dari *class* dalam melakukan enkapsulasi. Menurut Sahraoui, Godin dan Miceli dalam penulisannya mengungkapkan nilai CBO yang bagus adalah kurang dari 14 [28]. Sebuah Class X berkaitan (coupled) dengan class C apabila, X *operates on* (*affects*) C or C operates on X.

2.2.2.5 Response for A Class (RFC)

Metrik *Response For a Class* menghitung jumlah semua *method* yang dipanggil sebagai respon terhadap *object* luar dari sebuah *class* [30]. RFC mencakup semua *method* yang diakses dalam hirarki *class* tersebut. RFC digunakan untuk mengukur banyaknya komunikasi antar *object* dalam suatu *class*. Sehingga semakin tinggi nilai RFC maka, akan semakin banyak *method* yang digunakan untuk merespon *object* dari luar dan semakin kompleks sehingga akan meningkatkan waktu untuk *testing* [31].

Nilai RFC yang tinggi akan meningkatkan kemungkinan banyak kesalahan, karena *class* dengan RFC tinggi akan lebih kompleks dan sulit dimengerti. RFC yang tinggi akan

mengidentifikasi pengujian dan *debugging* yang rumit [28]. Rumus RFC ada pada persamaan 2.2.

*M_{ci} of methods called in response to a message
that invoked Method M_i*

$$RFC = \sum_{i=1}^n M_{c_i} \quad (2.2)$$

2.2.2.6 Lack Metrik Chidamber dan Kemerer of Cohesion Method (LCOM)

Metrik *Lack of Cohesion Method* mengukur kemiripan *method* dalam sebuah *class* dari *instance variabel* atau atribut. LCOM mengukur *method* yang tidak terhubung dengan *class* lain (mewakili bagian independen dalam *class*). Tingginya kohesi mengidentifikasi potensi yang baik pada *class* tersebut, mengindikasikan *class* tersebut sederhana dan memiliki sifat *reusability* yang tinggi. Sedangkan semakin rendah kohesi maka semakin kompleks *class* tersebut.

Apabila *p* adalah *method* yang tidak memiliki irisan atribut dengan *method* lainnya dan *q* adalah *method* yang memiliki irisan atribut dengan *method* lainnya maka, rumus LCOM ada pada persamaan 2.3.

$$\begin{aligned} & \text{if } p > q \text{ then, } LCOM = p - q \\ & \text{else } LCOM = 0 \end{aligned} \quad (2.3)$$

2.2.3. ISO/IEC 9126

ISO 9126 merupakan *best practice* dalam melakukan evaluasi terhadap kualitas produk perangkat lunak. ISO/IEC 9126 adalah standar internasional yang diterbitkan oleh ISO/IEC, standar ini dibagi menjadi empat bagian yang masing-masing

menjelaskan model kualitas, metrik eksternal, metrik internal, dan metrik kualitas [32]. Pada Tabel 2. 6 terdapat penjelasan singkat terkait pembagian ISO/IEC 9126 [33].

Tabel 2. 6 Pembagian ISO/IEC 9126

ISO/IEC 9126		Pembagian ISO/IEC 9126
<i>ISO/IEC 9126-1:2001 Software engineering</i>	Product Quality	<i>Part 1: Quality model</i>
<i>ISO/IEC TR 9126-2:2003 Software engineering</i>		<i>Part 2: External metrics</i>
<i>ISO/IEC TR 9126-3:2003 Software engineering</i>		<i>Part 3: Internal metrics</i>
<i>ISO/IEC TR 9126-4:2004 Software engineering</i>		<i>Part 4: Quality in use metrics</i>

ISO/IEC 9126 mempunyai enam model kualitas yang telah dicantumkan dalam ISO/IEC 9126-1. ISO/IEC 9126-1 (bagian pertama dari ISO 9126) membagi model kualitas perangkat lunak (software quality model) menjadi enam karakteristik yaitu fungsionalitas (*Functionability*), kehandalan (*Reliability*), kebergunaan (*Usability*), efisiensi (*Efficiency*), keterpeliharaan (*Maintainability*) dan portabilitas (*Portability*) [32]. Pada Tabel 2. 7 terdapat penjelasan definisi Kriteria dan sub kriteria dari ISO/IEC 9126 [33].

Tabel 2. 7 Deskripsi Kriteria dan Subkriteria ISO/IEC 9126-1

Kriteria	Deskripsi Kriteria	Sub Kriteria	Deskripsi Sub Kriteria
Functionality	Kemampuan produk software untuk memberikan fungsi yang memenuhi kebutuhan ketika perangkat lunak digunakan dalam kondisi tertentu	Suitability	Kemampuan produk software untuk memberikan set fungsi yang tepat untuk tugas-tugas tertentu dan tujuan pengguna
		Accuracy	Kemampuan produk software untuk memberikan ketepatan atau hasil atau pengaruh dengan tingkat yang dibutuhkan sesuai presisi yang disepakati
		Interoperability	Kemampuan produk software untuk berinteraksi dengan satu atau sistem tertentu
		Security	Kemampuan produk software untuk melindungi informasi dan data sehingga orang atau sistem yang tidak sah tidak dapat membaca atau memodifikasinya dan orang-orang atau sistem yang berwenang diberi aksesnya
		Functionality Compliance	Kemampuan produk software untuk mematuhi standar, konvensi atau peraturan dalam undang-undang yang berkaitan dengan fungsi
Reliability	Kemampuan produk software untuk mempertahankan	Maturity	Kemampuan produk software untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak
		Fault Tolerance	Kemampuan produk software untuk

Kriteria	Deskripsi Kriteria	Sub Kriteria	Deskripsi Sub Kriteria
	tingkat tertentu dari kinerja ketika digunakan dalam kondisi yang spesifik		mempertahankan tingkat tertentu dari kinerja dalam kasus kesalahan perangkat lunak atau ketidaksesuaian dari antarmuka yang ditentukan
		Recoverability	Kemampuan produk software untuk membangun kembali tingkat tertentu kinerja dan memulihkan data yang terkena dampak langsung dalam kasus kegagalan
		Reliability Compliance	Kemampuan produk software untuk mematuhi standar, konvensi atau peraturan yang berkaitan dengan keandalan
Usability	Kemampuan produk software untuk dipahami, dipelajari, digunakan dan atraktif bagi pengguna, bila digunakan dalam kondisi tertentu	Understandability	Kemampuan produk software yang memungkinkan pengguna untuk memahami apakah software tersebut cocok, dan bagaimana hal itu dapat digunakan untuk tugas-tugas dan ketentuan penggunaan tertentu
		Learnability	Kemampuan produk software yang memungkinkan pengguna untuk mempelajari penerapannya
		Operability	Kemampuan produk software yang memungkinkan pengguna untuk mengoperasikan dan mengendalikannya
		Attractiveness Usability	Kemampuan produk software menjadi menarik bagi pengguna

Kriteria	Deskripsi Kriteria	Sub Kriteria	Deskripsi Sub Kriteria
		Compliance	Kemampuan produk software untuk mematuhi standar, konvensi, panduan gaya atau peraturan yang berkaitan dengan kegunaan
Efficiency	Kemampuan produk software untuk memberikan kinerja yang tepat, relatif dengan jumlah sumber daya yang digunakan, dalam kondisi yang telah ditentukan	<i>Time Behaviour</i>	Kemampuan produk software untuk memberikan respon dan pengolahan waktu yang tepat dan tingkat throughput saat melakukan fungsinya, dalam kondisi yang telah ditentukan
		<i>Resource Utilisation</i>	Kemampuan produk software untuk menggunakan jumlah dan jenis sumber daya yang tepat ketika perangkat lunak melakukan fungsinya di bawah kondisi yang dinyatakan dalam kondisi yang telah ditentukan
		<i>Efficiency Compliance</i>	Kemampuan produk software untuk mengikuti standar atau konvensi yang berkaitan dengan efisiensi
Maintainability	Kemampuan produk software untuk dimodifikasi. Modifikasi dapat mencakup koreksi, perbaikan atau adaptasi dari	<i>Analyzability</i>	Kemampuan produk software untuk dapat didiagnosis kekurangan atau penyebab kegagalan dalam perangkat lunak, atau untuk bagian-bagian yang akan dimodifikasi untuk diidentifikasi
		<i>Changeability</i>	Kemampuan produk perangkat lunak untuk memungkinkan modifikasi tertentu untuk diimplementasikan
		<i>Stability</i>	Kemampuan produk software untuk menghindari efek tak terduga dari modifikasi perangkat lunak

Kriteria	Deskripsi Kriteria	Sub Kriteria	Deskripsi Sub Kriteria
	perangkat lunak untuk perubahan lingkungan, dan persyaratan dan spesifikasi fungsional	<i>Testability</i>	Kemampuan produk perangkat lunak untuk memungkinkan perangkat lunak dimodifikasi untuk divalidasi
		<i>Maintainability</i> <i>Compliance</i>	Kemampuan produk software untuk mengikuti standar atau konvensi yang berkaitan dengan pemeliharaan
Portability	Kemampuan produk software yang akan ditransfer dari satu lingkungan yang lain	<i>Adaptability</i> <i>Installability</i>	Kemampuan produk software harus disesuaikan untuk lingkungan tertentu yang berbeda tanpa menerapkan tindakan atau berarti selain yang disediakan untuk tujuan ini untuk perangkat lunak dipertimbangkan
		Co-Existence	Kemampuan produk software untuk hidup berdampingan dengan perangkat lunak independen lain dalam lingkungan yang sama berbagi sumber daya umum
		Replaceability	Kemampuan produk software yang akan digunakan di tempat produk software lain yang ditentukan untuk tujuan yang sama dalam lingkungan yang sama
		Portability Compliance	Kemampuan produk software untuk mengikuti standar atau konvensi yang berkaitan dengan portabilitas.

2.2.3.1. Hubungan ISO/IEC 9126-1 dengan Parameter Metrik Chidamber dan Kemerer

Kriteria kualitas dari perangkat lunak yang ada pada ISO/IEC 9126-1 dapat dipetakan dengan parameter yang ada pada Metrik Chidamber dan Kemerer [11]. Kriteria kualitas perangkat lunak yang ada pada Metrik Chidamber dan Kemerer tidak cukup dihitung dengan menggunakan satu metrik saja, sehingga hanya beberapa kriteria saja yang dapat diukur menggunakan metrik chidamber dan kemerer.

Tabel 2. 8 Pemetaan Quality Model ISO/IEC 9126-1 dengan Metrik Chidamber dan Kemerer

Properti Kualitas Software	Parameter Metric
<i>Efficiency</i>	LCOM, CBO, DIT, NOC
<i>Understandability</i>	WMC, RFC, DIT
<i>Reusability/ Replaceability</i>	WMC, LCOM, CBO, DIT, NOC
<i>Maintainability</i>	WMC, RFC, DIT, NOC.

Metrik Chidamber dan Kemerer hanya dapat mengukur *efficiency*, *understandability*, *reusability/ replaceability* dan *maintainability*.

Tabel 2. 9 Kriteria dan Subkriteria ISO/IEC 9126-1 yang akan diukur

Kriteria ISO/IEC 9126-1	Sub Kriteria ISO/IEC 9126-1
<i>Functionality</i>	<i>Suitability</i>
	<i>Accuracy</i>
	<i>Interoperability</i>
	<i>Security</i>
	<i>Functionality Compliance</i>
<i>Reliability</i>	<i>Maturity</i>
	<i>Fault Tolerance</i>
	<i>Recoverability</i>
	<i>Reliability Compliance</i>
<i>Usability</i>	<i>Understandability</i>
	<i>Learnability</i>

Kriteria ISO/IEC 9126-1	Sub Kriteria ISO/IEC 9126-1
	<i>Operability</i>
	<i>Attractiveness Usability</i>
	<i>Compliance</i>
<i>Efficiency</i>	<i>Time Behaviour</i>
	<i>Resource Utilisation</i>
	<i>Efficiency Compliance.</i>
<i>Maintainability</i>	<i>Analyzability</i>
	<i>Changeability</i>
	<i>Stability</i>
	<i>Testability</i>
	<i>MaintainabilityCompliance</i>
<i>Portability</i>	<i>Adaptability Installability</i>
	<i>Co-Existence</i>
	<i>Replaceability</i>
	<i>Portability Compliance.</i>

Metrik Chidamber dan Kemerer hanya dapat mengukur dua kriteria dan dua sub kriteria dari kualitas model ISO/IEC 9126-1, seperti yang telah diwarnai abu-abu pada Tabel 2. 9. Pada Tabel 2. 10 dijelaskan deskripsi masing-masing kriteria dan subkriteria yang akan diukur.

Tabel 2. 10 Deskrip Kriteria dan Subkriteria ISO/IEC 9126-1

Kriteria/Sub Kriteria ISO/IEC 9126-1	Deskripsi Kriteria/Sub Kriteria ISO/IEC 9126-1
<i>Understandability</i>	Tingkat kemudahan dalam mempelajari masukan dan keluaran dari program
<i>Efficiency</i>	Jumlah sumberdaya yang diproses dan kode yang diperlukan oleh program untuk melaksanakan fungsi tertentu
<i>Maintainability</i>	Tingkat kemampuan software untuk dimodifikasi. Modifikasi dapat mencakup koreksi, perbaikan atau adaptasi dari perangkat lunak untuk perubahan kebutuhan
<i>Replaceability</i>	Tingkat kemampuan program/bagian dari

Kriteria/Sub Kriteria ISO/IEC 9126-1	Deskripsi Kriteria/Sub Kriteria ISO/IEC 9126-1
	program yang dapat dipakai ulang dalam aplikasi lainnya, berkaitan dengan lingkup dari fungsi yang dilakukan oleh program tersebut

2.2.3.2. Pengaruh Parameter Metrik Chidamber dan Kemerer pada Properti Kualitas Code Software.

Hubungan Metrik Chidamber dan Kemerer terhadap kriteria kualitas model ISO/IEC 9126-1 berbanding terbalik [11]. Meskipun terdapat beberapa penulisan Linda H Roseberg pada tahun yang berbeda namun kedua penulisan tersebut menegaskan bahwa Metrik Chidamber dan Kemerer memiliki nilai yang berbanding terbalik dengan kualitas *Code code software*.

Tabel 2. 11 Korelasi Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-1

Metric	<i>Desirable Value</i>	<i>Efficiency</i>	<i>Understandability</i>	<i>Maintainability</i>	<i>Replaceability</i>
WMC	↓		↑	↑	↑
DIT	↓	↑	↑	↑	↑
NOC	↓	↑		↑	↑
CBO	↓	↑			↑
RFC	↓		↑	↑	
LCOM	↓	↓		↓	↓

2.2.4. Design Patterns

Dalam membuat desain sebuah perangkat lunak biasanya menggunakan asumsi atau pemahaman yang bersifat subjektif sehingga dibutuhkan gambaran secara formal dari masalah dan pemecahannya. *Design patterns* adalah unsur-unsur rancangan yang seringkali muncul pada berbagai sistem yang berbeda. Setiap pemakaian *patterns* akan menguji *pattern* tersebut di berbagai

situasi. Sebuah *design pattern* harus mendokumentasikan permasalahan, pemecahan, serta akibat-akibat penggunaannya. *Class diagram* adalah salah satu bentuk dari interpretasi dari suatu *pattern* dengan memanfaatkan kemampuan UML yang sudah berorientasi pada perancangan yang berbasiskan objek (OOP) [34].

Dalam pengembangan perangkat lunak sering terjadi permasalahan yang terjadi berulang-ulang, sehingga seorang arsitek mungkin akan banyak menghabiskan waktu dalam memberikan solusi dengan masalah yang serupa. Pada saat merancang perangkat lunak, *patterns* adalah suatu dokumen yang sangat penting untuk dimiliki dan dipahami. Dimana *design patterns* bukan sekedar menyediakan solusi terbaik dalam memecahkan suatu masalah, tetapi lebih dari pada itu *patterns* membuat desain perangkat lunak menjadi lebih efektif, fleksible dan waktu penyelesaian desain perangkat lunak juga lebih efisien.

Ada banyak *Design Patterns* yang sudah diakui kemampuannya, diterima dan diaplikasikan oleh banyak praktisi. *Design Patterns* yang cukup populer adalah yang diperkenalkan *The Gang of Four* (GoF) - Erich Gamma, Richard Helm, Ralph Johnson dan John Vlissides. Dalam *The Gang of Four* (GoF) terdapat 23 *Pattern* yang dibagi menjadi 3 kelompok besar.

Pada *Gang of Four Patterns*, terdapat tiga katalog *design patterns* yaitu *creational*, *structural*, dan *behavioral*. *Creational patterns* berhubungan dengan penciptaan objek. Pola-pola ini berkisar seputar objek mana yang diciptakan, siapa yang menciptakannya, serta berapa banyak objek diciptakan. *Structural patterns* berhubungan dengan struktur statis objek dan kelas. Pola-pola dalam *structural patterns* dapat dilihat pada saat program di-compile melalui struktur *inheritance*, *properties*, serta *agregasi* objek-objek. *Behavioral patterns* terkait perilaku *run-time program*. Pola-pola ini berkaitan dengan algoritma serta interaksi antar objek saat program berjalan.

Penekanan *behavioral patterns* lebih pada komposisi objek ketimbang *inheritance* [34].

Creational patterns adalah *pattern* yang berhubungan dengan Menginisialisasi dan mengkonfigurasi kelas serta objek. *Structural patterns* adalah *pattern* yang digunakan untuk memisahkan antarmuka (*interface*) dan implementasi kelas serta objek, *pattern* ini juga berhubungan dengan susunan jumlah kelas atau objek. *Behavioral patterns* adalah *pattern* yang berhubungan dengan interaksi dinamis antara kumpulan semua kelas dan objek, *pattern* ini juga terkait bagaimana *pattern* mendistribusikan tanggung jawab/tugasnya masing-masing [34].

Creational Patterns (cara class/object di-inisiasi)

1. *Abstract Factory* (Menciptakan sebuah *instance* dari beberapa keluarga kelas, digunakan untuk membangun objek terkait)
2. *Builder* (Memisahkan konstruksi objek dari perwakilannya, digunakan untuk membangun objek yang kompleks secara bertahap)
3. *Factory Method* (Menciptakan sebuah *instance* dari beberapa kelas yang diturunkan, *method* dalam suatu kelas turunan untuk menciptakan asosiasi)
4. *Prototype* (sebuah *instance* yang sepenuhnya diinisiasikan untuk disalin atau dikloning, digunakan untuk mengkloning instances dari prototipe)
5. *Singleton* (sebuah kelas yang mana hanya sebuah *instance* yang dapat *exist*, digunakan untuk *instance* yang tunggal).

Structural Patterns (struktur/relasi antar object/class)

1. *Adapter* (Mencocokkan *interface* dari kelas yang berbeda, *translator* yang beradaptasi terhadap suatu *server interface* untuk suatu *client*)
2. *Bridge* (Memisahkan antarmuka objek dari implementasinya, *abstraksi* yang digunakan untuk mengikat menjadi satu dari banyak implementasi yang dibuat)

3. *Composite* (Struktur *tree* dari objek sederhana dan komposit, struktur untuk membangun *agregasi rekursif*)
4. *Decorator* (Menambahkan responsibilities untuk objek dinamis, dekorator memperluas suatu objek secara terbuka)
5. *Facade* (Kelas tunggal yang merepresentasikan seluruh subsistem, menyederhanakan *interface* untuk subsistem)
6. *Flyweight* (*Instance* yang digunakan untuk berbagi efisiensi, banyak objek kecil/spesifik yang di-*share* secara efisien)
7. *Proxy* (Sebuah objek yang mewakili objek lain, satu objek yang mendekati objek lain).

Behavioral Patterns (tingkah laku atau fungsi dari *class/object*.)

1. *Chain of Responsibility* (sebuah cara untuk melewati permintaan antara rangkaian objek, Melakukan *request* yang didelegasikan kepada penyedia layanan yang menjadi tugas dan tanggung jawabnya)
2. *Command* (Enkapsulasasi permintaan *command request* sebagai objek, melakukan *request* objek *first-class*)
3. *Interpreter* (Sebuah cara untuk memasukkan elemen bahasa dalam program, menggabungkan elemen-elemen yang diakses secara skuensial)
4. *Iterator* (Secara berurutan mengakses elemen lokasi, *Language interpreter for a small grammar*, kompiler skala kecil)
5. *Mediator* (Mendefinisikan komunikasi sederhana antara kelas satu dengan lainnya, mengkoordinasi interaksi antar asosiasi yang ada)
6. *Memento* (Menangkap dan memulihkan state internal objek, gambaran melakukan *captures* dan *restores* keadaan suatu objek secara *private*)
7. *Observer* (Sebuah cara untuk memberitahukan perubahan sejumlah kelas, ketergantungan untuk *update* secara otomatis ketika seorang *programmer* melakukan perubahan pada kodenya)
8. *State* (Mengubah *behavior* objek ketika keadaannya berubah, objek yang perilakunya tergantung pada keadaannya)

9. *Strategy* (Melakukan enkapsulasi algoritma dalam kelas, *abstraksi* untuk memilih salah satu dari sekian banyak algoritma)
10. *Template Method* (Memperlambat langkah-langkah yang tepat dari suatu algoritma untuk *subclass*, algoritma dengan beberapa langkah yang telah disediakan oleh suatu kelas turunan)
11. *Visitor* (Mendefinisikan operasi baru untuk kelas tanpa perubahan, operasi yang diterapkan pada elemen-elemen dari struktur objek yang heterogen/ sangat beragam).

Pemetaan design *pattern* dari penelitian sebelumnya ada pada Tabel 2. 12 [35].

Tabel 2. 12 Pemetaan *Design Pattern*

		<i>Purpose/ arah tujuan pattern</i>		
		Creational	Structural	Behavioral
Scope/ Ruang Lingkup	Class	Factory <i>method</i>	adapter	Interpreter
				<i>Template method</i>
	Object	Abstract factory	Adapter (object)	Chain of responsibility
		Builder	Bridge	Command
		Prototype	Composite	Iterator
		Singleton	Decorator	Mediator
			Façade	Memento
			Flyweight	Observer
			Proxy	State
		Strategy		
		Visitor		

Secara singkatnya, *design pattern* adalah *best practice* pola-pola desain kelas pada perancangan berorientasi objek. Dengan mengaplikasikan *design pattern* ini, desain sebuah aplikasi dapat menjadi dinamis terhadap perubahan. Pada Tabel 2. 12 akan

menjelaskan *design pattern* yang sering dipakai pada penelitian sebelumnya [36].

Tabel 2. 13 *Design Pattern* yang Sering Digunakan

Nama <i>Pattern</i>	Panjelasan
Creational <i>Patterns</i>	berhubungan dengan penciptaan objek. Pola-pola ini berkisar seputar objek mana yang diciptakan, siapa yang menciptakan serta berapa banyak objek yang diciptakan
Abstract Factory	Menciptakan sekumpulan objek yang berhubungan atau saling tergantung dengan menyediakan suatu antarmuka tanpa menentukan kelas yang nyata
Factory <i>Method</i>	Design pattern ini memisahkan kode untuk membuat objek, sehingga ketika jenis objek yang dapat dibuat bertambah, kode yang berubah hanya kode pembuatan objeknya saja.
Singleton	Kelas yang hanya dapat diinstansiasi menjadi 1 objek. Setiap yang membutuhkan layanan kelas ini akan mendapatkan instans objek yang sama. Kelas ini memiliki konstruktor yang hak aksesnya adalah private (atau protected dalam beberapa kasus). Konstruktor hanya dapat dipanggil dari dalam kelas.
Behavioral <i>Patterns</i>	Design pattern yang terkait dengan kelakuan (behavior)
Observer	Mempertegas ketergantungan hubungan one to many antara objek-

Nama Pattern	Panjelasan
	objek agar ketika satu objek merubah keadaan (status), maka semua objek yang memiliki ketergantungan melaporkan dan meng-update secara otomatis
Strategy	Berguna untuk memecah algoritma menjadi bagian-bagian kecil, yang dapat diganti dengan mudah
<i>Structural Patterns</i>	Pattern yang masuk jenis ini adalah yang terkait dengan masalah struktural.
Adapter	Pattern untuk menyesuaikan interface dari sebuah kelas menjadi interface yang lain
Facade	Pattern untuk memisahkan kode rumit

2.2.5. Class Diagram

Pembuatan *class diagram* akan dilakukan setelah proses analisa kode program yang menghasilkan *code dependency*. Sebuah ketergantungan (*dependencies*) mengandung tiga unsur yaitu sumber (*source*), target dan tipe ketergantungan (*dependency kind*) [37]. Pemetaan *source*, *dependency kind*, *target*, dan diskripsinya ada pada Tabel 2. 14.

Tabel 2. 14 Code dependency

Source	Dependency Kind	Target	Deskripsi
<i>class/interface</i>	<i>extends</i>	<i>class/interface</i>	<i>class/interface extends</i> (perluasan) terhadap <i>class/interface/</i> yang lain.
<i>class/interface</i>	<i>contains</i>	<i>class/interface</i>	<i>class/interface</i> pada <i>source</i> berisi sebuah <i>class/interface</i> yang lain.
<i>class/in</i>	<i>contains</i>	<i>field</i>	<i>class/interface/</i> pada <i>source</i> berisi

<i>Source</i>	<i>Dependency Kind</i>	<i>Target</i>	Deskripsi
<i>terface</i>			<i>field</i> dari <i>class</i> yang lain.
<i>method</i>	<i>returns</i>	<i>class/in terface</i>	<i>method</i> pada <i>source</i> mengembalikan sebuah nilai berdasarkan <i>class/interface</i> yang lain.
<i>method</i>	<i>has param</i>	<i>class/in terface</i>	<i>method</i> pada <i>source</i> mendeklarasikan sebuah parameter berdasarkan <i>class/interface</i> yang lain.
<i>method</i>	<i>throws</i>	<i>class/in terface</i>	<i>method</i> pada <i>source</i> mendeklarasikan <i>class/interface/</i> yang lain dalam mengirim/melempar clause-nya.
<i>method</i>	<i>calls</i>	<i>method s</i>	<i>method</i> pada <i>source</i> memanggil <i>method</i> pada <i>class</i> yang lain.
<i>method s</i>	<i>accesses</i>	<i>field</i>	<i>method</i> pada <i>source</i> mengakses <i>field</i> pada <i>class</i> yang lain.
<i>field</i>	<i>is of type</i>	<i>class/in terface</i>	<i>field</i> pada <i>source</i> didasarkan pada <i>class/interface</i> yang lain.
<i>any</i>	<i>references</i>	<i>class/in terface</i>	<i>remote</i> objek yang dipanggil oleh objek lain melalui <i>remote object refernces</i> .

Class *diagram* merupakan bagian dari *Unifed Modelling Language* (UML), UML merupakan bahasa pemodelan standar. *Class diagram* digunakan untuk menampilkan kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang dikembangkan [37]. *Class diagram* memberi kita gambaran tentang perangkat lunak dan relasi-relasi yang ada di dalamnya. Terdapat unsur-unsur dari *class diagram* yaitu atribur, operasi dan hubungan antar *class*.

Atribut dan operasi dari *class diagram* memiliki salah satu sifat dari tiga sifat yaitu *private*, *protected*, dan *public*. Hubungan pada *class diagram* digambarkan sebagai *associations*, *dependencies*, *agregation*, *realization*, *generalization*, dan multiplisasi relasi [37].

Atribut atau operasi yang bersifat *private* berarti tersembunyi dari kelas-kelas lain sehingga kelas-kelas lain tidak dapat mengakses, sehingga atribut atau operasi tersebut tidak dapat dipanggil dari luar kelas yang bersangkutan dan hanya bisa dipakai dalam kelas yang bersangkutan. Atribut atau operasi yang bersifat *private* mempunyai simbol gembok dalam gambarnya.

Atribut atau operasi yang bersifat *protected* hanya dapat diakses oleh anggota kelas yang bersangkutan serta kelas-kelas lain yang menjadi turunannya dalam hirarki warisan (*inheritance*). Atribut atau operasi yang bersifat *protected* mempunyai simbol kunci dalam gambarnya. Atribut atau operasi yang bersifat *public* dapat diakses oleh semua kelas yang ada dan dapat dipanggil oleh siapa saja.

Associations Relationship adalah hubungan statis antar *class*. *Associations* umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain atau *class* yang harus mengetahui ekstensi *class* lain. Bentuk notasi digambarkan dengan garis lurus yang menghubungkan antar *class*.

Dependencies Relationship adalah hubungan antar *class* dimana sebuah *class* memiliki ketergantungan pada *class* lainnya tetapi tidak sebaliknya. Bentuk notasi dari *dependencies relationship* digambarkan dengan garis putus-putus dengan ujung panah terbuka yang menghubungkan dengan *class* lain.

Realization Relationship merupakan hubungan antar *class* dimana sebuah *class* memiliki keharusan untuk mengikuti aturan yang ditetapkan *class* yang lain. *Realization* digunakan untuk menspesifikasikan hubungan antara sebuah *interface* dengan *class* yang mengimplementasikan *interface* tersebut. Bentuk notasi *realization relationship* digambarkan dengan garis panah putus-putus dengan ujung panah tertutup yang menghubungkan antar kelas.

Generalization Relationship merupakan relasi antar *class* yang menunjukkan hubungan warisan (*inheritance*). Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari *class* yang berbeda dalam hirarki pewarisnya. Bentuk notasi dari *generalization relationship* adalah garis panah dengan ujung panah tertutup yang menghubungkan *class* induk dengan *class* turunannya.

Nesting Relationship adalah relasi antar *class*, dimana sebuah *class* (*inner class*) didefinisikan didalam *class* lain (*outer class*). Bentuk notasi dari *nesting relationship* adalah garis dengan ujung berbentuk lingkaran yang mengarah pada *outer class*.

Multiplisitas relasi adalah jumlah banyaknya objek sebuah *class* yang berelasi dengan sebuah *objek* lain pada *class* lain yang berasosiasi dengan *class* tersebut, Tabel 2. 15 mendeskripsikan arti simbol dari multiplisitas [37].

Tabel 2. 15 Multiplisitas

Multiplisitas	Arti
*	Banyak
0	Tepat nol
1	Tepat satu
0..*	Nol atau lebih
1..*	Satu atau lebih
0..1	Nol atau satu

Pembuatan *class diagram* dimudahkan dengan pemetaan yang dihasilkan dari *dependency kind* dan *relationship* yang ada pada *class diagram*. Pemetaan tersebut ada pada Tabel 2.16.

Tabel 2. 16 Pemetaan *Dependency Kind* dan *Relationship Class Diagram*

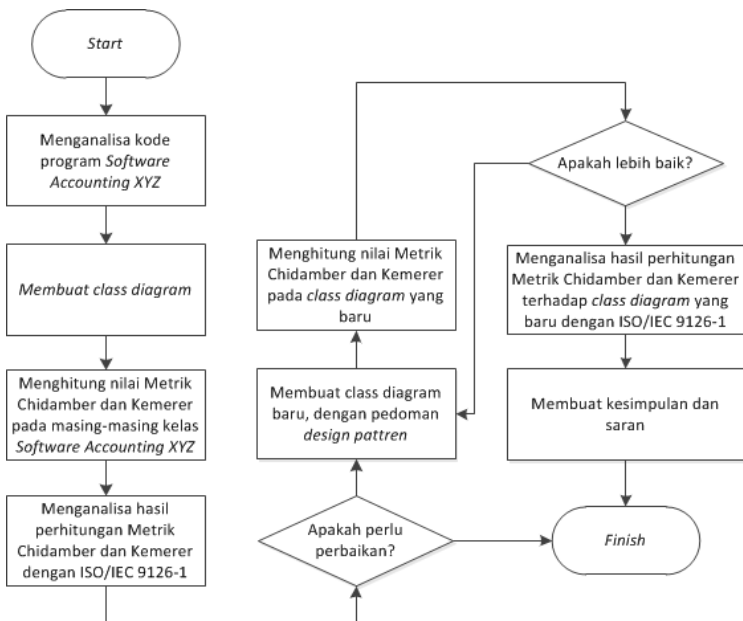
<i>Relationship</i>	<i>Deskripsi</i>	<i>Dependency Kind</i>
<i>Associations</i>	<i>Class</i> yang memiliki atribut	<i>Contains</i>

	berupa <i>class</i> lain atau <i>class</i> yang harus mengetahui ekstensi <i>class</i> lain	<i>Referencess</i> <i>Is Of Type</i>
<i>Dependencies</i>	Operasi suatu <i>class</i> yang menggunakan <i>class</i> lain atau <i>field</i> /atribut <i>class</i> lain	<i>Retruns</i>
		<i>Has Param</i>
		<i>Throws</i>
		<i>Calls</i>
		<i>Accesses</i>
<i>Nesting</i>	Sebuah <i>class</i> yang didefinisikan didalam <i>class (outer class)</i> lain.	
<i>Realization</i>	Sebuah <i>class</i> yang mengimplementasikan <i>interface</i>	<i>Implements (Interface)</i>
<i>Generalization</i>	<i>Class</i> yang menunjukkan hubungan warisan (<i>inheritance</i>) dengan <i>class</i> lainnya.	<i>Extendss</i>

BAB III METODOLOGI PENULISAN

Metodologi penulisan berisi tahapan-tahapan pengerjaan tugas akhir. Bagian terpenting yang digunakan sebagai panduan pengerjaan tugas akhir ini agar terarah, teratur dan sistematis.



Method Metodologi pengerjaan tugas akhir ini dapat dilihat pada Gambar 3. 1.



Gambar 3. 1 Metodologi Pengerjaan Tugas akhir

Pada metodologi pengerjaan tugas akhir ini terdapat tiga tahapan besar yaitu tahap perancangan, tahap implementasi dan tahap hasil dan pembahasan, detail dari tahapan-tahapan metodologi pengerjaan tugas akhir ini ada pada Tabel 3. 1.

Tabel 3. 1 Metodologi Pengerjaan Tugas akhir

INPUT	PROSES	OUTPUT
Tahap Perancangan		
Kode Program <i>Software Accounting XYZ</i>	Menganalisa kode program <i>Software Accounting XYZ</i>	<i>Code Dependency</i> dari masing-masing <i>package</i>
Kode Program <i>Software Accounting XYZ</i>	Membuat <i>class diagram</i>	<i>Class diagram</i> dari kode program <i>Software Accounting XYZ</i>
		
Tahap Implementasi		
<i>Code Dependency</i> dan <i>class diagram</i> dari <i>Software Accounting XYZ</i>	Menghitung nilai Metrik Chidamber dan Kemerer pada masing-masing <i>class Software Accounting XYZ</i>	Nilai WMC, DIT, NOC, CBO, RFC, dan LCOM
		
Tahap Pembahasan Hasil		
Nilai WMC, DIT, NOC, CBO, RFC, LCOM dan ISO/IEC 9126-1	Menganalisa hasil perhitungan Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-1	Kualitas <i>Software Accounting XYZ</i> berdasarkan ISO/IEC 9126-1
<i>Class Diagram</i> lama dan <i>Design Pattern</i>	Membuat <i>class diagram</i> baru	<i>Class diagram</i> baru
<i>Class diagram</i> baru	Menghitung nilai Metrik Chidamber dan Kemerer pada <i>class</i>	Nilai WMC, DIT, NOC, CBO, RFC, dan LCOM yang

INPUT	PROSES	OUTPUT
	<i>diagram</i> yang baru	baru
Nilai WMC, DIT, NOC, CBO, RFC, LCOM yang baru dan ISO/IEC 9126-1	Menganalisa hasil perhitungan Metrik Chidamber dan Kemerer yang baru dengan ISO/IEC 9126-1	Kualitas <i>Class Diagram</i> baru berdasarkan ISO/IEC 9126-1
Hasil perbandingan <i>class diagram</i> baru dan <i>class diagram</i> lama	Membuat Kesimpulan dan saran dari penulisan ini	Kesimpulan dan saran.

3.1. Tahap Perancangan

Tahap perancangan meliputi dua tahapan yaitu analisa kode program dan pembuatan *class diagram*.

3.1.1. Analisa Kode Program

Analisa kode program *Software Accounting XYZ* dengan memetakan *code* yang ada dan menemukan *dependencies* masing-masing *code*. Sehingga *code Software Accounting XYZ* akan lebih mudah untuk diukur nilainya terhadap Metrik Chidamber dan Kemerer.

3.1.2. Pembuatan *Class Diagram*

Pada tahapan perencanaan ini akan dibuat desain kode berupa *class diagram* dari kode program *Software Sccounting XYZ* untuk memudahkan perhitungan Metrik Chidamber dan Kemerer. Pembuatan *class diagram* dibantu dengan *Software Enterprise Architect 11*.

Class diagram dibuat berdasarkan hasil analisa dari tahapan sebelumnya yaitu analisa *code program* yang menghasilkan *code dependency*. *Code dependency* dapat memudahkan pembuatan

class diagram dengan bantuan pemetaan antara relasi pada *class diagram* dengan *dependency kind*.

3.2. Tahap Implementasi

Tahap implementasi meliputi perhitungan Metrik Chidamber dan Kemerer, yang terdiri dari Metrik WMC, DIT, NOC, CBO, RFC, dan LCOM.

Tahapan implementasi ini melakukan perhitungan Metrik Chidamber dan Kemerer. *Code Dependency* dan *Class diagram* dapat menjadi acuan untuk menghitung nilai Metrik Chidamber dan Kemerer. Setiap *class* akan diukur nilai Metrik Chidamber dan Kemerer-nya dan selanjutnya dihitung *average* (rata-rata) nya.

Pada Pembahasan pada Tabel 2. 4 telah ada patokan nilai yang dapat dijadikan parameter dalam pengukuran ditahapan ini.

3.3. Tahap Pembahasan Hasil

Tahap pembahasan hasil meliputi empat tahapan yaitu Analisa perhitungan Metrik Chidamber dan Kemerer, membuat *class diagram* baru, menghitung nilai Metrik Chidamber dan Kemerer, dan membuat kesimpulan serta saran.

3.3.1. Analisa Perhitungan Metrik Chidamber dan Kemerer

Tahapan pembahasan hasil yang pertama adalah menganalisa hasil perhitungan Metrik Chidamber dan Kemerer. Setelah mendapat nilai WMC, DIT, NOC, CBO, RFC, dan LCOM maka, nilai tersebut dapat menunjukkan kualitas *Software Accounting XYZ* berdasarkan Tabel 2. 8.

3.3.2. Membuat *Class Diagram* Baru

Tahapan pembahasan hasil yang kedua adalah membuat *class diagram* baru. Tahapan ini dilakukan untuk memperbaiki nilai metrik yang kurang bagus atau nilai metrik yang berhubungan dengan kriteria atau sub-kriteria ISO/IEC 9126-1. Pembuatan *class diagram* baru akan dilakukan berdasarkan pedoman *design pattern* dan *refactoring*. Pembuatan *class diagram* ini menggunakan bantuan *software* Enterprise Architect 11.

3.3.3. Perhitungan Nilai Metrik Chidamber dan Kemerer

Tahapan pembahasan hasil yang ketiga adalah melakukan perhitungan Metrik Chidamber dan Kemerer pada *class diagram* yang baru. *Class diagram* yang baru dapat menjadi acuan untuk menghitung nilai Metrik Chidamber dan Kemerer. Perhitungan akan dilakukan dengan menggunakan parameter nilai yang ada pada Tabel 2. 4. Setiap *class* akan diukur lagi nilai Metrik Chidamber dan Kemerer-nya dan selanjutnya dihitung *average* (rata-rata) nya.

Korelasi Metrik Chidamber dan Kemerer dengan ISO/IEC 9126-1 juga banyak diteliti pada penulisan lain dan menghasilkan hasil yang sama yaitu berbanding terbalik kecuali Metrik LCOM seperti yang sudah dijabarkan sebelumnya pada Tabel 2. 11 .

3.3.4. Pembuatan Kesimpulan dan Saran

Tahapan terakhir adalah pembuatan kesimpulan dan saran dari pengerjaan tugas akhir ini berdasarkan hasil yang diperoleh dari tahapan sebelumnya.

(Halaman ini sengaja dikosongkan)

BAB IV

PERANCANGAN KONSEPTUAL

Bab ini menjelaskan tentang perancangan penelitian dalam tugas akhir ini, yang meliputi: analisa kode program *Software Accounting XYZ* dan pembuatan *class diagram* berdasarkan *source code* dari *Software Accounting XYZ*.

4.1. Gambaran Umum *Software Accounting XYZ*

Gambaran umum *Software Accounting XYZ* membahas tentang *package* dan *class* secara umum yang ada pada kode program *Software Accounting XYZ*.

Package yang ada pada *Software Accounting XYZ* berjumlah lima yaitu *package appLayer*, *package dataLayer*, *package appLayer.taxRelated* dan *package appLayerTransactionRelated*.

Jumlah *class* dalam *Software Accounting XYZ* adalah 152 *class*. Setiap *package* memiliki jumlah *class* yang berbeda-beda. *Class* tersebut ada yang merupakan *superclass* (*parentclass*) ada yang merupakan *subclass* (*childclass*).

Package dataLayer yang mempunyai *class* berjumlah 14 *class* yaitu:

1. DB
2. HBCI
3. HBCIImporter
4. Messages
5. OBDXImporter
6. fileImporters
7. fileUtils
8. gnuPG
9. ProcessStreamReader (*Subclass dari class gnuPG*)
10. invalidImportFormatException
11. moneyplexImporter

12. prematureException
13. prepareDB
14. starmoneyImporter

Package appLayer yang mempunyai *class* berjumlah 45 *class* yaitu:

1. AccountNotFoundException
2. IItemListener
3. IcontextProvider
4. Messages
5. PaymentMethodNotFoundException
6. RoleNotFoundException
7. account
8. accountException
9. accountsList
10. application
11. asset
12. assets
13. bookingCellModifier
14. bookingContentProvider
15. bookingLabelProvider
16. client
17. configs
18. contact
19. contacts
20. contextComposite
21. document
22. documents
23. elementNotFoundException
24. entry
25. accountColumn (*Subclass dari class entry*)
26. Column (*Subclass dari class entry*)
27. columnNotFoundException (*Subclass dari class entry*)
28. dateColumn (*Subclass dari class entry*)
29. doubleColumn (*Subclass dari class entry*)
30. referenceColumn (*Subclass dari class entry*)

31. taxColumn *Subclass* dari class entry
32. undefinedNumberException (*Subclass* dari class entry)
33. entryNotInThisAccountException
34. item
35. itemTableList
36. itemTextList
37. items
38. moneyTransfer
39. pageActivator
40. placeholderManager
41. product
42. products
43. transactionContentProvider
44. typeNotFoundException
45. utils

Package appLayer.taxRelated yang mempunyai *class* berjumlah 7 *class* yaitu:

1. IRSoffice
2. IRSoffices
3. taxRelated.state
4. taxRelated.states
5. taxRelated.tax
6. taxRelated.taxList
7. taxRelated.taxNotFoundException

Package appLayer.transactionRelated yang mempunyai *class* berjumlah 9 *class* yaitu:

1. cancelation
2. creditmemo
3. invoice
4. offer
5. personalDrawing
6. reminder
7. transaction
8. transactionTypeNotFoundException

9. transactions

Package GUILayer yang mempunyai *class* berjumlah 99 *class* yaitu:

1. CComboBoxCellEditor
2. HBCIcallbackDialog
3. HSQLLockCheck
4. MainWindow
5. Messages
6. OOoCheck
7. TimeCellEditor
8. VATannouncementWindow
9. aboutWindow
10. accountDefinitionWindow
11. ContentProvider (*Subclass* dari *class* accountDefinition)
12. accountSelectionChangedListener (*Subclass* dari *class* accountDefinition)
13. accountingEditWindow
14. assetWindow
15. ContentProvider (*Subclass* dari *class* assetWindow)
16. assetSelectionChangedListener (*Subclass* dari *class* assetWindow)
17. balanceWindow
18. browserWindow
19. communityWindow
20. configWindow
21. contactsWindow
22. ContentProvider (*Subclass* dari *class* contactsWindow)
23. contactSelectionChangedListener (*Subclass* dari *class* contactsWindow)
24. databaseConnectivityTest
25. databaseDriverNameTestCase
26. designerWindow
27. documentsWindow
28. documentCellModifier (*Subclass* dari *class*

- documentsWindow)
- 29. documentLabelProvider (*Subclass* dari *class* documentsWindow)
- 30. documentsList (*Subclass* dari *class* documentsWindow)
- 31. entryDetailWindow
- 32. mainDirTestCase
- 33. newAccountingWizard
- 34. newAccountingWizardAdd
- 35. entryList (*Subclass* dari *class* newAccountingWizardAdd)
- 36. newAccountingWizardImport
- 37. newManualAccountingWizard
- 38. newTransactionSelectItemDetails
- 39. newTransactionSelectTransactionDetails
- 40. entryList (*Subclass* dari *class* newTransactionSelectTransactionDetails)
- 41. transactionCellModifier (*Subclass* dari *class* newTransactionSelectTransactionDetails)
- 42. transactionLabelProvider (*Subclass* dari *class* newTransactionSelectTransactionDetails)
- 43. newTransactionWizard
- 44. MailAuthenticator (*Subclass* dari *class* newTransactionWizard)
- 45. mailThread (*Subclass* dari *class* newTransactionWizard)
- 46. newTransactionWizardPreview
- 47. newTransactionWizardPrintAndSave
- 48. numbersWindow
- 49. ContentProvider (*Subclass* dari *class* numbersWindow)
- 50. transactionSelectionChangedListener (*Subclass* dari *class* numbersWindow)
- 51. passwordDialog
- 52. productsWindow
- 53. ContentProvider (*Subclass* dari *class* productsWindow)

54. productSelectionChangedListener (*Subclass* dari *class* productsWindow)
55. reportWizard
56. reportWizardSelect
57. reportWizardShow
58. tAccount (*Subclass* dari *class* reportWizardShow)
59. tAccountLine (*Subclass* dari *class* reportWizardShow)
60. selectionAdapterWithParent
61. settingsWindow
62. standardPositiveTestResult
63. tableCheck
64. tableWindow
65. taxesWindow
66. ContentProvider (*Subclass* dari *class* taxesWindow)
67. taxSelectionChangedListener (*Subclass* dari *class* taxesWindow)
68. templateCheck
69. testCase
70. testResult
71. testTest
72. timetrackerImportWindow
73. todoBalanceSelectionAdapter
74. todoWindow
75. transactionSelectionAdapter
76. viewerWindow
77. warningFocusAdapter

4.2. Code Dependencies

Pada pembahasan bab ini akan dijelaskan analisa *code* yang menghasilkan *code dependency* dengan acuan referensi pada Tabel 2. 14.

4.2.1. Extends Dependencies

Extend dependency merupakan *class* yang menjadi perluasan dari *class* lain. *Extend dependency* dapat diketahui dari kode program seperti pada Gambar 4. 1.

```
package appLayer.transactionRelated;
import java.sql.ResultSet;

public class cancelation extends transaction {
    public cancelation(transactions parent) {
        super(parent);
        // TODO Auto-generated constructor stub
    }
}
```

Class cancelation extends (perluasan dari) class transaction

Gambar 4. 1 Kode Program *Extends Dependencies* (Class *cancelation*)

4.2.2. Contains Dependency

Contains dependency merupakan sebuah *class* yang berisi *class* lain. *Contains dependency* dapat diketahui ketika suatu *class* berisi *class* lain, pada Gambar 4. 2 dijelaskan bahwa *class* *moneyTransfer* pada *package* *appLayer* berisi *class* *HBCI* yang ada pada *package* *dataLayer*.

```
public class moneyTransfer {
    private HBCI hbcI=null;
    public moneyTransfer() {
        hbcI=HBCI.getInstance();
    }
}
```

Class moneyTransfer berisi class HBCI yang ada pada package dataLayer

Gambar 4. 2 Kode Program *Contains Dependencies* (Class *moneyTransfer*)

4.2.3. Returns Dependency

Returns dependency merupakan sebuah *method* yang mengembalikan sebuah nilai atribut dari sebuah *class*. Pada Gambar 4. 3 menjelaskan bahwa *method* *getTransactions* pada *class* *client* di *package* *appLayer* mengembalikan sebuah nilai

transactionController berdasarkan *class* transaction pada *package* appLayer.transactionRelated.

```

package appLayer;

import java.text.DecimalFormat;

/**
 * This refers to the client of the application.
 * Customers are "contacts"
 */
public class client {
    public static transactions getTransactions() {
        if (transactionController==null) {
            transactionController=new transactions();
        }
        return transactionController;
    }
}

```

Method getTransactions
mengembalikan sebuah
atribut
transactionController
berdasarkan *class*
transactions

Gambar 4. 3 Kode Program Retrurns Dependency (Class moneyTransfer)

4.2.4. Has Param Dependency

Has param dependency merupakan sebuah *method* yang mendeklarasikan sebuah parameter berdasarkan sebuah *class*. Pada Gambar 4. 4 menjelaskan bahwa *method* itemTableList pada *class* itemTableList di *package* appLayer mendeklarasikan parameter transaction berdasarkan *class* transaction pada *package* appLayer.transactionRelated.

```

package appLayer;

import ag.ion.noa.text.XInterfaceObjectSelection;
import appLayer.transactionRelated.transaction;

public class itemTableList {
    private Shell shell = null;
    private ITextDocument document = null;
    private transaction theTransaction = null;
    private placeholderManager placeHolderManager = null;
    private ITextField[] placeholderFields = null;
}

```

Class itemTableList
mendeklarasikan
parameter transaction
berdasarkan *class*
transactions

Gambar 4. 4 Kode Program Has Param Dependency (Class itemTableList)

4.2.5. Throws Dependency

Throws dependency merupakan suatu *method* yang mendeklarasikan sebuah *class* dalam mengirim *clause*-nya. Pada Gambar 4. 5 menjelaskan bahwa *method* createDB pada *class* prepareDB di *package* dataLayer mendeklarasikan *class* AccountNotFoundException pada *package* appLayer dalam mengirim/melempar SQLException.

```

package dataLayer;

import java.io.BufferedReader;

public class prepareDB {
    public void createDB(Statement stmt, List listStatus) throws SQLException,
        AccountNotFoundException {
        String sql;
    }
}

```

Class prepareDB mengirim SQLException dengan mendeklarasikan class AccountNotFoundException

Gambar 4. 5 Kode Program Has Param Dependency (Class prepareDB)

4.2.6. Calls Dependency

Calls dependency merupakan suatu *method* memanggil *method* lainnya. Pada Gambar 4. 6 menjelaskan bahwa *method* onProductChange pada *class* pageActivator di *package* appLayer memanggil (calls) *method* checkPageComplete pada *class* newTransactionSelectedItemDetails di *package* GUILayer.

```

package appLayer;

import java.util.Vector;

class pageActivator implements appLayer.IItemListener {
    private newTransactionSelectItemDetails parentPage=null;

    public pageActivator(newTransactionSelectItemDetails parentPage) {
        this.parentPage=parentPage;
    }

    public void onProductChange(product newProduct) {
        // the wizard page can not be completed with the default product (<please
        if (parentPage!=null) {
            parentPage.checkPageComplete();
        }
    }
}

```

Methods onProductChange memanggil class checkPageComplete

Gambar 4. 6 Kode Program Calls Dependency (Class pageActivator)

4.2.7. Accesses Dependency

Accesses dependency adalah sebuah *method* yang mengakses sebuah *field* pada *class* lain. Pada Gambar 4. 7 menjelaskan bahwa *method* `performMoneyTransfer` pada *class* `moneyTransfer` di *package* `appLayer` mengakses *field* `RECEIVE` pada *class* `transaction` di *package* `appLayer.transactionRelated`.

```
package appLayer;
import org.kapott.hbci.GV.HBCIJob;

public class moneyTransfer {
    private HBCI hbci=null;

    public moneyTransfer() {
        hbci=HBCI.getInstance();
    }
    public void performMoneyTransfer(transaction theTransaction) {
        if (theTransaction.getCashFlow()==cashFlow.RECEIVE) {
            Konto own=new Konto(configs.getBankCode(), configs.getAccountCode());
            Konto other=new Konto(theTransaction.getRecipient().getBankCode(), theTransaction.
```

Methods
performMoneyTransfer
mengakses *fields* **RECEIVE**
pada *class* **transaction**.

Gambar 4. 7 Kode Program Access Dependency (Class moneyTransfer)

4.2.8. Is of Type Dependency

Is of type dependency adalah adanya *field* pada sebuah *class* didasarkan pada sebuah *class* lain. Pada Gambar 4. 8 menjelaskan bahwa *field* `hbci` pada *class* `moneyTransfer` di *package* `appLayer` didasarkan pada *class* `HBCI` di *package* `dataLayer`.

```
package appLayer;
import org.kapott.hbci.GV.HBCIJob;

public class moneyTransfer {
    private HBCI hbci=null;

    public moneyTransfer() {
        hbci=HBCI.getInstance();
```

Field hbci didasarkan pada
class HBCI.

Gambar 4. 8 Kode Program Access Dependency (Class moneyTransfer)

4.2.9. References Dependency

References Code dependency adalah suatu *methods/class/field* meremote sebuah objek dengan memanggil objek lain melalui *remote object references*. Pada Gambar 4. 9 menjelaskan bahwa *method* `importKTimeTracker` pada *class* `fileImporters` di *package* `dataLayer` me-Remote *class* `item` pada *package* `appLayer`.

```
import appLayer.item;
...

public class fileImporters {
    public static void importKTimeTracker(String importFileName, StatusLineManager
        File importFile = new File(importFileName);
        if (!importFile.exists()) {
            }
            item i=inv.addItem();
            i.setDescription(taskName);
            i.setQuantity(Double.valueOf(elements[elements.length-1]));
        }
    }
}
```

Method fileImporters meremote dengan memanggil class item.

Gambar 4. 9 Kode Program References Dependency (Class fileImporters)

4.3. Package dataLayer

Bagian ini membahas tentang *dependency code* antar *class* di *package* `dataLayer` dan juga *dependency code* antara *class* di *package* `dataLayer` dengan *class* di *package* lain. Tabel 4. 1 adalah *dependency code* antar *class* di `dataLayer`. Tabel 4. 2 adalah *dependency code* *package* `dataLayer` dengan *package* `appLayer`. Tabel 4. 3 adalah *dependency code* *package* `dataLayer` dengan *package* `appLayer.transactionRelated`. Tabel 4. 4 adalah *dependency code* *package* `dataLayer` dengan *package* `GUILayer`.

4.3.1. Code Dependency Antar Class dataLayer

Terdapat 38 *Code Dependency* yang ada pada *package* dataLayer, *dependency kind* terbanyak adalah *calls* dengan jumlah 31, *class* yang paling sering diakses adalah *class* DB.

Tabel 4. 1 *Code Dependency Antar Class dataLayer*

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	HBCIImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	HBCI	<i>getInstance()</i>
2	HBCIImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	HBCI	<i>newJob(String)</i>
3	HBCIImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	HBCI	<i>execute(String)</i>
4	HBCIImporter	<i>run(IPressMonitor)</i>	<i>references</i>	HBCI	
5	HBCIImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	DB	<i>getConnection()</i>
6	HBCIImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	DB	<i>insert(String, HashMap<String, String>)</i>
7	OBDXImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	Messages	<i>getString(String)</i>
8	OBDXImporter	<i>getAccountholder()</i>	<i>throws</i>	<i>prematuereException</i>	
9	OBDXImporter	<i>getAccountholder()</i>	<i>calls</i>	<i>prematuereException</i>	<i>prematuereException(String)</i>
10	OBDXImporter	<i>getBankaccount()</i>	<i>throws</i>	<i>prematuereException</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
11	OBDXImporter	<i>getBankaccount()</i>	<i>calls</i>	<i>prematureException</i>	<i>prematureException(String)</i>
12	OBDXImporter	<i>getBankcode()</i>	<i>throws</i>	<i>prematureException</i>	
13	OBDXImporter	<i>getBankcode()</i>	<i>calls</i>	<i>prematureException</i>	<i>prematureException(String)</i>
14	OBDXImporter	<i>getUsage()</i>	<i>throws</i>	<i>prematureException</i>	
15	OBDXImporter	<i>getUsage()</i>	<i>calls</i>	<i>prematureException</i>	<i>prematureException(String)</i>
16	OBDXImporter	<i>getValue()</i>	<i>throws</i>	<i>prematureException</i>	
17	OBDXImporter	<i>getValue()</i>	<i>calls</i>	<i>prematureException</i>	<i>prematureException(String)</i>
18	OBDXImporter	<i>run(IProgressMonitor)</i>	<i>calls</i>	<i>fileUtils</i>	<i>readFileAsString(String)</i>
19	fileImporters	<i>importKTimeTracker(String, StatusLineManager, Shell)</i>	<i>calls</i>	<i>fileUtils</i>	<i>parseCSVLineComma(String)</i>
20	fileImporters	<i>importTaskCoach(String, StatusLineManager,</i>	<i>calls</i>	<i>fileUtils</i>	<i>readFileAsString(String)</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		<i>Shell)</i>			
21	fileImporters	<i>importKTimeTracker(String, StatusLineManager, Shell)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
22	gnuPG	<i>createTempFile(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
23	gnuPG	<i>runGnuPG(String, String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
24	moneyplexImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>fileUtils</i>	<i>readFileAsString(String)</i>
25	moneyplexImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>DB</i>	<i>getConnection()</i>
26	moneyplexImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>DB</i>	<i>insert(String, HashMap<String, String>)</i>
27	prematureException	<i>prematureException(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
28	prepareDB	<i>createDB(Statement, List)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
29	prepareDB	<i>importAccountsFromXML()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
30	prepareDB	<i>createDB(Statement,</i>	<i>calls</i>	<i>DB</i>	<i>getDBVersionString()</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		<i>List</i>)			
31	prepareDB	<i>createDB(Statement, List)</i>	<i>calls</i>	<i>DB</i>	<i>getDatabaseOffset()</i>
32	prepareDB	<i>createDB(Statement, List)</i>	<i>calls</i>	<i>DB</i>	<i>insertRaw(String, HashMap<String, String>)</i>
33	prepareDB	<i>importAccountsFromXML()</i>	<i>calls</i>	<i>DB</i>	<i>getConnection()</i>
34	starmoneyImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>DB</i>	<i>getConnection()</i>
35	starmoneyImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>DB</i>	<i>insert(String, HashMap<String, String>)</i>
36	starmoneyImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>invalidImportFormatException</i>	<i>invalidImportFormatException()</i>
37	starmoneyImporter	<i>run(IPressMonitor)</i>	<i>references</i>	<i>invalidImportFormatException</i>	
38	starmoneyImporter	<i>run(IPressMonitor)</i>	<i>calls</i>	<i>fileUtils</i>	<i>parseCSVLineSemicolon(String)</i>

4.3.2. Code Dependency Package dataLayer ke Package appLayer

Terdapat 35 Code Dependency yang ada pada package dataLayer ke package appLayer, dependency kind terbanyak adalah *calls* dengan jumlah 10, class yang paling sering diakses adalah class configs.

Tabel 4. 2 Code Dependency Package dataLayer ke Package applayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	DB	getDatabaseOffset()	<i>calls</i>	configs	getDatabaseType()
2	DB	getIdentifierQuote()	<i>calls</i>	configs	getDatabaseType()
3	DB	getLastInsertID(Statement)	<i>calls</i>	configs	getDatabaseType()
4	HBCI	initPassport()	<i>calls</i>	client	getDataPath()
5	HBCI	initPassport()	<i>calls</i>	configs	getCtAPI()
6	HBCI	initPassport()	<i>calls</i>	configs	shallUseCardReaderPINPad()
7	HBCIimport er	run(IProgressMonitor)	<i>calls</i>	client	getClient()
8	HBCIimport er	run(IProgressMonitor)	<i>calls</i>	configs	getBankCode()
9	HBCIimport er	run(IProgressMonitor)	<i>calls</i>	configs	getAccountCode()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
10	OBDXImpor ter	run(IProgressMonitor)	<i>calls</i>	utils	round(Double, int)
11	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>calls</i>	client	getTransactions()
12	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>calls</i>	client	getTransactions()
13	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>calls</i>	item	setDescription(String)
14	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>calls</i>	item	setQuantity(Double)
15	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>calls</i>	item	setDescription(String)
16	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>calls</i>	item	setQuantity(Double)
17	gnuPG	gnuPG.()	<i>calls</i>	configs	getGPGPath()
18	moneyplexI mporter	run(IProgressMonitor)	<i>calls</i>	client	getClient()
19	prepareDB	createDB(Statement, List)	<i>calls</i>	account	getID()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
20	prepareDB	importAccountsFromXML()	<i>calls</i>	account	getTypeIDForString(String)
21	prepareDB	importAccountsFromXML()	<i>calls</i>	account	getSubAccountsTypeIDForString(String)
22	prepareDB	createDB(Statement, List)	<i>calls</i>	account	getStandardVATCreditAccount()
23	prepared	createDB(Statement, List)	<i>calls</i>	account	getStandardVATDebitAccount()
24	prepareDB	createDB(Statement, List) c	<i>calls</i>	application	getVersionString()
25	prepareDB	createDB(Statement, List)	<i>calls</i>	client	getClient()
26	prepareDB	createDB(Statement, List)	<i>calls</i>	client	getAccounts()
27	prepareDB	importAccountsFromXML()	<i>calls</i>	client	getClient()
28	prepareDB	createDB(Statement, List)	<i>calls</i>	configs	getDatabaseType()
29	prepareDB	createDB(Statement, List)	<i>calls</i>	configs	getSampleValue(String, int)
30	prepareDB	createDB(Statement, List)	<i>calls</i>	contacts	getRoleIDForString(String)
31	starmoneyIm porter	run(IProgressMonitor)	<i>calls</i>	client	getClient()
32	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>references</i>	item	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
33	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	references	item	
34	prepareDB	createDB(Statement, List)	references	RoleNotFou ndException	
35	prepareDB	createDB(Statement, List)	throws	AccountNot FoundExcep tion	

4.3.3. Code Dependency Package dataLayer ke Package appLayer.transactionRelated

Terdapat 8 *Code Dependency* yang ada pada *package* dataLayer ke *package* appLayer.transactionRelated, *dependency kind* calls berjumlah empat dan *dependency kind* references berjumlah empat, *class* yang paling sering diakses adalah *class* transactions.

Tabel 4. 3 Code Dependency Package dataLayer ke Package appLayer.transactionRelated

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	calls	transactions	getByTypeName(String)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
2	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>calls</i>	transactions	getByTypeName(String)
3	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>references</i>	transaction	
4	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>references</i>	transaction	
5	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>references</i>	transactionT ypeNotFoun dException	
6	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>references</i>	transactionT ypeNotFoun dException	
7	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	<i>calls</i>	transactions	getByTypeName(String)
8	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	<i>calls</i>	transactions	getByTypeName(String)

4.3.4. Code Dependency Package dataLayer ke Package GUILayer

Terdapat 7 Code Dependency yang ada pada package dataLayer ke package GUILayer, dependency kind terbanyak adalah calls dengan jumlah 4, class yang paling sering diakses adalah class newTransactionWizard.

Tabel 4. 4 Code Dependency Package dataLayer ke Package GUILayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	calls	newTransactionWizard	newTransactionWizard()
2	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)	references	newTransactionWizard	
3	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	calls	newTransactionWizard	newTransactionWizard()
4	fileImporters	importTaskCoach(String, StatusLineManager, Shell)	references	newTransactionWizard	
5	HBCI	HBCI()	calls	HBCIcallbackDialog	HBCIcallbackDialog()
6	HBCI	HBCI()	references	HBCIcallbackDialog	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
7	moneyplexI mporter	run(IProgressMonitor)	<i>calls</i>	Messages	getString(String)

4.4. Package appLayer

Bagian ini membahas tentang *dependency code* antar *class* di *package* appLayer dan juga *dependency code* antara *class* di *package* appLayer dengan *class* di *package* lain. Tabel 4. 5 adalah *dependency code* antar *class* di *package* appLayer. Tabel 4. 6 adalah *dependency code* *package* appLayer dengan *package* dataLayer. Tabel 4. 7 adalah *dependency code* *package* appLayer dengan *package* appLayer.taxRelated. Tabel 4. 8 adalah *dependency code* *package* appLayer dengan *package* appLayer.transactionRelated. Tabel 4. 9 adalah *dependency code* *package* appLayer dengan *package* GUILayer.

4.4.1. Code Dependency Antar Class appLayer

Terdapat 378 *Code Dependency* yang ada pada *package* appLayer, *dependency kind* terbanyak adalah *calls* dengan jumlah 168, *class* yang paling sering diakses adalah *class* entry.

Tabel 4. 5 *Code Dependency* Antar *Class* appLayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	ItemListener	<i>onProductChange(product)</i>	<i>has param</i>	<i>product</i>	
2	account	<i>account{...}</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
3	account	<i>getSubAccountsTypeIDForString(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
4	account	<i>getTypeIDForString(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
5	account	<i>isEmpty()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
6	account	<i>setReferTo(int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
7	account	<i>setType(int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
8	account		<i>contains</i>	<i>accountsList</i>	
9	account	<i>account(accountsList, int, String, String, int, int)</i>	<i>has param</i>	<i>accountsList</i>	
10	account	<i>delete()</i>	<i>calls</i>	<i>accountsList</i>	<i>signalChange(account)</i>
11	account	<i>parent (field)</i>	<i>is of type</i>	<i>accountsList</i>	
12	account	<i>save()</i>	<i>calls</i>	<i>accountsList</i>	<i>signalChange(account)</i>
13	account	<i>getSubAccountsTypeIDForString(String)</i>	<i>throws</i>	<i>typeNotFoundException</i>	
14	account	<i>getSubAccountsTypeIDForString(String)</i>	<i>calls</i>	<i>typeNotFoundException</i>	<i>typeNotFoundException (String)</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
15	account	<i>getSubAccountsTypeIDForString(String)</i>	<i>calls</i>	<i>typeNotFoundE xception</i>	<i>typeNotFoundException(String)</i>
16	account	<i>getTypeIDForString(String)</i>	<i>throws</i>	<i>typeNotFoundE xception</i>	
17	account	<i>getTypeIDForString(String)</i>	<i>calls</i>	<i>typeNotFoundE xception</i>	<i>typeNotFoundException(String)</i>
18	account	<i>isEmpty()</i>	<i>references</i>	<i>typeNotFoundE xception</i>	
19	account	<i>setType(int)</i>	<i>throws</i>	<i>typeNotFoundE xception</i>	
20	account	<i>setType(int)</i>	<i>calls</i>	<i>typeNotFoundE xception</i>	<i>typeNotFoundException(String)</i>
21	account	<i>setType(String)</i>	<i>references</i>	<i>typeNotFoundE xception</i>	
22	account	<i>setReferTo(int)</i>	<i>throws</i>	<i>accountExceptio n</i>	
23	account	<i>setReferTo(int)</i>	<i>calls</i>	<i>accountExceptio n</i>	<i>accountException(String)</i>
24	account	<i>addEntry(entry)</i>	<i>has param</i>	<i>entry</i>	
25	account	<i>entries</i>	<i>references</i>	<i>entry</i>	
26	account	<i>getBalance()</i>	<i>calls</i>	<i>entry</i>	<i>isCreditSide(account)</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
27	account	<i>getBalance()</i>	<i>calls</i>	<i>entry</i>	<i>getValue()</i>
28	account	<i>getBalance()</i>	<i>references</i>	<i>entry</i>	
29	account	<i>getEntries()</i>	<i>references</i>	<i>entry</i>	
30	account	<i>save()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
31	account	<i>getBalance()</i>	<i>references</i>	<i>entryNotInThisAccountException</i>	
32	accountsList	<i>addPleaseSelectAccount()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
33	accountsList	<i>addPleaseSelectAccount()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
34	accountsList	<i>getAccountForCode(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
35	accountsList	<i>getAccountForID(int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
36	accountsList	<i>getAccountForListID(int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
37	accountsList	<i>getAccountsFromDatabase()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
38	accountsList	<i>addPleaseSelectAccount()</i>	<i>references</i>	<i>typeNotFoundException</i>	
39	accountsList	<i>fillFromDB()</i>	<i>references</i>	<i>typeNotFoundException</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
40	accountsList	<i>getAccountsFromDatabase()</i>	<i>references</i>	<i>typeNotFoundException</i>	
41	accountsList	<i>getAccountForCode(String)</i>	<i>throws</i>	<i>AccountNotFoundException</i>	
42	accountsList	<i>getAccountForCode(String)</i>	<i>calls</i>	<i>AccountNotFoundException</i>	<i>AccountNotFoundException(String)</i>
43	accountsList	<i>getAccountForID(int)</i>	<i>throws</i>	<i>AccountNotFoundException</i>	
44	accountsList	<i>getAccountForID(int)</i>	<i>calls</i>	<i>AccountNotFoundException</i>	<i>AccountNotFoundException(String)</i>
45	accountsList	<i>getAccountForListID(int)</i>	<i>throws</i>	<i>AccountNotFoundException</i>	
46	accountsList	<i>getAccountForListID(int)</i>	<i>calls</i>	<i>AccountNotFoundException</i>	<i>AccountNotFoundException(String)</i>
47	accountsList	<i>getBankAccount()</i>	<i>throws</i>	<i>AccountNotFoundException</i>	
48	accountsList	<i>getEntriesFromDatabase()</i>	<i>references</i>	<i>AccountNotFoundException</i>	
49	accountsList	<i>getImportEntries()</i>	<i>references</i>	<i>AccountNotFoundException</i>	
50	accountsList	<i>getLiabilitiesAccount()</i>	<i>throws</i>	<i>AccountNotFou</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
				<i>ndException</i>	
51	accountsList	<i>getPersonalDrawAccount()</i>	<i>throws</i>	<i>AccountNotFou ndException</i>	
52	accountsList	<i>getReceivablesAccount()</i>	<i>throws</i>	<i>AccountNotFou ndException</i>	
53	accountsList	<i>getRevenuesAccount()</i>	<i>throws</i>	<i>AccountNotFou ndException</i>	
54	accountsList	<i>getStandardVATCreditAccount()</i>	<i>throws</i>	<i>AccountNotFou ndException</i>	
55	accountsList	<i>getStandardVATDebitAccount()</i>	<i>throws</i>	<i>AccountNotFou ndException</i>	
56	accountsList	<i>updateWithAccountTotal (String, Statement, HashMap<account, Double>, String, boolean)</i>	<i>references</i>	<i>AccountNotFou ndException</i>	
57	accountsList	<i>fillFromDB()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
58	accountsList	<i>getCompletePeriodEnd()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
59	accountsList	<i>getCompletePeriodStart())</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
60	accountsList	<i>getEntriesFromDatabase()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
61	accountsList	<i>getImportEntries()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
62	accountsList	<i>getImportEntries()</i>	<i>calls</i>	<i>client</i>	<i>getAccounts()</i>
63	accountsList	<i>getImportEntries()</i>	<i>calls</i>	<i>client</i>	<i>getTaxes()</i>
64	accountsList	<i>getYearsCovered()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
65	accountsList	<i>updateWithAccountTotal(String, Statement, HashMap<account, Double>, String, boolean)</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
66	accountsList	<i>accounts (field)</i>	<i>references</i>	<i>account</i>	
67	accountsList	<i>addPleaseSelectAccount()</i>	<i>calls</i>	<i>account</i>	<i>getTypeIDForString(String)</i>
68	accountsList	<i>addPleaseSelectAccount()</i>	<i>calls</i>	<i>account</i>	<i>account(String, String, int, int)</i>
69	accountsList	<i>addPleaseSelectAccount()</i>	<i>calls</i>	<i>account</i>	<i>setIndexInStringArray(int)</i>
70	accountsList	<i>addPleaseSelectAccount()</i>	<i>references</i>	<i>account</i>	
71	accountsList	<i>fillFromDB()</i>	<i>calls</i>	<i>account</i>	<i>account(accountsList, int,</i>

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
					<i>String, String, int, int)</i>
72	accountsList	<i>fillFromDB()</i>	<i>calls</i>	<i>account</i>	<i>setIndexInStringArray(int)</i>
73	accountsList	<i>fillFromDB()</i>	<i>calls</i>	<i>account</i>	<i>setType(int)</i>
74	accountsList	<i>fillFromDB()</i>	<i>calls</i>	<i>account</i>	<i>getAsString()</i>
57	accountsList	<i>fillFromDB()</i>	<i>references</i>	<i>account</i>	
58	accountsList	<i>getAccountForCode(String)</i>	<i>returns</i>	<i>account</i>	<i>getCode()</i>
59	accountsList	<i>getAccountForCode(String)</i>	<i>calls</i>	<i>account</i>	
60	accountsList	<i>getAccountForID(int)</i>	<i>returns</i>	<i>account</i>	<i>getID()</i>
61	accountsList	<i>getAccountForID(int)</i>	<i>calls</i>	<i>account</i>	
62	accountsList	<i>getAccountForListID(int)</i>	<i>returns</i>	<i>account</i>	<i>getIndexInStringArray()</i>
63	accountsList	<i>getAccountForListID(int)</i>	<i>calls</i>	<i>account</i>	
64	accountsList	<i>getAccounts()</i>	<i>calls</i>	<i>account</i>	<i>getTypeIDForString(String)</i>
65	accountsList	<i>getAccountsFromDatabase()</i>	<i>calls</i>	<i>account</i>	<i>account(accountsList, int, String, String, int, int)</i>
66	accountsList	<i>getAccountsFromDatabase()</i>	<i>calls</i>	<i>account</i>	<i>setIndexInStringArray(int)</i>
67	accountsList	<i>getAccountsFromDatabase()</i>	<i>references</i>	<i>account</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		<i>se()</i>			
68	accountsList	<i>getBankAccount()</i>	<i>returns</i>	<i>account</i>	
69	accountsList	<i>getDefaultAccount()</i>	<i>returns</i>	<i>account</i>	
70	accountsList	<i>getEntriesFromDatabase()</i>	<i>calls</i>	<i>account</i>	<i>removeAllEntries()</i>
71	accountsList	<i>getEntriesFromDatabase()</i>	<i>calls</i>	<i>account</i>	<i>addEntry(entry)</i>
72	accountsList	<i>getEntriesFromDatabase()</i>	<i>references</i>	<i>account</i>	
73	accountsList	<i>getImportEntries()</i>	<i>references</i>	<i>account</i>	
74	accountsList	<i>getLiabilitiesAccount()</i>	<i>returns</i>	<i>account</i>	
75	accountsList	<i>getPersonalDrawAccount()</i>	<i>returns</i>	<i>account</i>	
76	accountsList	<i>getReceivablesAccount()</i>	<i>returns</i>	<i>account</i>	
77	accountsList	<i>getRevenuesAccount()</i>	<i>returns</i>	<i>account</i>	
78	accountsList	<i>getStandardVATCreditAccount()</i>	<i>returns</i>	<i>account</i>	
79	accountsList	<i>getStandardVATDebitAccount()</i>	<i>returns</i>	<i>account</i>	
80	accountsList	<i>signalChange(account)</i>	<i>has param</i>	<i>account</i>	
81	accountsList	<i>updateWithAccountTotal</i>	<i>references</i>	<i>account</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		(String, Statement, HashMap<account, Double>, String, boolean)			
82	accountsList	getEntriesFromDatabase() ()	calls	entry	entry(int, Date, String, Double, account, account, tax, String, String)
83	accountsList	getEntriesFromDatabase() ()	calls	entry	setNumber(int)
84	accountsList	getEntriesFromDatabase() ()	references	entry	
85	accountsList	getEntryForImportID(int)	returns	entry	
86	accountsList	getEntryForImportID(int)	calls	entry	getImportID()
87	accountsList	getImportEntries() ()	calls	entry	entry(Date, String, Double, account, account, tax, String, String)
88	accountsList	getImportEntries() ()	calls	entry	setImportID(int)
89	accountsList	getImportEntries() ()	references	entry	
90	accountsList	getJournal() ()	references	entry	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
91	accountsList	<i>field journal</i>	<i>references</i>	<i>entry</i>	
92	application	<i>application{...}</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
93	application	<i>application{...}</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
94	asset		<i>contains</i>	<i>assets</i>	
95	asset	<i>asset(assets, int, int, String, Double, Date, Date, String)</i>	<i>has param</i>	<i>assets</i>	
96	asset	<i>delete()</i>	<i>calls</i>	<i>assets</i>	<i>signalChange(asset)</i>
97	asset	<i>field parent</i>	<i>is of type</i>	<i>assets</i>	
98	asset	<i>save()</i>	<i>calls</i>	<i>assets</i>	<i>signalChange(asset)</i>
100	assets	<i>getAssetsFromDB()</i>	<i>calls</i>	<i>asset</i>	<i>asset(assets, int, int, String, Double, Date, Date, String)</i>
101	assets	<i>getAssetsFromDB()</i>	<i>references</i>	<i>asset</i>	
102	assets	<i>signalChange(asset)</i>	<i>has param</i>	<i>asset</i>	
103	assets	<i>getAssetsFromDB()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
104	asset	<i>getValuesHashMap()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
105	assets	<i>getAssetsFromDB()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
106	assets	<i>Field assets</i>	<i>references</i>	<i>asset</i>	
107	assets	<i>getAssets()</i>	<i>references</i>	<i>asset</i>	
108	bookingCellM	<i>modify(Object, String,</i>	<i>calls</i>	<i>client</i>	<i>getAccounts()</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	odifier	<i>Object</i>)			
109	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>calls</i>	<i>client</i>	<i>getTaxes()</i>
110	bookingCellModifier	<i>getValue(Object, String)</i>	<i>calls</i>	<i>entry</i>	<i>getColumn(String)</i>
111	bookingCellModifier	<i>getValue(Object, String)</i>	<i>references</i>	<i>entry</i>	
112	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>calls</i>	<i>entry</i>	<i>getColumnType(String)</i>
113	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>calls</i>	<i>entry</i>	<i>setColumn(String, Object)</i>
114	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>references</i>	<i>entry</i>	
115	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>references</i>	<i>AccountNotFoundException</i>	
116	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>calls</i>	<i>accountsList</i>	<i>getAccountForListID(int)</i>
117	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
118	bookingCellModifier	<i>modify(Object, String, Object)</i>	<i>references</i>	<i>account</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
119	bookingContentProvider	<i>addEntry(entry)</i>	<i>has param</i>	<i>entry</i>	
220	bookingContentProvider	<i>addEntry(entry)</i>	<i>calls</i>	<i>entry</i>	<i>bindContentProvider (bookingContentProvider)</i>
221	bookingContentProvider	<i>field entries</i>	<i>references</i>	<i>entry</i>	
222	bookingContentProvider	<i>removeEntry(entry)</i>	<i>has param</i>	<i>entry</i>	
223	bookingLabelProvider	<i>getForeground(Object, int)</i>	<i>references</i>	<i>AccountNotFoundException</i>	
224	bookingLabelProvider	<i>getForeground(Object, int)</i>	<i>calls</i>	<i>accountsList</i>	<i>getBankAccount()</i>
225	bookingLabelProvider	<i>getColumnText(Object, int)</i>	<i>calls</i>	<i>entry</i>	<i>getColumnString(int)</i>
226	bookingLabelProvider	<i>getColumnText(Object, int)</i>	<i>references</i>	<i>entry</i>	
227	bookingLabelProvider	<i>getForeground(Object, int)</i>	<i>calls</i>	<i>entry</i>	<i>getDebitAccount()</i>
228	bookingLabelProvider	<i>getForeground(Object, int)</i>	<i>references</i>	<i>entry</i>	
229	bookingLabel	<i>getForeground(Object,</i>	<i>calls</i>	<i>client</i>	<i>getAccounts()</i>

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	Provider	<i>int</i>)			
230	client		<i>contains</i>	<i>documents</i>	
231	client	<i>field</i>	<i>is of type</i>	<i>documents</i>	
232	client	<i>getDocuments()</i>	<i>returns</i>	<i>documents</i>	
233	client	<i>getDocuments()</i>	<i>calls</i>	<i>documents</i>	<i>documents()</i>
234	client		<i>contains</i>	<i>accountsList</i>	
235	client	<i>field allAccounts</i>	<i>is of type</i>	<i>accountsList</i>	
236	client	<i>getAccounts()</i>	<i>returns</i>	<i>accountsList</i>	
237	client	<i>getAccounts()</i>	<i>calls</i>	<i>accountsList</i>	<i>accountsList()</i>
238	client	<i>getAccounts()</i>	<i>calls</i>	<i>accountsList</i>	<i>getAccountsFromDatabase()</i>
239	client	<i>getAccounts()</i>	<i>calls</i>	<i>accountsList</i>	<i>getEntriesFromDatabase()</i>
240	configs	<i>disposeOfficeApplication</i> <i>()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
241	configs	<i>etSampleValue(String,</i> <i>int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
242	configs	<i>readConfiguration()</i>	<i>calls</i>	<i>client</i>	<i>getConfigFilename()</i>
243	configs	<i>readConfiguration()</i>	<i>calls</i>	<i>client</i>	<i>getConfigPath()</i>
244	configs	<i>readDatabaseCredential</i> <i>s()</i>	<i>calls</i>	<i>client</i>	<i>getGlobalDataPath()</i>
245	configs	<i>readSettings()</i>	<i>calls</i>	<i>client</i>	<i>getSettingsFilename()</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
246	configs	<i>readSettings()</i>	<i>calls</i>	<i>client</i>	<i>getDataPath()</i>
247	configs	<i>writeConfiguration()</i>	<i>calls</i>	<i>client</i>	<i>getConfigFilename()</i>
248	configs	<i>writeSettings()</i>	<i>calls</i>	<i>client</i>	<i>getSettingsFilename()</i>
249	contact	<i>bindToID(int)</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
250	contact		<i>contains</i>	<i>contacts</i>	
251	contact	<i>contact(contacts, int, String, String, String, String, String, String, String, String, int, String, String, String, int)</i>	<i>has param</i>	<i>contacts</i>	
252	contact	<i>delete()</i>	<i>calls</i>	<i>contacts</i>	<i>signalChange(contact)</i>
253	contact	<i>field parent</i>	<i>is of type</i>	<i>contacts</i>	
254	contact	<i>save()</i>	<i>calls</i>	<i>contacts</i>	<i>signalChange(contact)</i>
255	contacts	<i>getContactsFromDB()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
256	contacts	<i>contacts{...}</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
257	contacts	<i>getContactsFromDB()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
258	contacts	<i>getPaymentIDForString(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
259	contacts	<i>getRoleIDForString(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
260	contacts	<i>getPaymentIDForString(String)</i>	<i>throws</i>	<i>PaymentMethodNotFoundException</i>	
261	contacts	<i>getPaymentIDForString(String)</i>	<i>calls</i>	<i>PaymentMethodNotFoundException</i>	<i>PaymentMethodNotFoundException(String)</i>
262	contacts	<i>getRoleIDForString(String)</i>	<i>throws</i>	<i>RoleNotFoundException</i>	
263	contacts	<i>getRoleIDForString(String)</i>	<i>calls</i>	<i>RoleNotFoundException</i>	<i>RoleNotFoundException(String)</i>
264	contacts	<i>field contactsList</i>	<i>references</i>	<i>contact</i>	
265	contacts	<i>getContacts()</i>	<i>references</i>	<i>contact</i>	
266	contacts	<i>getContactsFromDB()</i>	<i>calls</i>	<i>contact</i>	<i>contact(contacts, int, String, String, String, String, String, String, String, String, int, String, String, String, int)</i>
267	contacts	<i>getContactsFromDB()</i>	<i>references</i>	<i>contact</i>	<i>getContactsFromDB()</i>
268	contacts	<i>signalChange(contact)</i>	<i>has param</i>	<i>contact</i>	
269	contextComposite		<i>implements</i>	<i>Interface IcontextProvide</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
				<i>r</i>	
270	document	<i>getChangeValueForColumnName(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
271	document	<i>getValueForColumnIndex(Object, int)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
272	document	<i>save()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
273	document	<i>setChangedValueForColumnName(String, Object)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
274	document		<i>contains</i>	<i>documents</i>	
275	document	<i>document(documents, String)</i>	<i>has param</i>	<i>documents</i>	
276	document	<i>field parent</i>	<i>is of type</i>	<i>documents</i>	
277	documents	<i>getDocumentsFromDatabase()</i>	<i>calls</i>	<i>Messages</i>	<i>getString (String)</i>
278	documents	<i>getDocumentsFromDatabase()</i>	<i>calls</i>	<i>client</i>	<i>getClient()</i>
279	documents	<i>Field allDocuments</i>	<i>references</i>	<i>document</i>	
280	documents	<i>getAsArray()</i>	<i>returns</i>	<i>document</i>	
281	documents	<i>getDocumentForNumber</i>	<i>returns</i>	<i>document</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		<i>(String)</i>			
282	documents	<i>getDocumentForNumber (String)</i>	<i>calls</i>	<i>document</i>	<i>getNumber()</i>
283	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setID(int)</i>
284	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setDateEntry(Date)</i>
285	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setDateDocument(Date)</i>
286	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setNumber(String)</i>
287	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setSourceMD5(String)</i>
288	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setEntryID(int)</i>
289	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	<i>setIsLinked (boolean)</i>
290	documents	<i>getDocumentsFromData base()</i>	<i>references</i>	<i>document</i>	
291	documents	<i>getDocumentsFromData</i>	<i>returns</i>	<i>document</i>	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		<i>base()</i>			
292	documents	<i>getDocumentsFromData base()</i>	<i>calls</i>	<i>document</i>	
293	documents	<i>getUnlinkedNumbers()</i>	<i>calls</i>	<i>document</i>	<i>isLinked()</i>
294	documents	<i>getUnlinkedNumbers()</i>	<i>calls</i>	<i>document</i>	<i>getNumber()</i>
295	documents	<i>getUnlinkedNumbers()</i>	<i>references</i>	<i>document</i>	
296	item	<i>item {...}</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
297	item	<i>getPrice()</i>	<i>calls</i>	<i>utils</i>	<i>round(Double, int)</i>
298	item	<i>getQuantity()</i>	<i>calls</i>	<i>utils</i>	<i>round(Double, int)</i>
299	item	<i>getTotal()</i>	<i>calls</i>	<i>utils</i>	<i>round(Double, int)</i>
300	item	<i>addListener(IItemListene r)</i>	<i>has param</i>	<i>IItemListener</i>	
301	item	<i>itemListeners</i>	<i>references</i>	<i>IItemListener</i>	
302	item	<i>setProduct(product)</i>	<i>calls</i>	<i>IItemListener</i>	<i>onProductChange(product)</i>
303	item	<i>setProduct(product)</i>	<i>references</i>	<i>IItemListener</i>	
304	item	<i>item()</i>	<i>contains</i>	<i>products</i>	
305	item	<i>item()</i>	<i>calls</i>	<i>products</i>	<i>products()</i>
306	item	<i>item()</i>	<i>calls</i>	<i>products</i>	<i>getProductsFromDB()</i>
307	item	<i>item()</i>	<i>calls</i>	<i>products</i>	<i>addPleaseSelectProduct()</i>
308	item	<i>item()</i>	<i>calls</i>	<i>products</i>	<i>getDefaultProduct()</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
309	item	<i>item()</i>	<i>references</i>	<i>products</i>	
310	item	<i>Field availableProducts</i>	<i>is of type</i>	<i>products</i>	
311	item	<i>isEmpty()</i>	<i>calls</i>	<i>products</i>	<i>getDefaultProduct()</i>
312	item	<i>setProduct(int)</i>	<i>calls</i>	<i>products</i>	<i>getProduct(int)</i>
313	item	<i>getColumnString(int)</i>	<i>calls</i>	<i>product</i>	<i>getName()</i>
314	item	<i>getNominalPrice()</i>	<i>calls</i>	<i>product</i>	<i>getNominalPrice()</i>
315	item	<i>getOBDXString()</i>	<i>calls</i>	<i>product</i>	<i>getVAT()</i>
316	item	<i>getPriceGross()</i>	<i>calls</i>	<i>product</i>	<i>getVAT()</i>
317	item	<i>getProduct()</i>	<i>returns</i>	<i>product</i>	
318	item	<i>getProductName()</i>	<i>calls</i>	<i>product</i>	<i>getName()</i>
319	item	<i>getTotalGross()</i>	<i>calls</i>	<i>product</i>	<i>getVAT()</i>
320	item	<i>isEmpty()</i>	<i>calls</i>	<i>product</i>	<i>getIndexInStringArray()</i>
321	item	<i>saveForTransaction(int)</i>	<i>accesses</i>	<i>product</i>	<i>Field id</i>
322	item	<i>saveForTransaction(int)</i>	<i>references</i>	<i>product</i>	
323	item	<i>setProduct(product)</i>	<i>has param</i>	<i>product</i>	
324	item		<i>contains</i>	<i>transactionContentProvider</i>	
325	item	<i>bindContentProvider</i>	<i>has param</i>	<i>transactionContentProvider</i>	
326	item	<i>Field contentProvider</i>	<i>is of type</i>	<i>transactionContent</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
				<i>entProvider</i>	
327	<i>item</i>	<i>removeItem()</i>	<i>calls</i>	<i>transactionCont entProvider</i>	<i>removeEntry(item)</i>
328	<i>itemTableList</i>	<i>processSingleTable(IText Table, ITextTableCell, ITextTableCell)</i>	<i>references</i>	<i>items</i>	
329	<i>itemTableList</i>	<i>process()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
330	<i>itemTableList</i>	<i>processSingleTable(IText Table, ITextTableCell, ITextTableCell)</i>	<i>references</i>	<i>item</i>	
331	<i>itemTableList</i>	<i>processSingleTable(IText Table, ITextTableCell, ITextTableCell)</i>	<i>calls</i>	<i>transactionCont entProvider</i>	<i>size()</i>
332	<i>itemTableList</i>	<i>processSingleTable(IText Table, ITextTableCell, ITextTableCell)</i>	<i>calls</i>	<i>transactionCont entProvider</i>	<i>elementAt(int)</i>
333	<i>itemTableList</i>		<i>contains</i>	<i>placeholderMan ager</i>	
334	<i>itemTableList</i>	<i>itemTableList(Shell, ITextDocument, transaction,</i>	<i>has param</i>	<i>placeholderMan ager</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		<i>placeholderManager, ITextField[]</i>			
335	itemTableList	<i>Field placeHolderManager</i>	<i>is of type</i>	<i>placeholderManager</i>	
336	itemTableList	<i>processSingleTable(ITextTable, ITextTableCell, ITextTableCell)</i>	<i>calls</i>	<i>placeholderManager</i>	<i>getPlaceholderNames()</i>
337	itemTableList	<i>processSingleTable(ITextTable, ITextTableCell, ITextTableCell)</i>	<i>calls</i>	<i>placeholderManager</i>	<i>getPlaceholderValues(item, transaction)</i>
338	itemTextList	<i>process()</i>	<i>references</i>	<i>items</i>	
339	itemTextList	<i>getItemTemplate()</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
340	itemTextList	<i>insertItem(item, transaction)</i>	<i>has param</i>	<i>item</i>	
341	itemTextList	<i>process()</i>	<i>references</i>	<i>item</i>	
342	itemTextList	<i>process()</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>size()</i>
343	itemTextList	<i>process()</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>elementAt(int)</i>
344	itemTextList		<i>contains</i>	<i>placeholderManager</i>	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
345	itemTextList	<i>itemTextList(Shell, ITextDocument, transaction, placeholderManager, ITextField[])</i>	<i>has param</i>	<i>placeholderManager</i>	
346	itemTextList	<i>insertItem(item, transaction)</i>	<i>calls</i>	<i>placeholderManager</i>	<i>getPlaceholderNames()</i>
347	itemTextList	<i>insertItem(item, transaction)</i>	<i>calls</i>	<i>placeholderManager</i>	<i>etPlaceholderValues(item, transaction)</i>
348	itemTextList	<i>Field placeholderManager</i>	<i>is of type</i>	<i>placeholderManager</i>	
349	items	<i>setCurrentPrice(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
350	items	<i>setCurrentPrice(String)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>
351	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>item</i>	<i>item()</i>
352	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>item</i>	<i>setDescription(String)</i>
353	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>item</i>	<i>setPrice(double)</i>
354	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>item</i>	<i>setQuantity(Double)</i>
355	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>item</i>	<i>setProduct(int)</i>
356	items	<i>loadFromDB(int)</i>	<i>references</i>	<i>item</i>	
357	items	<i>setCurrentDescription(String)</i>	<i>calls</i>	<i>item</i>	<i>setDescription(String)</i>

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
358	items	<i>setCurrentDescription(String)</i>	<i>references</i>	<i>item</i>	
359	items	<i>setCurrentDescription(String)</i>	<i>calls</i>	<i>item</i>	<i>setPrice(double)</i>
360	items	<i>setCurrentDescription(String)</i>	<i>references</i>	<i>item</i>	
361	items	<i>setCurrentQuantity(String)</i>	<i>calls</i>	<i>item</i>	<i>setQuantity(Double)</i>
362	items	<i>setCurrentQuantity(String)</i>	<i>references</i>	<i>item</i>	
363	items		<i>extends</i>	<i>transactionContentProvider</i>	
364	items	<i>items()</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>transactionContentProvider()</i>
365	items	<i>emptyAllItems()</i>	<i>accesses</i>	<i>transactionContentProvider</i>	<i>Field entries</i>
366	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>clear()</i>
367	items	<i>loadFromDB(int)</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>addEntry(item)</i>
368	items	<i>setCurrentDescription(String)</i>	<i>calls</i>	<i>transactionContent</i>	<i>elementAt(int)</i>

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		<i>ring</i>)		<i>entProvider</i>	
369	items	<i>setCurrentDescription(String)</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>elementAt(int)</i>
370	items	<i>setCurrentDescription(String)</i>	<i>calls</i>	<i>transactionContentProvider</i>	<i>elementAt(int)</i>
371	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>contact</i>	<i>getBankCode()</i>
372	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>contact</i>	<i>getBankAccount()</i>
373	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>contact</i>	<i>getAccountholder()</i>
374	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>configs</i>	<i>getBankCode()</i>
375	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>configs</i>	<i>getAccountCode()</i>
376	moneyTransfer		<i>calls</i>	<i>configs</i>	<i>getBankCode()</i>
377	moneyTransfer		<i>calls</i>	<i>configs</i>	<i>getAccountCode()</i>
378	moneyTransfer	<i>performMoneyTransfer(transaction)</i>	<i>calls</i>	<i>Messages</i>	<i>getString(String)</i>

4.4.2. Code Dependency Package appLayer ke Package dataLayer

Terdapat 47 Code Dependency yang ada pada package appLayer ke package dataLayer, dependency kind terbanyak adalah *calls* dengan jumlah 45, class yang paling sering diakses adalah class DB.

Tabel 4. 6 Code Dependency Package appLayer ke Package dataLayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	account	delete()	<i>calls</i>	DB	deleteLogical(String, String, int)
2	account	save()	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
3	account	save()	<i>calls</i>	DB	insert(String, HashMap<String, String>)
4	accountsList	fillFromDB()	<i>calls</i>	DB	getConnection()
5	accountsList	getCompletePeriod End()	<i>calls</i>	DB	getConnection()
6	accountsList	getCompletePeriod Start()	<i>calls</i>	DB	getConnection()
7	accountsList	getEntriesFromData base()	<i>calls</i>	DB	getConnection()
8	accountsList	getImportEntries()	<i>calls</i>	DB	getConnection()
9	accountsList	getYearsCovered()	<i>calls</i>	DB	getConnection()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
10	asset	delete()	<i>calls</i>	DB	deleteLogical(String, String, int)
11	asset	save()	<i>calls</i>	DB	insert(String, HashMap<String, String>)
12	asset	save()	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
13	asset	getAssetsFromDB()	<i>calls</i>	DB	getConnection()
14	configs	readDatabaseCredentials()	<i>calls</i>	DB	getDBVersionString()
15	configs	readConfiguration()	<i>calls</i>	fileUtils	readFileAsString(String)
16	configs	readSettings()	<i>calls</i>	fileUtils	readFileAsString(String)
17	contact	bindToID(int)	<i>calls</i>	DB	DB.getConnection()
18	contact	delete()	<i>calls</i>	DB	deleteLogical(String, String, int)
19	contact	save()	<i>calls</i>	DB	insert(String, HashMap<String, String>)
20	contact	save()	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
21	contact	getContactsFromDB()	<i>calls</i>	DB	getConnection()
22	document	save()	<i>calls</i>	DB	insertRaw(String, HashMap<String, String>)
23	document	save()	<i>calls</i>	DB	updateRaw(String, String, int,

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
					HashMap<String, String>)
24	document	getDocumentsFromDatabase()	<i>calls</i>	DB	getConnection()
25	entry	delete()	<i>calls</i>	DB	deleteLogical(String, String, int)
26	entry	getColumnInsertStrDefinitions()	<i>calls</i>	DB	getIdentifierQuote()
27	entry	insert()	<i>calls</i>	DB	getConnection()
28	entry	insert()	<i>calls</i>	DB	getLastInsertID(Statement)
29	entry	load(int)	<i>calls</i>	DB	getConnection()
30	entry	removeFromImport(boolean)	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
31	entry	removeFromImport(boolean)	<i>calls</i>	DB	deleteLogical(String, String, int)
32	entry	update()	<i>calls</i>	DB	getConnection()
33	entry	accountColumn.getDBInsertStrDefinition()	<i>calls</i>	DB	getIdentifierQuote()
34	entry	column.getDBUpdateStr()	<i>calls</i>	DB	getIdentifierQuote()
35	item	saveForTransaction(int)	<i>calls</i>	DB	insert(String, HashMap<String, String>)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
36	items	loadFromDB(int)	<i>calls</i>	DB	getConnection()
37	moneyTransfer	moneyTransfer()	<i>calls</i>	HBCI	getInstance()
38	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	HBCI	newJob(String)
39	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	HBCI	execute(String)
40	moneyTransfer	transferMoneyTo(String, String, String, String, double)	<i>calls</i>	HBCI	newJob(String)
41	moneyTransfer	transferMoneyTo(String, String, String, String, double)	<i>calls</i>	HBCI	execute(String)
42	product	delete()	<i>calls</i>	DB	deleteLogical(String, String, int)
43	product	save()	<i>calls</i>	DB	insert(String, HashMap<String, String>)
44	product	save()	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
45	product	getProductsFromDB()	<i>calls</i>	DB	getConnection()
46	moneyTransfer	moneyTransfer	<i>contains</i>	HBCI	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	er				
47	moneyTransfer	field hbci	<i>is of type</i>	HBCI	

4.4.3. Code Dependency Package appLayer ke Package appLayer.taxRelated

Terdapat 50 Code Dependency yang ada pada package appLayer ke package appLayer.taxRelated, dependency kind terbanyak adalah *calls* dengan jumlah 26, class yang paling sering diakses adalah class tax.

Tabel 4.7 Code Dependency Package appLayer ke Package appLayer.taxRelated

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	accountsList	getImportEntries()	<i>calls</i>	taxList	.getDefault()
2	bookingCellModifier	modify(Object, String, Object)	<i>calls</i>	taxList	getVATAtListIndex(int)
3	client	getTaxes()	<i>calls</i>	taxList	<i>method</i> appLayer.taxRelated.taxList.()
4	client	getTaxes()	<i>calls</i>	taxList	getTaxesFromDatabase()
5	entry	book()	<i>calls</i>	tax	getIDinList()
6	entry	book()	<i>calls</i>	tax	getFactor()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
7	entry	book()	<i>calls</i>	tax	getSelectedAccount()
8	entry	book()	<i>calls</i>	tax	isOnCreditSide()
9	entry. taxColumn	getStringValue()	<i>calls</i>	tax	getDescription()
10	entry	taxColumn.getValue()	<i>calls</i>	tax	getIDinList()
11	entry	book()	<i>calls</i>	taxList	addEmptyTax()
12	entry	book()	<i>calls</i>	taxList	getDefault()
13	entry	load(int)	<i>calls</i>	taxList	getDefault()
14	entry.taxCol umn.	getStringValue()	<i>calls</i>	taxList	getDefault()
15	entry.taxCol umn.	getTaxValue()	<i>calls</i>	taxList	getDefault()
16	entry.taxCol umn.	getValue()	<i>calls</i>	taxList	getDefault()
17	item	getOBDXString()	<i>calls</i>	tax	getFactor()
18	item	getPriceGross()	<i>calls</i>	tax	getFactor()
19	item	getTotalGross()	<i>calls</i>	tax	getFactor()
20	placeholder Manager	getPlaceholderValues(item, transaction)	<i>calls</i>	tax	getDescription()
21	product	save()	<i>calls</i>	tax	getID()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
22	product	addPleaseSelectProduct()	<i>calls</i>	taxList	taxList()
23	product	addPleaseSelectProduct()	<i>calls</i>	taxList	getTaxesFromDatabase()
24	product	addPleaseSelectProduct()	<i>calls</i>	taxList	getDefault()
25	product	getProductsFromDB()	<i>calls</i>	taxList	getDefault()
26	product	getProductsFromDB()	<i>calls</i>	taxList	getVATByID(int)
27	client		<i>contains</i>	taxList	
28	product		<i>contains</i>	tax	
29	entry	entry(int, Date, String, Double, account, account, tax, String, String)	<i>has param</i>	tax	
30	entry	entry(Date, String, Double, account, account, tax, String, String)	<i>has param</i>	tax	
31	entry	assignBasicValues(Date, String, Double, account, account, tax, String, String)	<i>has param</i>	tax	
32	entry	setVAT(tax)	<i>has param</i>	tax	
33	entry	taxColumn.(entry, taxList, String, String)	<i>has param</i>	taxList	
34	product	product(products, int, tax, String, double)	<i>has param</i>	tax	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
35	product	setTax(tax)	<i>has param</i>	tax	
36	client	field allTaxes	<i>is of type</i>	taxList	
37	product	field vat	<i>is of type</i>	tax	
38	bookingCell Modifier	bookingCellModifier.modify(Object, String, Object)	<i>references</i>	tax	
39	bookingCell Modifier	bookingCellModifier.modify(Object, String, Object)	<i>references</i>	taxNotFoundException	
40	entry.taxColumn	.getStringValue()	<i>references</i>	tax	
41	entry.taxColumn	taxColumn.getValue()	<i>references</i>	tax	
42	entry.taxColumn	setValue(Object)	<i>references</i>	tax	
43	entry	book()	<i>references</i>	taxList	
44	products	<i>method</i> appLayer.products.addPleaseSelectProduct()	<i>references</i>	taxList	
45	products	<i>method</i> appLayer.products.getProductsFromDB()	<i>references</i>	taxList	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
46	products	<i>method</i> appLayer.products.getProducts FromDB()	<i>references</i>	taxNotF oundEx ception	
47	client	getTaxes()	<i>returns</i>	taxList	
48	entry	getVAT()	<i>returns</i>	tax	
49	entry.taxCol umn	getTaxValue()	<i>returns</i>	tax	
50	product	getVAT()	<i>returns</i>	tax	

4.4.4. Code Dependency Package appLayer ke Package appLayer.transactionRelated

Terdapat 34 *Code Dependency* yang ada pada *package* appLayer ke *package* appLayer.transactionRelated, *dependency kind* terbanyak adalah *calls* dengan jumlah 19, *class* yang diakses adalah *class* transaction.

Tabel 4. 8 *Code Dependency Package* appLayer ke *Package* appLayer.transactionRelated

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	moneyTransfe r	performMoneyTransfer(transact ion)	<i>aceses</i>	transaction	field cashFlow.RECEIVE
2	moneyTransfe r	performMoneyTransfer(transact ion)	<i>aceses</i>	transaction	field cashFlow.SPEND

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
3	client	getTransactions()	<i>calls</i>	transaction	transactions.()
4	entry	removeFromImport(boolean)	<i>calls</i>	transaction	getNumber()
5	entry	book()	<i>calls</i>	transaction	getIDForNumber(String)
6	entry	removeFromImport(boolean)	<i>calls</i>	transaction	load(int)
7	itemTableList	processSingleTable(ITextTable, ITextTableCell, ITextTableCell)	<i>calls</i>	transaction	getItems()
8	itemTextList	process()	<i>calls</i>	transaction	getItems()
9	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	transaction	getCashFlow()
10	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	transaction	getRecipient()
11	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	transaction	getTotalGross()
12	moneyTransfer	performMoneyTransfer(transaction)	<i>calls</i>	transaction	getNumber()
13	placeholderManager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getDueDate()
14	placeholderManager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getIssueDate()
15	placeholderManager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getRecipient()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	nager	transaction)			
16	placeholderMa nager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getNumber()
17	placeholderMa nager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getRemarks()
18	placeholderMa nager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getTotalString()
19	placeholderMa nager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getTotalGrossString()
20	placeholderMa nager	placeholderManager.getPlaceh olderValues(item, transaction)	<i>calls</i>	transaction	getVatTotal()
21	placeholderMa nager	getPlaceholderValues(item, transaction)	<i>calls</i>	transaction	getVatList()
22	client		<i>contains</i>	transaction	
23	itemTableList		<i>contains</i>	transaction	
24	itemTableList		<i>contains</i>	transaction	
25	itemTableList	itemTableList(Shell, ITextDocument, transaction, placeholderManager, ITextField[])	<i>has param</i>	transaction	
26	itemTextList	itemTextList(Shell,	<i>has param</i>	transaction	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		ITextDocument, transaction, placeholderManager, ITextField[])			
27	itemTextList	insertItem(item, transaction)	<i>has param</i>	transaction	
28	moneyTransfer	performMoneyTransfer(transaction)	<i>has param</i>	transaction	
29	placeholderManager	getPlaceholderValues(item, transaction)	<i>has param</i>	transaction	
30	client	field transactionController	<i>is of type</i>	transaction	
31	itemTableList	field theTransaction	<i>is of type</i>	transaction	
32	itemTextList	field theTransaction	<i>is of type</i>	transaction	
33	entry	removeFromImport(boolean)	<i>references</i>	transaction	
34	client	getTransactions()	<i>returns</i>	transaction	

4.4.5. Code Dependency Package appLayer ke Package GUILayer

Terdapat 7 Code Dependency yang ada pada package appLayer ke package GUILayer, dependency kind terbanyak adalah *contains*, *has param* dan *is of type* dengan jumlah masing-masing dua, class diakses adalah class newTransactionSelectItemDetails.

Tabel 4. 9 Code Dependency Package appLayer ke Package GUILayer

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	pageActivator	<i>method</i> appLayer.pageActivator.onProductChange(product)	<i>calls</i>	newTransactionSelectItemDetails	<i>method</i> GUILayer.newTransactionSelectItemDetails.checkPageComplete()
2	pageActivator	<i>class</i> appLayer.pageActivator	<i>contains</i>	newTransactionSelectItemDetails	<i>class</i> GUILayer.newTransactionSelectItemDetails
3	transactionContentProvider	<i>class</i> appLayer.transactionContentProvider	<i>contains</i>	newTransactionSelectItemDetails	
4	pageActivator	<i>method</i> appLayer.pageActivator.(newTransactionSelectItemDetails)	<i>has param</i>	newTransactionSelectItemDetails	
5	transactionContentProvider	<i>method</i> appLayer.transactionContentProvider.bindToPage(newTransactionSelectItemDetails)	<i>has param</i>	newTransactionSelectItemDetails	
6	pageActivator	field appLayer.pageActivator.parentPage	<i>is of type</i>	newTransactionSelectItemDetails	
7	transactionContentProvider	field	<i>is of type</i>	newTransactionSelectItemDetails	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	nContent Provider	appLayer.transactionContentProvider.parentPage		SelectItemDetails	

4.5. Package appLayer.taxRelated

Bagian ini membahas tentang *dependency code* antar *class* di *package* appLayer.taxRelated dan juga *dependency code* antara *class* di *package* appLayer.transactionRelated dengan *class* di *package* lain. Tabel 4. 10 adalah *dependency code* antar *class* di *package* appLayer.taxRelated. Tabel 4. 11 adalah *dependency code package* appLayer.taxRelated dengan *package* dataLayer. Tabel 4. 12 adalah *dependency code package* appLayer.taxRelated dengan *package* appLayer.

4.5.1. Code Dependency Antar Class appLayer.taxRelated

Terdapat 39 *Code Dependency* yang ada pada *package* appLayer.taxRelated, *dependency kind* terbanyak adalah *calls* dengan jumlah 17, *class* yang paling sering diakses adalah *class* tax.

Tabel 4. 10 code dependency antar class appLayer.taxRelated

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	IRSOffices	IRSOffices{...}	contains	IRSOffice	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
2	IRSOffices	IRSOffices{...}	<i>calls</i>	IRSOffice	<i>IRSOffice(String, String)</i>
3	IRSOffices	getIRSOffices()	<i>references</i>	IRSOffice	
4	IRSOffices	getIRSOffices()	<i>returns</i>	IRSOffice	
5	IRSOffices	Field offices	<i>is of type</i>	IRSOffice	
6	state	getOrderNumber(IRSOffice)	<i>has param</i>	IRSOffice	
7	state	getOrderNumber(IRSOffice)	<i>calls</i>	IRSOffice	getID()
8	states	states()	<i>calls</i>	state	state(String, String, String)
9	states	states()	<i>calls</i>	state	addTaxCode(String)
10	states	states()	<i>references</i>	state	
11	states	getStates()	<i>returns</i>	state	
12	states	Field states	<i>references</i>	state	
13	tax		<i>contains</i>	taxList	
14	tax	tax(taxList, int, String, double, account, account)	<i>has param</i>	taxList	
15	tax	delete()	<i>calls</i>	taxList	signalChange(tax)
16	tax	Field parent	<i>is of type</i>	taxList	
17	tax	save()	<i>calls</i>	taxList	signalChange(tax)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
18	taxList	getDefault()	<i>returns</i>	tax	
19	taxList	getStringArray()	<i>calls</i>	tax	getDescription()
20	taxList	getStringArray()	<i>references</i>	tax	
21	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	tax(taxList, int, String, double, account, account)
22	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	bindToList(int)
23	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	setCreditTaxField(int)
24	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	setDebitTaxField(int)
25	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	setDescription(String)
26	taxList	getTaxesFromDatabase()	<i>calls</i>	tax	setSelectedAccountCredit(boolean)
27	taxList	getTaxesFromDatabase()	<i>references</i>	tax	
28	taxList	getVATArray()	<i>returns</i>	tax	
29	taxList	getVATAtListIndex(int)	<i>returns</i>	tax	
30	taxList	getVATAtListIndex(int)	<i>calls</i>	tax	getIDinList()
31	taxList	getVATByID(int)	<i>returns</i>	tax	
32	taxList	getVATByID(int)	<i>calls</i>	tax	getID()
33	taxList	getVATs()	<i>references</i>	tax	
34	taxList	signalChange(tax)	<i>has param</i>	tax	
35	taxList	Field VATs	<i>references</i>	tax	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
36	taxList	getVATAtListIndex(int)	throws	taxNotFoundException	
37	taxList	getVATAtListIndex(int)	calls	taxNotFoundException	taxNotFoundException()
38	taxList	getVATByID(int)	throws	taxNotFoundException	
39	taxList	getVATByID(int)	calls	taxNotFoundException	taxNotFoundException()

4.5.2. Code Dependency Package `appLayer.taxRelated` ke Package `dataLayer`

Terdapat 4 Code Dependency yang ada pada package `appLayer.taxRelated` ke package `dataLayer`, dependency kind yang ada adalah *calls*, class yang paling sering diakses adalah class `DB`.

Tabel 4. 11 Code Dependency Package `appLayer.taxRelated` ke Package `dataLayer`

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	tax	delete()	calls	DB	deleteLogical(String, String, int)
2	tax	save()	calls	DB	insert(String, HashMap<String, String>)
3	tax	save()	calls	DB	update(String, String, int, HashMap<String, String>)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
4	taxList	getTaxesFromDatabase()	calls	DB	getConnection()

4.5.3. Code Dependency Package `appLayer.taxRelated` ke Package `appLayer`

Terdapat 19 *Code Dependency* yang ada pada *package* `appLayer.taxRelated` ke *package* `appLayer`, *dependency kind* terbanyak adalah *calls* dengan jumlah 9, *class* yang paling sering diakses adalah *class* `account`.

Tabel 4. 12 *Code Dependency Package* `appLayer.taxRelated` ke *Package* `appLayer`

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	state	getTaxIDWithoutFirstSlice()	calls	configs	getTaxID()
2	states	states()	calls	Messages	getString(String)
3	tax	save()	calls	account	getID()
4	tax	save()	calls	client	getClient()
5	taxList	taxList()	calls	Messages	getString(String)
6	taxList	getTaxesFromDatabase()	calls	Messages	getString(String)

7	taxList	getTaxesFromDatabase() ()	<i>calls</i>	accountsList	getAccountForID(int)
8	taxList	getTaxesFromDatabase() ()	<i>calls</i>	client	getClient()
9	taxList	getTaxesFromDatabase() ()	<i>calls</i>	client	getAccounts()
10	tax		<i>contains</i>	account	
11	tax	tax(taxList, int, String, double, account, account)	<i>has param</i>	account	
12	tax	setCreditAccount(account)	<i>has param</i>	account	
13	tax	setDebitAccount(account)	<i>has param</i>	account	
14	tax	field creditAccount	<i>is of type</i>	account	
15	tax	field debitAccount	<i>is of type</i>	account	
16	taxList	getTaxesFromDatabase() ()	<i>references</i>	AccountNotFoundException	
17	tax	getCreditAccount()	<i>returns</i>	account	
18	tax	getDebitAccount()	<i>returns</i>	account	

19	tax	getSelectedAccount()	returns	account	
----	-----	----------------------	---------	---------	--

4.6. Package appLayer.transactionRelated

Bagian ini membahas tentang *dependency code* antar *class* di *package* appLayer.transactionRelated dan juga *dependency code* antara *class* di *package* appLayer.transactionRelated dengan *class* di *package* lain. Tabel 4. 13 adalah *dependency code* antar *class* di *package* appLayer.transactionRelated. Tabel 4. 14 adalah *dependency code* *package* appLayer.transactionRelated dengan *package* dataLayer. Tabel 4. 15 adalah *dependency code* *package* appLayer.transactionRelated dengan *package* appLayer. Tabel 4. 16 adalah *dependency code* *package* appLayer.transactionRelated dengan *package* appLayer.taxRelated. Tabel 4. 17 adalah *dependency code* *package* appLayer.transactionRelated dengan *package* GUILayer.

4.6.1 Code Dependency Antar Class appLayer.transactionRelated

Terdapat 23 *Code Dependency* yang ada pada *package* appLayer.transactionRelated, *dependency kind* terbanyak adalah *calls* dengan jumlah 16, *class* yang paling sering diakses adalah *class* transaction.

Tabel 4. 13 Code Dependency antar class appLayer.transactionRelated

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	cancelation		<i>extends</i>	transaction	
2	cancelation	cancelation(transactions)	<i>calls</i>	transaction	transaction(transactions)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
3	cancelation	book()	<i>calls</i>	transaction	getParent()
4	cancelation	book()	<i>calls</i>	transaction	getPrecedingTransactionID()
5	cancelation	book()	<i>calls</i>	transaction	getNumber()
6	cancelation	book()	<i>calls</i>	transaction	getIssueDate()
7	cancelation	book()	<i>calls</i>	transaction	getDueDate()
8	cancelation	book()	<i>calls</i>	transaction	getIssueDate()
9	cancelation	book()	<i>calls</i>	transaction	getTotalGross()
10	cancelation	book()	<i>references</i>	transaction	
11	cancelation	getCashFlow()	<i>returns</i>	transaction	Enum cashFlow
12	cancelation	getCashFlow()	<i>accesses</i>	transaction	Field NO
13	cancelation	getOriginalTransactions Entry()	<i>calls</i>	transaction	getPrecedingTransactionID()
14	cancelation	cancelation(transactions)	<i>has param</i>	transactions	
15	cancelation	book()	<i>calls</i>	transactions	load(int)
16	cancelation	book()	<i>calls</i>	transactions	getCurrentTransaction()
17	creditmemo	creditmemo(transactions)	<i>has param</i>	transactions	
18	creditmemo		<i>extends</i>	transaction	
19	creditmemo	creditmemo(transactions)	<i>calls</i>	transaction	transaction(transactions)
20	creditmemo	book()	<i>calls</i>	transaction	getRecipient()
21	creditmemo	book()	<i>calls</i>	transaction	getIssueDate()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
22	creditmemo	book()	<i>calls</i>	transaction	getNumber()
23	creditmemo	book()	<i>calls</i>	transaction	getTotalGross()

4.6.2 Code Dependency Package `appLayer.transactionRelated` ke Package `dataLayer`

Terdapat 20 *Code Dependency* yang ada pada *package* `appLayer.transactionRelated` ke *package* `dataLayer`, *dependency kind* yang ada adalah *calls*, *class* yang diakses adalah *class* `DB`.

Tabel 4. 14 *Code Dependency Package* `appLayer.transactionRelated` ke *Package* `dataLayer`

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	cancelation	getOriginalTransactionsEntry()	<i>calls</i>	DB	getConnection()
2	cancelation	getType()	<i>calls</i>	DB	getDatabaseOffset()
3	creditmemo	getTodoItems()	<i>calls</i>	DB	getConnection()
4	creditmemo	getType()	<i>calls</i>	DB	getDatabaseOffset()
5	invoice	getTodoItems()	<i>calls</i>	DB	getConnection()
6	invoice	getType()	<i>calls</i>	DB	getDatabaseOffset()
7	offer	getType()	<i>calls</i>	DB	getDatabaseOffset()
8	personalDra	getTodoItems()	<i>calls</i>	DB	getConnection()

	wing				
9	personalDrawing	getType()	<i>calls</i>	DB	getDatabaseOffset()
10	reminder	getType()	<i>calls</i>	DB	getDatabaseOffset()
11	transaction	getNewTransactionNumber()	<i>calls</i>	DB	getConnection()
12	transaction	getNewTransactionNumber()	<i>calls</i>	DB	getIdentifierQuote()
13	transaction	save()	<i>calls</i>	DB	getConnection()
14	transaction	save()	<i>calls</i>	DB	getLastInsertID(Statement)
15	transaction	saveType()	<i>calls</i>	DB	update(String, String, int, HashMap<String, String>)
16	transactions	getIDForNumber(String)	<i>calls</i>	DB	getConnection()
17	transactions	getIDForNumber(String)	<i>calls</i>	DB	getIdentifierQuote()
18	transactions	getTransactionTypesFromDB()	<i>calls</i>	DB	getConnection()
19	transactions	getTransactionTypesFromDB()	<i>calls</i>	DB	getIdentifierQuote()
20	transactions	load(int)	<i>calls</i>	DB	getConnection()

4.6.3 Code Dependency Package `appLayer.transactionRelated` ke `Package appLayer`

Terdapat 163 *Code Dependency* yang ada pada *package* `appLayer.transactionRelated` ke *package* `appLayer`, *dependency kind* terbanyak adalah *calls* dengan jumlah 98, *class* yang paling sering diakses adalah *class* `account`.

Tabel 4. 15 *Code Dependency Package* `appLayer.transactionRelated` ke *Package* `appLayer`

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	cancelation	book() calls	<i>calls</i>	Messages	getString(String)
2	cancelation	getOriginalTransactionsEntry()	<i>calls</i>	client	getClient()
3	cancelation	book()	<i>calls</i>	entry	entry(int)
4	cancelation	book()	<i>calls</i>	entry	getDebitAccount()
5	cancelation	book()	<i>calls</i>	entry	getCreditAccount()
6	cancelation	book()	<i>calls</i>	entry	getVAT()
7	cancelation	book()	<i>calls</i>	entry	entry(Date, String, Double, account, account, tax, String, String)
8	cancelation	book()	<i>calls</i>	entry	setReferToTransaction(int)
9	cancelation	book()	<i>calls</i>	entry	entrybook()
10	creditmemo	book()	<i>calls</i>	account	setReferTo(int)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
11	creditmemo	getTodoItems()	<i>calls</i>	account	getID()
12	creditmemo	creditmemo(transactions)	<i>calls</i>	accountsList	getRevenuesAccount()
13	creditmemo	creditmemo(transactions)	<i>calls</i>	accountsList	getBankAccount()
14	creditmemo	creditmemo(transactions)	<i>calls</i>	client	getAccounts()
15	creditmemo	book()	<i>calls</i>	client	getTaxes()
16	creditmemo	getTodoItems()	<i>calls</i>	client	getClient()
17	creditmemo	book()	<i>calls</i>	contact	getID()
18	creditmemo	book()	<i>calls</i>	entry	entry(Date, String, Double, account, account, tax, String, String)
19	creditmemo	book()	<i>calls</i>	entry	setReferToTransaction(int)
20	creditmemo	book()	<i>calls</i>	entry	book()
21	invoice	book()	<i>calls</i>	account	setReferTo(int)
22	invoice	getTodoItems()	<i>calls</i>	account	getID()
23	invoice	invoice(transactions)	<i>calls</i>	accountsList	getReceivablesAccount()
24	invoice	invoice(transactions)	<i>calls</i>	accountsList	getRevenuesAccount()
25	invoice	invoice(transactions)	<i>calls</i>	accountsList	getBankAccount()
26	invoice	book()	<i>calls</i>	accountsList	getRevenuesAccount()
27	invoice	invoice(transactions)	<i>calls</i>	client	getAccounts()
28	invoice	book()	<i>calls</i>	client	getTaxes()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
29	invoice	book()	<i>calls</i>	client	getAccounts()
30	invoice	getTodoItems()	<i>calls</i>	client	getClient()
31	invoice	book()	<i>calls</i>	contact	getID()
32	invoice	book()	<i>calls</i>	entry	entry.(Date, String, Double account, account, tax, String, String)
33	invoice	book()	<i>calls</i>	entry	setReferToTransaction(int)
34	invoice	book()	<i>calls</i>	entry	book()
35	invoice	book()	<i>calls</i>	item	getTotal()
36	invoice	book()	<i>calls</i>	item	getProduct()
37	invoice	book()	<i>calls</i>	item	getTotalGross()
38	invoice	book()	<i>calls</i>	product	getVAT()
39	invoice	book()	<i>calls</i>	transactionContent Provider	getItems()
40	personalDrawing	book()	<i>calls</i>	account	setReferTo(int)
41	personalDrawing	getTodoItems()	<i>calls</i>	account	getID()
42	personalDrawing	personalDrawing(transactions)	<i>calls</i>	accountsList	getPersonalDrawAccount()
43	personalDrawing	personalDrawing(transactions)	<i>calls</i>	accountsList	getBankAccount()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	ng	s)			
44	personalDrawi ng	personalDrawing(transaction s)	<i>calls</i>	client	getAccounts()
45	personalDrawi ng	book()	<i>calls</i>	client	getTaxes()
46	personalDrawi ng	getTodoItems()	<i>calls</i>	client	getClient()
47	personalDrawi ng	book()	<i>calls</i>	contact	getID()
48	personalDrawi ng	book()	<i>calls</i>	entry	entry(Date, String, Double, account, account, tax, String, String)
49	personalDrawi ng	book()	<i>calls</i>	entry	setReferToTransaction(int)
50	personalDrawi ng	book()	<i>calls</i>	entry	book()
51	transaction	save()	<i>calls</i>	application	getOBDXVersionDouble()
52	transaction	getNewTransactionNumber()	<i>calls</i>	client	getClient()
53	transaction	getNewTransactionNumber()	<i>calls</i>	client	getClientPath()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
54	transaction	getNewTransactionNumber())	<i>calls</i>	client	getConfigPath()
55	transaction	save()	<i>calls</i>	client	getTransactions()
56	transaction	save()	<i>calls</i>	client	getClient()
57	transaction	getFilenamePDFGPG()	<i>calls</i>	configs	getGPGPath()
58	transaction	save()	<i>calls</i>	configs	getOrganisationName()
59	transaction	save()	<i>calls</i>	configs	getTaxID()
60	transaction	save()	<i>calls</i>	configs	getAccountCode()
61	transaction	save()	<i>calls</i>	configs	getBankCode()
62	transaction	save()	<i>calls</i>	configs	getBankName()
63	transaction	save()	<i>calls</i>	configs	getHolderName()
64	transaction	transaction(transactions)	<i>calls</i>	contact	contact()
65	transaction	bindToCustomer(int)	<i>calls</i>	contact	bindToID(int)
66	transaction	cloneFrom(transaction)	<i>calls</i>	contact	getID()
67	transaction	save()	<i>calls</i>	contact	getID()
68	transaction	save()	<i>calls</i>	contact	getName()
69	transaction	save()	<i>calls</i>	contact	getNumber()
70	transaction	save()	<i>calls</i>	contact	getCO()
71	transaction	save()	<i>calls</i>	contact	getCountry()
72	transaction	save()	<i>calls</i>	contact	getStreet()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
73	transaction	save()	calls	contact	getLocation()
74	transaction	save()	calls	contact	getPhone()
75	transaction	save()	calls	contact	getFax()
76	transaction	setTransactionAttributes(int, String, String, String, int, Date, Date, int, String)	calls	contact	contact(int)
77	transaction	setTransactionAttributes(int, String, String, String, int, Date, Date, int, String)	calls	contact	bindToID(int)
78	transaction	addItem()	calls	item	item()
79	transaction	getTotal()	calls	item	getTotal()
80	transaction	getTotalGross()	calls	item	getTotalGross()
81	transaction	save()	calls	item	saveForTransaction(int)
82	transaction	save()	calls	item	getOBDXString()
83	transaction	transaction(transactions)	calls	item	items()
84	transaction	emptyItems()	calls	item	emptyAllItems()
85	transaction	loadItemsFromDB()	calls	item	loadFromDB(int)
86	transaction	addItem()	calls	transactionContent Provider	addEntry(item)
87	transaction	getTotal()	calls	transactionContent Provider	elementAt(int)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
88	transaction	getTotal()	<i>calls</i>	transactionContentProvider	size()
89	transaction	getTotalGross()	<i>calls</i>	transactionContentProvider	elementAt(int)
90	transaction	getTotalGross()	<i>calls</i>	transactionContentProvider	size()
91	transaction	save()	<i>calls</i>	transactionContentProvider	elementAt(int)
92	transaction	save()	<i>calls</i>	transactionContentProvider	size()
93	transaction	getVatTotal()	<i>calls</i>	utils	round(Double, int)
94	transaction	transactionTypeNotFoundException(String)	<i>calls</i>	Messages	getString(String)
95	transactions	finishCurrentTransaction()	<i>calls</i>	accountsList	getEntriesFromDatabase()
96	transactions	finishCurrentTransaction()	<i>calls</i>	client	getAccounts()
97	transactions	getIDForNumber(String)	<i>calls</i>	client	getClient()
98	transactions	getTransactionTypesFromDB()	<i>calls</i>	client	getClient()
99	creditmemo		<i>contains</i>	account	
100	invoice		<i>contains</i>	account	
101	personalDrawi		<i>contains</i>	account	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	ng				
102	transaction		<i>contains</i>	contact	
103	transaction		<i>contains</i>	items	
104	creditmemo	field finishCreditAccount	<i>is of type</i>	account	
105	creditmemo	field finishDebitAccount	<i>is of type</i>	account	
106	creditmemo	field pendingCreditAccount	<i>is of type</i>	account	
107	creditmemo	field pendingDebitAccount	<i>is of type</i>	account	
108	invoice	field finishCreditAccount	<i>is of type</i>	account	
109	invoice	field finishDebitAccount	<i>is of type</i>	account	
110	invoice	field pendingCreditAccount	<i>is of type</i>	account	
111	invoice	field pendingDebitAccount	<i>is of type</i>	account	
112	personalDrawi ng	field finishCreditAccount	<i>is of type</i>	account	
113	personalDrawi ng	field finishDebitAccount	<i>is of type</i>	account	
114	personalDrawi ng	field pendingCreditAccount	<i>is of type</i>	account	
115	personalDrawi ng	field pendingDebitAccount	<i>is of type</i>	account	
116	transaction	field recipient	<i>is of type</i>	contact	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
117	transaction	field allItems	<i>is of type</i>	items	
118	cancelation	book()	<i>references</i>	account	
119	cancelation	book()	<i>references</i>	entry	
120	creditmemo	creditmemo(transactions)	<i>references</i>	AccountNotFound Exception	
121	creditmemo	book()	<i>references</i>	account	
122	creditmemo	book()	<i>references</i>	accountException	
123	creditmemo	book()	<i>references</i>	entry	
124	invoice	invoice(transactions)	<i>references</i>	AccountNotFound Exception	
125	invoice	book()	<i>references</i>	AccountNotFound Exception	
126	invoice	book()	<i>references</i>	account	
127	invoice	getTodoItems()	<i>references</i>	account	
128	invoice	book()	<i>references</i>	accountException	
129	invoice	book()	<i>references</i>	entry	
130	invoice	book()	<i>references</i>	item	
131	invoice	book()	<i>references</i>	items	
132	personalDrawi ng	personalDrawing(transaction s)	<i>references</i>	AccountNotFound Exception	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
133	personalDrawi ng	personalDrawingbook()	references	account	
134	personalDrawi ng	personalDrawingbook()	references	accountException	
135	personalDrawi ng	personalDrawingbook()	references	entry	
136	transaction	transaction(transactions)	references	contact	
137	transaction	setTransactionAttributes(int, String, String, String, int, Date, Date, int, String)	references	contact	
138	transaction	getTotal()	references	item	
139	transaction	getTotalGross()	references	item	
140	transaction	save()	references	item	
141	transaction	transaction(transactions)	references	items	
142	transaction	addItem()	references	items	
143	transaction	getTotal()	references	items	
144	transaction	getTotalGross()	references	items	
145	transaction	save()	references	items	
146	cancelation	getCreditAccountForWorkfl owStep(int)	returns	account	
147	cancelation	getDebitAccountForWorkflo	returns	account	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		wStep(int)			
148	creditmemo	getCreditAccountForWorkfl owStep(int)	<i>returns</i>	account	
149	creditmemo	getDebitAccountForWorkflo wStep(int)	<i>returns</i>	account	
150	invoice	getCreditAccountForWorkfl owStep(int)	<i>returns</i>	account	
151	invoice	getDebitAccountForWorkflo wStep(int)	<i>returns</i>	account	
152	offer	getCreditAccountForWorkfl owStep(int)	<i>returns</i>	account	
153	offer	getDebitAccountForWorkflo wStep(int)	<i>returns</i>	account	
154	personalDrawi ng	personalDrawing.getCreditA ccountForWorkflowStep(int)	<i>returns</i>	account	
155	personalDrawi ng	personalDrawing.getDebitA ccountForWorkflowStep(int)	<i>returns</i>	account	
156	reminder	getCreditAccountForWorkfl owStep(int)	<i>returns</i>	account	
157	reminder	getDebitAccountForWorkflo wStep(int)	<i>returns</i>	account	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
158	transaction	getCreditAccountForWorkflowStep(int)	returns	account	
159	transaction	getDebitAccountForWorkflowStep(int)	returns	account	
160	transaction	getRecipient()	returns	contact	
161	transaction	addItem()	returns	item	
162	transaction	getAllProductsAndItems()	returns	items	
163	transaction	getItems()	returns	items	

4.6.4 Code Dependency Package `appLayer.transactionRelated` ke `Package appLayer.taxRelated`

Terdapat 11 *Code Dependency* yang ada pada *package* `appLayer.transactionRelated` ke *package* `appLayer.taxRelated`, *dependency kind* terbanyak adalah *calls* dengan jumlah 8, *class* yang paling sering diakses adalah *class* `taxList`.

Tabel 4. 16 *Code Dependency Package* `appLayer.transactionRelated` ke `Package appLayer.taxRelated`

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	creditmemo	book()	calls	taxList	taxList()
2	creditmemo	book()	calls	taxList	getTaxesFromDatabase()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
3	creditmemo	book()	<i>calls</i>	taxList	getDefault()
4	invoice	book()	<i>calls</i>	tax	getDebitAccount()
5	invoice	book()	<i>calls</i>	taxList	getDefault()
6	personalDrawing	book()	<i>calls</i>	taxList	taxList()
7	personalDrawing	book()	<i>calls</i>	taxList	getTaxesFromDatabase()
8	personalDrawing	book()	<i>calls</i>	taxList	getDefault()
9	creditmemo	book()	<i>references</i>	taxList	
10	invoice	book()	<i>references</i>	taxList	
11	personalDrawing	book()	<i>references</i>	taxList	

4.6.5 Code Dependency Package `appLayer.transactionRelated` ke Package `GUILayer`

Terdapat 1 *Code Dependency* yang ada pada *package* `appLayer.transactionRelated` ke *package* `GUILayer`, *dependency kind* yang ada adalah *calls*, *class* yang diakses adalah *class* `todoWindow`.

Tabel 4. 17 *Code Dependency Package* `appLayer.transactionRelated` ke *Package* `GUILayer`

No.	Source	Dependen	Target
-----	--------	----------	--------

	<i>Class</i>	<i>Method</i>	<i>cy kind</i>	<i>Class</i>	<i>Method</i>
1	transactions	finishCurrentTransaction ()	calls	todoWindow	refreshToDoList()

4.7. Package GUILayer

Bagian ini membahas tentang *dependency code* antar *class* di *package* GUILayer dan juga *dependency code* antara *class* di *package* GUILayer dengan *class* di *package* lain. Tabel 4. 18 adalah *dependency code* antar *class* di *package* GUILayer. Tabel 4. 19 adalah *dependency code* *package* GUILayer dengan *package* dataLayer. Tabel 4. 20 adalah *dependency code* *package* GUILayer dengan *package* appLayer. Tabel 4. 21 adalah *dependency code* *package* GUILayer dengan *package* appLayer.taxRelated. Tabel 4. 22 adalah *dependency code* *package* GUILayer dengan *package* appLayer.transactionRelated.

4.7.1 Code Dependency Antar Class GUILayer

Terdapat 5 *Code Dependency* yang ada pada *package* GUILayer, *dependency kind* terbanyak adalah *calls* dengan jumlah 2, *class* yang paling sering diakses adalah *class* testResult.

Tabel 4. 18 Code Dependency Antar Class GUILayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	HSQLErrorCh	HSQLErrorCheck(config	has param	configWindow	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	eck	Window, String)			
2	HSQLLockCheck	account{...}	references	configWindow	
3	HSQLLockCheck	doTest()	returns	testResult	
4	HSQLLockCheck	doTest()	calls	testResult	testResult(boolean, String)
5	HSQLLockCheck	doTest()	calls	testResult	setFatal()

4.7.2 Code Dependency Package GUILayer ke Package dataLayer

Terdapat 43 *Code Dependency* yang ada pada *package* GUILayer ke *package* dataLayer, *dependency kind* terbanyak adalah *calls* dengan jumlah 35, *class* yang paling sering diakses adalah *class* DB.

Tabel 4. 19 *Code Dependency Package* GUILayer ke *Package* dataLayer

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	HSQLLockCheck	doTest()	calls	DB	getConnection()
2	HSQLLockCheck	doTest()	calls	DB	getDBVersionString()
3	MainWindow	createContents(Compo	calls	DB	getDBVersionDouble()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		site)			
4	MainWindow	createContents(Compo site)	<i>calls</i>	DB	insertRaw(String, HashMap<String, String>)
5	MainWindow	widgetDisposed(Dispo seEvent)	<i>calls</i>	DB	updateRaw(String, String, int, HashMap<String, String>)
6	MainWindow	widgetDisposed(Dispo seEvent)	<i>calls</i>	DB	getConnection()
7	aboutWindow	createContents(Compo site)	<i>calls</i>	DB	getDBVersionString()
8	configWindow	testConnect(Driver, Properties, String, String)	<i>calls</i>	DB	setConnection(Connection)
9	configWindow	getOOMMajorVersion(String)	<i>calls</i>	fileUtils	readFileAsString(String)
10	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	DB	getConnection()
11	contactsWindow	checkBankData()	<i>calls</i>	HBCI	getInstance()
12	entryDetailWindo w	widgetSelected(Selecti onEvent)	<i>calls</i>	OBDXImp orter	OBDXImporter(String)
13	entryDetailWindo w	widgetSelected(Selecti onEvent)	<i>calls</i>	OBDXImp orter	getBankcode()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
14	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	OBDXImporter	getBankaccount()
15	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	OBDXImporter	getAccountholder()
16	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	OBDXImporter	getUsage()
17	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	OBDXImporter	getValue()
18	newAccountingWizardImport	performHBCIImport()	<i>calls</i>	HBCIImporter	HBCIImporter(Date, Date)
19	newAccountingWizardImport	performMoneyplexImport(String)	<i>calls</i>	moneyplexImporter	moneyplexImporter(String)
20	newAccountingWizardImport	performStarmoneyImport(String)	<i>calls</i>	starmoneyImporter	starmoneyImporter(String)
21	newAccountingWizardImport	createControl(Composite)	<i>calls</i>	DB	getConnection()
22	productsWindow	productsWindow.getVATIDs(int)	<i>calls</i>	DB	getConnection()
23	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	DB	getConnection()
24	settingsWindow	checkBankDetails()	<i>calls</i>	HBCI	getInstance()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
25	tableCheck	tableCheck.doTest()	<i>calls</i>	DB	getConnection()
26	tableCheck	tableCheck.doTest()	<i>calls</i>	prepareDB	prepareDB()
27	tableCheck	tableCheck.doTest()	<i>calls</i>	prepareDB	createDB(Statement, List)
28	taxesWindow	widgetSelected(SelectonEvent)	<i>calls</i>	DB	getConnection()
29	templateCheck	templateCheck.doTest() ()	<i>calls</i>	DB	getConnection()
30	templateCheck	templateCheck.doTest() ()	<i>calls</i>	DB	getIdentifierQuote()
31	timetrackerImportWindow	widgetSelected(SelectonEvent)	<i>calls</i>	fileImporters	importKTimeTracker(String, StatusLineManager, Shell)
32	timetrackerImportWindow	widgetSelected(SelectonEvent)	<i>calls</i>	fileImporters	importTaskCoach(String, StatusLineManager, Shell)
33	todoBalanceSelectionAdapter	widgetSelected(SelectonEvent)	<i>calls</i>	DB	getConnection()
34	todoWindow	checkTableForDefaultValue(boolean, String, String, String, SelectionAdapter)	<i>calls</i>	DB	getConnection()
35	todoWindow	checkUnprocessedOffers(boolean)	<i>calls</i>	DB	getConnection()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
36	entryDetailWindow		<i>contains</i>	OBDXImporter	
37	entryDetailWindow	field oi	<i>is of type</i>	OBDXImporter	
38	entryDetailWindow	widgetSelected(SelectIonEvent)	<i>references</i>	OBDXImporter	
39	entryDetailWindow	widgetSelected(SelectIonEvent)	<i>references</i>	prematureException	
40	newAccountingWizardImport	performHBCIImport()	<i>references</i>	HBCIImporter	
41	newAccountingWizardImport	performMoneyplexImport(String)	<i>references</i>	moneyplexImporter	
42	newAccountingWizardImport	performStarMoneyImport(String)	<i>references</i>	starMoneyImporter	
43	tableCheck	doTest()	<i>references</i>	prepareDB	

4.7.3 Code Dependency Package GUILayer ke Package appLayer

Terdapat 575 Code Dependency yang ada pada package GUILayer ke package appLayer, dependency kind terbanyak adalah *calls* dengan jumlah 466, class yang paling sering diakses adalah class configs.

Tabel 4. 20 Code Dependency Package GUILayer ke Package appLayer

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	MainWindow	getVersionHistory()	<i>accesses</i>	application	field onHistory
2	HSQLErrorCheck	doTest()	<i>calls</i>	client	getGlobalDataPath()
3	MainWindow	configureShell(Shell) calls	<i>calls</i>	application	getAppName()
4	MainWindow	createContents(Compo site)	<i>calls</i>	application	getVersionDouble()
5	MainWindow	<i>method</i> GUILayer.MainWindo w.createContents(Com posite)	<i>calls</i>	application	<i>method</i> appLayer.application.getVersionS tring()
6	MainWindow	widgetDisposed(Dispo seEvent)	<i>calls</i>	application	getAppName()
7	MainWindow	createContents(Compo site)	<i>calls</i>	client	getClient()
8	MainWindow	setConfigPath(String)	<i>calls</i>	client	setConfigPath(String)
9	MainWindow	createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
10	MainWindow	createContents(Compo site)	<i>calls</i>	configs	startWithCommunity()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
11	MainWindow	widgetDisposed(DisposeEvent) calls	<i>calls</i>	configs	disposeOfficeApplication()
12	MainWindow	widgetDisposed(DisposeEvent)	<i>calls</i>	configs	getDatabaseType()
13	MainWindow	createContents(Composite)	<i>calls</i>	contextComposite	contextComposite(Composite, int)
14	MainWindow	createContents(Composite)	<i>calls</i>	contextComposite	setCurrentHelpURL(String)
15	OOoCheck	doTest()	<i>calls</i>	configs	getOOoPath()
16	exportAction	widgetSelected(SelectionEvent)	<i>calls</i>	account	getID()
17	exportAction	widgetSelected(SelectionEvent)	<i>calls</i>	account	getBalance()
18	VATAnnouncementWindow	createContents(Composite)	<i>calls</i>	account	accountsList.()
19	VATAnnouncementWindow	createContents(Composite)		accountsList	getAccountsFromDatabase()
20	VATAnnouncementWindow	createContents(Composite)	<i>calls</i>	accountsList	getYearsCovered()
21	VATAnnouncementWindow	isOpenable()	<i>calls</i>	accountsList	isEmpty()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
22	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	accountsList	setPeriod(Date, Date)
23	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	accountsList	getEntriesFromDatabase()
24	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	accountsList	getAccountForID(int)
25	VATannouncementWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
26	VATannouncementWindow	isOpenable()	<i>calls</i>	client	getAccounts()
27	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	client	getAccounts()
28	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	client	getTaxes()
29	VATannouncementWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
30	VATannouncementWindow	createContents(Composite)	<i>calls</i>	configs	getVATPeriod()
31	VATannouncementWindow	widgetSelected(SelectonEvent)	<i>calls</i>	configs	getTaxOfficeListIDX()
32	VATannouncement	widgetSelected(SelectonEvent)	<i>calls</i>	configs	getStateListIDX()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	ntWindow	onEvent)			
33	VATannounceme ntWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	getTaxID()
34	VATannounceme ntWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	getOrganisationName()
35	VATannounceme ntWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	getVATPeriod()
36	VATannounceme ntWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	getWinstonPath()
37	VATannounceme ntWindow	getValueForField(int, Double, tax)	<i>calls</i>	utils	utils.round(Double, int)
38	aboutWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
39	aboutWindow	createContents(Compo site)	<i>calls</i>	application	getVersionString()
40	aboutWindow	createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
41	accountDefinition Window	createContents(Compo site)	<i>calls</i>	account	getTypes()
42	accountDefinition Window	createContents(Compo site)	<i>calls</i>	account	getSubAccounts()
43	accountDefinition	updateControlsFromSe	<i>calls</i>	account	getID()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	Window	lection()			
44	accountDefinition Window	updateControlsFromSe lection()	<i>calls</i>	account	getCode()
45	accountDefinition Window	updateControlsFromSe lection()	<i>calls</i>	account	getDescription()
46	accountDefinition Window	updateControlsFromSe lection()	<i>calls</i>	account	getType()
47	accountDefinition Window	updateControlsFromSe lection()	<i>calls</i>	account	getSubAccountTypesCode()
48	accountDefinition Window	widgetSelected(Selecti onEvent)	<i>calls</i>	account	update(String, String, int, int)
49	accountDefinition Window	widgetSelected(Selecti onEvent)	<i>calls</i>	account	save()
50	accountDefinition Window	widgetSelected(Selecti onEvent)	<i>calls</i>	account	delete()
51	accountDefinition Window	updateControlsFromSe lection()	<i>calls</i>	accountsLis t	getDefaultAccount()
52	accountDefinition Window	widgetSelected(Selecti onEvent)	<i>calls</i>	accountsLis t	getNumAccounts()
53	accountDefinition Window	ContentProvider(accou ntDefinitionWindow)	<i>calls</i>	accountsLis t	addEmptyAccount()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
54	accountDefinition Window	getElements(Object)	<i>calls</i>	accountsLis t	getAccounts()
55	accountDefinition Window	configureShell(Shell)	<i>calls</i>	application	getAppName()
56	accountDefinition Window	ContentProvider(accou ntDefinitionWindow)	<i>calls</i>	client	getAccounts()
57	accountDefinition Window	getAllAccounts()	<i>calls</i>	client	getAccounts()
58	accountDefinition Window	getElements(Object)	<i>calls</i>	client	getAccounts()
59	accountDefinition Window	createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
60	accountingEditWi ndow	createContents(Compo site)	<i>calls</i>	accountsLis t	getStringArray()
61	accountingEditWi ndow	refreshAccountingEntri es()	<i>calls</i>	accountsLis t	getEntriesFromDatabase()
62	accountingEditWi ndow	refreshAccountingEntri es()	<i>calls</i>	accountsLis t	getJournal()
63	accountingEditWi ndow	configureShell(Shell)	<i>calls</i>	application	getAppName()
64	accountingEditWi	createContents(Compo	<i>calls</i>	bookingCel	bookingCellModifier(TableViewe

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	indow	site)		lModifier	r)
65	accountingEditWindow	createContents(Composite)	<i>calls</i>	bookingContentProvider	bookingContentProvider()
66	accountingEditWindow	refreshAccountingEntries()	<i>calls</i>	bookingContentProvider	removeAllEntries()
67	accountingEditWindow	refreshAccountingEntries()	<i>calls</i>	bookingContentProvider	addEntry(entry)
68	accountingEditWindow	createContents(Composite)	<i>calls</i>	bookingContentProvider	bookingLabelProvider(Display)
69	accountingEditWindow	createContents(Composite)	<i>calls</i>	client	getAccounts()
70	accountingEditWindow	refreshAccountingEntries()	<i>calls</i>	client	getAccounts()
71	accountingEditWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
72	accountingEditWindow	createContents(Composite)	<i>calls</i>	entry	entry()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
73	accountingEditWi ndow	createContents(Compo site)	<i>calls</i>	entry	removeColumn(String)
74	accountingEditWi ndow	createContents(Compo site)	<i>calls</i>	entry	getColumnNames()
75	accountingEditWi ndow	refreshAccountingEntri es()	<i>calls</i>	entry	setAutoUpdate(boolean)
76	accountingEditWi ndow	refreshAccountingEntri es()	<i>calls</i>	entry	hasColumn(String)
77	accountingEditWi ndow	refreshAccountingEntri es()	<i>calls</i>	entry	removeColumn(String)
78	accountingEditWi ndow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	delete()
79	assetWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
80	assetWindow	updateControlsFromSe lection()	<i>calls</i>	asset	getName()
81	assetWindow	updateControlsFromSe lection()	<i>calls</i>	asset	getValue()
82	assetWindow	updateControlsFromSe lection()	<i>calls</i>	asset	getID()
83	assetWindow	updateControlsFromSe lection()	<i>calls</i>	asset	getDepreciationStart()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
84	assetWindow	updateControlsFromSelection()	<i>calls</i>	asset	getDepreciationEnd()
85	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	setName(String)
86	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	setValue(Double)
87	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	setDepreciationStart(Date)
88	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	setDepreciationEnd(Date)
89	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	save()
90	assetWindow	widgetSelected(SelectionEvent)	<i>calls</i>	asset	delete()
91	assetWindow	ContentProvider(assetWindow)	<i>calls</i>	asset	assets()
92	assetWindow	ContentProvider(assetWindow)	<i>calls</i>	asset	addEmptyAsset()
93	assetWindow	ContentProvider(assetWindow)	<i>calls</i>	asset	getAssetsFromDB()
94	assetWindow	getElements(Object)	<i>calls</i>	asset	getAssets()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
95	assetWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
96	balanceWindow	load(int)	<i>calls</i>	account	getAsString()
97	balanceWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
98	balanceWindow	load(int)	<i>calls</i>	client	getTransactions()
99	balanceWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
100	balanceWindow	widgetSelected(SelectonEvent)	<i>calls</i>	configs	<i>method</i> appLayer.configs.getBankCode()
101	balanceWindow	widgetSelected(SelectonEvent)	<i>calls</i>	configs	getAccountCode()
102	balanceWindow	widgetSelected(SelectonEvent)	<i>calls</i>	configs	getHolderName()
103	balanceWindow	load(int)	<i>calls</i>	contact	getName()
104	balanceWindow	load(int)	<i>calls</i>	contact	getPayment <i>method</i> ()
105	balanceWindow	widgetSelected(SelectonEvent)	<i>calls</i>	contact	getAccountholder()
106	balanceWindow	<i>method</i> widgetSelected(SelectonEvent)	<i>calls</i>	contact	getBankCode()
107	balanceWindow	widgetSelected(SelectonEvent)	<i>calls</i>	contact	getBankAccount()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		onEvent)			
108	balanceWindow	load(int)	<i>calls</i>	contact	getPaymentIDForString(String)
109	balanceWindow	load(int)	<i>calls</i>	entry	entry(int)
110	balanceWindow	load(int)	<i>calls</i>	entry	getReferToTransaction()
111	balanceWindow	load(int)	<i>calls</i>	entry	getValue()
112	balanceWindow	load(int)	<i>calls</i>	entry	getReference()
113	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	clone()
114	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	getReferToTransaction()
115	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	setCreditAccount(account)
116	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	setDebitAccount(account)
117	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	setDate(Date)
118	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	book()
119	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	moneyTran sfer	<i>method</i> appLayer.moneyTransfer()
120	balanceWindow	widgetSelected(Selecti	<i>calls</i>	moneyTran	performMoneyTransfer(transactio

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		onEvent)		sfer	n)
121	browserWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
122	communityWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
123	communityWindow	createContents(Compo site)	<i>calls</i>	application	getVersionDouble()
124	communityWindow	createContents(Compo site)	<i>calls</i>	configs	startWithCommunity()
125	configWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
126	configWindow	test()	<i>calls</i>	client	getDataPath()
127	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseType()
128	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseServer()
129	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseName()
130	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseDriverFileName()
131	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseDriverName()
132	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabasePassword()
133	configWindow	connectDatabase()	<i>calls</i>	configs	getDatabaseUser()
134	configWindow	createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
135	configWindow	<i>method</i>	<i>calls</i>	configs	getWinstonPath()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		GUILayer.configWindow.createContents(Composite)			
136	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	shallUseExternalDB()
137	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseName()
138	configWindow	<i>method</i> selectValuesFromConfigFile()	<i>calls</i>	configs	getPrinterName()
139	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseUser()
140	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabasePassword()
141	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseDriverName()
142	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseType()
143	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseDriverFileName()
144	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getDatabaseServer()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		gFile()			
145	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	shallPrintPoweredBy()
146	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getOOoPath()
147	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getCtAPI()
148	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	shallUseCardReaderPINPad()
149	configWindow	selectValuesFromConfigFile()	<i>calls</i>	configs	getGPGPath()
150	configWindow	testConnect(Driver, Properties, String, String)	<i>calls</i>	configs	getDatabaseType()
151	configWindow	testConnect(Driver, Properties, String, String)	<i>calls</i>	configs	getDatabaseName()
152	configWindow	widgetSelected(SelectionEvent)	<i>calls</i>	configs	setPrinterName(String)
153	configWindow	widgetSelected(SelectionEvent)	<i>calls</i>	configs	setDatabaseName(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
154	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setDatabaseDriverName(String)
155	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setDatabaseDriverFileName(Strin g)
156	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	<i>method</i> appLayer.configs.setDatabasePass word(String)
157	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setDatabaseServer(String)
158	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setDatabaseUser(String)
159	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setDatabaseType(String)
160	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setOOoPath(String)
161	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setPrintPoweredBy(boolean)
162	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setUseExternalDB(boolean)
163	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setWinstonPath(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
164	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setGPGPath(String)
165	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setCtAPI(String)
166	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setUseCardReaderPINpad(boolea n)
167	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	writeConfiguration()
168	configWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	readConfiguration()
169	configWindow	handleEvent(Event)	<i>calls</i>	contextCo mposite	getCurrentHelpURL()
170	contactsWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
171	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	client	getClient()
172	contactsWindow	<i>method</i> createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
173	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getID()
174	contactsWindow	updateControlsFromSe	<i>calls</i>	contact	getName()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		lection()			
175	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getCO()
176	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getStreet()
177	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getZIP()
178	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getLocation()
179	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getCountry()
180	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getPhone()
181	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getFax()
182	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getEmail()
183	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	getVATID()
184	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	contact.getAccountholder()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
185	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	contact.getBankAccount()
186	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	contact.getBankCode()
187	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	contact.getRole()
188	contactsWindow	updateControlsFromSe lection()	<i>calls</i>	contact	contact.getPaymentmethod()
189	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	contact.setName(String)
190	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	contact.setStreet(String)
191	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setZIP(String)
192	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setLocation(String)
193	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setCountry(String)
194	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setCO(String)
195	contactsWindow	widgetSelected(Selecti	<i>calls</i>	contact	setEmail(String)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		onEvent)			
196	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setPhone(String)
197	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setFax(String)
198	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setVATID(String)
199	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setRole(int)
200	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setBankaccount(String)
201	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setBankcode(String)
202	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setAccountholder(String)
203	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	setPaymentmethod(int)
204	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	save()
205	contactsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	contact	delete()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
206	contactsWindow	createContents(Compo site)	<i>calls</i>	contacts	getPaymentMethods()
207	contactsWindow	createContents(Compo site)	<i>calls</i>	contacts	getRoles()
208	ContentProvider	ContentProvider(contact sWindow)	<i>calls</i>	contacts	contacts()
209	ContentProvider	ContentProvider(contact sWindow)	<i>calls</i>	contacts	addEmptyContact()
210	ContentProvider	ContentProvider(contact sWindow)	<i>calls</i>	contacts	getContactsFromDB()
211	ContentProvider	getElements(Object)	<i>calls</i>	contacts	getContacts()
212	databaseDriverNameTestCase	doTest()	<i>calls</i>	configs	getDatabaseDriverName()
213	designerWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
214	designerWindow	<i>method</i> GUILayer.designerWindow.createContents(Compo site)	<i>calls</i>	client	getTransactions()
215	designerWindow	loadTemplates()	<i>calls</i>	client	getDataPath()
216	designerWindow	createContents(Compo site)	<i>calls</i>	configs	getOfficeApplication()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
217	designerWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
218	designerWindow	createContents(Composite)	<i>calls</i>	contextComposite	contextComposite(Composite, int)
219	designerWindow	createContents(Composite)	<i>calls</i>	contextComposite	setCurrentHelpURL(String)
220	documentsWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
221	documentsWindow	createContents(Composite)	<i>calls</i>	client	getDocuments()
222	documentsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getDocuments()
223	documentsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	document	save()
224	documentsWindow.documentCellModifier.	canModify(Object, String)	<i>calls</i>	document	getChangeValueForColumnName(String)
225	documentsWindow.documentCellModifier	getValue(Object, String)	<i>calls</i>	document	getChangeValueForColumnName(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
226	documentsWindow.documentCellModifier	modify(Object, String, Object)	<i>calls</i>	document	setChangedValueForColumnName(String, Object)
227	documentsWindow.documentCellModifier	getColumnText(Object, int)	<i>calls</i>	document	getValueForColumnIndex(Object, int)
228	documentsWindow	createContents(Composite)	<i>calls</i>	document	getDocumentsFromDatabase()
229	documentsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	document	getNewDocument(String)
230	entryDetailWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
231	entryDetailWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
232	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	moneyTransfer	moneyTransfer()
233	entryDetailWindow	widgetSelected(SelectionEvent)	<i>calls</i>	moneyTransfer	transferMoneyTo(String, String, String, String, double)
234	mainDirTestCase	doTest()	<i>calls</i>	client	getDataPath()
235	newAccountingWizardAdd	newAccountingWizardAdd(newAccountingW	<i>calls</i>	accountsList	getAccountsFromDatabase()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		izard)			
236	newAccountingWizardAdd	newAccountingWizardAdd(newAccountingWizard)	<i>calls</i>	accountsList	addPleaseSelectAccount()
237	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	accountsList	accountsList.getStringArray()
238	newAccountingWizardAdd	setVisible(boolean)	<i>calls</i>	accountsList	accountsList.getImportEntries()
239	newAccountingWizardAdd	widgetSelected(SelectionEvent)	<i>calls</i>	accountsList	accountsList.getDefaultAccount()
240	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	bookingCellModifier	bookingCellModifier(TableViewer)
241	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	bookingContentProvider	bookingContentProvider()
242	newAccountingWizardAdd	setVisible(boolean)	<i>calls</i>	bookingContentProvider	addEntry(entry)
243	newAccountingWizardAdd	widgetSelected(SelectionEvent)	<i>calls</i>	bookingContentProvider	addEntry(entry)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
244	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	bookingLabelProvider	bookingLabelProvider(Display)
245	newAccountingWizardAdd	newAccountingWizardAdd(newAccountingWizard)	<i>calls</i>	client	getTaxes()
246	newAccountingWizardAdd	newAccountingWizardAdd(newAccountingWizard)	<i>calls</i>	client	getAccounts()
247	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	client	getDocuments()
248	newAccountingWizardAdd	setVisible(boolean)	<i>calls</i>	client	<i>method</i> appLayer.client.getAccounts()
249	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	configs	getDefaultFont()
250	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	documents	getUnlinkedNumbers()
251	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	entry	entry()
252	newAccountingWizardAdd	createControl(Composite)	<i>calls</i>	entry	getColumnNames()
253	newAccountingWizardAdd	performBookings()	<i>calls</i>	entry	isEmpty()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	zardAdd				
254	newAccountingWizardAdd	performBookings()	<i>calls</i>	entry	book()
255	newAccountingWizardAdd	performBookings()	<i>calls</i>	entry	removeFromImport(boolean)
256	newAccountingWizardAdd	widgetSelected(SelectionEvent)	<i>calls</i>	entry	entry(Date, String, Double, account, account, tax, String, String)
257	newAccountingWizardAdd	widgetSelected(SelectionEvent)	<i>calls</i>	entry	removeFromImport(boolean)
258	newAccountingWizardAdd	dragSetData(DragSourceEvent)	<i>calls</i>	entry	getImportID()
259	newAccountingWizardImport	createControl(Composite)	<i>calls</i>	configs	getDefaultFont()
260	newAccountingWizardImport	performHBCIImport()	<i>calls</i>	configs	getCtAPI()
261	newTransactionSelectItemDetails	createControl(Composite)	<i>calls</i>	client	getTransactions()
262	newTransactionSelectItemDetails	createControl(Composite)	<i>calls</i>	configs	getDefaultFont()
263	newTransactionSe	checkPageComplete()	<i>calls</i>	item	isEmpty()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	lectItemDetails				
264	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	item	getColumnNames()
265	newTransactionSe lectItemDetails	widgetSelected(Selecti onEvent)	<i>calls</i>	item	removeItem()
266	newTransactionSe lectItemDetails	widgetSelected(Selecti onEvent)	<i>calls</i>	item	item()
267	newTransactionSe lectItemDetails	getValue(Object, String)	<i>calls</i>	item	getColumnNames()
268	newTransactionSe lectItemDetails	getValue(Object, String)	<i>calls</i>	item	getColumn(int)
269	newTransactionSe lectItemDetails	modify(Object, String, Object)	<i>calls</i>	item	getColumnNames()
270	newTransactionSe lectItemDetails	modify(Object, String, Object)	<i>calls</i>	item	getColumnType(int)
271	newTransactionSe lectItemDetails	modify(Object, String, Object)	<i>calls</i>	item	setColumn(int, Object)
272	newTransactionSe lectItemDetails	getColumnText(Object, int)	<i>calls</i>	item	getColumnString(int)
273	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	item	bindTableView(TableView)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
274	newTransactionSelectItemDetails	createControl(Composite)	calls	products	products()
275	newTransactionSelectItemDetails	createControl(Composite)	calls	products	getProductsFromDB()
276	newTransactionSelectItemDetails	createControl(Composite)	calls	products	addPleaseSelectProduct()
277	newTransactionSelectItemDetails	createControl(Composite)	calls	products	getStringArray()
278	newTransactionSelectItemDetails	modify(Object, String, Object)	calls	products	getProductForListID(int)
279	newTransactionSelectItemDetails	checkPageComplete()	calls	transactionContentProvider	size()
280	newTransactionSelectItemDetails	checkPageComplete()	calls	transactionContentProvider	getItems()
281	newTransactionSelectItemDetails	createControl(Composite)	calls	transactionContentProvider	bindToPage(newTransactionSelectItemDetails)
282	newTransactionSelectItemDetails	widgetSelected(SelectionEvent)	calls	transactionContentPro	addEntry(item)

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
				vider	
283	newTransactionSe lectItemDetails	getValue(Object, String)	<i>calls</i>	utils	findIndexOfStringInStringArray(String[], String)
284	newTransactionSe lectItemDetails	modify(Object, String, Object)	<i>calls</i>	utils	findIndexOfStringInStringArray(String[], String)
285	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	client	getTransactions()
286	newTransactionSe lectItemDetails	fillTransactionChoice()	<i>calls</i>	client	getTransactions()
287	newTransactionSe lectItemDetails	getNextPage()	<i>calls</i>	client	getTransactions()
288	newTransactionSe lectItemDetails	loadFromDB(int)	<i>calls</i>	client	getTransactions()
289	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	configs	getDefaultFont()
290	newTransactionSe lectItemDetails	fillCustomerChoice(Co nnection)	<i>calls</i>	contact	getID()
291	newTransactionSe lectItemDetails	fillCustomerChoice(Co nnection)	<i>calls</i>	contact	getName()
292	newTransactionSe lectItemDetails	loadFromDB(int)	<i>calls</i>	contact	getID()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
293	newTransactionSelectItemDetails	fillCustomerChoice(Connection)	calls	contact	contacts()
294	newTransactionSelectItemDetails	fillCustomerChoice(Connection)	calls	contact	getContactsFromDB()
295	newTransactionSelectItemDetails	fillCustomerChoice(Connection)	calls	contact	getContacts()
296	newTransactionWizard	run(IProgressMonitor)	calls	application	getAppName()
297	newTransactionWizard	performFinish()	calls	client	getTransactions()
298	newTransactionWizard	performFinish()	calls	configs	getPrinterName()
299	newTransactionWizard	performFinish()	calls	configs	getSMTPServer()
300	newTransactionWizard	performFinish()	calls	configs	getSenderEmail()
301	newTransactionWizard	performFinish()	calls	configs	getSMTPUsername()
302	newTransactionWizard	performFinish()	calls	configs	getSMTPPassword()
303	newTransactionW	performFinish()	calls	configs	shallUseSMTPAuth()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	izard				
304	newTransactionW izard	run(IProgressMonitor)	<i>calls</i>	configs	getSMTPServer()
305	newTransactionW izard	run(IProgressMonitor)	<i>calls</i>	configs	shallUseSMTPAuth()
306	newTransactionW izard	run(IProgressMonitor)	<i>calls</i>	configs	getSenderEmail()
307	newTransactionW izard	performFinish()	<i>calls</i>	contact	getEmail()
308	newTransactionW izardPreview	insertHeader()	<i>calls</i>	client	getTransactions()
309	newTransactionW izardPreview	insertHeader()	<i>calls</i>	client	getDataPath()
310	newTransactionW izardPreview	process()	<i>calls</i>	client	getTransactions()
311	newTransactionW izardPreview	createControl(Composi te)	<i>calls</i>	configs	getOfficeApplication()
312	newTransactionW izardPreview	process()	<i>calls</i>	configs	shallPrintPoweredBy()
313	newTransactionW izardPreview	process()	<i>calls</i>	itemTableL ist	itemTableList(Shell, ITextDocument, transaction,

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
					placeholderManager, ITextField[])
314	newTransactionWizardPreview	process()	calls	itemTableList	itemTableList.process()
315	newTransactionWizardPreview	process()	calls	itemTextList	itemTextList(Shell, ITextDocument, transaction, placeholderManager, ITextField[])
316	newTransactionWizardPreview	process()	calls	itemTextList	itemTextList.process()
317	newTransactionWizardPreview	doReplaceItemless(placeholderManager, transaction)	calls	placeholderManager	getPlaceholderNames()
318	newTransactionWizardPreview	doReplaceItemless(placeholderManager, transaction)	calls	placeholderManager	getPlaceholderValues(item, transaction)
319	newTransactionWizardPreview	process()	calls	placeholderManager	placeholderManager.()
320	newTransactionWizardPrintAndSave	getNextPage()	calls	client	getTransactions()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
321	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	client	getDocuments()
322	newTransactionWizardPrintAndSave	createControl(Composite)	<i>calls</i>	configs	getDefaultFont()
323	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	configs	getGPGPath()
324	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	document	setNumber(String)
325	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	document	setEntryID(int)
326	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	document	save()
327	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	document	getNewDocument(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
328	numbersWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
329	numbersWindow	<i>method</i> GUILayer.numbersWindow.updateControlsFromSelection()	<i>calls</i>	client	getTransactions()
330	numbersWindow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getTransactions()
331	numbersWindow	ContentProvider(numbersWindow)	<i>calls</i>	client	getTransactions()
332	numbersWindow		<i>calls</i>		getDefaultFont()
333	passwordDialog	createDialogArea(Composite)	<i>calls</i>	configs	getDefaultFont()
334	productsWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
335	productsWindow	productsWindow.getVATIDs(int)	<i>calls</i>	client	getClient()
336	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getClient()
337	productsWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
338	productsWindow	updateControlsFromSelection()	<i>calls</i>	product	getID()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
339	productsWindow	updateControlsFromSelection()	<i>calls</i>	product	getName()
340	productsWindow	updateControlsFromSelection()	<i>calls</i>	product	getPrice()
341	productsWindow	updateControlsFromSelection()	<i>calls</i>	product	getVAT()
342	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	product	setPrice(double)
343	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	product	setName(String)
344	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	product	setTax(tax)
345	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	product	save()
346	productsWindow	<i>method</i> GUILayer.productsWindow.2.widgetSelected(SelectionEvent)	<i>calls</i>	product	delete()
347	productsWindow	ContentProvider(productsWindow)	<i>calls</i>	products	products()
348	productsWindow	ContentProvider(productsWindow)	<i>calls</i>	products	addEmptyProduct()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		ctsWindow)			
349	productsWindow	ContentProvider(productsWindow)	<i>calls</i>	products	getProductsFromDB()
350	productsWindow	getElements(Object)	<i>calls</i>	products	getProducts()
351	productsWindow	isOpenable()	<i>calls</i>	accountsList	accountsList.isEmpty()
352	reportWizard	isOpenable()	<i>calls</i>	client	getAccounts()
353	reportWizardSelect	createControl(Composite)	<i>calls</i>	accountsList	getCompletePeriodStart()
354	reportWizardSelect	createControl(Composite)	<i>calls</i>	accountsList	getCompletePeriodEnd()
355	reportWizardSelect	getNextPage()	<i>calls</i>	accountsList	setPeriod(Date, Date)
356	reportWizardSelect	createControl(Composite)	<i>calls</i>	client	getAccounts()
357	reportWizardSelect	getNextPage()	<i>calls</i>	client	getAccounts()
358	reportWizardSelect	reportAccountsList()	<i>calls</i>	configs	getDefaultFont()
359	.reportWizardShow	<i>method</i> GUILayer.reportWizard	<i>calls</i>	account	getCode()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		dShow.reportAccountsList()			
360	reportWizardSelect	reportAccountsList()	<i>calls</i>	account	getDescription()
361	reportWizardShow	reportAccountsList()	<i>calls</i>	account	getTypeString()
362	reportWizardShow	reportAccountsList()	<i>calls</i>	account	getBalance()
363	reportWizardShow	reportLists()	<i>calls</i>	account	getEntries()
364	reportWizardShow	reportLists()	<i>calls</i>	account	getAsString()
365	reportWizardShow	reportTAccounts()	<i>calls</i>	account	hasEntries()
366	reportWizardShow	reportTAccounts()	<i>calls</i>	account	getOpeningBalance()
367	reportWizardShow.tAccount	Account(reportWizardShow, account)	<i>calls</i>	account	getAsString()
368	reportWizardShow.tAccount	tAccount(reportWizardShow, account)	<i>calls</i>	account	getOpeningBalance()
369	reportWizardShow	<i>method</i>	<i>calls</i>	account	getEntries()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	w.tAccount	GUILayer.reportWizardShow.tAccount.(reportWizardShow, account)			
370	reportWizardShow.tAccount	tAccount(reportWizardShow, account)	<i>calls</i>	account	getBalance()
371	reportWizardShow.tAccount	reportAccountsList()	<i>calls</i>	account	getAccounts()
372	reportWizardShow.tAccount	reportJournal()	<i>calls</i>	account	getJournal()
373	reportWizardShow.tAccount	reportLists()	<i>calls</i>	accountsList	getAccounts()
374	reportWizardShow.tAccount	reportTAccounts()	<i>calls</i>	accountsList	getAccounts()
375	reportWizardShow.tAccount	setVisible(boolean)	<i>calls</i>	accountsList	getEntriesFromDatabase()
376	reportWizardShow.tAccount	reportAccountsList()	<i>calls</i>	client	getAccounts()
377	reportWizardShow.tAccount	reportJournal()	<i>calls</i>	client	getAccounts()
378	reportWizardShow.tAccount	reportLists()	<i>calls</i>	client	getAccounts()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
379	reportWizardShow	reportTAccounts()	<i>calls</i>	client	getAccounts()
380	reportWizardShow	setVisible(boolean)	<i>calls</i>	client	getAccounts()
381	reportWizardShow	createControl(Composite)	<i>calls</i>	configs	getOfficeApplication()
382	reportWizardShow	reportJournal()	<i>calls</i>	entry	getID()
383	reportWizardShow	reportJournal()	<i>calls</i>	entry	getNumber()
384	reportWizardShow	reportJournal()	<i>calls</i>	entry	getDate()
385	reportWizardShow	reportJournal()	<i>calls</i>	entry	getAbsoluteValue()
386	reportWizardShow	reportJournal()	<i>calls</i>	entry	getAccountAsString()
387	reportWizardShow	reportJournal()	<i>calls</i>	entry	getContraAccountAsString()
388	reportWizardShow	reportJournal()	<i>calls</i>	entry	getReference()
389	reportWizardShow	reportJournal()	<i>calls</i>	entry	getDescription()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	w				
390	reportWizardShow	reportJournal()	<i>calls</i>	entry	getComment()
391	reportWizardShow	reportLists()	<i>calls</i>	entry	getID()
392	reportWizardShow	reportLists()	<i>calls</i>	entry	getNumber()
393	reportWizardShow	reportLists()	<i>calls</i>	entry	getDate()
394	reportWizardShow	reportLists()	<i>calls</i>	entry	getAccountAsString()
395	reportWizardShow	reportLists()	<i>calls</i>	entry	getContraAccountAsString()
396	reportWizardShow	reportLists()	<i>calls</i>	entry	getReference()
397	reportWizardShow	reportLists()	<i>calls</i>	entry	getDescription()
398	reportWizardShow	reportLists()	<i>calls</i>	entry	getComment()
399	reportWizardShow	reportLists()	<i>calls</i>	entry	isCreditSide(account)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
400	reportWizardShow	reportLists()	<i>calls</i>	entry	getAbsoluteValue()
401	reportWizardShow.tAccount	tAccount(reportWizardShow, account)	<i>calls</i>	entry	getAbsoluteValue()
402	reportWizardShow.tAccount	tAccount(reportWizardShow, account)	<i>calls</i>	entry	getID()
403	reportWizardShow.tAccount	tAccount(reportWizardShow, account)	<i>calls</i>	entry	isCreditSide(account)
404	reportWizardShow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getDataPath()
405	settingsWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
406	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
407	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getOrganisationName()
408	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getTaxID()
409	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getStateListIDX()
410	settingsWindow	<i>method</i> GUILayer.settingsWin	<i>calls</i>	configs	getTaxOfficeListIDX()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		dow.createContents(Composite)			
411	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getVATPeriod()
412	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getSMTPServer()
413	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getSMTPUsername()
414	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getSMTPPassword()
415	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getSenderEmail()
416	settingsWindow	createContents(Composite)	<i>calls</i>	configs	shallUseSMTPAuth()
417	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getBankName()
418	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getBankCode()
419	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getAccountCode()
420	settingsWindow	createContents(Composite)	<i>calls</i>	configs	getHolderName()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		site)			
421	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setOrganizationName(String)
422	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setTaxID(String)
423	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setStateListIDX(Integer)
424	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setTaxOfficeListIDX(Integer)
425	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setVATPeriod(String)
426	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setSMTPServer(String)
427	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setSMTPUsername(String)
428	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setSMTPPassword(String)
429	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setUseSMTPAuth(boolean)
430	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setEmailSender(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
431	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setAccountCode(String)
432	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	<i>method</i> appLayer.configs.setHolderName (String)
433	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setBankCode(String)
434	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	setBankName(String)
435	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	writeSettings()
436	settingsWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	configs	readSettings()
437	tableCheck	tableCheck.doTest()	<i>calls</i>	client	getClient()
438	taxesWindow	createContents(Compo site)	<i>calls</i>	account	getAsString()
439	taxesWindow	updateControlsFromSe lection()	<i>calls</i>	account	getIndexInStringArray()
440	taxesWindow	createContents(Compo site)	<i>calls</i>	accountsLis t	addEmptyAccount()
441	taxesWindow	createContents(Compo	<i>calls</i>	accountsLis	getAccountsFromDatabase()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		site)		t	
442	taxesWindow	createContents(Composite)	<i>calls</i>	accountsList	.getAccounts()
443	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	accountsList	getAccountForListID(int)
444	taxesWindow	<i>method</i> GUILayer.taxesWindow.configureShell(Shell)	<i>calls</i>	application	getAppName()
445	taxesWindow	createContents(Composite)	<i>calls</i>	client	getAccounts()
446	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getAccounts()
447	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	client	getClient()
448	taxesWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
449	taxesWindow	updateControlsFromSelection()	<i>calls</i>	utils	round(Double, int)
450	templateCheck	templateCheck.doTest()	<i>calls</i>	client	getClient()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
451	templateCheck	templateCheck.doTest()	<i>calls</i>	client	getDataPath()
452	timetrackerImport Window	configureShell(Shell)	<i>calls</i>	application	getAppName()
453	timetrackerImport Window	createContents(Compo site)	<i>calls</i>	configs	getDefaultFont()
454	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	client	getTransactions()
455	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	entry.int)
456	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	entry	getReferToTransaction()
457	todoWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
458	todoWindow	checkTableForDefault Value(boolean, String, String, String, SelectionAdapter)	<i>calls</i>	client	getClient()
459	todoWindow	checkTemplates(boolea n, String, String, SelectionAdapter)	<i>calls</i>	client	getDataPath()
460	todoWindow	<i>method</i>	<i>calls</i>	client	getTransactions()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		GUILayer.todoWindow.checkUnbalancedCreditmemos(boolean)			
461	todoWindow	checkUnbalancedInvoices(boolean)	<i>calls</i>	client	getTransactions()
462	todoWindow	checkUnprocessedOffers(boolean)	<i>calls</i>	client	getClient()
463	todoWindow	checkUnprocessedOffers(boolean)	<i>calls</i>	client	getTransactions()
464	todoWindow	checkTableForDefaultValue(boolean, String, String, String, SelectionAdapter)	<i>calls</i>	configs	getSampleValue(String, int)
465	todoWindow	createContents(Composite)	<i>calls</i>	configs	getDefaultFont()
466	transactionSelectionAdapter	widggetSelected(SelectionEvent)	<i>calls</i>	client	getTransactions()
467	viewerWindow	configureShell(Shell)	<i>calls</i>	application	getAppName()
468	viewerWindow	createContents(Composite)	<i>calls</i>	configs	getOfficeApplication()
469	accountingEditWi		<i>contains</i>	bookingCo	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	ndow			ntentProvid er	
470	assetWindow.Cont entProvider		<i>contains</i>	assets	
471	balanceWindow		<i>contains</i>	entry	
472	contactsWindow. ContentProvider		<i>contains</i>	contacts	
473	newAccountingWi zardAdd		<i>contains</i>	accountsLis t	
474	newAccountingWi zardAdd		<i>contains</i>	bookingCo ntentProvid er	
475	newTransactionSe lectItemDetails		<i>contains</i>	items	
476	newTransactionSe lectItemDetails		<i>contains</i>	products	
477	newTransactionSe lectItemDetails.tra nsactionCellModif ier		<i>contains</i>	products	
478	newTransactionSe		<i>contains</i>	transaction	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	lectItemDetails			ContentProvider	
479	productsWindow.ContentProvider		<i>contains</i>	products	
480	reportWizardShow.tAccount		<i>contains</i>	account	
481	newTransactionSelectItemDetails	transactionCellModifier(newTransactionSelectItemDetails, products)	<i>has param</i>	products	
482	newTransactionWizardPreview	doReplaceItemless(placeholderManager, transaction)	<i>has param</i>	placeholderManager	
483	tAccount	tAccount(reportWizardShow, account)	<i>has param</i>	account	
484	accountingEditWindow	field contentProvider	<i>is of type</i>	bookingContentProvider	
485	assetWindow.ContentProvider	field allAssets	<i>is of type</i>	assets	
486	balanceWindow	field theEntry	<i>is of type</i>	entry	
487	contactsWindow.	field allContacts	<i>is of type</i>	contacts	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	ContentProvider				
488	newAccountingWizardAdd	field accounts	<i>is of type</i>	accountsList	
489	newAccountingWizardAdd	field contentProvider	<i>is of type</i>	bookingContentProvider	
490	newTransactionSelectItemDetails	field allItems	<i>is of type</i>	items	
491	newTransactionSelectItemDetails	field allProducts	<i>is of type</i>	products	
492	newTransactionSelectItemDetails.transactionCellModifier	field allProducts	<i>is of type</i>	products	
493	newTransactionSelectItemDetails	field contentProvider	<i>is of type</i>	transactionContentProvider	
494	ContentProvider	field allProducts	<i>is of type</i>	products	
495	tAccount	field accountObject	<i>is of type</i>	account	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
496	MainWindow	createContents(Compo site)	references	contextCo mposite	
497	exportAction	widgetSelected(Selecti onEvent)	references	AccountNo tFoundExc eption	
498	exportAction	widgetSelected(Selecti onEvent)	references	account	
499	VATannounceme ntWindow	createContents(Compo site)	references	accountsLis t	
500	VATannounceme ntWindow	isOpenable()	references	accountsLis t	
501	accountDefinition Window	updateControlsFromSe lection()	references	account	
502	accountDefinition Window	widgetSelected(Selecti onEvent)	references	account	
503	accountDefinition Window	widgetSelected(Selecti onEvent)	references	account	
504	accountingEditWi ndow	createContents(Compo site)	references	bookingCel lModifier	
505	accountingEditWi ndow	createContents(Compo site)	references	bookingCo ntentProvid	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
				er	
506	accountingEditWi ndow	createContents(Compo site)	<i>references</i>	bookingLa belProvider	
507	accountingEditWi ndow	createContents(Compo site)	<i>references</i>	entry	
508	accountingEditWi ndow	refreshAccountingEntri es()	<i>references</i>	entry	
509	accountingEditWi ndow	widgetSelected(Selecti onEvent)	<i>references</i>	entry	
510	assetWindow	updateControlsFromSe lection()	<i>references</i>	asset	
511	assetWindow	widgetSelected(Selecti onEvent)	<i>references</i>	asset	
512	assetWindow	widgetSelected(Selecti onEvent)	<i>references</i>	asset	
513	assetWindow	ContentProvider(asset Window)	<i>references</i>	asset	
514	balanceWindow	balanceWindow.load(i nt)	<i>references</i>	PaymentMe thodNotFo undExcepti on	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
515	balanceWindow	balanceWindow.load(int)	references	account	
516	balanceWindow	widgetSelected(SelectionEvent)	references	account	
517	balanceWindow	load(int)	references	entry	
518	balanceWindow	widgetSelected(SelectionEvent)	references	entry	
519	balanceWindow	widgetSelected(SelectionEvent)	references	moneyTransfer	
520	configWindow	handleEvent(Event)	references	contextComposite	
521	contactsWindow	updateControlsFromSelection()	references	contact	
522	contactsWindow	widgetSelected(SelectionEvent)	references	contact	
523	contactsWindow	widgetSelected(SelectionEvent)	references	contact	
524	contactsWindow	ContentProvider(contactsWindow)	references	contacts	
525	designerWindow	createContents(Composite)	references	contextComposite	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
526	documentsWindow	widgetSelected(Selecti onEvent)	references	document	
527	documentsWindow.documentCellM odifier	canModify(Object, String)	references	document	
528	documentsWindow.documentCellM odifier	getValue(Object, String)	references	document	
529	documentsWindow.documentCellM odifier	modify(Object, String, Object)	references	document	
530	documentsWindow.documentLabel Provider	getColumnText(Object, int)	references	document	
531	entryDetailWindow	widgetSelected(Selecti onEvent)	references	moneyTran sfer	
532	newAccountingWi zardAdd	createControl(Composi te)	references	bookingCel lModifier	
533	newAccountingWi zardAdd	createControl(Composi te)	references	bookingCo ntentProvid er	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
534	newAccountingWizardAdd	createControl(Composi te)	references	bookingCo ntentProvid er	
535	newAccountingWizardAdd	createControl(Composi te)	references	entry	
536	newAccountingWizardAdd	performBookings()	references	entry	
537	newAccountingWizardAdd	setVisible(boolean)	references	entry	
538	newAccountingWizardAdd	widgetSelected(Selecti onEvent)	references	entry	
539	newAccountingWizardAdd	widgetSelected(Selecti onEvent)	references	entry	
540	newAccountingWizardAdd	dragSetData(DragSour ceEvent)	references	entry	
541	newTransactionSe lectItemDetails.tra nsactionCellModif ier	getValue(Object, String)	references	elementNot FoundExce ption	
542	newTransactionSe lectItemDetails.tra	modify(Object, String, Object)	references	elementNot FoundExce	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	nsactionCellModifier			ption	
543	newTransactionSelectItemDetails	checkPageComplete()	references	item	
544	newTransactionSelectItemDetails	widgetSelected(SelectionEvent)	references	item	
545	newTransactionSelectItemDetails	widgetSelected(SelectionEvent)	references	item	
546	newTransactionSelectItemDetails.transactionCellModifier	getValue(Object, String)	references	item	
547	newTransactionSelectItemDetails.transactionCellModifier	modify(Object, String, Object)	references	item	
548	newTransactionSelectItemDetails.transactionLabelProvider	getColumnText(Object, int)	references	item	
549	newTransactionSe	checkPageComplete()	references	items	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	lectItemDetails				
550	newTransactionSe lectItemDetails	createControl(Composi te)	references	items	
551	newTransactionSe lectItemDetails	modify(Object, String, Object)	references	product	
552	newTransactionSe lectItemDetails	createControl(Composi te)	references	products	
553	newTransactionSe lectTransactionDe tails	fillCustomerChoice(Co nnection)	references	contact	
554	newTransactionSe lectTransactionDe tails	fillCustomerChoice(Co nnection)	references	contacts	
555	newTransactionW izardPreview	process()	references	itemTableL ist	
556	newTransactionW izardPreview	process()	references	itemTextLi st	
557	newTransactionW izardPreview	process()	references	placeholder Manager	
558	newTransactionW izardPreview	getNextPage()	references	document	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
559	productsWindow	updateControlsFromSe lection()	references	product	
560	productsWindow	widgetSelected(Selecti onEvent)	references	product	
561	productsWindow	widgetSelected(Selecti onEvent)	references	product	
562	productsWindow	ContentProvider(produ ctsWindow)	references	product	
563	reportWizardSho w	reportAccountsList()	references	account	
564	reportWizardSho w	reportLists()	references	account	
565	reportWizardSho w	reportTAccoun references ts()	references	account	
566	reportWizardSho w	reportJournal()	references	entry	
567	reportWizardSho w	reportLists()	references	entry	
568	reportWizardSho w	tAccount(reportWizard Show, account)	references	entry	
569	reportWizardSho	tAccount(reportWizard	references	entryNotIn	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	w	Show, account)		ThisAccou ntExceptio n	
570	tableCheck	doTest()	references	AccountNo tFoundExc eption	
571	taxesWindow	widgetSelected(Selecti onEvent)	references	AccountNo tFoundExc eption	
572	taxesWindow	createContents(Compo site)	references	account	
573	taxesWindow	widgetSelected(Selecti onEvent)	references	account	
574	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	references	entry	
575	accountDefinition Window.ContentP rovider	getAllAccounts()	returns	accountsLis t	

4.7.4 Code Dependency Package GUILayer ke Package appLayer.taxRelated

Terdapat 71 *Code Dependency* yang ada pada *package* GUILayer ke *package* appLayer.taxRelated, *dependency kind* terbanyak adalah *calls* dengan jumlah 47, *class* yang paling sering diakses adalah *class* taxList.

Tabel 4. 21 Code Dependency Package GUILayer ke Package appLayer.taxRelated

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
1	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	IRSoffices	getIRSOffices()
2	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	state	getOrderNumber(IRSoffice)
3	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	states	states()
4	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	states	getStates()
5	exportAction	getValueForField(int, Double, tax)	<i>calls</i>	tax	getValue()
6	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	getCreditTaxField()
7	exportAction	widgetSelected(Selecti	<i>calls</i>	tax	getCreditAccount()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		onEvent)			
8	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	getDebitTaxField()
9	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	getDebitAccount()
10	exportAction	widgetSelected(Selecti onEvent)	<i>calls</i>	taxList	getVATArray()
11	newAccountingWi zardAdd	newAccountingWizard Add(newAccountingW izard)	<i>calls</i>	taxList	addEmptyTax()
12	newAccountingWi zardAdd	newAccountingWizard Add(newAccountingW izard)	<i>calls</i>	taxList	setEmptyElementName(String)
13	newAccountingWi zardAdd	newAccountingWizard Add(newAccountingW izard)	<i>calls</i>	taxList	enableCreditDebit(boolean)
14	newAccountingWi zardAdd	newAccountingWizard Add(newAccountingW izard)	<i>calls</i>	taxList	getTaxesFromDatabase()
15	newAccountingWi zardAdd	createControl(Composi te)	<i>calls</i>	taxList	getStringArray()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
16	newAccountingWizardAdd	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	getDefault()
17	productsWindow	updateControlsFromSelection()	<i>calls</i>	tax	getID()
18	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	taxList.()
19	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	getTaxesFromDatabase()
20	productsWindow	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	getVATByID(int)
21	settingsWindow	createContents(Composite)	<i>calls</i>	IRSooffice	getName()
22	settingsWindow	<i>method</i> createContents(Composite)	<i>calls</i>	IRSoffices	getIRSOffices()
23	settingsWindow	createContents(Composite)	<i>calls</i>	state	getName()
24	settingsWindow	createContents(Composite)	<i>calls</i>	states	states()
25	settingsWindow	createContents(Composite)	<i>calls</i>	states	getStates()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
26	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getID()
27	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getCreditTaxField()
28	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getDebitTaxField()
29	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getDescription()
30	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getValue()
31	taxesWindow	<i>method</i> GUILayer.taxesWindow.updateControlsFromSelection()	<i>calls</i>	tax	getCreditAccount()
32	taxesWindow	updateControlsFromSelection()	<i>calls</i>	tax	getDebitAccount()
33	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	tax	setValue(double)
34	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	tax	setCreditAccount(account)
35	taxesWindow	widgetSelected(SelectionEvent)	<i>calls</i>	tax	setDebitAccount(account)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		onEvent)			
36	taxesWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	setCreditTaxField(int)
37	taxesWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	setDebitTaxField(int)
38	taxesWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	setDescription(String)
39	taxesWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	save()
40	taxesWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	tax	delete()
41	taxesWindow	taxesWindow.()		taxList	<i>method</i> appLayer.taxRelated.taxList.()
42	taxesWindow	updateControlsFromSe lection()	<i>calls</i>	taxList	getDefault()
43	taxesWindow	ContentProvider.taxes Window)	<i>calls</i>	taxList	taxList.)
44	taxesWindow	ContentProvider.taxes Window)	<i>calls</i>	taxList	addEmptyTax()
45	taxesWindow	ContentProvider.taxes Window)	<i>calls</i>	taxList	getTaxesFromDatabase()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
46	taxesWindow	ContentProvider.getElements(Object)	<i>calls</i>	taxList	taxList.getVATs()
47	todoBalanceSelectionAdapter	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	taxList()
48	todoBalanceSelectionAdapter	widgetSelected(SelectionEvent)	<i>calls</i>	taxList	getTaxesFromDatabase()
49	newAccountingWizardAdd		<i>contains</i>	taxList	
50	taxesWindow		<i>contains</i>	taxList	
51	ContentProvider		<i>contains</i>	taxList	
52	exportAction	getValueForField(int, Double, tax)	<i>has param</i>	tax	
53	newAccountingWizardAdd	field taxes	<i>is of type</i>	taxList	
54	taxesWindow	field vats	<i>is of type</i>	taxList	
55	taxesWindow	field axes	<i>is of type</i>	taxList	
56	exportAction	widgetSelected(SelectionEvent)	<i>references</i>	IRSoffice	
57	exportAction	widgetSelected(SelectionEvent)	<i>references</i>	state	
58	exportAction	widgetSelected(SelectionEvent)	<i>references</i>	states	

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
		onEvent)			
59	exportAction	widgetSelected(Selecti onEvent)	references	tax	
60	exportAction	widgetSelected(Selecti onEvent)	references	taxList	
61	productsWindow	widgetSelected(Selecti onEvent)	references	taxList	
62	productsWindow	widgetSelected(Selecti onEvent)	references	taxNotFoun dException	
63	settingsWindow	createContents(Compo site)	references	IRSoffice	
64	settingsWindow	createContents(Compo site)	references	state	
65	settingsWindow	createContents(Compo site)	references	states	
66	taxesWindow	updateControlsFromSe lection()	references	tax	
67	taxesWindow	widgetSelected(Selecti onEvent)	references	tax	
68	taxesWindow	widgetSelected(Selecti onEvent)	references	tax	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
69	taxesWindow	taxesWindow()	references	taxList	
70	taxesWindow	ContentProvider.(taxesWindow)	references	taxList	
71	todoBalanceSelectionAdapter	widgetSelected(SelectionEvent)	references	taxList	

4.7.5 Code Dependency Package GUILayer ke Package appLayer.transactionRelated

Terdapat 100 Code Dependency yang ada pada package GUILayer ke package appLayer.transactionRelated, dependency kind terbanyak adalah *calls* dengan jumlah 80, class yang paling sering diakses adalah class transaction.

Tabel 4. 22 Code Dependency Package GUILayer ke Package appLayer.transactionRelated

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
1	balanceWindow		contains	transaction	
2	balanceWindow		contains	transactions	
3	balanceWindow	balanceWindow()	calls	transactions	transactions()
4	balanceWindow	balanceWindow()	references	transactions	
5	balanceWindow	widgetSelected(SelectionEvent)	calls	transactions	getCreditAccountForWorkflowSt

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
		onEvent)			ep(int)
6	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	getDebitAccountForWorkflowStep(int)
7	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	load(int)
8	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	getCurrentTransaction()
9	balanceWindow	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	getRecipient()
10	balanceWindow	balanceWindow.load(int)	<i>calls</i>	transaction	getCreditAccountForWorkflowStep(int)
11	balanceWindow	load(int)	<i>calls</i>	transaction	getDebitAccountForWorkflowStep(int)
12	balanceWindow	load(int)	<i>calls</i>	transaction	getRecipient()
13	balanceWindow	load(int)	<i>calls</i>	transaction	getDueDate()
14	balanceWindow	load(int)	<i>calls</i>	transaction	getVatList()
15	balanceWindow	load(int)	<i>calls</i>	transaction	getCashFlow()
16	balanceWindow	load(int)	<i>accesses</i>	transaction	field NO
17	balanceWindow	load(int)	<i>accesses</i>	transaction	field RECEIVE
18	balanceWindow	load(int)	<i>calls</i>	transactions	load(int)
19	balanceWindow	load(int)	<i>calls</i>	transactions	getCurrentTransaction()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
20	balanceWindow	field theTransaction	<i>is of type</i>	transaction	
21	balanceWindow	field transactionStage	<i>is of type</i>	transactions	
22	designerWindow	createContents(Compo site)	<i>calls</i>	transaction	getTypeName()
23	designerWindow	createContents(Compo site)	<i>calls</i>	transaction	getTypePrefix()
24	designerWindow	createContents(Compo site)	<i>references</i>	transaction	
25	designerWindow	createContents(Compo site)	<i>calls</i>	transactions	getAllTypes()
26	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	transaction	getItems()
27	newTransactionSe lectItemDetails	createControl(Composi te)	<i>calls</i>	transactions	getCurrentTransaction()
28	newTransactionSe lectItemDetails	createContents(Compo site)	<i>calls</i>	transaction	getRemarks()
29	newTransactionSe lectItemDetails	createContents(Compo site)	<i>calls</i>	transactions	getCurrentTransaction()
30	newTransactionSe lectItemDetails	fillTransactionChoice()	<i>calls</i>	transaction	getTypeName()
31	newTransactionSe	fillTransactionChoice()	<i>calls</i>	transaction	getType()

No.	Source		Dependen cy kind	Target	
	Class	Method		Class	Method
	lectItemDetails				
32	newTransactionSe lectItemDetails	fillTransactionChoice()	references	transaction	
33	newTransactionSe lectItemDetails	fillTransactionChoice()	calls	transactions	getAllTypes()
34	newTransactionSe lectItemDetails	fillTransactionChoice()	calls	transactions	getCurrentTransaction()
35	newTransactionSe lectItemDetails	getNextPage()	calls	transaction	setRemarks(String)
36	newTransactionSe lectItemDetails	getNextPage()	calls	transaction	setIssueDate(Date)
37	newTransactionSe lectItemDetails	getNextPage()	calls	transaction	bindToCustomer(int)
38	newTransactionSe lectItemDetails	getNextPage()	calls	transactions	getCurrentTransaction()
39	newTransactionSe lectItemDetails	loadFromDB(int)	calls	transaction	getRecipient()
40	newTransactionSe lectItemDetails	loadFromDB(int)	references	transaction	
41	newTransactionSe lectItemDetails	loadFromDB(int)	calls	transactions	load(int)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
42	newTransactionSelectItemDetails	loadFromDB(int)	<i>calls</i>	transactions	getCurrentTransaction()
43	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getRecipient()
44	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getTransactionName()
45	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getNumber()
46	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getFilenamePDF()
47	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getFilenamePDFGPG()
48	newTransactionWizard	performFinish()	<i>calls</i>	transaction	getFilenameOBDX()
49	newTransactionWizard	performFinish()	<i>calls</i>	transactions	getCurrentTransaction()
50	newTransactionWizard	performFinish()	<i>calls</i>	transactions	finishCurrentTransaction()
51	newTransactionWizard	doReplaceItemless(plac eholderManager, transaction)	<i>has param</i>	transaction	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
52	newTransactionWizard	insertHeader()	calls	transaction	getTypePrefix()
53	newTransactionWizard	insertHeader()	calls	transactions	getCurrentTransaction()
54	newTransactionWizard	process()	calls	transactions	getCurrentTransaction()
55	newTransactionWizardPrintAndSave	getNextPage()	calls	transaction	getFilenameODT()
56	newTransactionWizardPrintAndSave	getNextPage()	calls	transaction	getFilenamePDF()
57	newTransactionWizardPrintAndSave	getNextPage()	calls	transaction	getNumber()
58	newTransactionWizardPrintAndSave	getNextPage()	calls	transaction	getID()
59	newTransactionWizardPrintAndSave	getNextPage()	calls	transaction	save()

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
60	newTransactionWizardPrintAndSave	getNextPage()	<i>calls</i>	transactions	getCurrentTransaction()
61	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	setTypeNames(String)
62	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	setTypePrefix(String)
63	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	setTypeFormat(String)
64	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	setTypeIndex(int)
65	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	setTypePeriod(int)
66	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transaction	saveType()
67	widgetSelected	widgetSelected(SelectonEvent)	<i>references</i>	transaction	
68	widgetSelected	widgetSelected(SelectonEvent)	<i>calls</i>	transactions	getDefaultTransaction()
69	ContentProvider		<i>contains</i>	transactions	
70	ContentProvider	field allTransactions	<i>is of type</i>	transactions	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
71	ContentProvider	getElements(Object)	<i>calls</i>	transactions	getAllTypes()
72	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypeID()
73	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypeName()
74	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypePrefix()
75	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypeFormat()
76	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypeIndex()
77	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transaction	getTypePeriod()
78	numbersWindow	updateControlsFromSelection()	<i>references</i>	transaction	
79	numbersWindow	updateControlsFromSelection()	<i>calls</i>	transactions	getDefaultTransaction()
80	todoBalanceSelectionAdapter	widgetSelected(SelectionEvent)	<i>calls</i>	invoice	invoice.transactions)
81	todoBalanceSelectionAdapter	widgetSelected(SelectionEvent)	<i>references</i>	invoice	

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
82	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	setReferTo(int)
83	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	getNumber()
84	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	getIssueDate()
85	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	getDueDate()
86	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	setRemarks(String)
87	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transaction	cloneFrom(transaction)
88	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>references</i>	transaction	
89	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>references</i>	transaction TypeNotFo undExcepti on	
90	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	load(int)
91	todoBalanceSelect	widgetSelected(Selecti	<i>calls</i>	transactions	getTypeIDByName(String)

No.	Source		Dependency kind	Target	
	Class	Method		Class	Method
	ionAdapter	onEvent)			
92	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	setTransactionType(int)
93	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	getCurrentTransaction()
94	todoBalanceSelect ionAdapter	widgetSelected(Selecti onEvent)	<i>references</i>	transactions	
95	todoBalanceSelect ionAdapter	checkUnbalancedCredi tmemos(boolean)	<i>calls</i>	transactions	getUnbalancedCreditMemos()
96	todoBalanceSelect ionAdapter	checkUnbalancedInvoi ces(boolean)	<i>calls</i>	transactions	getUnbalancedInvoices()
97	todoBalanceSelect ionAdapter	checkUnprocessedOffe rs(boolean)	<i>calls</i>	transaction	getType()
98	todoBalanceSelect ionAdapter	checkUnprocessedOffe rs(boolean)	<i>references</i>	transaction TypeNotFo undExcepti on	
99	todoBalanceSelect ionAdapter	checkUnprocessedOffe rs(boolean)	<i>calls</i>	transactions	getByTypeName(String)
100	transactionSelecti onAdapter	widgetSelected(Selecti onEvent)	<i>calls</i>	transactions	setTransactionType(int)

4.8. Pembuatan *Class Diagram*

Berdasarkan hasil analisa pada tahapan sebelumnya telah menghasilkan *code dependency* yang dapat digunakan untuk memudahkan pembuatan *class diagram* sesuai dengan referensi Tabel 2. 16. *Class diagram* digunakan untuk memudahkan perhitungan Metrik Chidamber dan Kemerer.

Berdasarkan *code dependency* pada tabel Tabel 4. 1 sampai Tabel 4. 22 dapat diketahui *relationship* antar *class* sesuai dengan referensi pada Tabel 2. 16. Tabel 4. 23 adalah pemetaan hasil dari *code dependency* ke *relationship class diagram*. Hasil keseluruhan berupa *class diagram* terdapat pada Gambar 4. 10 sampai Gambar 4. 18.

Tabel 4. 23 Pemetaan *Code Dependency* ke *Relationship Class Diagram*

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
1	dataLayer	HBCIImporter	<i>calls</i>	dataLayer	<i>HBCI</i>	<i>Dependencies</i>
2	dataLayer	HBCIImporter	<i>references</i>	dataLayer	<i>HBCI</i>	<i>Associations</i>
3	dataLayer	HBCIImporter	<i>calls</i>	dataLayer	<i>DB</i>	<i>Dependencies</i>
4	dataLayer	OBDXImporter	<i>calls</i>	dataLayer	<i>Messages</i>	<i>Dependencies</i>
5	dataLayer	OBDXImporter	<i>throws</i>	dataLayer	<i>prematureException</i>	<i>Dependencies</i>
6	dataLayer	OBDXImporter	<i>calls</i>	dataLayer	<i>prematureException</i>	<i>Dependencies</i>
7	dataLayer	OBDXImporter	<i>calls</i>	dataLayer	<i>fileUtils</i>	<i>Dependencies</i>
8	dataLayer	fileImporters	<i>calls</i>	dataLayer	<i>fileUtils</i>	<i>Dependencies</i>
9	dataLayer	fileImporters	<i>calls</i>	dataLayer	<i>Messages</i>	<i>Dependencies</i>
10	dataLayer	gnuPG	<i>calls</i>	dataLayer	<i>Messages</i>	<i>Dependencies</i>
11	dataLayer	moneyplexImporter	<i>calls</i>	dataLayer	<i>fileUtils</i>	<i>Dependencies</i>
12	dataLayer	moneyplexImporter	<i>calls</i>	dataLayer	<i>DB</i>	<i>Dependencies</i>
13	dataLayer	prematureException	<i>calls</i>	dataLayer	<i>Messages</i>	<i>Dependencies</i>
14	dataLayer	prepareDB	<i>calls</i>	dataLayer	<i>Messages</i>	<i>Dependencies</i>
15	dataLayer	prepareDB	<i>calls</i>	dataLayer	<i>DB</i>	<i>Dependencies</i>
16	dataLayer	starmoneyImporte	<i>calls</i>	dataLayer	<i>DB</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		r				
17	dataLayer	starmoneyImporter	<i>calls</i>	dataLayer	<i>invalidImportFormatException</i>	<i>Dependencies</i>
18	dataLayer	starmoneyImporter	<i>references</i>	dataLayer	<i>invalidImportFormatException</i>	<i>Associations</i>
19	dataLayer	starmoneyImporter	<i>calls</i>	dataLayer	<i>fileUtils</i>	<i>Dependencies</i>
20	dataLayer	DB	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
21	dataLayer	HBCI	<i>calls</i>	appLayer	client	<i>Dependencies</i>
22	dataLayer	HBCI	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
23	dataLayer	HBCIImporter	<i>calls</i>	appLayer	client	<i>Dependencies</i>
24	dataLayer	HBCIImporter	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
25	dataLayer	OBDXImporter	<i>calls</i>	appLayer	utils	<i>Dependencies</i>
26	dataLayer	fileImporters	<i>calls</i>	appLayer	client	<i>Dependencies</i>
27	dataLayer	fileImporters	<i>calls</i>	appLayer	item	<i>Dependencies</i>
28	dataLayer	gnuPG	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
29	dataLayer	moneyplexImporter	<i>calls</i>	appLayer	client	<i>Dependencies</i>
30	dataLayer	prepared	<i>calls</i>	appLayer	account	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
31	dataLayer	prepareDB	<i>calls</i>	appLayer	application	<i>Dependencies</i>
32	dataLayer	prepareDB	<i>calls</i>	appLayer	client	<i>Dependencies</i>
33	dataLayer	prepareDB	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
34	dataLayer	prepareDB	<i>calls</i>	appLayer	contacts	<i>Dependencies</i>
35	dataLayer	starmoneyImporte r	<i>calls</i>	appLayer	client	<i>Dependencies</i>
36	dataLayer	fileImporters	<i>references</i>	appLayer	item	<i>Associations</i>
37	dataLayer	prepareDB	<i>references</i>	appLayer	RoleNotFou ndException	<i>Associations</i>
38	dataLayer	prepareDB	<i>throws</i>	appLayer	AccountNot FoundExcep tion	<i>Dependencies</i>
39	dataLayer	fileImporters	<i>calls</i>	appLayer.tra nsactionRela ted	transactions	<i>Dependencies</i>
40	dataLayer	fileImporters	<i>references</i>	appLayer.tra nsactionRela ted	transaction	<i>Associations</i>
41	dataLayer	fileImporters	<i>references</i>	appLayer.tra nsactionRela ted	transactionT ypeNotFoun dException	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
42	dataLayer	fileImporters	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
43	dataLayer	fileImporters	<i>calls</i>	GUILayer	newTransactionWizard	<i>Dependencies</i>
44	dataLayer	fileImporters	<i>references</i>	GUILayer	newTransactionWizard	<i>Associations</i>
45	dataLayer	HBCI	<i>calls</i>	GUILayer	HBCIcallbackDialog	<i>Dependencies</i>
46	dataLayer	HBCI	<i>references</i>	GUILayer	HBCIcallbackDialog	<i>Associations</i>
47	dataLayer	moneyplexImporter	<i>calls</i>	GUILayer	Messages	<i>Dependencies</i>
48	appLayer	ItemListener	<i>has param</i>	appLayer	<i>product</i>	<i>Dependencies</i>
49	appLayer	account	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
50	appLayer	account	<i>contains</i>	appLayer	<i>accountsList</i>	<i>Associations</i>
51	appLayer	account	<i>has param</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>
52	appLayer	account	<i>calls</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>
53	appLayer	account	<i>is of type</i>	appLayer	<i>accountsList</i>	<i>Associations</i>
54	appLayer	account	<i>throws</i>	appLayer	<i>typeNotFoundException</i>	<i>Dependencies</i>
55	appLayer	account	<i>calls</i>	appLayer	<i>typeNotFoundException</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
56	appLayer	account	<i>references</i>	appLayer	<i>typeNotFoun dException</i>	<i>Associations</i>
57	appLayer	account	<i>throws</i>	appLayer	<i>accountExce ption</i>	<i>Dependencies</i>
58	appLayer	account	<i>calls</i>	appLayer	<i>accountExce ption</i>	<i>Dependencies</i>
59	appLayer	account	<i>has param</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
60	appLayer	account	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
61	appLayer	account	<i>calls</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
62	appLayer	account	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
63	appLayer	account	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
64	appLayer	account	<i>references</i>	appLayer	<i>entryNotInT hisAccountE xception</i>	<i>Associations</i>
65	appLayer	accountsList	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
66	appLayer	accountsList	<i>references</i>	appLayer	<i>typeNotFoun dException</i>	<i>Associations</i>
67	appLayer	accountsList	<i>throws</i>	appLayer	<i>AccountNot FoundExcep tion</i>	<i>Dependencies</i>
68	appLayer	accountsList	<i>calls</i>	appLayer	<i>AccountNot FoundExcep tion</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
69	appLayer	accountsList	<i>references</i>	appLayer	<i>AccountNot FoundException</i>	<i>Associations</i>
70	appLayer	accountsList	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
71	appLayer	accountsList	<i>references</i>	appLayer	<i>account</i>	<i>Associations</i>
72	appLayer	accountsList	<i>calls</i>	appLayer	<i>account</i>	<i>Dependencies</i>
73	appLayer	accountsList	<i>returns</i>	appLayer	<i>account</i>	<i>Dependencies</i>
74	appLayer	accountsList	<i>has param</i>	appLayer	<i>account</i>	<i>Dependencies</i>
75	appLayer	accountsList	<i>calls</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
76	appLayer	accountsList	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
77	appLayer	accountsList	<i>returns</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
78	appLayer	application	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
79	appLayer	asset	<i>contains</i>	appLayer	<i>assets</i>	<i>Associations</i>
80	appLayer	asset	<i>has param</i>	appLayer	<i>assets</i>	<i>Dependencies</i>
81	appLayer	asset	<i>calls</i>	appLayer	<i>assets</i>	<i>Dependencies</i>
82	appLayer	asset	<i>is of type</i>	appLayer	<i>assets</i>	<i>Associations</i>
83	appLayer	assets	<i>references</i>	appLayer	<i>asset</i>	<i>Associations</i>
84	appLayer	assets	<i>has param</i>	appLayer	<i>asset</i>	<i>Dependencies</i>
85	appLayer	asset	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
86	appLayer	assets	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
87	appLayer	assets	<i>references</i>	appLayer	<i>asset</i>	<i>Associations</i>
88	appLayer	bookingCellModi	<i>calls</i>	appLayer	<i>entry</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		fier				
89	appLayer	bookingCellModi fier	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
90	appLayer	bookingCellModi fier	<i>references</i>	appLayer	<i>AccountNot FoundExcep tion</i>	<i>Associations</i>
91	appLayer	bookingCellModi fier	<i>calls</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>
92	appLayer	bookingCellModi fier	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
93	appLayer	bookingCellModi fier	<i>references</i>	appLayer	<i>account</i>	<i>Associations</i>
94	appLayer	bookingContentPr ovider	<i>has param</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
95	appLayer	bookingContentPr ovider	<i>calls</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
96	appLayer	bookingContentPr ovider	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
97	appLayer	bookingLabelPro vider	<i>references</i>	appLayer	<i>AccountNot FoundExcep tion</i>	
98	appLayer	bookingLabelPro vider	<i>calls</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
99	appLayer	bookingLabelProvider	<i>calls</i>	appLayer	<i>entry</i>	<i>Dependencies</i>
100	appLayer	bookingLabelProvider	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
101	appLayer	bookingLabelProvider	<i>references</i>	appLayer	<i>entry</i>	<i>Associations</i>
102	appLayer	bookingLabelProvider	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
103	appLayer	client	<i>contains</i>	appLayer	<i>documents</i>	<i>Associations</i>
104	appLayer	client	<i>is of type</i>	appLayer	<i>documents</i>	<i>Associations</i>
105	appLayer	client	<i>returns</i>	appLayer	<i>documents</i>	<i>Dependencies</i>
106	appLayer	client	<i>calls</i>	appLayer	<i>documents</i>	<i>Dependencies</i>
107	appLayer	client	<i>contains</i>	appLayer	<i>accountsList</i>	<i>Associations</i>
108	appLayer	client	<i>is of type</i>	appLayer	<i>accountsList</i>	<i>Associations</i>
109	appLayer	client	<i>returns</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>
110	appLayer	client	<i>calls</i>	appLayer	<i>accountsList</i>	<i>Dependencies</i>
111	appLayer	configs	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
112	appLayer	configs	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
113	appLayer	contact	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
114	appLayer	contact	<i>contains</i>	appLayer	<i>contacts</i>	<i>Associations</i>
115	appLayer	contact	<i>has param</i>	appLayer	<i>contacts</i>	<i>Dependencies</i>
116	appLayer	contact	<i>calls</i>	appLayer	<i>contacts</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
117	appLayer	contact	<i>is of type</i>	appLayer	<i>contacts</i>	<i>Associations</i>
118	appLayer	contact	<i>calls</i>	appLayer	<i>contacts</i>	<i>Dependencies</i>
119	appLayer	contacts	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
120	appLayer	contacts	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
121	appLayer	contacts	<i>throws</i>	appLayer	<i>PaymentMet hodNotFoun dException</i>	<i>Dependencies</i>
122	appLayer	contacts	<i>calls</i>	appLayer	<i>PaymentMet hodNotFoun dException</i>	<i>Dependencies</i>
123	appLayer	contacts	<i>throws</i>	appLayer	<i>RoleNotFou ndException</i>	<i>Dependencies</i>
124	appLayer	contacts	<i>calls</i>	appLayer	<i>RoleNotFou ndException</i>	<i>Dependencies</i>
125	appLayer	contacts	<i>references</i>	appLayer	<i>contact</i>	<i>Associations</i>
126	appLayer	contacts	<i>calls</i>	appLayer	<i>contact</i>	<i>Dependencies</i>
127	appLayer	contacts	<i>has param</i>	appLayer	<i>contact</i>	<i>Dependencies</i>
128	appLayer	contextComposite	<i>implements</i>	appLayer	<i>Interface IcontextProv ider</i>	<i>Realization</i>
129	appLayer	document	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
130	appLayer	document	<i>contains</i>	appLayer	<i>documents</i>	<i>Associations</i>
131	appLayer	document	<i>has param</i>	appLayer	<i>documents</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
132	appLayer	document	<i>is of type</i>	appLayer	<i>documents</i>	<i>Associations</i>
133	appLayer	documents	<i>calls</i>	appLayer	<i>client</i>	<i>Dependencies</i>
134	appLayer	documents	<i>references</i>	appLayer	<i>document</i>	<i>Associations</i>
135	appLayer	documents	<i>returns</i>	appLayer	<i>document</i>	<i>Dependencies</i>
136	appLayer	documents	<i>calls</i>	appLayer	<i>document</i>	<i>Dependencies</i>
137	appLayer	documents	<i>returns</i>	appLayer	<i>document</i>	<i>Dependencies</i>
138	appLayer	item	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
139	appLayer	item	<i>calls</i>	appLayer	<i>utils</i>	<i>Dependencies</i>
140	appLayer	item	<i>has param</i>	appLayer	<i>ItemListener</i>	<i>Dependencies</i>
141	appLayer	item	<i>references</i>	appLayer	<i>ItemListener</i>	<i>Associations</i>
142	appLayer	item	<i>calls</i>	appLayer	<i>ItemListener</i>	<i>Dependencies</i>
143	appLayer	item	<i>contains</i>	appLayer	<i>products</i>	<i>Associations</i>
144	appLayer	item	<i>calls</i>	appLayer	<i>products</i>	<i>Dependencies</i>
145	appLayer	item	<i>references</i>	appLayer	<i>products</i>	<i>Associations</i>
146	appLayer	item	<i>is of type</i>	appLayer	<i>products</i>	<i>Associations</i>
147	appLayer	item	<i>calls</i>	appLayer	<i>product</i>	<i>Dependencies</i>
148	appLayer	item	<i>returns</i>	appLayer	<i>product</i>	<i>Dependencies</i>
149	appLayer	item	<i>calls</i>	appLayer	<i>product</i>	<i>Dependencies</i>
150	appLayer	item	<i>accesses</i>	appLayer	<i>product</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
151	appLayer	item	<i>references</i>	appLayer	<i>product</i>	<i>Associations</i>
152	appLayer	item	<i>has param</i>	appLayer	<i>product</i>	<i>Dependencies</i>
153	appLayer	item	<i>contains</i>	appLayer	<i>transactionContentProvider</i>	<i>Associations</i>
154	appLayer	item	<i>has param</i>	appLayer	<i>transactionContentProvider</i>	<i>Dependencies</i>
155	appLayer	item	<i>is of type</i>	appLayer	<i>transactionContentProvider</i>	<i>Associations</i>
156	appLayer	item	<i>calls</i>	appLayer	<i>transactionContentProvider</i>	<i>Dependencies</i>
157	appLayer	itemTableList	<i>references</i>	appLayer	<i>items</i>	<i>Associations</i>
158	appLayer	itemTableList	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
159	appLayer	itemTableList	<i>references</i>	appLayer	<i>item</i>	<i>Associations</i>
160	appLayer	itemTableList	<i>calls</i>	appLayer	<i>transactionContentProvider</i>	<i>Dependencies</i>
161	appLayer	itemTableList	<i>contains</i>	appLayer	<i>placeholderManager</i>	<i>Associations</i>
162	appLayer	itemTableList	<i>has param</i>	appLayer	<i>placeholder</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
					Manager	
163	appLayer	itemTableList	<i>is of type</i>	appLayer	<i>placeholder Manager</i>	<i>Associations</i>
164	appLayer	itemTableList	<i>calls</i>	appLayer	<i>placeholder Manager</i>	<i>Dependencies</i>
165	appLayer	itemTextList	<i>references</i>	appLayer	<i>items</i>	<i>Associations</i>
166	appLayer	itemTextList	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
167	appLayer	itemTextList	<i>has param</i>	appLayer	<i>item</i>	<i>Dependencies</i>
168	appLayer	itemTextList	<i>references</i>	appLayer	<i>item</i>	<i>Associations</i>
169	appLayer	itemTextList	<i>calls</i>	appLayer	<i>transactionC ontentProvid er</i>	<i>Dependencies</i>
170	appLayer	itemTextList	<i>contains</i>	appLayer	<i>placeholder Manager</i>	<i>Associations</i>
171	appLayer	itemTextList	<i>has param</i>	appLayer	<i>placeholder Manager</i>	<i>Associations</i>
172	appLayer	itemTextList	<i>calls</i>	appLayer	<i>placeholder Manager</i>	<i>Dependencies</i>
173	appLayer	itemTextList	<i>is of type</i>	appLayer	<i>placeholder Manager</i>	<i>Associations</i>
174	appLayer	items	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
175	appLayer	items	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
176	appLayer	items	<i>calls</i>	appLayer	<i>item</i>	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
177	appLayer	items	<i>references</i>	appLayer	<i>item</i>	<i>Associations</i>
178	appLayer	items	<i>extends</i>	appLayer	<i>transactionContentProvider</i>	<i>Generalization</i>
179	appLayer	items	<i>calls</i>	appLayer	<i>transactionContentProvider</i>	<i>Dependencies</i>
180	appLayer	items	<i>accesses</i>	appLayer	<i>transactionContentProvider</i>	<i>Dependencies</i>
181	appLayer	moneyTransfer	<i>calls</i>	appLayer	<i>contact</i>	<i>Dependencies</i>
182	appLayer	moneyTransfer	<i>calls</i>	appLayer	<i>configs</i>	<i>Dependencies</i>
183	appLayer	moneyTransfer	<i>calls</i>	appLayer	<i>configs</i>	<i>Dependencies</i>
184	appLayer	moneyTransfer	<i>calls</i>	appLayer	<i>Messages</i>	<i>Dependencies</i>
185	appLayer	account	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
186	appLayer	accountsList	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
187	appLayer	asset	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
188	appLayer	configs	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
189	appLayer	configs	<i>calls</i>	dataLayer	fileUtils	<i>Dependencies</i>
190	appLayer	contact	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
191	appLayer	document	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
192	appLayer	entry	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
193	appLayer	item	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
194	appLayer	items	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
195	appLayer	moneyTransfer	<i>calls</i>	dataLayer	HBCI	<i>Dependencies</i>
196	appLayer	product	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
197	appLayer	moneyTransfer	<i>contains</i>	dataLayer	HBCI	<i>Associations</i>
198	appLayer	moneyTransfer	<i>is of type</i>	dataLayer	HBCI	<i>Associations</i>
199	appLayer	accountsList	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
200	appLayer	bookingCellModi fier	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
201	appLayer	client	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
202	appLayer	entry	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
203	appLayer	entry. taxColumn	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
204	appLayer	entry	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
205	appLayer	entry.taxColumn.	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
206	appLayer	item	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
207	appLayer	placeholderMana ger	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
208	appLayer	product	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
209	appLayer	product	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
210	appLayer	client	<i>contains</i>	appLayer.tax Related	taxList	<i>Associations</i>
211	appLayer	product	<i>contains</i>	appLayer.tax Related	tax	<i>Associations</i>
212	appLayer	entry	<i>has param</i>	appLayer.tax Related	tax	<i>Dependencies</i>
213	appLayer	entry	<i>has param</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
214	appLayer	product	<i>has param</i>	appLayer.tax Related	tax	<i>Dependencies</i>
215	appLayer	client	<i>is of type</i>	appLayer.tax Related	taxList	<i>Associations</i>
216	appLayer	product	<i>is of type</i>	appLayer.tax Related	tax	<i>Associations</i>
217	appLayer	bookingCellModi fier	<i>references</i>	appLayer.tax Related	tax	<i>Associations</i>
218	appLayer	bookingCellModi fier	<i>references</i>	appLayer.tax Related	taxNotFound Exception	<i>Associations</i>
219	appLayer	entry. taxColumn	<i>references</i>	appLayer.tax	tax	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
				Related		
220	appLayer	entry	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
221	appLayer	products	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
222	appLayer	products	<i>references</i>	appLayer.tax Related	taxNotFound Exception	<i>Associations</i>
223	appLayer	client	<i>returns</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
224	appLayer	entry	<i>returns</i>	appLayer.tax Related	tax	<i>Dependencies</i>
225	appLayer	entry.taxColumn	<i>returns</i>	appLayer.tax Related	tax	<i>Dependencies</i>
226	appLayer	product	<i>returns</i>	appLayer.tax Related	tax	<i>Dependencies</i>
227	appLayer	moneyTransfer	<i>aceses</i>	GUILayer	transaction	<i>Dependencies</i>
228	appLayer	client	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>
229	appLayer	entry	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>
230	appLayer	itemTableList	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>
231	appLayer	itemTextList	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>
232	appLayer	moneyTransfer	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>
233	appLayer	placeholderManager	<i>calls</i>	GUILayer	transaction	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
234	appLayer	client	<i>contains</i>	GUILayer	transaction	<i>Dependencies</i>
235	appLayer	itemTableList	<i>contains</i>	GUILayer	transaction	<i>Dependencies</i>
236	appLayer	itemTableList	<i>has param</i>	GUILayer	transaction	<i>Dependencies</i>
237	appLayer	itemTextList	<i>has param</i>	GUILayer	transaction	<i>Dependencies</i>
238	appLayer	moneyTransfer	<i>has param</i>	GUILayer	transaction	<i>Dependencies</i>
239	appLayer	placeholderManager	<i>has param</i>	GUILayer	transaction	<i>Dependencies</i>
240	appLayer	client	<i>is of type</i>	GUILayer	transaction	<i>Associations</i>
241	appLayer	itemTableList	<i>is of type</i>	GUILayer	transaction	<i>Associations</i>
242	appLayer	itemTextList	<i>is of type</i>	GUILayer	transaction	<i>Associations</i>
243	appLayer	entry	<i>references</i>	GUILayer	transaction	<i>Associations</i>
244	appLayer	client	<i>returns</i>	GUILayer	transaction	<i>Dependencies</i>
245	appLayer	pageActivator	<i>calls</i>	GUILayer	newTransactionSelectItemDetails	<i>Dependencies</i>
246	appLayer	pageActivator	<i>contains</i>	GUILayer	newTransactionSelectItemDetails	<i>Associations</i>
247	appLayer	transactionContentProvider	<i>contains</i>	GUILayer	newTransactionSelectItemDetails	<i>Associations</i>
248	appLayer	pageActivator	<i>has param</i>	GUILayer	newTransactionSelectItem	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
					mDetails	
249	appLayer	transactionContentProvider	<i>has param</i>	GUILayer	newTransactionSelectedItemDetails	<i>Dependencies</i>
250	appLayer	pageActivator	<i>is of type</i>	GUILayer	newTransactionSelectedItemDetails	<i>Associations</i>
251	appLayer	transactionContentProvider	<i>is of type</i>	GUILayer	newTransactionSelectedItemDetails	<i>Associations</i>
252	appLayer.taxRelated	IRSoffices	<i>contains</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
253	appLayer.taxRelated	IRSoffices	<i>calls</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
254	appLayer.taxRelated	IRSoffices	<i>references</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
255	appLayer.taxRelated	IRSoffices	<i>returns</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
256	appLayer.taxRelated	IRSoffices	<i>is of type</i>	appLayer.taxRelated	IRSoffice	<i>Associations</i>
257	appLayer.taxRelated	state	<i>has param</i>	appLayer.taxRelated	IRSoffice	<i>Dependencies</i>
258	appLayer.t	state	<i>calls</i>	appLayer.tax	IRSoffice	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	axRelated			Related		
259	appLayer.t axRelated	states	<i>calls</i>	appLayer.tax Related	state	<i>Dependencies</i>
260	appLayer.t axRelated	states	<i>references</i>	appLayer.tax Related	state	<i>Associations</i>
261	appLayer.t axRelated	states	<i>returns</i>	appLayer.tax Related	state	<i>Dependencies</i>
262	appLayer.t axRelated	tax	<i>contains</i>	appLayer.tax Related	taxList	<i>Associations</i>
263	appLayer.t axRelated	tax	<i>has param</i>	appLayer.tax Related	taxList	<i>Associations</i>
264	appLayer.t axRelated	tax	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
265	appLayer.t axRelated	tax	<i>is of type</i>	appLayer.tax Related	taxList	<i>Associations</i>
266	appLayer.t axRelated	tax	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
267	appLayer.t axRelated	taxList	<i>returns</i>	appLayer.tax Related	tax	<i>Dependencies</i>
268	appLayer.t axRelated	taxList	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
269	appLayer.t axRelated	taxList	<i>references</i>	appLayer.tax Related	tax	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
270	appLayer.taxRelated	taxList	<i>calls</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
271	appLayer.taxRelated	taxList	<i>references</i>	appLayer.taxRelated	tax	<i>Associations</i>
272	appLayer.taxRelated	taxList	<i>returns</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
273	appLayer.taxRelated	taxList	<i>has param</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
274	appLayer.taxRelated	taxList	<i>throws</i>	appLayer.taxRelated	taxNotFound Exception	<i>Dependencies</i>
275	appLayer.taxRelated	taxList	<i>calls</i>	appLayer.taxRelated	taxNotFound Exception	<i>Dependencies</i>
276	appLayer.taxRelated	tax	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
277	appLayer.taxRelated	taxList	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
278	appLayer.taxRelated	state	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
279	appLayer.taxRelated	states	<i>calls</i>	appLayer	Messages	<i>Dependencies</i>
280	appLayer.taxRelated	tax	<i>calls</i>	appLayer	account	<i>Dependencies</i>
281	appLayer.t	taxList	<i>calls</i>	appLayer	Messages	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	axRelated					
282	appLayer.t axRelated	taxList	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
283	appLayer.t axRelated	taxList	<i>calls</i>	appLayer	client	<i>Dependencies</i>
284	appLayer.t axRelated	tax	<i>contains</i>	appLayer	account	<i>Associations</i>
285	appLayer.t axRelated	tax	<i>has param</i>	appLayer	account	<i>Dependencies</i>
286	appLayer.t axRelated	tax	<i>is of type</i>	appLayer	account	<i>Associations</i>
287	appLayer.t axRelated	taxList	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
288	appLayer.t axRelated	tax	<i>returns</i>	appLayer	account	<i>Dependencies</i>
289	appLayer.r ansactionR elated	cancelation	<i>extends</i>	appLayer.ra nsactionRela ted	transaction	<i>Generalization</i>
290	appLayer.r ansactionR elated	cancelation	<i>calls</i>	appLayer.ra nsactionRela ted	transaction	<i>Dependencies</i>
291	appLayer.r	cancelation	<i>references</i>	appLayer.ra	transaction	

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	ansactionR elated			nsactionRela ted		
292	appLayer.r ansactionR elated	cancelation	<i>returns</i>	appLayer.ra nsactionRela ted	transaction	<i>Dependencies</i>
293	appLayer.r ansactionR elated	cancelation	<i>accesses</i>	appLayer.ra nsactionRela ted	transaction	<i>Dependencies</i>
294	appLayer.r ansactionR elated	cancelation	<i>has param</i>	appLayer.ra nsactionRela ted	transactions	<i>Dependencies</i>
295	appLayer.r ansactionR elated	cancelation	<i>calls</i>	appLayer.ra nsactionRela ted	transactions	<i>Dependencies</i>
296	appLayer.r ansactionR elated	creditmemo	<i>has param</i>	appLayer.ra nsactionRela ted	transactions	<i>Dependencies</i>
297	appLayer.r ansactionR elated	creditmemo	<i>extends</i>	appLayer.ra nsactionRela ted	transaction	<i>Generalization</i>
298	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	appLayer.ra nsactionRela ted	transaction	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
299	appLayer.r ansactionR elated	cancelation	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
300	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
301	appLayer.r ansactionR elated	invoice	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
302	appLayer.r ansactionR elated	offer	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
303	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
304	appLayer.r ansactionR elated	reminder	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
305	appLayer.r ansactionR elated	transaction	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
306	appLayer.r ansactionR	transaction	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	related					
307	appLayer.r ansactionR elated	transactions	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
308	appLayer.r ansactionR elated	cancelation	<i>calls</i>	appLayer	Messages	<i>Dependencies</i>
309	appLayer.r ansactionR elated	cancelation	<i>calls</i>	appLayer	client	<i>Dependencies</i>
310	appLayer.r ansactionR elated	cancelation	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
311	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	appLayer	account	<i>Dependencies</i>
312	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
313	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	appLayer	client	<i>Dependencies</i>
314	appLayer.r	creditmemo	<i>calls</i>	appLayer	contact	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	ansactionR elated					
315	appLayer.r ansactionR elated	creditmemo	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
316	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	account	<i>Dependencies</i>
317	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
318	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	client	<i>Dependencies</i>
319	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
320	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
321	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	item	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
322	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	product	<i>Dependencies</i>
323	appLayer.r ansactionR elated	invoice	<i>calls</i>	appLayer	transactionC ontentProvid er	<i>Dependencies</i>
324	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	appLayer	account	<i>Dependencies</i>
325	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
326	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	appLayer	client	<i>Dependencies</i>
327	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
328	appLayer.r ansactionR elated	personalDrawing	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
329	appLayer.r ansactionR	transaction	<i>calls</i>	appLayer	application	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	related					
330	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	client	<i>Dependencies</i>
331	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
332	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
333	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	item	<i>Dependencies</i>
334	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	transactionC ontentProvid er	<i>Dependencies</i>
335	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	utils	<i>Dependencies</i>
336	appLayer.r ansactionR elated	transaction	<i>calls</i>	appLayer	Messages	<i>Dependencies</i>
337	appLayer.r	transactions	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	ansactionR elated					
338	appLayer.r ansactionR elated	transactions	<i>calls</i>	appLayer	client	<i>Dependencies</i>
339	appLayer.r ansactionR elated	creditmemo	<i>contains</i>	appLayer	account	<i>Associations</i>
340	appLayer.r ansactionR elated	invoice	<i>contains</i>	appLayer	account	<i>Associations</i>
341	appLayer.r ansactionR elated	personalDrawing	<i>contains</i>	appLayer	account	<i>Associations</i>
342	appLayer.r ansactionR elated	transaction	<i>contains</i>	appLayer	contact	<i>Associations</i>
343	appLayer.r ansactionR elated	transaction	<i>contains</i>	appLayer	items	<i>Associations</i>
344	appLayer.r ansactionR elated	creditmemo	<i>is of type</i>	appLayer	account	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
345	appLayer.r ansactionR elated	invoice	<i>is of type</i>	appLayer	account	<i>Associations</i>
346	appLayer.r ansactionR elated	personalDrawing	<i>is of type</i>	appLayer	account	<i>Associations</i>
347	appLayer.r ansactionR elated	transaction	<i>is of type</i>	appLayer	contact	<i>Associations</i>
348	appLayer.r ansactionR elated	transaction	<i>is of type</i>	appLayer	items	<i>Associations</i>
349	appLayer.r ansactionR elated	cancelation	<i>references</i>	appLayer	account	<i>Associations</i>
350	appLayer.r ansactionR elated	cancelation	<i>references</i>	appLayer	entry	<i>Associations</i>
351	appLayer.r ansactionR elated	creditmemo	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
352	appLayer.r ansactionR	creditmemo	<i>references</i>	appLayer	account	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	related					
353	appLayer.r ansactionR elated	creditmemo	<i>references</i>	appLayer	accountExcep tion	<i>Associations</i>
354	appLayer.r ansactionR elated	creditmemo	<i>references</i>	appLayer	entry	<i>Associations</i>
355	appLayer.r ansactionR elated	invoice	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
356	appLayer.r ansactionR elated	invoice	<i>references</i>	appLayer	account	<i>Associations</i>
357	appLayer.r ansactionR elated	invoice	<i>references</i>	appLayer	accountExcep tion	<i>Associations</i>
358	appLayer.r ansactionR elated	invoice	<i>references</i>	appLayer	entry	<i>Associations</i>
359	appLayer.r ansactionR elated	invoice	<i>references</i>	appLayer	item	<i>Associations</i>
360	appLayer.r	invoice	<i>references</i>	appLayer	items	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	ansactionR elated					
361	appLayer.r ansactionR elated	personalDrawing	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
362	appLayer.r ansactionR elated	personalDrawing	<i>references</i>	appLayer	account	<i>Associations</i>
363	appLayer.r ansactionR elated	personalDrawing	<i>references</i>	appLayer	accountExce ption	<i>Associations</i>
364	appLayer.r ansactionR elated	personalDrawing	<i>references</i>	appLayer	entry	<i>Associations</i>
365	appLayer.r ansactionR elated	transaction	<i>references</i>	appLayer	contact	<i>Associations</i>
366	appLayer.r ansactionR elated	transaction	<i>references</i>	appLayer	item	<i>Associations</i>
367	appLayer.r ansactionR elated	transaction	<i>references</i>	appLayer	items	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
368	appLayer.r ansactionR elated	cancelation	<i>returns</i>	appLayer	account	<i>Dependencies</i>
369	appLayer.r ansactionR elated	creditmemo	<i>returns</i>	appLayer	account	<i>Dependencies</i>
370	appLayer.r ansactionR elated	invoice	<i>returns</i>	appLayer	account	<i>Dependencies</i>
371	appLayer.r ansactionR elated	offer	<i>returns</i>	appLayer	account	<i>Dependencies</i>
372	appLayer.r ansactionR elated	personalDrawing	<i>returns</i>	appLayer	account	<i>Dependencies</i>
373	appLayer.r ansactionR elated	reminder	<i>returns</i>	appLayer	account	<i>Dependencies</i>
374	appLayer.r ansactionR elated	transaction	<i>returns</i>	appLayer	account	<i>Dependencies</i>
375	appLayer.r ansactionR	transaction	<i>returns</i>	appLayer	contact	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	related					
376	appLayer.transactionRelated	transaction	<i>returns</i>	appLayer	item	<i>Dependencies</i>
377	appLayer.transactionRelated	transaction	<i>returns</i>	appLayer	items	<i>Dependencies</i>
378	appLayer.transactionRelated	creditmemo	<i>calls</i>	appLayer.taxRelated	taxList	<i>Dependencies</i>
379	appLayer.transactionRelated	invoice	<i>calls</i>	appLayer.taxRelated	tax	<i>Dependencies</i>
380	appLayer.transactionRelated	invoice	<i>calls</i>	appLayer.taxRelated	taxList	<i>Dependencies</i>
381	appLayer.transactionRelated	personalDrawing	<i>calls</i>	appLayer.taxRelated	taxList	<i>Dependencies</i>
382	appLayer.transactionRelated	creditmemo	<i>references</i>	appLayer.taxRelated	taxList	<i>Associations</i>
383	appLayer.r	invoice	<i>references</i>	appLayer.tax	taxList	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
	ansactionR elated			Related		
384	appLayer.r ansactionR elated	personalDrawing	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
385	appLayer.r ansactionR elated	transactions	<i>calls</i>	GUILayer	todoWindow	<i>Dependencies</i>
386	GUILayer	HSQLErrorCheck	<i>has param</i>	GUILayer	configWindo w	<i>Dependencies</i>
387	GUILayer	HSQLErrorCheck	<i>references</i>	GUILayer	configWindo w	<i>Associations</i>
388	GUILayer	HSQLErrorCheck	<i>returns</i>	GUILayer	testResult	<i>Dependencies</i>
389	GUILayer	HSQLErrorCheck	<i>calls</i>	GUILayer	testResult	<i>Dependencies</i>
390	GUILayer	HSQLErrorCheck	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
391	GUILayer	MainWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
392	GUILayer	aboutWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
393	GUILayer	configWindow	<i>calls</i>	dataLayer	fileUtils	<i>Dependencies</i>
394	GUILayer	contactsWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
395	GUILayer	contactsWindow	<i>calls</i>	dataLayer	HBCI	<i>Dependencies</i>
396	GUILayer	entryDetailWindo w	<i>calls</i>	dataLayer	OBDXImpor ter	<i>Dependencies</i>
397	GUILayer	newAccountingW	<i>calls</i>	dataLayer	HBCIImport	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		izardImport			er	
398	GUILayer	newAccountingW izardImport	<i>calls</i>	dataLayer	moneyplexI mporter	<i>Dependencies</i>
399	GUILayer	newAccountingW izardImport	<i>calls</i>	dataLayer	starMoneyIm porter	<i>Dependencies</i>
400	GUILayer	newAccountingW izardImport	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
401	GUILayer	productsWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
402	GUILayer	settingsWindow	<i>calls</i>	dataLayer	HBCI	<i>Dependencies</i>
403	GUILayer	tableCheck	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
404	GUILayer	tableCheck	<i>calls</i>	dataLayer	prepareDB	<i>Dependencies</i>
405	GUILayer	taxesWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
406	GUILayer	templateCheck	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
407	GUILayer	timetrackerImport Window	<i>calls</i>	dataLayer	fileImporters	<i>Dependencies</i>
408	GUILayer	todoBalanceSele ctionAdapter	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
409	GUILayer	todoWindow	<i>calls</i>	dataLayer	DB	<i>Dependencies</i>
410	GUILayer	entryDetailWindo w	<i>contains</i>	dataLayer	OBDXImpor ter	<i>Associations</i>
411	GUILayer	entryDetailWindo w	<i>is of type</i>	dataLayer	OBDXImpor ter	<i>Associations</i>
412	GUILayer	entryDetailWindo	<i>references</i>	dataLayer	OBDXImpor	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		w			ter	
413	GUILayer	entryDetailWindow	<i>references</i>	dataLayer	prematureException	<i>Associations</i>
414	GUILayer	newAccountingWizardImport	<i>references</i>	dataLayer	HBCIImporter	<i>Associations</i>
415	GUILayer	newAccountingWizardImport	<i>references</i>	dataLayer	moneyplexImporter	<i>Associations</i>
416	GUILayer	newAccountingWizardImport	<i>references</i>	dataLayer	starmoneyImporter	<i>Associations</i>
417	GUILayer	tableCheck	<i>references</i>	dataLayer	prepareDB	<i>Associations</i>
418	GUILayer	MainWindow	<i>accesses</i>	appLayer	application	<i>Dependencies</i>
419	GUILayer	HSQLLockCheck	<i>calls</i>	appLayer	client	<i>Dependencies</i>
420	GUILayer	MainWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
421	GUILayer	MainWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
422	GUILayer	MainWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
423	GUILayer	MainWindow	<i>calls</i>	appLayer	contextComposite	<i>Dependencies</i>
424	GUILayer	OOoCheck	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
425	GUILayer	exportAction	<i>calls</i>	appLayer	account	<i>Dependencies</i>
426	GUILayer	VATannouncementWindow	<i>calls</i>	appLayer	account	<i>Dependencies</i>
427	GUILayer	VATannouncementWindow	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
428	GUILayer	VATannounceme ntWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
429	GUILayer	VATannounceme ntWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
430	GUILayer	VATannounceme ntWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
431	GUILayer	VATannounceme ntWindow	<i>calls</i>	appLayer	utils	<i>Dependencies</i>
432	GUILayer	aboutWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
433	GUILayer	aboutWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
434	GUILayer	accountDefinition Window	<i>calls</i>	appLayer	account	<i>Dependencies</i>
435	GUILayer	accountDefinition Window	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
436	GUILayer	accountDefinition Window	<i>calls</i>	appLayer	application	<i>Dependencies</i>
437	GUILayer	accountDefinition Window	<i>calls</i>	appLayer	client	<i>Dependencies</i>
438	GUILayer	accountDefinition Window	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
439	GUILayer	accountingEditWi ndow	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
440	GUILayer	accountingEditWi	<i>calls</i>	appLayer	application	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		ndow				
441	GUILayer	accountingEditWindow	<i>calls</i>	appLayer	bookingCellModifier	<i>Dependencies</i>
442	GUILayer	accountingEditWindow	<i>calls</i>	appLayer	bookingContentProvider	<i>Dependencies</i>
443	GUILayer	accountingEditWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
444	GUILayer	accountingEditWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
445	GUILayer	accountingEditWindow	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
446	GUILayer	assetWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
447	GUILayer	assetWindow	<i>calls</i>	appLayer	asset	<i>Dependencies</i>
448	GUILayer	assetWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
449	GUILayer	balanceWindow	<i>calls</i>	appLayer	account	<i>Dependencies</i>
450	GUILayer	balanceWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
451	GUILayer	balanceWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
452	GUILayer	balanceWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
453	GUILayer	balanceWindow	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
454	GUILayer	balanceWindow	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
455	GUILayer	balanceWindow	<i>calls</i>	appLayer	moneyTransfer	<i>Dependencies</i>
456	GUILayer	browserWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
457	GUILayer	communityWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
458	GUILayer	communityWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
459	GUILayer	configWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
460	GUILayer	configWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
461	GUILayer	configWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
462	GUILayer	configWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
463	GUILayer	configWindow	<i>calls</i>	appLayer	contextComposite	<i>Dependencies</i>
464	GUILayer	contactsWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
465	GUILayer	contactsWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
466	GUILayer	contactsWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
467	GUILayer	contactsWindow	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
468	GUILayer	ContentProvider	<i>calls</i>	appLayer	contacts	<i>Dependencies</i>
469	GUILayer	databaseDriverNameTestCase	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
470	GUILayer	designerWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
471	GUILayer	designerWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
472	GUILayer	designerWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
473	GUILayer	designerWindow	<i>calls</i>	appLayer	contextComposite	<i>Dependencies</i>
474	GUILayer	documentsWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		w				
475	GUILayer	documentsWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
476	GUILayer	documentsWindow	<i>calls</i>	appLayer	document	<i>Dependencies</i>
477	GUILayer	documentsWindow.documentCellModifier	<i>calls</i>	appLayer	document	<i>Dependencies</i>
478	GUILayer	documentsWindow	<i>calls</i>	appLayer	document	<i>Dependencies</i>
479	GUILayer	entryDetailWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
480	GUILayer	entryDetailWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
481	GUILayer	entryDetailWindow	<i>calls</i>	appLayer	moneyTransfer	<i>Dependencies</i>
482	GUILayer	mainDirTestCase	<i>calls</i>	appLayer	client	<i>Dependencies</i>
483	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
484	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	bookingCellModifier	<i>Dependencies</i>
485	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	bookingContentProvider	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
486	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	client	<i>Dependencies</i>
487	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
488	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	documents	<i>Dependencies</i>
489	GUILayer	newAccountingWizardAdd	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
490	GUILayer	newAccountingWizardImport	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
491	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	client	<i>Dependencies</i>
492	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
493	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	item	<i>Dependencies</i>
494	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	products	<i>Dependencies</i>
495	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	transactionContentProvider	<i>Dependencies</i>
496	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	utils	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
497	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	client	<i>Dependencies</i>
498	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
499	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
500	GUILayer	newTransactionWizard	<i>calls</i>	appLayer	application	<i>Dependencies</i>
501	GUILayer	newTransactionWizard	<i>calls</i>	appLayer	client	<i>Dependencies</i>
502	GUILayer	newTransactionWizard	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
503	GUILayer	newTransactionWizard	<i>calls</i>	appLayer	contact	<i>Dependencies</i>
504	GUILayer	newTransactionWizardPreview	<i>calls</i>	appLayer	client	<i>Dependencies</i>
505	GUILayer	newTransactionWizardPreview	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
506	GUILayer	newTransactionWizardPreview	<i>calls</i>	appLayer	itemTableList	<i>Dependencies</i>
507	GUILayer	newTransactionWizardPreview	<i>calls</i>	appLayer	itemTextList	<i>Dependencies</i>
508	GUILayer	newTransactionW	<i>calls</i>	appLayer	placeholder	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		izardPreview			Manager	
509	GUILayer	newTransactionW izardPrintAndSave	<i>calls</i>	appLayer	client	<i>Dependencies</i>
510	GUILayer	newTransactionW izardPrintAndSave	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
511	GUILayer	newTransactionW izardPrintAndSave	<i>calls</i>	appLayer	document	<i>Dependencies</i>
512	GUILayer	numbersWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
513	GUILayer	numbersWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
514	GUILayer	passwordDialog	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
515	GUILayer	productsWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
516	GUILayer	productsWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
517	GUILayer	productsWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
518	GUILayer	productsWindow	<i>calls</i>	appLayer	product	<i>Dependencies</i>
519	GUILayer	productsWindow	<i>calls</i>	appLayer	products	<i>Dependencies</i>
520	GUILayer	productsWindow	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
521	GUILayer	reportWizard	<i>calls</i>	appLayer	client	<i>Dependencies</i>
522	GUILayer	reportWizardSele ct	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
523	GUILayer	reportWizardSele	<i>calls</i>	appLayer	client	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		ct				
524	GUILayer	reportWizardSelect	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
525	GUILayer	.reportWizardShow	<i>calls</i>	appLayer	account	<i>Dependencies</i>
526	GUILayer	reportWizardShow.tAccount	<i>calls</i>	appLayer	account	<i>Dependencies</i>
527	GUILayer	reportWizardShow.tAccount	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
528	GUILayer	reportWizardShow.tAccount	<i>calls</i>	appLayer	client	<i>Dependencies</i>
529	GUILayer	reportWizardShow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
530	GUILayer	reportWizardShow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
531	GUILayer	reportWizardShow	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
532	GUILayer	reportWizardShow.tAccount	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
533	GUILayer	reportWizardShow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
534	GUILayer	settingsWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
535	GUILayer	settingsWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
536	GUILayer	tableCheck	<i>calls</i>	appLayer	client	<i>Dependencies</i>
537	GUILayer	taxesWindow	<i>calls</i>	appLayer	account	<i>Dependencies</i>
538	GUILayer	taxesWindow	<i>calls</i>	appLayer	accountsList	<i>Dependencies</i>
539	GUILayer	taxesWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
540	GUILayer	taxesWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
541	GUILayer	taxesWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
542	GUILayer	taxesWindow	<i>calls</i>	appLayer	utils	<i>Dependencies</i>
543	GUILayer	templateCheck	<i>calls</i>	appLayer	client	<i>Dependencies</i>
544	GUILayer	timetrackerImport Window	<i>calls</i>	appLayer	application	<i>Dependencies</i>
545	GUILayer	timetrackerImport Window	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
546	GUILayer	todoBalanceSelec tionAdapter	<i>calls</i>	appLayer	client	<i>Dependencies</i>
547	GUILayer	todoBalanceSelec tionAdapter	<i>calls</i>	appLayer	entry	<i>Dependencies</i>
548	GUILayer	todoWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
549	GUILayer	todoWindow	<i>calls</i>	appLayer	client	<i>Dependencies</i>
550	GUILayer	todoWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>
551	GUILayer	transactionSelecti onAdapter	<i>calls</i>	appLayer	client	<i>Dependencies</i>
552	GUILayer	viewerWindow	<i>calls</i>	appLayer	application	<i>Dependencies</i>
553	GUILayer	viewerWindow	<i>calls</i>	appLayer	configs	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
554	GUILayer	accountingEditWindow	<i>contains</i>	appLayer	bookingContentProvider	<i>Associations</i>
555	GUILayer	assetWindow.ContentProvider	<i>contains</i>	appLayer	assets	<i>Associations</i>
556	GUILayer	balanceWindow	<i>contains</i>	appLayer	entry	<i>Associations</i>
557	GUILayer	contactsWindow.ContentProvider	<i>contains</i>	appLayer	contacts	<i>Associations</i>
558	GUILayer	newAccountingWizardAdd	<i>contains</i>	appLayer	accountsList	<i>Associations</i>
559	GUILayer	newAccountingWizardAdd	<i>contains</i>	appLayer	bookingContentProvider	<i>Associations</i>
560	GUILayer	newTransactionSelectItemDetails	<i>contains</i>	appLayer	items	<i>Associations</i>
561	GUILayer	newTransactionSelectItemDetails	<i>contains</i>	appLayer	products	<i>Associations</i>
562	GUILayer	newTransactionSelectItemDetails.transactionCellModifier	<i>contains</i>	appLayer	products	<i>Associations</i>
563	GUILayer	newTransactionSelectItemDetails	<i>contains</i>	appLayer	transactionContentProvider	<i>Associations</i>
564	GUILayer	productsWindow.	<i>contains</i>	appLayer	products	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		ContentProvider				
565	GUILayer	reportWizardShow.tAccount	<i>contains</i>	appLayer	account	<i>Associations</i>
566	GUILayer	newTransactionSelectItemDetails	<i>has param</i>	appLayer	products	<i>Dependencies</i>
567	GUILayer	newTransactionWizardPreview	<i>has param</i>	appLayer	placeholderManager	<i>Dependencies</i>
568	GUILayer	tAccount	<i>has param</i>	appLayer	account	<i>Dependencies</i>
569	GUILayer	accountingEditWindow	<i>is of type</i>	appLayer	bookingContentProvider	<i>Associations</i>
570	GUILayer	assetWindow.ContentProvider	<i>is of type</i>	appLayer	assets	<i>Associations</i>
571	GUILayer	balanceWindow	<i>is of type</i>	appLayer	entry	<i>Associations</i>
572	GUILayer	contactsWindow.ContentProvider	<i>is of type</i>	appLayer	contacts	<i>Associations</i>
573	GUILayer	newAccountingWizardAdd	<i>is of type</i>	appLayer	accountsList	<i>Associations</i>
574	GUILayer	newAccountingWizardAdd	<i>is of type</i>	appLayer	bookingContentProvider	<i>Associations</i>
575	GUILayer	newTransactionSelectItemDetails	<i>is of type</i>	appLayer	items	<i>Associations</i>
576	GUILayer	newTransactionSelectItemDetails	<i>is of type</i>	appLayer	products	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
577	GUILayer	newTransactionSelectItemDetails.transactionCellModifier	<i>is of type</i>	appLayer	products	<i>Associations</i>
578	GUILayer	newTransactionSelectItemDetails	<i>is of type</i>	appLayer	transactionContentProvider	<i>Associations</i>
579	GUILayer	ContentProvider	<i>is of type</i>	appLayer	products	<i>Associations</i>
580	GUILayer	tAccount	<i>is of type</i>	appLayer	account	<i>Associations</i>
581	GUILayer	MainWindow	<i>references</i>	appLayer	contextComposite	<i>Associations</i>
582	GUILayer	exportAction	<i>references</i>	appLayer	AccountNotFoundExcep tion	<i>Associations</i>
583	GUILayer	exportAction	<i>references</i>	appLayer	account	<i>Associations</i>
584	GUILayer	VATannouncementWindow	<i>references</i>	appLayer	accountsList	<i>Associations</i>
585	GUILayer	VATannouncementWindow	<i>references</i>	appLayer	accountsList	<i>Associations</i>
586	GUILayer	accountDefinitionWindow	<i>references</i>	appLayer	account	<i>Associations</i>
587	GUILayer	accountingEditWindow	<i>references</i>	appLayer	bookingCellModifier	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
588	GUILayer	accountingEditWindow	references	appLayer	bookingContentProvider	Associations
589	GUILayer	accountingEditWindow	references	appLayer	bookingLabelProvider	Associations
590	GUILayer	accountingEditWindow	references	appLayer	entry	Associations
591	GUILayer	assetWindow	references	appLayer	asset	Associations
592	GUILayer	balanceWindow	references	appLayer	PaymentMethodNotFoundException	Associations
593	GUILayer	balanceWindow	references	appLayer	account	Associations
594	GUILayer	balanceWindow	references	appLayer	entry	Associations
595	GUILayer	balanceWindow	references	appLayer	moneyTransfer	Associations
596	GUILayer	configWindow	references	appLayer	contextComposite	Associations
597	GUILayer	contactsWindow	references	appLayer	contact	Associations
598	GUILayer	contactsWindow	references	appLayer	contacts	Associations
599	GUILayer	designerWindow	references	appLayer	contextComposite	Associations
600	GUILayer	documentsWindow	references	appLayer	document	Associations
601	GUILayer	documentsWindow	references	appLayer	document	Associations

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		w.documentCell Modifier				
602	GUILayer	documentsWindo w.documentLabel Provider	<i>references</i>	appLayer	document	<i>Associations</i>
603	GUILayer	entryDetailWindo w	<i>references</i>	appLayer	moneyTrans fer	<i>Associations</i>
604	GUILayer	newAccountingW izardAdd	<i>references</i>	appLayer	bookingCell Modifier	<i>Associations</i>
605	GUILayer	newAccountingW izardAdd	<i>references</i>	appLayer	bookingCont entProvider	<i>Associations</i>
606	GUILayer	newAccountingW izardAdd	<i>references</i>	appLayer	entry	<i>Associations</i>
607	GUILayer	newAccountingW izardAdd	<i>references</i>	appLayer	entry	<i>Associations</i>
608	GUILayer	newTransactionSe lectItemDetails.tra nsactionCellModi fier	<i>references</i>	appLayer	elementNotF oundExcepti on	<i>Associations</i>
609	GUILayer	newTransactionSe lectItemDetails	<i>references</i>	appLayer	item	<i>Associations</i>
610	GUILayer	newTransactionSe lectItemDetails.tra	<i>references</i>	appLayer	item	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		nsactionCellModifier				
611	GUILayer	newTransactionSelectItemDetails.transactionLabelProvider	<i>references</i>	appLayer	item	<i>Associations</i>
612	GUILayer	newTransactionSelectItemDetails	<i>references</i>	appLayer	items	<i>Associations</i>
613	GUILayer	newTransactionSelectItemDetails	<i>references</i>	appLayer	product	<i>Associations</i>
614	GUILayer	newTransactionSelectItemDetails	<i>references</i>	appLayer	products	<i>Associations</i>
615	GUILayer	newTransactionSelectTransactionDetails	<i>references</i>	appLayer	contact	<i>Associations</i>
616	GUILayer	newTransactionSelectTransactionDetails	<i>references</i>	appLayer	contacts	<i>Associations</i>
617	GUILayer	newTransactionWizardPreview	<i>references</i>	appLayer	itemTableList	<i>Associations</i>
618	GUILayer	newTransactionWizardPreview	<i>references</i>	appLayer	itemTextList	<i>Associations</i>
619	GUILayer	newTransactionW	<i>references</i>	appLayer	placeholder	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		izardPreview			Manager	
620	GUILayer	newTransactionW izardPreview	<i>references</i>	appLayer	document	<i>Associations</i>
621	GUILayer	productsWindow	<i>references</i>	appLayer	product	<i>Associations</i>
622	GUILayer	reportWizardSho w	<i>references</i>	appLayer	account	<i>Associations</i>
623	GUILayer	reportWizardSho w	<i>references</i>	appLayer	entry	<i>Associations</i>
624	GUILayer	reportWizardSho w	<i>references</i>	appLayer	entryNotInT hisAccountE xception	<i>Associations</i>
625	GUILayer	tableCheck	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
626	GUILayer	taxesWindow	<i>references</i>	appLayer	AccountNot FoundExcep tion	<i>Associations</i>
627	GUILayer	taxesWindow	<i>references</i>	appLayer	account	<i>Associations</i>
628	GUILayer	todoBalanceSelec tionAdapter	<i>references</i>	appLayer	entry	<i>Associations</i>
629	GUILayer	accountDefinition Window.ContentP rovider	<i>returns</i>	appLayer	accountsList	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
630	GUILayer	exportAction	<i>calls</i>	appLayer.tax Related	IRSoffices	<i>Dependencies</i>
631	GUILayer	exportAction	<i>calls</i>	appLayer.tax Related	state	<i>Dependencies</i>
632	GUILayer	exportAction	<i>calls</i>	appLayer.tax Related	states	<i>Dependencies</i>
633	GUILayer	exportAction	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
634	GUILayer	exportAction	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
635	GUILayer	newAccountingW izardAdd	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
636	GUILayer	productsWindow	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
637	GUILayer	productsWindow	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
638	GUILayer	settingsWindow	<i>calls</i>	appLayer.tax Related	IRSoffice	<i>Dependencies</i>
639	GUILayer	settingsWindow	<i>calls</i>	appLayer.tax Related	IRSoffices	<i>Dependencies</i>
640	GUILayer	settingsWindow	<i>calls</i>	appLayer.tax Related	state	<i>Dependencies</i>
641	GUILayer	settingsWindow	<i>calls</i>	appLayer.tax	states	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
				Related		
642	GUILayer	taxesWindow	<i>calls</i>	appLayer.tax Related	tax	<i>Dependencies</i>
643	GUILayer	taxesWindow	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
644	GUILayer	todoBalanceSelec tionAdapter	<i>calls</i>	appLayer.tax Related	taxList	<i>Dependencies</i>
645	GUILayer	newAccountingW izardAdd	<i>contains</i>	appLayer.tax Related	taxList	<i>Associations</i>
646	GUILayer	taxesWindow	<i>contains</i>	appLayer.tax Related	taxList	<i>Associations</i>
647	GUILayer	ContentProvider	<i>contains</i>	appLayer.tax Related	taxList	<i>Associations</i>
648	GUILayer	exportAction	<i>has param</i>	appLayer.tax Related	tax	<i>Dependencies</i>
649	GUILayer	newAccountingW izardAdd	<i>is of type</i>	appLayer.tax Related	taxList	<i>Associations</i>
650	GUILayer	taxesWindow	<i>is of type</i>	appLayer.tax Related	taxList	<i>Associations</i>
651	GUILayer	exportAction	<i>references</i>	appLayer.tax Related	IRSoffice	<i>Associations</i>
652	GUILayer	exportAction	<i>references</i>	appLayer.tax Related	state	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
653	GUILayer	exportAction	<i>references</i>	appLayer.tax Related	states	<i>Associations</i>
654	GUILayer	exportAction	<i>references</i>	appLayer.tax Related	tax	<i>Associations</i>
655	GUILayer	exportAction	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
656	GUILayer	productsWindow	<i>references</i>	appLayer.tax Related	taxNotFound Exception	<i>Associations</i>
657	GUILayer	settingsWindow	<i>references</i>	appLayer.tax Related	IRSoffice	<i>Associations</i>
658	GUILayer	settingsWindow	<i>references</i>	appLayer.tax Related	state	<i>Associations</i>
659	GUILayer	settingsWindow	<i>references</i>	appLayer.tax Related	states	<i>Associations</i>
660	GUILayer	taxesWindow	<i>references</i>	appLayer.tax Related	tax	<i>Associations</i>
661	GUILayer	taxesWindow	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
662	GUILayer	todoBalanceSelec tionAdapter	<i>references</i>	appLayer.tax Related	taxList	<i>Associations</i>
663	GUILayer	balanceWindow	<i>contains</i>	appLayer.tra nsactionRela ted	transaction	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
664	GUILayer	balanceWindow	<i>contains</i>	appLayer.transactionRelated	transactions	<i>Associations</i>
665	GUILayer	balanceWindow	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
666	GUILayer	balanceWindow	<i>references</i>	appLayer.transactionRelated	transactions	<i>Associations</i>
667	GUILayer	balanceWindow	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
668	GUILayer	balanceWindow	<i>aceses</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
669	GUILayer	balanceWindow	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
670	GUILayer	balanceWindow	<i>is of type</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
671	GUILayer	designerWindow	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
				ted		
672	GUILayer	designerWindow	<i>references</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
673	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
674	GUILayer	newTransactionSelectItemDetails	<i>references</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
675	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
676	GUILayer	newTransactionSelectItemDetails	<i>references</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
677	GUILayer	newTransactionSelectItemDetails	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
678	GUILayer	newTransactionWizard	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
679	GUILayer	newTransactionW	<i>has param</i>	appLayer.tra	transaction	<i>Dependencies</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		izard		nsactionRela ted		
680	GUILayer	newTransactionW izard	<i>calls</i>	appLayer.tra nsactionRela ted	transaction	<i>Dependencies</i>
681	GUILayer	newTransactionW izard	<i>calls</i>	appLayer.tra nsactionRela ted	transactions	<i>Dependencies</i>
682	GUILayer	newTransactionW izardPrintAndSav e	<i>calls</i>	appLayer.tra nsactionRela ted	transaction	<i>Dependencies</i>
683	GUILayer	newTransactionW izardPrintAndSav e	<i>calls</i>	appLayer.tra nsactionRela ted	transactions	<i>Dependencies</i>
684	GUILayer	widgetSelected	<i>calls</i>	appLayer.tra nsactionRela ted	transaction	<i>Dependencies</i>
685	GUILayer	widgetSelected	<i>references</i>	appLayer.tra nsactionRela ted	transaction	<i>Associations</i>
686	GUILayer	widgetSelected	<i>calls</i>	appLayer.tra nsactionRela ted	transactions	<i>Dependencies</i>

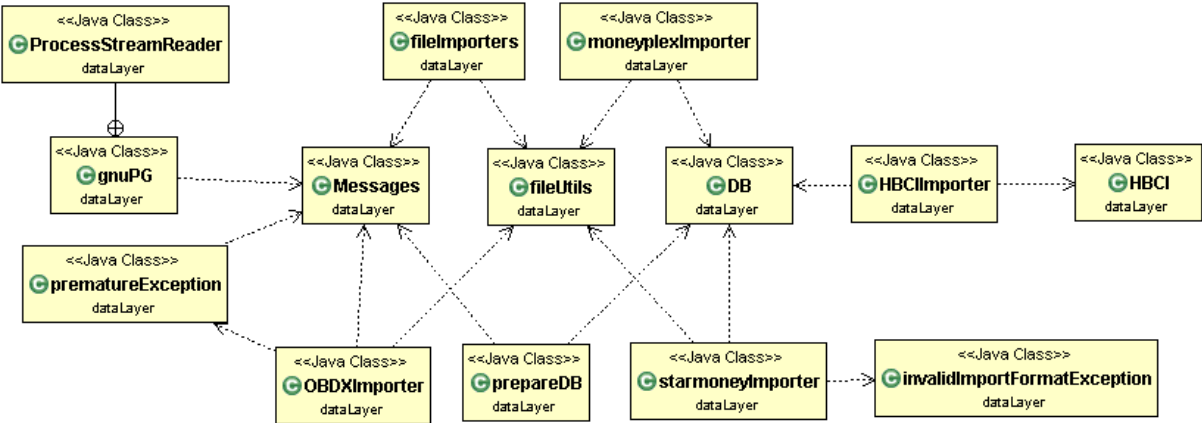
No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
687	GUILayer	ContentProvider	<i>contains</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
688	GUILayer	ContentProvider	<i>is of type</i>	appLayer.transactionRelated	transactions	<i>Associations</i>
689	GUILayer	ContentProvider	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
690	GUILayer	numbersWindow	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
691	GUILayer	numbersWindow	<i>references</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
692	GUILayer	numbersWindow	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
693	GUILayer	todoBalanceSelectionAdapter	<i>calls</i>	appLayer.transactionRelated	invoice	<i>Dependencies</i>
694	GUILayer	todoBalanceSelectionAdapter	<i>references</i>	appLayer.transactionRelated	invoice	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
				ted		
695	GUILayer	todoBalanceSelectionAdapter	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
696	GUILayer	todoBalanceSelectionAdapter	<i>references</i>	appLayer.transactionRelated	transaction	<i>Associations</i>
697	GUILayer	todoBalanceSelectionAdapter	<i>references</i>	appLayer.transactionRelated	transactionTypeNotFoundException	<i>Associations</i>
698	GUILayer	todoBalanceSelectionAdapter	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
699	GUILayer	todoBalanceSelectionAdapter	<i>references</i>	appLayer.transactionRelated	transactions	<i>Associations</i>
700	GUILayer	todoBalanceSelectionAdapter	<i>calls</i>	appLayer.transactionRelated	transactions	<i>Dependencies</i>
701	GUILayer	todoBalanceSelectionAdapter	<i>calls</i>	appLayer.transactionRelated	transaction	<i>Dependencies</i>
702	GUILayer	todoBalanceSelec	<i>references</i>	appLayer.tra	transactionT	<i>Associations</i>

No.	Source		Dependency kind	Target		Relasi
	Package	Class		Package	Class	
		tionAdapter		nsactionRela ted	ypeNotFoun dException	
703	GUILayer	todoBalanceSelec tionAdapter	<i>calls</i>	appLayer.tran sactionRela ted	transactions	<i>Dependencies</i>
704	GUILayer	transactionSelecti onAdapter	<i>calls</i>	appLayer.tran sactionRela ted	transactions	<i>Dependencies</i>

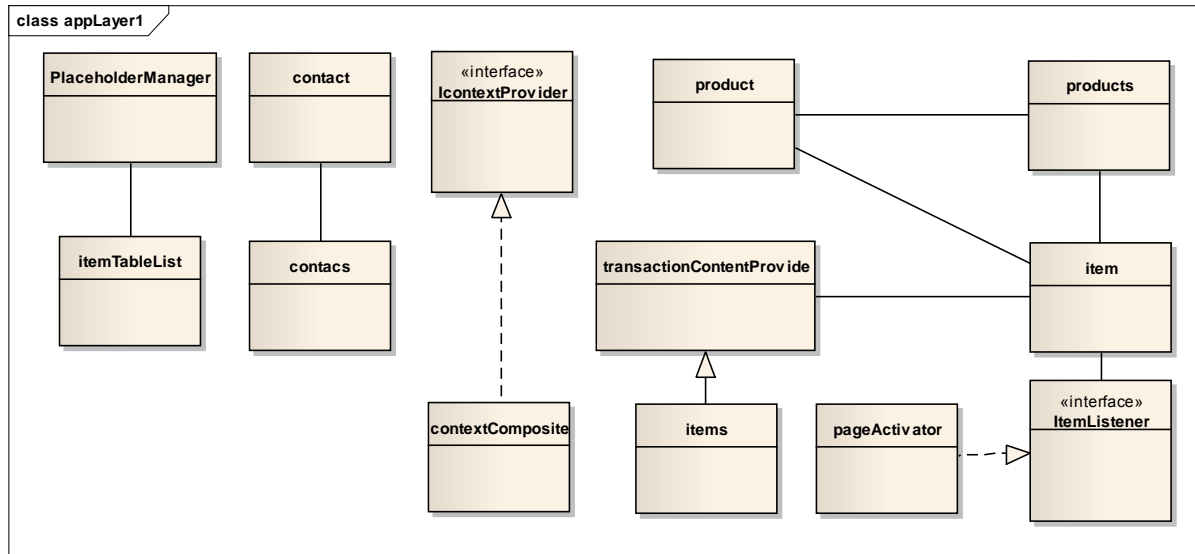
Setelah melakukan pemetaan antara *code dependency* dan *relationship* pada *class diagram*, tahapan selanjutnya adalah membuat *class diagram*. *Class diagram* terdapat pada Gambar 4. 10 sampai Gambar 4. 18.

Gambar 4. 1 adalah *class diagram* pada *package* dataLayer, terdapat relasi *dependencies relationship* dan *nesting relationship*.



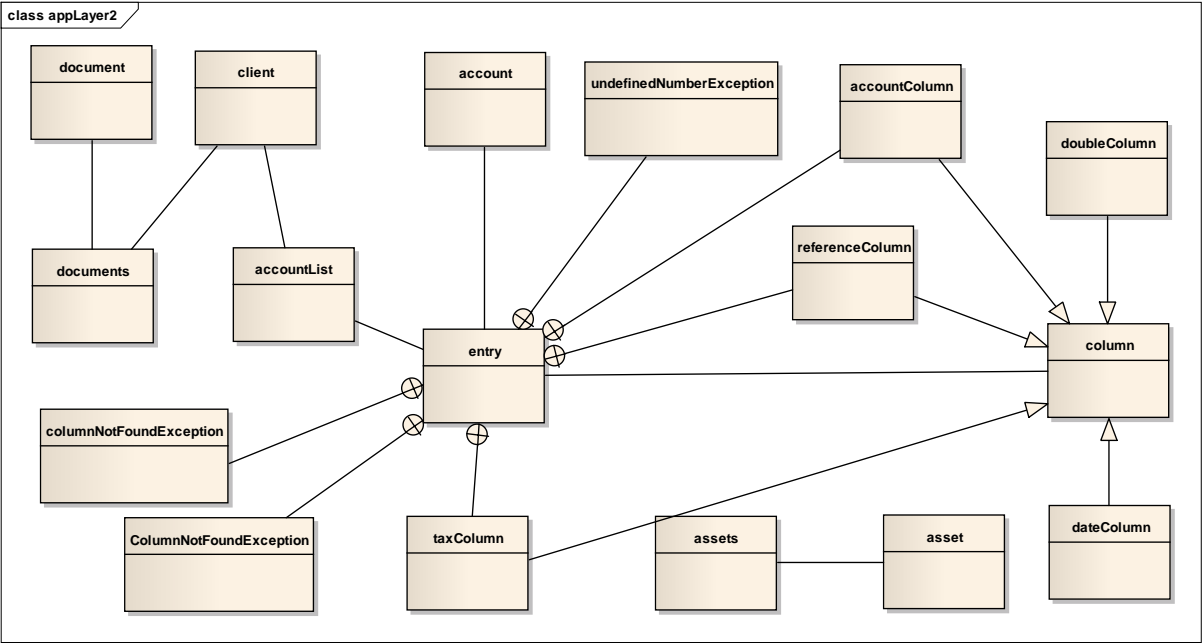
Gambar 4. 10 *Class Diagram Package dataLayer*

Error! Reference source not found. adalah *class diagram* pada *package* appLayer, terdapat relasi *ssociation relationship*, *generalization relationship* dan *realization relationship*. *Dependencies relationship package* appLayer dapat dilihat pada Gambar 4. 13.

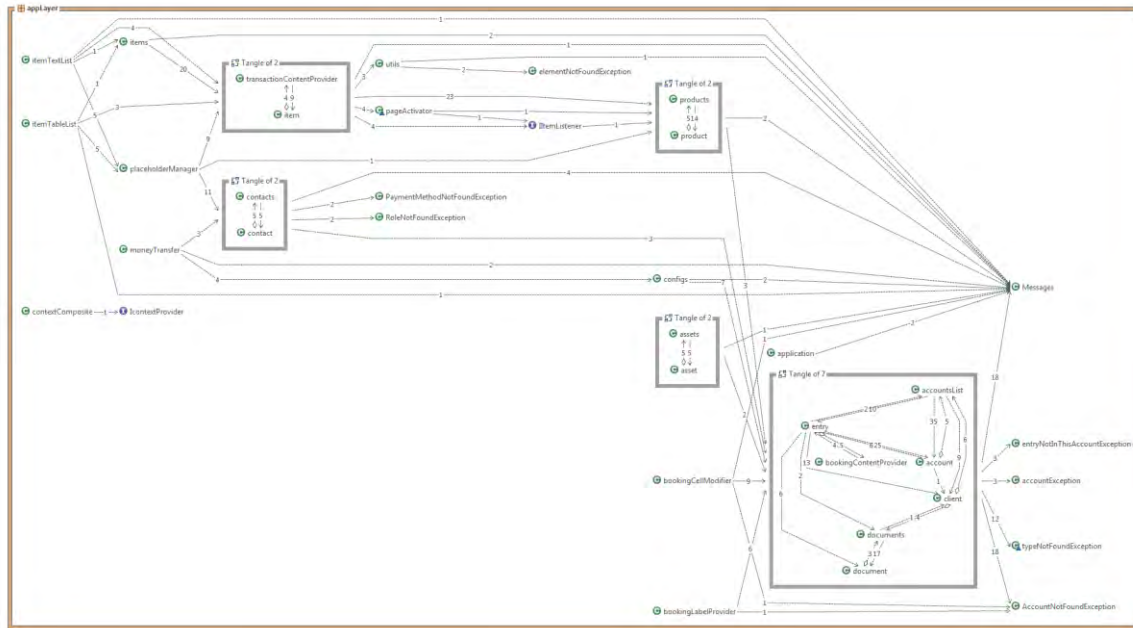


Gambar 4. 11 Class Diagram Package appLayer part 1

Gambar 4. 12 adalah *class diagram* pada *package* appLayer, terdapat *relasi association relationship*, *generalization relationship* dan *realization relationship*. *Dependencies relationship package* appLayer dapat dilihat pada Gambar 4. 13.

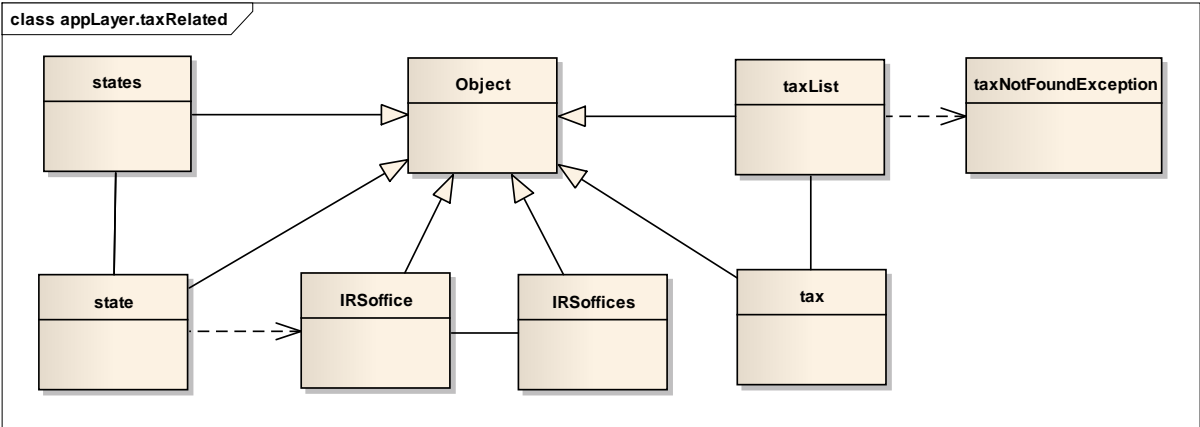


Gambar 4. 12 Class Diagram Package appLayer part 2



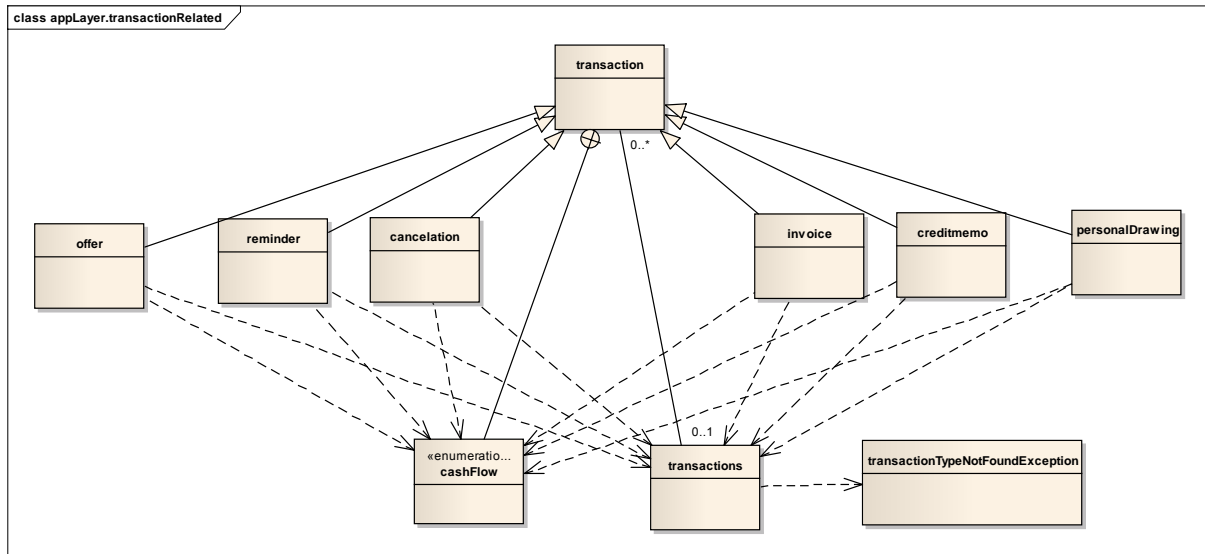
Gambar 4. 13 Dependencies Relationship pada package appLayer

Gambar 4. 14 adalah *class diagram* pada *package* appLayer.taxRelated, terdapat relasi *association relationship*, *dependencies relationship* dan *generalization relationship*.



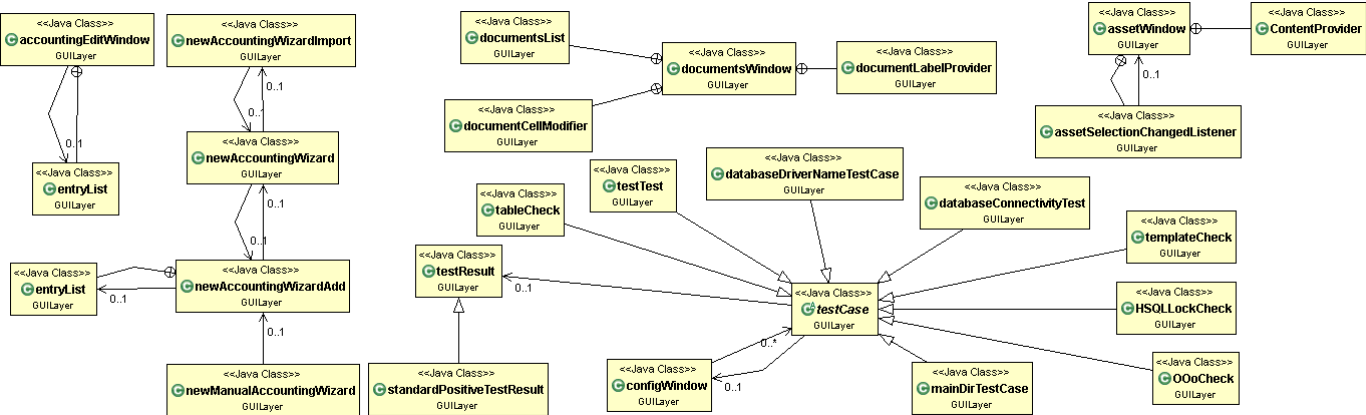
Gambar 4. 14 *Class Diagram Package* appLayer.taxRelated

Gambar 4. 15 adalah *class diagram* pada *package* `appLayer.transactionRelated`, terdapat *relasi association relationship*, *dependencies relationship* dan *generalization relationship*.



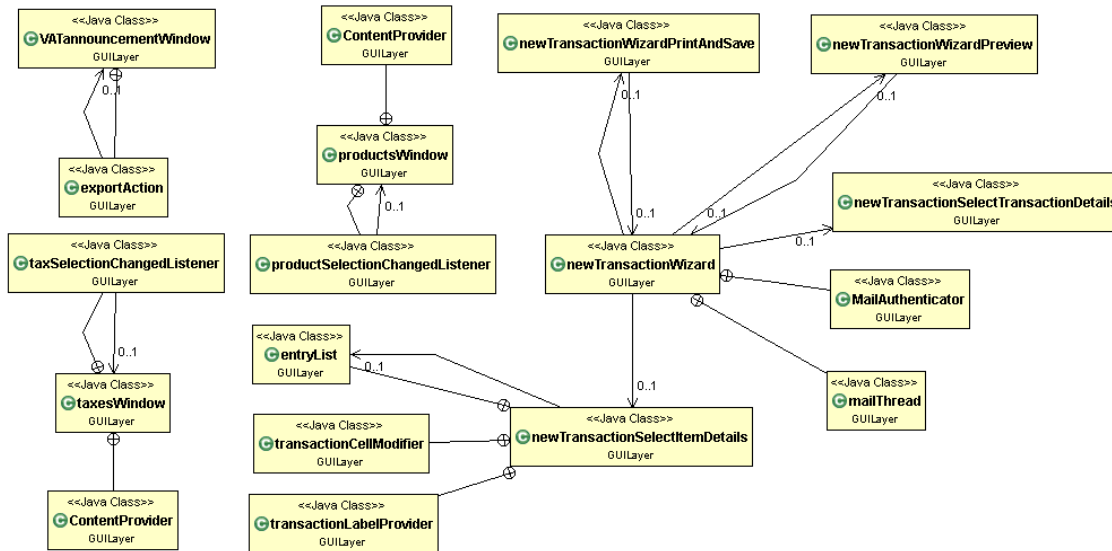
Gambar 4. 15 Class Diagram Package `appLayer.transactionRelated`

Gambar 4. 16 adalah class diagram pada package GUILayer, terdapat relasi association relationship, dependencies relationship dan nesting relationship.



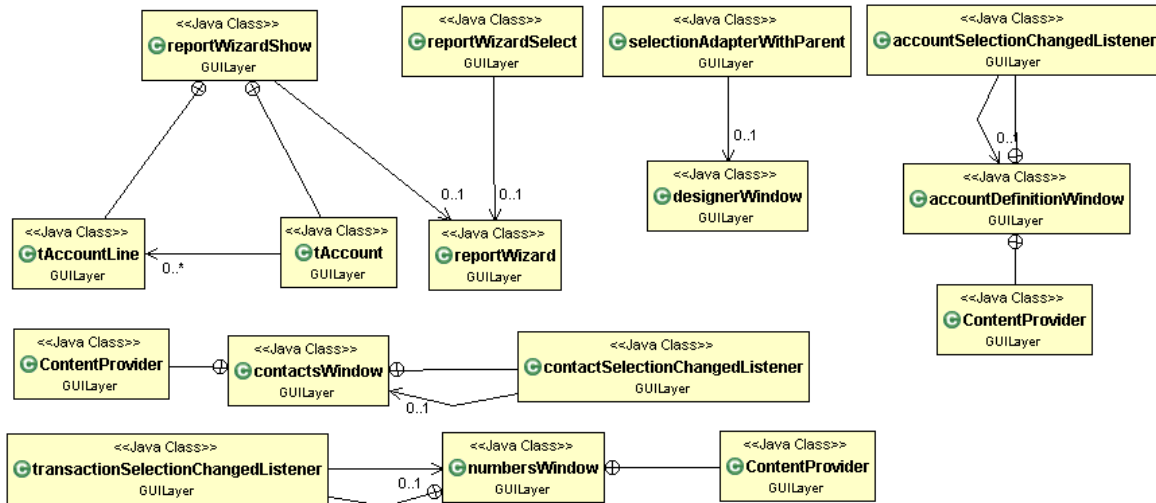
Gambar 4. 16 Class Diagram Package GUILayer part 1

Gambar 4. 17 adalah *class diagram* pada *package* GUILayer, terdapat relasi *association relationship*, *dependencies relationship* dan *nesting relationship*.



Gambar 4. 17 Class Diagram Package GUILayer part 2

Gambar 4. 18 adalah *class diagram* pada *package* GUILayer, terdapat relasi *association relationship*, *dependencies relationship* dan *nesting relationship*.



Gambar 4. 18 Class Diagram Package GUILayer part 3

(Halaman ini sengaja dikosongkan)

BAB V IMPLEMENTASI

Bab ini menjelaskan tentang tahap implementasi yang dilakukan pada tugas akhir ini. Tahapan ini berisi perhitungan Metrik Chidamber dan Kemerer yang meliputi perhitungan WMC, perhitungan DIT, perhitungan NOC, perhitungan RFC, perhitungan CBO, dan perhitungan LCOM.

Code dependency dan *class diagram* menjadi bahan dalam melakukan perhitungan Metrik Chidamber dan Kemerer. Terdapat enam metrik yang diukur yaitu *Weighted Method per Class* (WMC), *Depth of Inheritance Tree* (DIT), *Number OF Children* (NOC), *Response For a Class* (RFC), *Coupling Between Object Classes* (CBO) dan *Lack of Cohesion Method* (LCOM).

5.1. Weighted Method per Class (WMC)

Metrik *Weighted Method per Class* adalah jumlah semua *method* dan jumlah *decision points* (if/for/while) pada suatu *class*.

Contoh perhitungan Metrik WMC ada pada Gambar 5. 1 dan Gambar 5. 2. Gambar 5. 1 menjelaskan bahwa *Class contentBookingProvider* mempunyai nilai Metrik WMC tujuh karena memiliki tujuh *method* dan tidak ada *decision points*.


```

public class bookingContentProvider implements IStructuredContentProvider, Serializable {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    Vector<entry> entries=new Vector<entry>();

    public void addEntry(entry e) {
        entries.add(e);
        e.bindContentProvider(this);
    }

    public void removeEntry(entry e) {
        entries.remove(e);
    }

    public void removeAllEntries() {
        entries.removeAllElements();
    }

    public Object[] getElements(Object arg0) {
        return entries.toArray();
    }

    public void dispose() {
        // TODO Auto-generated method stub
    }

    public void inputChanged(Viewer arg0, Object arg1, Object arg2) {
        // TODO Auto-generated method stub
    }
}

```

Class bookingContentProvider

Gambar 5. 1 Source Code Contoh Metrik WMC (Class contentBookingProvider)

Pada Gambar 5. 2 terdapat contoh *code* yang memiliki *decision point*. Gambar 5. 2 menjelaskan bahwa HBCIcallbackdialog mempunyai nilai WMC empat, terdapat satu *methods* yaitu callback dan tiga *decision point* yaitu NEED_CONECTION, CLOSE CONNECTION, dan super.callback.

```

package GUIayer;
import org.kapott.hbci.callback.HBCIcallbackSwing;
public class HBCIcallbackDialog extends HBCIcallbackSwing {
    @Override
    public void callback(final HBCIPassport passport,int reason,String msg,int datatype,StringBuffer retData) {
        if (reason==NEED_CONNECTION || reason==CLOSE_CONNECTION) {
            // System.out.println("internet connection is always available"); //$NON-NLS-1$
        } else {
            super.callback(passport,reason,msg,datatype,retData);
        }
    }
}

```

Class HBCIcallbackDialog

Gambar 5. 2 Souce Code Metrik WMC (Class HBCIcallbackDialog)

5.1.1. Nilai Metrik WMC *Package dataLayer*

Tabel 5. 1 merupakan hasil perhitungan nilai Metrik WMC pada *package dataLayer*.

Tabel 5. 1 Nilai Metrik WMC *Package dataLayer*

No	Class name	Nilai WMC
1	DB	37
2	HBCI	26
3	HBCIImporter	7
4	Messages	4
5	OBDXImporter	31
6	fileImporters	22
7	fileUtils	18
8	gnuPG	32
9	ProcessStreamReader	6
10	invalidImportFormatException	1
11	moneyplexImporter	22
12	prematureException	1
13	prepareDB	28
14	starmoneyImporter	17
Total		252

Nilai WMC yang ada pada Tabel 5. 1 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package dataLayer* mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.1.2. Nilai Metrik WMC *Package appLayer*

Tabel 5. 2 merupakan hasil perhitungan nilai Metrik WMC pada *package appLayer*.

Tabel 5. 2 Nilai Metrik WMC *Package appLayer*

No	Class name	Nilai WMC
1	AccountNotFoundException	1
2	ItemListener	0
3	IcontextProvider	0

No	Class name	Nilai WMC
4	Messages	4
5	PaymentMethodNotFoundException	1
6	RoleNotFoundException	1
7	account	48
8	accountException	1
9	accountsList	69
10	application	7
11	asset	19
12	assets	8
13	bookingCellModifier	13
14	bookingContentProvider	7
15	bookingLabelProvider	7
16	client	18
17	configs	222
18	contact	42
19	contacts	17
20	contextComposite	3
21	document	39
22	documents	18
23	elementNotFoundException	1
24	entry	132
33	AccountNotFoundException	16
34	IItemListener	20
35	IcontextProvider	1
36	Messages	10
37	PaymentMethodNotFoundException	14
38	RoleNotFoundException	8
39	account	13
40	accountException	1
41	entryNotInThisAccountException	1
42	item	46
43	itemTableList	34
44	itemTextList	16
45	items	15
33	moneyTransfer	7
34	pageActivator	4

No	Class name	Nilai WMC
35	placeholderManager	4
36	product	15
37	products	18
38	transactionContentProvider	12
39	typeNotFoundException	1
40	utils	5
Total		938

Keseluruhan nilai WMC yang ada pada Tabel 5. 2 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class* accountsList berada pada *range* kuning atau *medium* dan terdapat dua *class* yaitu *class* configs dan entry berada pada *range* merah atau *bad*, berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.1.3. Nilai Metrik WMC *Package* appLayer.taxRelated

Tabel 5. 3 merupakan hasil perhitungan nilai Metrik WMC pada *package* appLayer.taxRelated.

Tabel 5. 3 Nilai Metrik WMC *Package* appLayer.taxRelated

No	Class name	Nilai WMC
1	IRSoffice	3
2	IRSoffices	3
3	state	10
4	states	3
5	tax	26
6	taxList	28
7	taxNotFoundException	1
Total		74

Nilai WMC yang ada pada Tabel 5. 3 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.1.4. Nilai Metrik WMC *Package* `appLayer.transactionRelated`

Tabel 5. 4 merupakan hasil perhitungan nilai Metrik WMC pada *package* `appLayer.transactionRelated`.

Tabel 5. 4 Nilai Metrik WMC *Package* `appLayer.transactionRelated`

No	Class name	Nilai WMC
1	cancelation	11
2	creditmemo	17
3	invoice	24
4	offer	9
5	personalDrawing	17
6	reminder	9
7	transaction	74
8	transactionTypeNotFoundException	1
9	transactions	25
Total		187

Keseluruhan nilai WMC yang ada pada Tabel 5. 4 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class* `transaction` berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* `dataLayer` mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.1.5. Nilai Metrik WMC *Package* `GUILayer`

Tabel 5. 5 merupakan hasil perhitungan nilai Metrik WMC pada *package* `GUILayer`.

Tabel 5. 5 Nilai Metrik WMC *Package* `GUILayer`

No	Class name	Nilai WMC
1	CComboBoxCellEditor	12
2	HBCIcallbackDialog	4
3	HSQLErrorCheck	8
4	MainWindow	20
5	Messages	4

No	Class name	Nilai WMC
6	OOoCheck	8
7	TimeCellEditor	12
8	VATannouncementWindow	15
9	aboutWindow	19
10	accountDefinitionWindow	15
11	ContentProvider	5
12	accountSelectionChangedListener	2
13	accountingEditWindow	19
14	assetWindow	13
15	ContentProvider	4
16	assetSelectionChangedListener	2
17	balanceWindow	17
18	browserWindow	10
19	communityWindow	11
20	configWindow	54
21	contactsWindow	24
22	ContentProvider	4
23	contactSelectionChangedListener	2
24	databaseConnectivityTest	3
25	databaseDriverNameTestCase	3
26	designerWindow	21
27	documentsWindow	10
28	documentCellModifier	5
29	documentLabelProvider	4
30	documentsList	1
31	entryDetailWindow	10
32	mainDirTestCase	3
33	newAccountingWizard	8
34	newAccountingWizardAdd	13
35	entryList	1
36	newAccountingWizardImport	15
37	newManualAccountingWizard	3
38	newTransactionSelectItemDetails	10
39	newTransactionSelectTransactionDetails	13
40	entryList	1
41	transactionCellModifier	10
42	transactionLabelProvider	3
43	newTransactionWizard	25

No	Class name	Nilai WMC
44	mailAuthenticator	2
45	mailThread	7
46	newTransactionWizardPreview	26
47	newTransactionWizardPrintAndSave	10
48	numbersWindow	12
49	ContentProvider	4
50	transactionSelectionChangedListener	2
51	passwordDialog	7
52	productsWindow	18
53	ContentProvider	4
54	productSelectionChangedListener	2
55	reportWizard	8
56	reportWizardSelect	5
57	reportWizardShow	27
58	tAccount	33
59	tAccountLine	3
60	selectionAdapterWithParent	3
61	settingsWindow	20
62	standardPositiveTestResult	1
63	tableCheck	5
64	tableWindow	0
65	taxesWindow	17
66	ContentProvider	4
67	taxSelectionChangedListener	2
68	templateCheck	5
69	testCase	11
70	testResult	5
71	testTest	2
72	timetrackerImportWindow	8
73	todoBalanceSelectionAdapter	10
74	todoWindow	42
75	transactionSelectionAdapter	2
76	viewerWindow	11
77	warningFocusAdapter	3
Total		767

Keseluruhan nilai WMC yang ada pada Tabel 5. 5 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class configWindow* berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package dataLayer* mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.1.6. Nilai *Average* Metrik WMC

Nilai *Average* Metrik WMC didapatkan dengan menjumlah semua nilai Metrik WMC dan dibagi dengan jumlah semua *class*. Perhitungan *Average* nilai Metrik WMC terdapat pada Tabel 5. 6 Nilai *Average* Metrik WMC.

Tabel 5. 6 Nilai *Average* Metrik WMC

Package	Jumlah Kelas	Nilai WMC
GUILayer	77	767
appLayer	45	938
appLayer.taxRelated	7	74
appLayer.transactionRelated	9	187
dataLayer	14	252
Total	152	1565
Average WMC	$= \frac{\sum WMC}{\sum class}$ $= \mathbf{10,28}$	

Hasil perhitungan pada Tabel 5. 6 menunjukkan bahwa keseluruhan nilai Metrik WMC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package dataLayer* mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2. Depth of Inheritance Tree (DIT)

Depth of Inheritance Tree *metric* berasal dari *inheritance tree* yang berisi *class* dengan menentukan panjang *path* dari *class* kepada *root* (yaitu, *java.lang*), *java.lang* dianggap selalu 0.

Kedalaman dari *class* dalam *hierarki inheritance* adalah panjang maksimum dari *node-node class* ke *root* dari *tree* dan diukur oleh jumlah dari induk *class*. adapun contohnya dijelaskan pada Gambar 5. 3 bahwa *class items* extends terhadap *class transactionContentProvider*. Pada Gambar 5. 4 terdapat *class diagram* dari *class items* dan *class transactionContentProvider* yang mempunyai relasi *generalization*.

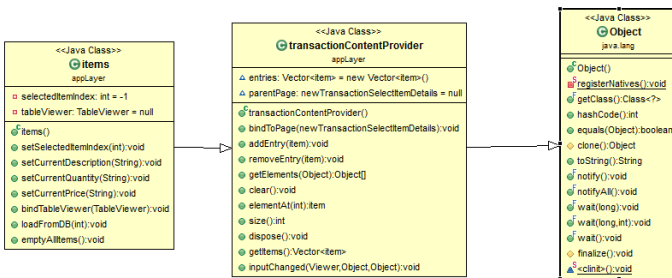
```
public class items extends transactionContentProvider {
    private int selectedIndex = -1;
    private TableView tableViewer=null;

    public void setSelectedItemIndex(int index) {
        selectedIndex=index;
    }
}
```

Class items

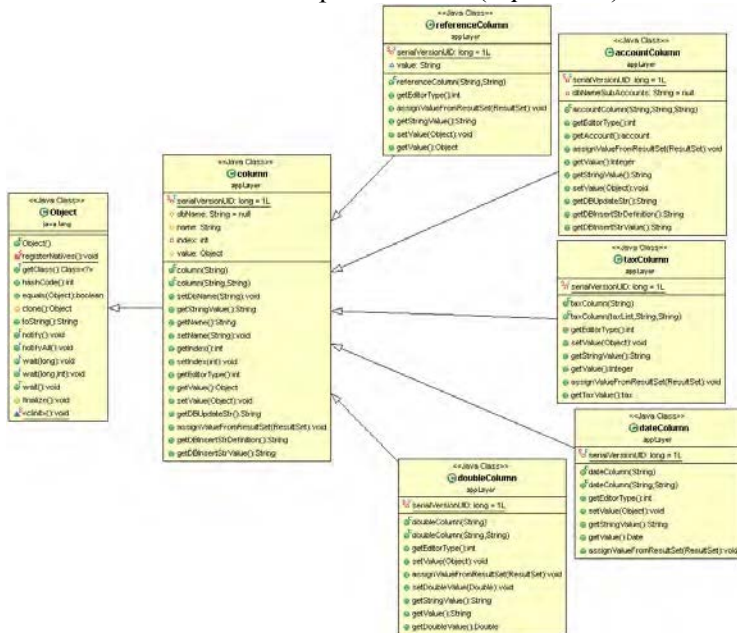
Gambar 5. 3 Source Code Contoh Metrik DIT (Class items)

Class items diturunkan dari *class transactionContentProvider* (*Superclass*), *class items* mewarisi data/behavior dari *class transactionContentProvider*.



Gambar 5. 4 Class Diagram contoh Metrik DIT (Class items)

Nilai Metrik DIT untuk *class* items adalah 2 karena *class* items merupakan *root* yang menjadi turunan ke-2 dari *leaves*, nilai Metrik DIT *class* transaction Content Provider adalah 1 karena *class* tersebut merupakan turunan dari *class* object (java.lang) dan *class* object mempunyai nilai Metrik DIT 0 karena *class* tersebut merupakan *leaves* (*superclass*).



Gambar 5. 5 Class Diagram contoh Metrik DIT

Contoh Perhitungan *average* Metrik DIT pada *class diagram* yang digambarkan pada Gambar 5. 5 didapatkan dengan menggunakan rumus pada persamaan 5.1 [30].

$$AvgDIT = \frac{(0 \times n_0) + (1 \times n_1) + (2 \times n_2) + \dots}{\sum n} \quad (5.1)$$

Contoh Perhitungan *average* Metrik DIT dari persamaan 5.1 menghasilkan nilai 1,57 sesuai dengan perhitungan pada Gambar 5. 6.

$$AvgDIT = \frac{(0 \times 1) + (1 \times 1) + (2 \times 5)}{7} = 1,57$$

Gambar 5. 6 Contoh Perhitungan *Average* Metrik DIT

5.2.1. Nilai Metrik DIT *Package* dataLayer

Tabel 5. 7 merupakan hasil perhitungan nilai Metrik DIT pada *package* dataLayer. *Inheritance tree* sebagai pedoman perhitungan DIT dapat dilihat pada LAMPIRAN B – *Generaization (Inheritance) Relationship*.

Tabel 5. 7 Nilai Metrik DIT *Package* dataLayer

No	Nama Class	Nilai DIT
1	DB	1
2	HBCI	1
3	HBCIImporter	1
4	Messages	1
5	OBDXImporter	1
6	fileImporters	1
7	fileUtils	1
8	gnuPG	1
9	ProcessStreamReader	1
10	invalidImportFormatException	1
11	moneyplexImporter	1
12	prematureException	1
13	prepareDB	1
14	starmoneyImporter	1

Nilai DIT yang ada pada Tabel 5. 7 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2.2. Nilai Metrik DIT *Package* appLayer

Tabel 5. 8 merupakan hasil perhitungan nilai Metrik DIT pada *package* appLayer. *Inheritance tree* sebagai pedoman

perhitungan DIT dapat dilihat pada LAMPIRAN B – *Generaization (Inheritance) Relationship*.

Tabel 5. 8 Nilai Metrik DIT Package appLayer

No	Nama Class	Nilai DIT
1	AccountNotFoundException	1
2	ItemListener	0
3	IcontextProvider	0
4	Messages	1
5	PaymentMethodNotFoundException	1
6	RoleNotFoundException	1
7	account	1
8	accountException	1
9	accountsList	1
10	application	1
11	asset	1
12	assets	1
13	bookingCellModifier	1
14	bookingContentProvider	1
15	bookingLabelProvider	1
16	client	1
17	configs	1
18	contact	1
19	contacts	1
20	contextComposite	1
21	document	1
22	documents	1
23	elementNotFoundException	1
24	entry	1
25	accountColumn	2
26	column	1
27	columnNotFoundException	1
28	dateColumn	2
29	doubleColumn	2
30	referenceColumn	2
31	taxColumn	2
32	undefinedNumberException	1

No	Nama Class	Nilai DIT
33	entryNotInThisAccountException	1
34	item	1
35	itemTableList	1
36	itemTextList	1
37	items	2
38	moneyTransfer	1
39	pageActivator	1
40	placeholderManager	1
41	product	1
42	products	1
43	transactionContentProvider	1
44	typeNotFoundException	1
45	utils	1
Total		49

Nilai DIT yang ada pada Tabel 5. 8 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* *appLayer* mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2.3. Nilai Metrik DIT *Package appLayer.taxRelated*

Tabel 5. 9 merupakan hasil perhitungan nilai Metrik DIT pada *package appLayer.taxRelated*. *Inheritance tree* sebagai pedoman perhitungan DIT dapat dilihat pada LAMPIRAN B – *Generaization (Inheritance) Relationship*.

Tabel 5. 9 Nilai Metrik DIT *Package appLayer*

No	Nama Class	Nilai DIT
1	IRSooffice	1
2	IRSoffices	1
3	state	1
4	states	1
5	tax	1
6	taxList	1
7	taxNotFoundException	1

No	Nama Class	Nilai DIT
	Total	7

Nilai DIT yang ada pada Tabel 5. 9 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* `appLayer.taxRelated` mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2.4. Nilai Metrik DIT Package `appLayer.transactionRelated`

Tabel 5. 10 merupakan hasil perhitungan nilai Metrik DIT pada *package* `appLayer.transaxtionRelated`. *Inheritance tree* sebagai pedoman perhitungan DIT dapat dilihat pada LAMPIRAN B – *Generaization (Inheritance) Relationship*.

Tabel 5. 10 Nilai Metrik DIT Package `appLayer`

No	Nama Class	Nilai DIT
1	<code>cancelation</code>	2
2	<code>creditmemo</code>	2
3	<code>invoice</code>	2
4	<code>offer</code>	2
5	<code>personalDrawing</code>	2
6	<code>reminder</code>	2
7	<code>transaction</code>	1
8	<code>transactionTypeNotFoundException</code>	1
9	<code>transactions</code>	1
	Total	15

Nilai DIT yang ada pada Tabel 5. 10 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* `appLayer.transactionRelated` mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2.5. Nilai Metrik DIT *Package* GUILayer

Tabel 5. 11 merupakan hasil perhitungan nilai Metrik DIT pada *package* GUILayer. *Inheritance tree* sebagai pedoman perhitungan DIT dapat dilihat pada LAMPIRAN B – *Generaization (Inheritance) Relationship*.

Tabel 5. 11 Nilai Metrik DIT *Package* GUILayer

No	Nama Class	Nilai DIT
1	CComboBoxCellEditor	1
2	HBCIcallbackDialog	1
3	HSQLLockCheck	2
4	MainWindow	1
5	Messages	1
6	OOoCheck	2
7	TimeCellEditor	1
8	VATannouncementWindow	1
9	aboutWindow	1
10	accountDefinitionWindow	1
11	ContentProvider	1
12	accountSelectionChangedListener	1
13	accountingEditWindow	1
14	assetWindow	1
15	ContentProvider	1
16	assetSelectionChangedListener	1
17	balanceWindow	1
18	browserWindow	1
19	communityWindow	1
20	configWindow	1
21	contactsWindow	1
22	ContentProvider	1
23	contactSelectionChangedListener	1
24	databaseConnectivityTest	2
25	databaseDriverNameTestCase	2
26	designerWindow	1
27	documentsWindow	1
28	documentCellModifier	1
29	documentLabelProvider	1
30	documentsList	1

No	Nama Class	Nilai DIT
31	entryDetailWindow	1
32	mainDirTestCase	2
33	newAccountingWizard	1
34	newAccountingWizardAdd	1
35	entryList	1
36	newAccountingWizardImport	1
37	newManualAccountingWizard	1
38	newTransactionSelectItemDetails	1
39	newTransactionSelectTransactionDetails	1
40	entryList	1
41	transactionCellModifier	1
42	transactionLabelProvider	1
43	newTransactionWizard	1
44	mailAuthenticator	1
45	mailThread	1
46	newTransactionWizardPreview	1
47	newTransactionWizardPrintAndSave	1
48	numbersWindow	1
49	ContentProvider	1
50	transactionSelectionChangedListener	1
51	passwordDialog	1
52	productsWindow	1
53	ContentProvider	1
54	productSelectionChangedListener	1
55	reportWizard	1
56	reportWizardSelect	1
57	reportWizardShow	1
58	tAccount	1
59	tAccountLine	1
60	selectionAdapterWithParent	1
61	settingsWindow	1
62	standardPositiveTestResult	2
63	tableCheck	2
64	tableWindow	0
65	taxesWindow	1
66	ContentProvider	1
67	taxSelectionChangedListener	1
68	templateCheck	2

No	Nama Class	Nilai DIT
69	testCase	1
70	testResult	1
71	testTest	2
72	timetrackerImportWindow	1
73	todoBalanceSelectionAdapter	1
74	todoWindow	1
75	transactionSelectionAdapter	1
76	viewerWindow	1
77	warningFocusAdapter	1
Total		86

Keseluruhan nilai DIT yang ada pada Tabel 5. 11 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* GUILayer mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.2.6. Nilai AVERAGE Metrik DIT

Nilai *Average* Metrik Metrik DIT didapatkan dengan menjumlah semua nilai Metrik DIT dan dibagi dengan jumlah semua *class*. Perhitungan *Average* nilai Metrik DIT terdapat pada

$$AvgDIT = \frac{(0 \times 3) + (1 \times 128) + (2 \times 21)}{7} = 1,1$$

Gambar 5. 7 *Average* Metrik DIT

Hasil perhitungan pada Gambar 5. 7 menunjukkan bahwa nilai Metrik DIT pada *Software Accounting* XYZ terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.3.1. Nilai Metrik NOC *Package dataLayer*

Tabel 5. 12 merupakan hasil perhitungan nilai Metrik NOC pada *package dataLayer*.

Tabel 5. 12 Nilai Metrik NOC *Package dataLayer*

No	Nama Class	Nilai NOC
1	DB	0
2	HBCI	0
3	HBCIImporter	0
4	Messages	0
5	OBDXImporter	0
6	fileImporters	0
7	fileUtils	0
8	gnuPG	0
9	ProcessStreamReader	0
10	invalidImportFormatException	0
11	moneyplexImporter	0
12	prematureException	0
13	prepareDB	0
14	starmoneyImporter	0
Total		0

Nilai NOC yang ada pada Tabel 5. 12 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package dataLayer* mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.3.2. Nilai Metrik NOC *Package appLayer*

Tabel 5. 13 merupakan hasil perhitungan nilai Metrik NOC pada *package appLayer*.

Tabel 5. 13 Nilai Metrik NOC *Package appLayer*

No	Nama Class	Nilai NOC
1	AccountNotFoundException	0
2	ItemListener	0
3	IcontextProvider	0

No	Nama Class	Nilai NOC
4	Messages	0
5	PaymentMethodNotFoundException	0
6	RoleNotFoundException	0
7	account	0
8	accountException	0
9	accountsList	0
10	application	0
11	asset	0
12	assets	0
13	bookingCellModifier	0
14	bookingContentProvider	0
15	bookingLabelProvider	0
16	client	0
17	configs	0
18	contact	0
19	contacts	0
20	contextComposite	0
21	document	0
22	documents	0
23	elementNotFoundException	0
24	entry	0
25	accountColumn	0
26	column	5
27	columnNotFoundException	0
28	dateColumn	0
29	doubleColumn	0
30	referenceColumn	0
31	taxColumn	0
32	undefinedNumberException	0
33	entryNotInThisAccountException	0
34	item	0
35	itemTableList	0
36	itemTextList	0
37	items	0
38	moneyTransfer	0
39	pageActivator	0

No	Nama Class	Nilai NOC
40	placeholderManager	0
41	product	0
42	products	0
43	transactionContentProvider	1
44	typeNotFoundException	0
45	utils	0
Total		6

Keseluruhan nilai NOC yang ada pada Tabel 5. 13 berada di *range* hijau atau *good* namun, terdapat satu class yaitu *class* column yang berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Nilai Metrik NOC *Package* appLayer.taxRelated

Tabel 5. 14 merupakan hasil perhitungan nilai Metrik NOC pada *package* appLayer.taxRelated.

Tabel 5. 14 Nilai Metrik NOC *Package* appLayer.taxRelated

No	Nama Class	Nilai NOC
1	IRSooffice	0
2	IRSooffices	0
3	state	0
4	states	0
5	tax	0
6	taxList	0
7	taxNotFoundException	0
Total		0

Nilai NOC yang ada pada Tabel 5. 14 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.3.3. Nilai Metrik NOC Package *appLayer.transactionRelated*

Tabel 5. 15 merupakan hasil perhitungan nilai Metrik NOC pada *package appLayer.transaxtionRelated*.

Tabel 5. 15 Nilai Metrik NOC Package *appLayer.transactionRelated*

No	Nama Class	Nilai NOC
1	cancelation	0
2	creditmemo	0
3	invoice	0
4	offer	0
5	personalDrawing	0
6	reminder	0
7	transaction	6
8	transactionTypeNotFoundException	0
9	transactions	0
Total		6

Keseluruhan nilai NOC yang ada pada Tabel 5. 15 berada di *range* hijau atau *good* namun, terdapat satu class yaitu *class transaction* yang berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package dataLayer* mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.3.4. Nilai Metrik NOC Package *GUILayer*

Tabel 5. 16 merupakan hasil perhitungan nilai Metrik NOC pada *package GUILayer*.

Tabel 5. 16 Nilai Metrik NOC Package *GUILayer*

No	Nama Class	Nilai NOC
1	CComboBoxCellEditor	0
2	HBCIcallbackDialog	0
3	HSQLLockCheck	0
4	MainWindow	0
5	Messages	0

No	Nama Class	Nilai NOC
6	OOoCheck	0
7	TimeCellEditor	0
8	VATannouncementWindow	0
9	aboutWindow	0
10	accountDefinitionWindow	0
11	ContentProvider	0
12	accountSelectionChangedListener	0
13	accountingEditWindow	0
14	assetWindow	0
15	ContentProvider	0
16	assetSelectionChangedListener	0
17	balanceWindow	0
18	browserWindow	0
19	communityWindow	0
20	configWindow	0
21	contactsWindow	0
22	ContentProvider	0
23	contactSelectionChangedListener	0
24	databaseConnectivityTest	0
25	databaseDriverNameTestCase	0
26	designerWindow	0
27	documentsWindow	0
28	documentCellModifier	0
29	documentLabelProvider	0
30	documentsList	0
31	entryDetailWindow	0
32	mainDirTestCase	0
33	newAccountingWizard	0
34	newAccountingWizardAdd	0
35	entryList	0
36	newAccountingWizardImport	0
37	newManualAccountingWizard	0
38	newTransactionSelectItemDetails	0
39	newTransactionSelectTransactionDetails	0
40	entryList	0
41	transactionCellModifier	0
42	transactionLabelProvider	0
43	newTransactionWizard	0

No	Nama Class	Nilai NOC
44	mailAuthenticator	0
45	mailThread	0
46	newTransactionWizardPreview	0
47	newTransactionWizardPrintAndSave	0
48	numbersWindow	0
49	ContentProvider	0
50	transactionSelectionChangedListener	0
51	passwordDialog	0
52	productsWindow	0
53	ContentProvider	0
54	productSelectionChangedListener	0
55	reportWizard	0
56	reportWizardSelect	0
57	reportWizardShow	0
58	tAccount	0
59	tAccountLine	0
60	selectionAdapterWithParent	0
61	settingsWindow	0
62	standardPositiveTestResult	0
63	tableCheck	0
64	tableWindow	0
65	taxesWindow	0
66	ContentProvider	0
67	taxSelectionChangedListener	0
68	templateCheck	0
69	testCase	8
70	testResult	1
71	testTest	0
72	timetrackerImportWindow	0
73	todoBalanceSelectionAdapter	0
74	todoWindow	0
75	transactionSelectionAdapter	0
76	viewerWindow	0
77	warningFocusAdapter	0
Total		9

Keseluruhan nilai NOC yang ada pada Tabel 5. 16 berada di *range* hijau atau *good* namun, terdapat satu class yaitu *class*

testCase yang berada pada range kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.3.5. Nilai *Average* Metrik NOC

Nilai *Average* Metrik Metrik NOC didapatkan dengan menjumlah semua nilai Metrik NOC dan dibagi dengan jumlah semua *class*. Perhitungan *Average* nilai Metrik NOC terdapat pada Tabel 5. 17.

Tabel 5. 17 Nilai *Average* Metrik NOC

Package	Jumlah Kelas	Nilai NOC
GUILayer	77	9
appLayer	45	6
appLayer.taxRelated.IRSoffice	7	0
appLayer.transactionRelated	9	6
dataLayer	14	0
Total	152	21
Average NOC	$= \frac{\sum NOC}{\sum class}$ $= 0,12$	

Hasil perhitungan pada Tabel 5. 17 menunjukkan bahwa keseluruhan nilai Metrik NOC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4. Coupling Between Object Classes (CBO)

Coupling between Objects *metric* diperoleh dengan melihat pada *code dependency* dengan menghitung jumlah *class* lain

(kecuali *class* yang di-*extend* atau di-*implement*) yang digunakan di dalam *class* tersebut.

Class dikatakan *coupled* jika ada panggilan *method* dari salah satu *class* untuk *method class* lain dan jika tidak ada warisan hubungan antara dua *class*, tidak hanya anggota *method* yang dihitung, tetapi juga *method nested classes*. *Konstruktor* dan *initializers* kelas diabaikan.

Adapun contohnya adalah pada *package* `appLayer.transactionRElated` terdapat *method* dari *class* `creditmemo` yang memanggil *class* pada *package* `appLayer` dan `appLayer.taxRelated`, lebih detailnya dapat dilihat pada *code dependency* Tabel 4. 16 dan Tabel 4. 15. *Class* tersebut *coupled* dengan *class* `account`, *class* `transactions`, *class* `AccountNotFoundException`, *class* `accountException`, *class* `taxList` dan *class* `entry`. Sehingga *class* `appLayer.transactionRelated.creditmemo` mempunyai nilai CBO 6.

5.4.1. Nilai Metrik CBO *Package dataLayer*

Tabel 5. 18 merupakan hasil perhitungan nilai Metrik CBO pada *package* `dataLayer`.

Tabel 5. 18 Nilai Metrik CBO *Package dataLayer*

No	Nama Class	Nilai CBO
1	DB	40
2	HBCI	6
3	HBCIImporter	4
4	Messages	4
5	OBDXImporter	4
6	fileImporters	7
7	fileUtils	6
8	gnuPG	2
9	ProcessStreamReader	-
10	invalidImportFormatException	0
11	moneyplexImporter	4

No	Nama Class	Nilai CBO
12	prematureException	0
13	prepareDB	9
14	starmoneyImporter	3
Total		77

Keseluruhan nilai CBO yang ada pada Tabel 5. 18 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class* DB berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4.2. Nilai Metrik CBO Package appLayer

Tabel 5. 19 merupakan hasil perhitungan nilai Metrik CBO pada *package* appLayer.

Tabel 5. 19 Nilai Metrik CBO Package appLayer

No	Nama Class	Nilai CBO
1	AccountNotFoundException	0
2	ItemListener	1
3	IcontextProvider	0
4	Messages	18
5	PaymentMethodNotFoundException	0
6	RoleNotFoundException	0
7	account	15
8	accountException	0
9	accountsList	19
10	application	25
11	asset	4
12	assets	5
13	bookingCellModifier	5
14	bookingContentProvider	3
15	bookingLabelProvider	3
16	client	56
17	configs	41

No	Nama Class	Nilai CBO
18	contact	13
19	contacts	8
20	contextComposite	3
21	document	6
22	documents	8
23	elementNotFoundException	0
24	entry	24
25	accountColumn	-
26	column	-
27	columnNotFoundException	-
28	dateColumn	-
29	doubleColumn	-
30	referenceColumn	-
31	taxColumn	-
32	undefinedNumberException	-
33	entryNotInThisAccountException	0
34	item	13
35	itemTableList	5
36	itemTextList	5
37	items	5
38	moneyTransfer	7
39	pageActivator	1
40	placeholderManager	8
41	product	8
42	products	8
43	transactionContentProvider	6
44	typeNotFoundException	0
45	utils	7
Total		330

Keseluruhan nilai CBO yang ada pada Tabel 5. 19 berada di *range* hijau atau *good* namun, terdapat tujuh *class* yaitu *class* messages, *class* transaction, *class* account, *class* accountsList, *class* application, *class* client, *class* configs, *class* entry berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package*

dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4.3. Nilai Metrik CBO *Package appLayer.taxRelated*

Tabel 5. 20 merupakan hasil perhitungan nilai Metrik CBO pada *package appLayer.taxRelated*.

Tabel 5. 20 Nilai Metrik CBO *Package appLayer.taxRelated*

No	Nama Class	Nilai CBO
1	IRSooffice	2
2	IRSooffices	2
3	state	4
4	states	2
5	tax	12
6	taxList	16
7	taxNotFoundException	0
Total		38

Keseluruhan nilai CBO yang ada pada Tabel 5. 20 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class taxList* berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package dataLayer* mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4.4. Nilai Metrik CBO *Package appLayer.transactionRelated*

Tabel 5. 21 merupakan hasil perhitungan nilai Metrik CBO pada *package appLayer.transaxtionRelated*.

Tabel 5. 21 Nilai Metrik CBO *Package appLayer.transactionRelated*

No	Nama Class	Nilai CBO
1	cancelation	5
2	creditmemo	6
3	invoice	11
4	offer	1
5	personalDrawing	6

No	Nama Class	Nilai CBO
6	reminder	1
7	transaction	26
8	transactionTypeNotFoundException	0
9	transactions	18
Total		74

Keseluruhan nilai CBO yang ada pada Tabel 5. 21 berada di *range* hijau atau *good* namun, terdapat dua *class* yaitu *class* transaction dan *class* transactions berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4.5. Nilai Metrik CBO Package GUILayer

Tabel 5. 22 merupakan hasil perhitungan nilai Metrik CBO pada *package* GUILayer.

Tabel 5. 22 Nilai Metrik CBO Package GUILayer

No	Nama Class	Nilai CBO
1	CComboBoxCellEditor	0
2	HBCIcallbackDialog	0
3	HSQLErrorCheck	4
4	MainWindow	13
5	Messages	40
6	OOoCheck	4
7	TimeCellEditor	2
8	VATannouncementWindow	16
9	aboutWindow	7
10	accountDefinitionWindow	7
11	ContentProvider	-
12	accountSelectionChangedListener	-
13	accountingEditWindow	11
14	assetWindow	6
15	ContentProvider	-
16	assetSelectionChangedListener	-
17	balanceWindow	15

No	Nama Class	Nilai CBO
18	browserWindow	2
19	communityWindow	3
20	configWindow	16
21	contactsWindow	10
22	ContentProvider	-
23	contactSelectionChangedListener	-
24	databaseConnectivityTest	3
25	databaseDriverNameTestCase	3
26	designerWindow	10
27	documentsWindow	6
28	documentCellModifier	-
29	documentLabelProvider	-
30	documentsList	-
31	entryDetailWindow	7
32	mainDirTestCase	2
33	newAccountingWizard	2
34	newAccountingWizardAdd	
35	entryList	13
36	newAccountingWizardImport	5
37	newManualAccountingWizard	1
38	newTransactionSelectItemDetails	13
39	newTransactionSelectTransactionDetails	-
40	entryList	-
41	transactionCellModifier	-
42	transactionLabelProvider	8
43	newTransactionWizard	10
44	mailAuthenticator	-
45	mailThread	-
46	newTransactionWizardPreview	9
47	newTransactionWizardPrintAndSave	8
48	numbersWindow	8
49	ContentProvider	-
50	transactionSelectionChangedListener	-
51	passwordDialog	3
52	productsWindow	11
53	ContentProvider	-
54	productSelectionChangedListener	-
55	reportWizard	6

No	Nama Class	Nilai CBO
56	reportWizardSelect	4
57	reportWizardShow	8
58	tAccount	-
59	tAccountLine	-
60	selectionAdapterWithParent	3
61	settingsWindow	10
62	standardPositiveTestResult	0
63	tableCheck	4
64	tableWindow	0
65	taxesWindow	13
66	ContentProvider	-
67	taxSelectionChangedListener	-
68	templateCheck	2
69	testCase	2
70	testResult	6
71	testTest	0
72	timetrackerImportWindow	6
73	todoBalanceSelectionAdapter	9
74	todoWindow	14
75	transactionSelectionAdapter	2
76	viewerWindow	3
77	warningFocusAdapter	1
Total		371

Keseluruhan nilai CBO yang ada pada Tabel 5. 22 berada di *range* hijau atau *good* namun, terdapat lima *class* yaitu *class* messages, *class* VATannouncementWindow, *class* balanceWindow, *class* configWindow dan *class* todoWindow berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.4.6. Nilai AVERAGE Metrik CBO

Nilai AVERAGE Metrik Metrik CBO didapatkan dengan menjumlah semua nilai Metrik CBO dan dibagi dengan jumlah

semua *class*. Perhitungan *Average* nilai Metrik CBO terdapat pada

Tabel 5. 23 Nilai *Average* Metrik CBO

Package	Jumlah Kelas	Nilai CBO
GUILayer	54	371
appLayer	37	330
appLayer.taxRelated.IRSoffice	7	38
appLayer.transactionRelated	9	74
dataLayer	13	77
Total	120	890
Average CBO	$= \frac{\sum CBO}{\sum class}$ $= 7,52$	

Hasil perhitungan pada

menunjukkan bahwa keseluruhan nilai Metrik CBO pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5. Response For a Class (RFC)

Metrik *Response for a Class* digunakan untuk menghitung jumlah *method* pada kelas ditambah jumlah *method* pada kelas lain dari *code base*, yang dipanggil langsung dari salah satu *method* di kelas tersebut (setiap *method* dihitung sekali, walaupun telah dipanggil beberapa kali).

Metrik RFC dihitung dari jumlah dari fungsi atau *procedure* yang dapat secara potensial dieksekusi dalam sebuah *class*.

Secara khusus, ini adalah jumlah dari operasi yang secara langsung diminta oleh member operasi dalam *class* ditambah member operasi mereka sendiri.

Cara menghitung RFC dengan melihat *class diagram* dan *code dependency* dengan menghitung banyaknya *method* lokal dan *method* yang dipanggil oleh *method* lokal termasuk semua *method* dalam kelas hirarki.

Pada *class* *starmoneyImporter* terdapat dua *method* yaitu *starmoneyImporter* dan *method* *run*. Pada *code dependency* Tabel 4. 2 dijelaskan bahwa *class* *starmoneyImporter* *calls method* *getClient(),getConnection, insert, invalidImportFormatException, dan parseCSVLineSemicolon*. Terdapat lima *method* yang dipanggil dan dua *method class* tersebut. Sehingga nilai RFC *class* *starmoneyImporter* adalah tujuh.

5.5.1. Nilai Metrik RFC *Package dataLayer*

Tabel 5. 24 merupakan hasil perhitungan nilai Metrik RFC pada *package dataLayer*.

Tabel 5. 24 Nilai Metrik RFC *Package dataLayer*

No	Nama Class	Nilai RFC
1	DB	16
2	HBCI	16
3	HBCIImporter	10
4	Messages	3
5	OBDXImporter	17
6	fileImporters	12
7	fileUtils	5
8	gnuPG	15
9	ProcessStreamReader	3
10	invalidImportFormatException	1
11	moneyplexImporter	8
12	prematureException	2
13	prepareDB	19
14	starmoneyImporter	7

No	Nama Class	Nilai RFC
Total		134

Nilai RFC yang ada pada Tabel 5. 24 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understanbility* dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5.2. Nilai Metrik RFC Package appLayer

Tabel 5. 25 merupakan hasil perhitungan nilai Metrik RFC pada *package* appLayer.

Tabel 5. 25 Nilai Metrik RFC Package appLayer

No	Nama Class	Nilai RFC
1	AccountNotFoundException	1
2	ItemListener	1
3	IcontextProvider	2
4	Messages	3
5	PaymentMethodNotFoundException	1
6	RoleNotFoundException	1
7	account	42
8	accountException	1
9	accountsList	53
10	application	8
11	asset	22
12	assets	9
13	bookingCellModifier	12
14	bookingContentProvider	8
15	bookingLabelProvider	9
16	client	21
17	configs	84
18	contact	45
19	contacts	16
20	contextComposite	3
21	document	28
22	documents	22

No	Nama Class	Nilai RFC
23	elementNotFoundException	1
24	entry	112
25	AccountNotFoundException	18
26	ItemListener	16
27	IcontextProvider	2
28	Messages	9
29	PaymentMethodNotFoundException	11
30	RoleNotFoundException	8
31	account	14
32	accountException	1
33	entryNotInThisAccountException	1
34	item	46
35	itemTableList	10
36	itemTextList	10
37	items	19
38	moneyTransfer	16
39	pageActivator	3
40	placeholderManager	33
41	product	20
42	products	21
43	transactionContentProvider	14
44	typeNotFoundException	1
45	utils	5
Total		783

Keseluruhan nilai RFC yang ada pada Tabel 5. 25 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class* entry berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understanbility* dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5.3. Nilai Metrik RFC *Package* appLayer.taxRelated

Tabel 5. 26 merupakan hasil perhitungan nilai Metrik RFC pada *package* appLayer.taxRelated.

Tabel 5. 26 Nilai Metrik RFC Package appLayer.taxRelated

No	Nama Class	Nilai RFC
1	IRSooffice	3
2	IRSoffices	4
3	state	8
4	states	5
5	tax	30
6	taxList	28
7	taxNotFoundException	1
Total		79

Nilai RFC yang ada pada Tabel 5. 26 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understanbility* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5.4. Nilai Metrik RFC Package appLayer.transaxtionRelated

Tabel 5. 27 merupakan hasil perhitungan nilai Metrik RFC pada *package* appLayer.transaxtionRelated

Tabel 5. 27 Nilai Metrik RFC Package appLayer.transactionRelated

No	Nama Class	Nilai RFC
1	cancelation	30
2	creditmemo	31
3	invoice	36
4	offer	11
5	personalDrawing	31
6	reminder	11
7	transaction	101
8	transactionTypeNotFoundExpection	2
9	transactions	36
Total		289

Keseluruhan nilai RFC yang ada pada Tabel 5. 27 berada di *range* hijau atau *good* namun, terdapat satu *class* yaitu *class*

transaction berada pada *range* kuning atau kategori *medium* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understability* dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5.5. Nilai Metrik RFC *Package* GUILayer

Tabel 5. 28 merupakan hasil perhitungan nilai Metrik RFC pada *package* GUILayer.

Tabel 5. 28 Nilai Metrik RFC *Package* GUILayer

No	Nama Class	Nilai RFC
1	CComboBoxCellEditor	6
2	HBCIcallbackDialog	2
3	HSQLLockCheck	10
4	MainWindow	54
5	Messages	3
6	OoOCheck	9
7	TimeCellEditor	12
8	VATannouncementWindow	20
9	aboutWindow	19
10	accountDefinitionWindow	27
11	ContentProvider	8
12	accountSelectionChangedListener	3
13	accountingEditWindow	35
14	assetWindow	24
15	ContentProvider	8
16	assetSelectionChangedListener	3
17	balanceWindow	34
18	browserWindow	12
19	communityWindow	13
20	configWindow	60
21	contactsWindow	41
22	ContentProvider	8
23	contactSelectionChangedListener	3
24	databaseConnectivityTest	8
25	databaseDriverNameTestCase	8
26	designerWindow	27

No	Nama Class	Nilai RFC
27	documentsWindow	18
28	documentCellModifier	6
29	documentLabelProvider	6
30	documentsList	1
31	entryDetailWindow	14
32	mainDirTestCase	7
33	newAccountingWizard	11
34	newAccountingWizardAdd	41
35	entryList	1
36	newAccountingWizardImport	16
37	newManualAccountingWizard	5
38	newTransactionSelectItemDetails	24
39	newTransactionSelectTransactionDetails	30
40	entryList	1
41	transactionCellModifier	11
42	transactionLabelProvider	4
43	newTransactionWizard	37
44	mailAuthenticator	2
45	mailThread	8
46	newTransactionWizardPreview	27
47	newTransactionWizardPrintAndSave	23
48	numbersWindow	24
49	ContentProvider	6
50	transactionSelectionChangedListener	3
51	passwordDialog	8
52	productsWindow	28
53	ContentProvider	8
54	productSelectionChangedListener	3
55	reportWizard	11
56	reportWizardSelect	9
57	reportWizardShow	36
58	tAccount	19
59	tAccountLine	3
60	selectionAdapterWithParent	6
61	settingsWindow	37
62	standardPositiveTestResult	3
63	tableCheck	8
64	tableWindow	1

No	Nama Class	Nilai RFC
65	taxesWindow	33
66	ContentProvider	8
67	taxSelectionChangedListener	3
68	templateCheck	9
69	testCase	7
70	testResult	5
71	testTest	4
72	timetrackerImportWindow	14
73	todoBalanceSelectionAdapter	24
74	todoWindow	38
75	transactionSelectionAdapter	4
76	viewerWindow	13
77	warningFocusAdapter	3
78		1128

Nilai RFC yang ada pada Tabel 5. 28 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *understanbility* dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

5.5.6. Nilai AVERAGE Metrik RFC

Nilai *Average* Metrik Metrik RFC didapatkan dengan menjumlah semua nilai Metrik RFC dan dibagi dengan jumlah semua *class*. Perhitungan *Average* nilai Metrik RFC terdapat pada Tabel 5. 29.

Tabel 5. 29 Nilai *Average* Metrik RFC

Package	Jumlah Kelas	Nilai RFC
GUILayer	77	1128
appLayer	45	783
appLayer.taxRelated.IRSoffice	7	79
appLayer.transactionRelated	9	289
dataLayer	14	134
Total	152	2413

Average RFC	$= \frac{\sum RFC}{\sum class}$ $= 11,58$
-------------	---

Hasil perhitungan pada Tabel 5. 29 menunjukkan bahwa keseluruhan nilai Metrik RFC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* *dataLayer* mempunyai *understanbility* dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

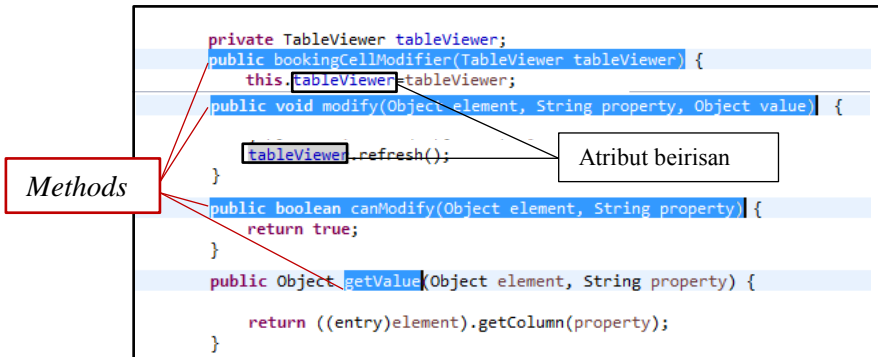
5.6. Lack of Cohesion Method (LCOM)

Metrik *Lack of Cohesion Method* digunakan untuk mengukur derajat kemiripan *method* oleh variabel input data atau atribut dalam *class*. *Class* dengan metrik LCOM yang tinggi dapat dibagi menjadi dua tau lebih *subclass* sehingga dapat meningkatkan nilai *cohesi* dalam *class*. Rumus metrik LCOM terdapat pada Gambar 5. 9 Rumus LCOM. *p* adalah *method* yang tidak memiliki irisan atribut dengan *method* lainnya, *q* adalah *method* yang memiliki irisan atribut dengan *method* lainnya

$$\begin{array}{l} \text{if } p > q \text{ then LCOM} = p - q \\ \text{else LCOM} = 0 \end{array}$$

Gambar 5. 9 Rumus LCOM

Class *bookingCellModifier* pada *package* *appLayer* memiliki empat *method* yaitu *method* *bookingCellModifier*, *method* *modify*, *method* *canModify*, *method* *getValue*. Hanya *method* *bookingCellModifier* yang memiliki irisan atribut *tableviewer* dengan *method* *modify*. Sehingga $p=5$ dan $q=1$, $LCOM = 4$.



Gambar 5. 10 Source Code contoh Metrik LCOM (Class bookingCellModifier)

5.6.1. Nilai Metrik LCOM Package dataLayer

Tabel 5. 30 merupakan hasil perhitungan nilai Metrik LCOM pada package dataLayer.

Tabel 5. 30 Nilai Metrik LCOM Package dataLayer

No	Nama Class	Nilai LCOM
1	DB	0
2	HBCI	33
3	HBCIImporter	0
4	Messages	0
5	OBDXImporter	14
6	fileImporters	0
7	fileUtils	0
8	gnuPG	0
9	ProcessStreamReader	0
10	invalidImportFormatException	0
11	moneyplexImporter	0
12	prematureException	0
13	prepareDB	3
14	starmoneyImporter	0
Total		50

Keseluruhan nilai LCOM yang ada pada Tabel 5. 30 menunjukkan keseluruhan package dataLayer mempunyai

efficiency, maintainability dan replaceability yang bagus berdasarkan referensi Tabel 2. 11.

5.6.2. Nilai Metrik LCOM *Package appLayer*

Tabel 5. 31 merupakan hasil perhitungan nilai Metrik LCOM pada *package appLayer*.

Tabel 5. 31 Nilai Metrik LCOM *Package appLayer*

No	Nama Class	Nilai LCOM
1	AccountNotFoundException	0
2	ItemListener	0
3	IcontextProvider	0
4	Messages	0
5	PaymentMethodNotFoundException	0
6	RoleNotFoundException	0
7	account	213
8	accountException	0
9	accountsList	305
10	application	0
11	asset	112
12	assets	0
13	bookingCellModifier	4
14	bookingContentProvider	1
15	bookingLabelProvider	4
16	client	0
17	configs	0
18	contact	597
19	contacts	0
20	contextComposite	1
21	document	254
22	documents	6
23	elementNotFoundException	0
24	entry	1317
25	AccountNotFoundException	39
26	ItemListener	45
27	IcontextProvider	0
28	Messages	21

No	Nama Class	Nilai LCOM
29	PaymentMethodNotFoundException	36
30	RoleNotFoundException	3
31	account	28
32	accountException	0
33	entryNotInThisAccountException	0
34	item	204
35	itemTableList	2
36	itemTextList	0
37	items	2
38	moneyTransfer	0
39	pageActivator	0
40	placeholderManager	3
41	product	53
42	products	0
43	transactionContentProvider	0
44	typeNotFoundException	0
45	utils	0
Total		3310

Keseluruhan nilai LCOM yang ada pada Tabel 5. 31 berada di *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency, maintainability dan replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.6.3. Nilai Metrik LCOM Package appLayer.taxRelated

Tabel 5. 32 merupakan hasil perhitungan nilai Metrik LCOM pada *package* appLayer.taxRelated.

Tabel 5. 32 Nilai Metrik LCOM Package appLayer.taxRelated

No	Nama Class	Nilai LCOM
1	IRSooffice	0
2	IRSoffices	0
3	state	9
4	states	0
5	tax	212

No	Nama Class	Nilai LCOM
6	taxList	32
7	taxNotFoundException	0
Total		253

Keseluruhan nilai LCOM yang ada pada Tabel 5. 32 menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency, maintainability dan replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.6.4. Nilai Metrik LCOM *Package* appLayer.transactionRelated

Tabel 5. 33 merupakan hasil perhitungan nilai Metrik LCOM pada *package* appLayer.transactionRelated.

Tabel 5. 33 Nilai Metrik LCOM *Package* appLayer.transactionRelated

No	Nama Class	Nilai LCOM
1	appLayer.transactionRelated.cancelation	45
2	appLayer.transactionRelated.creditmemo	26
3	appLayer.transactionRelated.invoice	32
4	appLayer.transactionRelated.offer	36
5	appLayer.transactionRelated.personalDrawing	26
6	appLayer.transactionRelated.reminder	36
7	appLayer.transactionRelated.transaction	1017
8	appLayer.transactionRelated.transactionTypeNotFoundException	0
9	appLayer.transactionRelated.transactions	30
Total		1248

Keseluruhan nilai LCOM yang ada pada Tabel 5. 33 menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency, maintainability dan replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.6.5. Nilai Metrik LCOM *Package* GUILayer

Tabel 5. 34 merupakan hasil perhitungan nilai Metrik LCOM pada *package* GUILayer.

Tabel 5. 34 Nilai Metrik LCOM *Package* GUILayer

No	Nama Class	Nilai LCOM
1	CComboBoxCellEditor	10
2	HBCIcallbackDialog	1
3	HSQLLockCheck	1
4	MainWindow	34
5	Messages	0
6	OOoCheck	1
7	TimeCellEditor	16
8	VATannouncementWindow	36
9	aboutWindow	36
10	accountDefinitionWindow	30
11	ContentProvider	10
12	accountSelectionChangedListener	0
13	accountingEditWindow	28
14	assetWindow	30
15	ContentProvider	4
16	assetSelectionChangedListener	0
17	balanceWindow	30
18	browserWindow	26
19	communityWindow	28
20	configWindow	106
21	contactsWindow	35
22	ContentProvider	4
23	contactSelectionChangedListener	0
24	databaseConnectivityTest	1
25	databaseDriverNameTestCase	1
26	designerWindow	40
27	documentsWindow	28
28	documentCellModifier	6
29	documentLabelProvider	4
30	documentsList	0
31	entryDetailWindow	26
32	mainDirTestCase	1

No	Nama Class	Nilai LCOM
33	newAccountingWizard	0
34	newAccountingWizardAdd	0
35	entryList	0
36	newAccountingWizardImport	15
37	newManualAccountingWizard	0
38	newTransactionSelectItemDetails	0
39	newTransactionSelectTransactionDetails	14
40	entryList	0
41	transactionCellModifier	4
42	transactionLabelProvider	3
43	newTransactionWizard	6
44	mailAuthenticator	0
45	mailThread	0
46	newTransactionWizardPreview	0
47	newTransactionWizardPrintAndSave	3
48	numbersWindow	30
49	ContentProvider	4
50	transactionSelectionChangedListener	0
51	passwordDialog	0
52	productsWindow	37
53	ContentProvider	4
54	productSelectionChangedListener	0
55	reportWizard	3
56	reportWizardSelect	0
57	reportWizardShow	14
58	tAccount	2
59	tAccountLine	0
60	selectionAdapterWithParent	0
61	settingsWindow	34
62	standardPositiveTestResult	0
63	tableCheck	1
64	tableWindow	0
65	taxesWindow	30
66	ContentProvider	4
67	taxSelectionChangedListener	0
68	templateCheck	1
69	testCase	0
70	testResult	0

No	Nama Class	Nilai LCOM
71	testTest	1
72	timetrackerImportWindow	26
73	todoBalanceSelectionAdapter	0
74	todoWindow	45
75	transactionSelectionAdapter	0
76	viewerWindow	30
77	warningFocusAdapter	0
78		884

Nilai LCOM yang ada pada Tabel 5. 34 menunjukkan keseluruhan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

5.6.6. Nilai AVERAGE Metrik LCOM

Nilai *Average* Metrik Metrik LCOM didapatkan dengan menjumlah semua nilai Metrik LCOM dan dibagi dengan jumlah semua *class*. Perhitungan *Average* nilai Metrik LCOM terdapat pada Tabel 5. 35.

Tabel 5. 35 Nilai AVERAGE Metrik LCOM

Package	Jumlah Kelas	Nilai LCOM
GUILayer	77	884
appLayer	45	3310
appLayer.taxRelated	7	253
appLayer.transactionRelated	9	1248
dataLayer	14	50
Total	152	2413
Average LCOM	$= \frac{\sum RFC}{\sum class}$ $= 20,2$	

Hasil perhitungan pada Gambar 5. 7 menunjukkan bahwa Keseluruhan nilai Metrik LCOM pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* *dataLayer* mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

BAB VI

HASIL DAN PEMBAHASAN

Bab ini berisi hasil dari implementasi dan perbaikan yang perlu dilakukan pada *Software Accounting XYZ*.

6.1. Analisa Hasil Perhitungan Metrik Chidamber dan Kemerer

Perhitungan pada bab sebelumnya telah menghasilkan nilai metrik metrik chidamber dan kemerer pada *Software Accounting XYZ*. Metrik Chidamber dan Kemerer telah dipetakan terhadap kualitas *software* menurut ISO 9126-1 namun, hanya kualitas *efficiency* dan *maintainability* serta sub kriteria *understandability* dan *reusability/replaceability*.

Software Accounting XYZ dikatakan mempunyai tingkat *efficiency* yang bagus ketika jumlah sumberdaya yang diproses (memori) dan kode program yang diperlukan oleh program untuk melaksanakan fungsi tertentu semakin sedikit.

Software Accounting XYZ dikatakan mempunyai tingkat *understandability* yang bagus ketika semakin mudah untuk memodifikasi *software*. Modifikasi dapat mencakup koreksi, perbaikan atau adaptasi dari perangkat lunak untuk keperluan perubahan kebutuhan.

Software Accounting XYZ dikatakan mempunyai tingkat *understandability* yang bagus ketika kode program dapat dengan mudah dipelajari *input* dan *output*-nya atau mempunyai tingkat kompleksitas yang rendah.

Software Accounting XYZ dikatakan mempunyai tingkat *replaceability* yang bagus ketika program/bagian dari program dapat dipakai ulang dalam aplikasi lainnya, pada lingkup fungsi yang dilakukan oleh program tersebut.

Hasil perhitungan WMC menghasilkan nilai 10,28; hal ini menunjukkan bahwa keseluruhan nilai Metrik WMC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Hasil perhitungan DIT menghasilkan nilai 1,1; hal ini menunjukkan bahwa nilai Metrik DIT pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Hasil perhitungan NOC menghasilkan nilai 0,12; hal ini menunjukkan bahwa keseluruhan nilai Metrik NOC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Hasil perhitungan CBO menghasilkan nilai 7,52; hal ini menunjukkan bahwa keseluruhan nilai Metrik CBO pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *efficiency* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Hasil perhitungan RFC menghasilkan nilai 11,58; hal ini menunjukkan bahwa keseluruhan nilai Metrik RFC pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* dataLayer mempunyai *understanbility*

dan *maintainability* yang bagus berdasarkan referensi Tabel 2. 11.

Hasil perhitungan LCOM menghasilkan nilai 20,2; hal ini menunjukkan bahwa Keseluruhan nilai Metrik LCOM pada *Software Accounting XYZ* terletak pada *range* hijau atau *good* berdasarkan referensi dari Tabel 2. 4. Hal ini menunjukkan *package* *dataLayer* mempunyai *efficiency*, *understanbility*, *maintainability* dan *replaceability* yang bagus berdasarkan referensi Tabel 2. 11.

Selain nilai keseluruhan Metrik Chidamber dan Kemerer telah didapatkan nilai *class* yang memerlukan perbaikan. Terdapat dua *class* dengan warna merah yang menunjukkan kategori *bad*, sehingga harus dirubah. Terdapat 22 *class* yang menunjukkan kategori *medium*, sehingga diperlukan perbaikan hingga mencapai status *good*.

Tabel 6. 1 Class yang Memerlukan Perbaikan

Nama Package	Class	Metrik	Nilai	Kategori
appLayer	accountsList	WMC	69	<i>Medium</i>
appLayer	configs	WMC	222	<i>Bad</i>
appLayer	appLayer.entry	WMC	132	<i>Bad</i>
dataLayer	dataLayer.D B	CBO	40	<i>Medium</i>
appLayer	appLayer.M essages	CBO	18	<i>Medium</i>
appLayer	account	CBO	15	<i>Medium</i>
appLayer	accountsList	CBO	19	<i>Medium</i>
appLayer	client	CBO	56	<i>Medium</i>
appLayer	configs	CBO	41	<i>Medium</i>
appLayer	entry	CBO	24	<i>Medium</i>
appLayer.tax Related	taxList	CBO	16	<i>Medium</i>
appLayer.tra nsactionRela ted	transaction	CBO	26	<i>Medium</i>

appLayer.transactionRelated	transactions	CBO	18	<i>Medium</i>
GUILayer	Messages	CBO	40	<i>Medium</i>
GUILayer	balanceWindow	CBO	15	<i>Medium</i>
GUILayer	todoWindow	CBO	14	<i>Medium</i>
appLayer	entry	RFC	112	<i>Medium</i>

6.2. Membuat *Class Diagram* Baru

Pada bagian ini dijelaskan bagaimana membuat *class diagram* baru sehingga dapat memperoleh nilai Metrik Chidamber dan Kemerer yang lebih baik. Pembuatan *class diagram* berdasarkan pedoman *design pattern* bagian *refactoring*.

Design pattern yang dipilih dalam membuat *class diagram* baru adalah *strategy pattern*. Dalam melakukan *strategy pattern* digunakan *tools refactoring*. *Refactoring* dan *Design Pattern* memiliki hubungan yang erat. *Design Pattern* adalah pedoman, sedangkan *Refactoring* adalah cara untuk memperbaiki *code*.

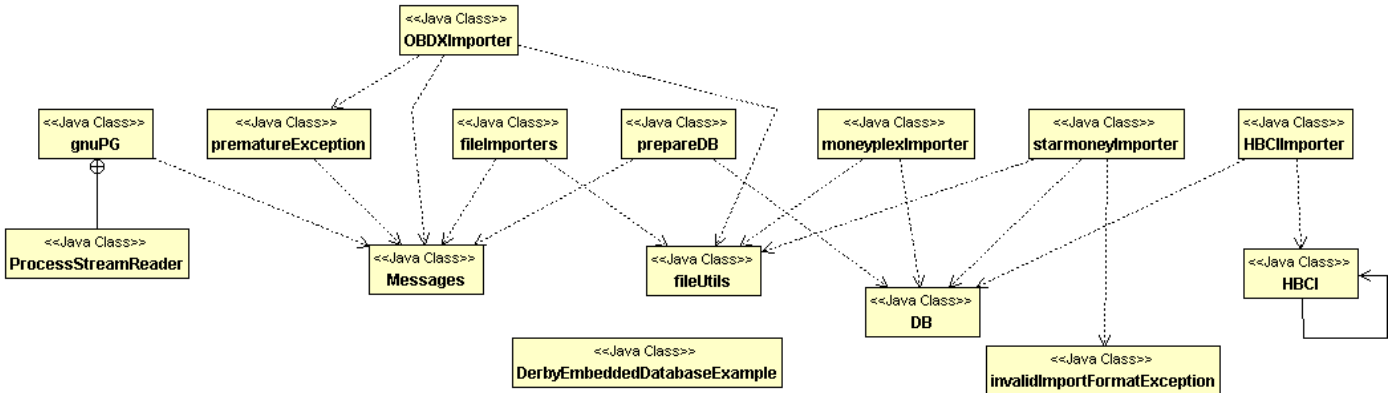
Pada pembuatan *class diagram* baru ini terdapat beberapa *refactoring* yang digunakan.

Tabel 6. 2 Refactoring yang Digunakan

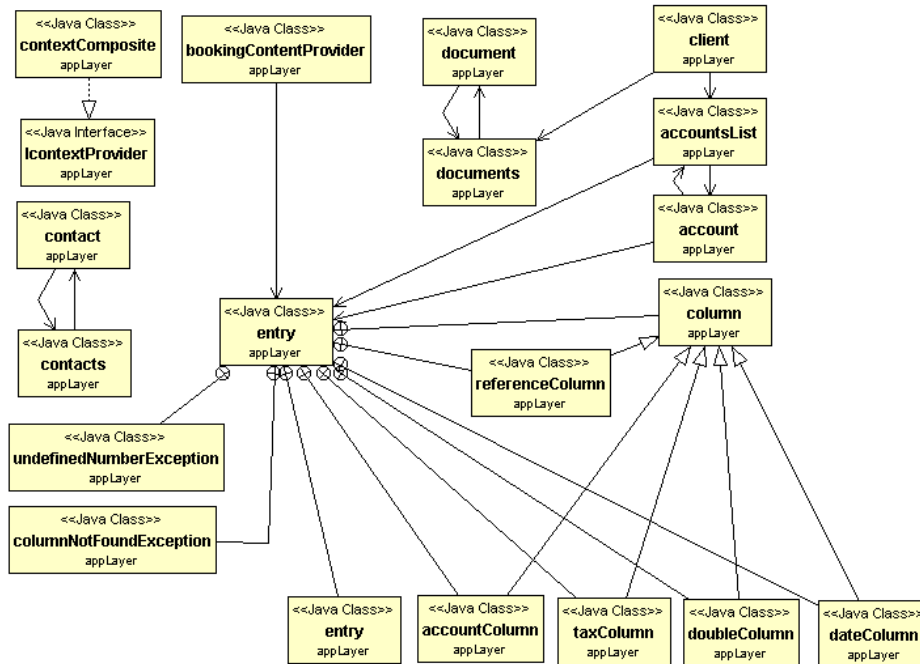
No	Jenis <i>Bad Smell</i>	Diskripsi	Solusi
1	<i>Duplicated Code</i>	Terdapat <i>code</i> yang sama dan berulang di lokasi yang berbeda.	Menghapus atau diimplementasikan sebagai satu <i>method</i> atau <i>function</i> tunggal yang akan dipanggil saat diperlukan
2	<i>Long Method</i>	<i>Method</i> yang ditulis terlalu panjang	Diubah dan dikelompokkan menjadi <i>method</i>

			yang lebih pendek.
3	<i>Large Class</i>	<i>Class</i> yang besar (terlalu banyak <i>method</i>) sehingga sulit di dibaca, di mengerti, dan di <i>maintenance</i> .	Dikelompokkan menjadi beberapa <i>class</i> atau dijadikan <i>partial class</i>
4	<i>Long Parameter List</i>	Parameter sebuah <i>method</i> terlalu banyak sehingga sulit di mengerti	menggunakan tipe objek sebagai parameter
5	<i>Feature Envy</i>	Sebuah <i>method</i> membutuhkan banyak informasi dari <i>class</i> lain daripada <i>class</i> nya sendiri	Mengelompokkan <i>method</i> yang sesuai
6	<i>Data Clumping</i>	Kelompok data yang sama (<i>field</i> dalam <i>class</i> , parameter dalam <i>method</i>) terulang kembali di beberapa tempat dalam sebuah program.	Mengganti dengan enkapsulasi semua data dalam satu objek
7	<i>Switch Statement</i>	Switch sebagai pengganti “if” biasanya bertebaran dimana-mana	Menggunakan <i>Polymorphism</i> untuk menggantikannya
8	<i>Lazy Class</i>	<i>Class</i> yang tidak mempunyai fungsi atau <i>class</i> yang jarang digunakan.	Menggabungkan <i>class-class</i> yang hanya memiliki sedikit tanggung jawab, atau menghilangkan <i>class</i> yang <i>method-method</i> nya tidak dipanggil sama sekali

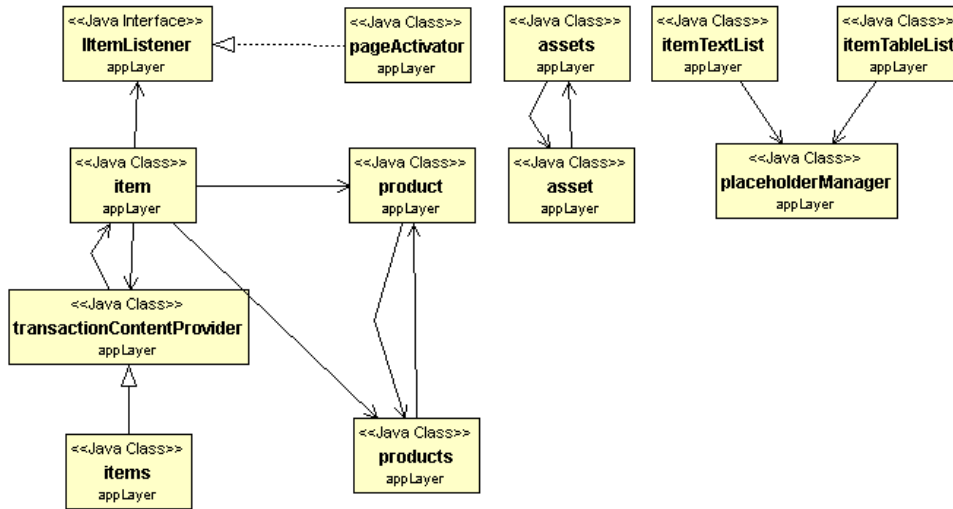
Class diagram baru ada pada Gambar 6. 1 sampai Gambar 6. 7.



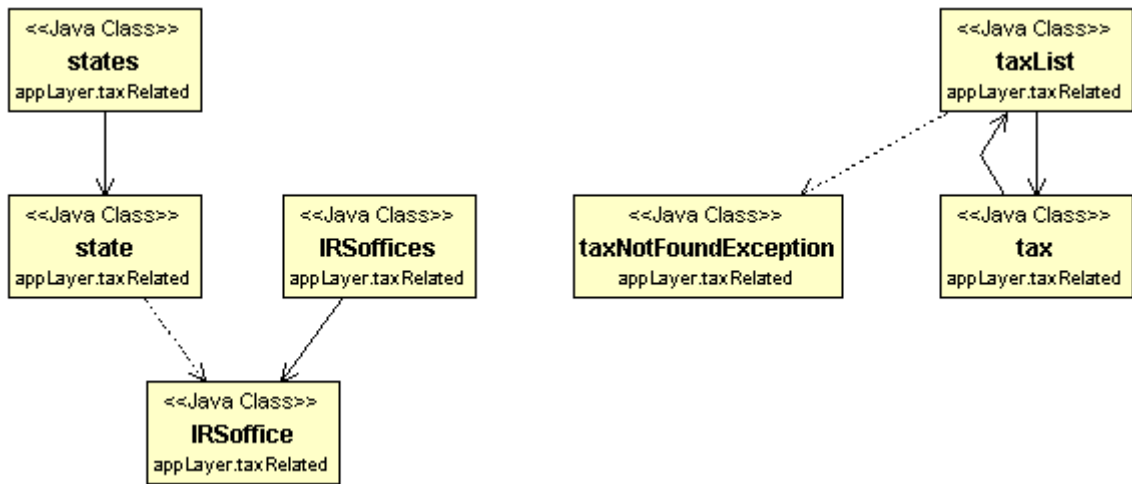
Gambar 6. 1 Class Diagram baru Package dataLayer



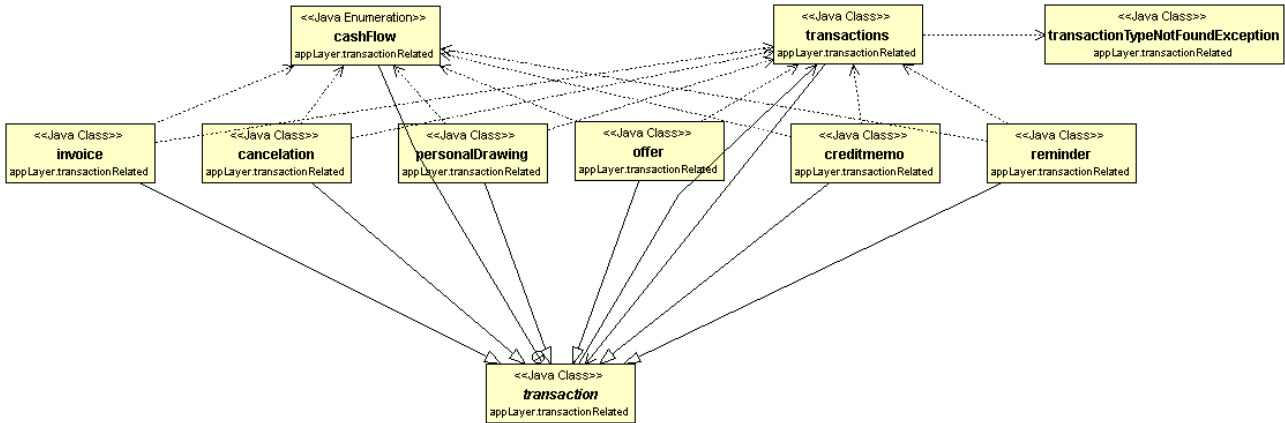
Gambar 6. 2 Class Diagram baru Package appLayer part 1



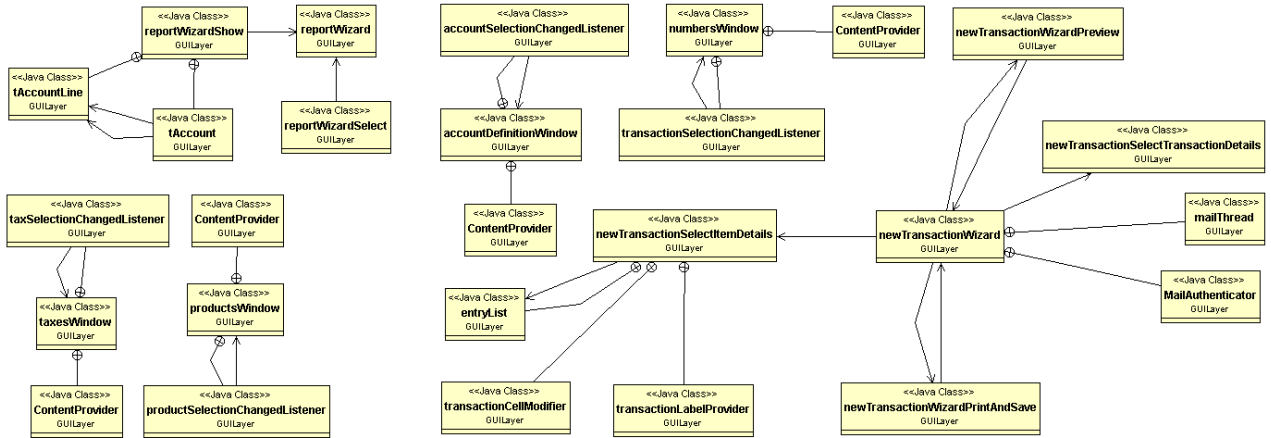
Gambar 6. 3 *Class Diagram* baru Package appLayer part 2



Gambar 6. 4 *Class Diagram* baru Package `appLayer.taxRelated`



Gambar 6. 5 Class Diagram baru Package appLayer.transactionRelated



Gambar 6.7 Class Diagram baru Package GUILayer part 2

6.3. Perhitungan Metrik Chidamber dan Kemerer pada Class Diagram Baru

Pada bagian ini dilakukan pengukuran Metrik Chidamber dan Kemerer yang baru. Nilai Metrik Chidamber dan Kemerer pada semua *class* telah berubah. Pada Tabel 6. 3 terdapat nilai Metrik Chidamber dan Kemerer yang baru pada *class* yang perlu diperbaiki.

Tabel 6. 3 Pengukuran Matrik Chidamber dan Kemerer terhadap *Class* yang Diperbaiki

Nama Package	Class	Metrik	Nilai	Kategori
appLayer	accountsList	WMC	53	<i>Medium</i>
appLayer	configs	WMC	75	<i>Bad</i>
appLayer	appLayer.entry	WMC	1	<i>Bad</i>
dataLayer	dataLayer.DB	CBO	37	<i>Medium</i>
appLayer	appLayer.Messages	CBO	14	<i>Medium</i>
appLayer	account	CBO	13	<i>Medium</i>
appLayer	accountsList	CBO	14	<i>Medium</i>
appLayer	client	CBO	49	<i>Medium</i>
appLayer	configs	CBO	35	<i>Medium</i>
appLayer	entry	CBO	0	<i>Medium</i>
appLayer.tax Related	taxList	CBO	9	<i>Medium</i>
appLayer.transaction Related	transaction	CBO	24	<i>Medium</i>
appLayer.transaction Related	transactions	CBO	25	<i>Medium</i>
GUILayer	Messages	CBO	14	<i>Medium</i>
GUILayer	balanceWindow	CBO	8	<i>Medium</i>
GUILayer	todoWindow	CBO	12	<i>Medium</i>
appLayer	entry	RFC	1	<i>Medium</i>

Menghitung Metrik Chidamber dan Kemerer pada *class diagram* yang baru menggunakan cara yang sama dengan tahapan sebelumnya. Pada tahapan perhitungan *class diagram* baru ini di dapatkan hasil keseluruhan nilai Metrik Chidamber dan Kemerer.

Tabel 6. 4 Nilai Metrik Chidamber dan Kemerer Baru

No.	Metrik	Nilai	Keterangan
1	WMC	8,83	Turun
2	DIT	1,08	Turun
3	NOC	0,09	Turun
4	CBO	6,27	Turun
5	RFC	10,46	Turun
6	LCOM	32,35	Naik

Pada Tabel 6. 4 telah ada nilai Metrik Chidamber dan Kemerer yang baru. Metrik WMC, DIT, NOC, CBO, dan RFC menurun, hal ini menunjukkan kualitas perangkat lunak yang semakin baik karena nilai metrik tersebut berbanding terbalik dengan kriteria kualitas *software* pada ISO/IEC 9126-1. Nilai Matrik LCOM lebih tinggi dibandingkan nilai yang sebelumnya, hal ini menunjukkan kualitas perangkat lunak yang semakin baik karena nilai metrik tersebut berbanding lurus dengan kriteria kualitas *software* pada ISO/IEC 9126-1.

BAB VII PENUTUP

Bab ini akan menjelaskan kesimpulan dari penulisan tugas akhir ini, beserta saran yang dapat bermanfaat untuk perbaikan di penulisan selanjutnya.

7.1. Kesimpulan

Berdasarkan pada pengerjaan tugas akhir ini, maka didapatkan kesimpulan dari tugas akhir ini yaitu :

1. *Class* diagram baru yang diperbaiki berdasarkan *refactoring* tetap mempunyai 152 *class* namun, terdapat perubahan *method* sehingga tidak ada *class* yang *lazy* atau *class* yang kelebihan *method*.
2. Nilai Metrik Chidamber dan Kemerer lebih baik pada *class diagram* yang baru,
 - Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik WMC yang semula 10,28 menjadi 8,83; hal ini menunjukkan kenaikan atau perbaikan *understandability*, *maintainability*, dan *replaceability* sebesar 14,11% berdasarkan jumlah *method* atau kompleksitas setiap *class*.
 - Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik DIT yang semula 1,1 menjadi 1,08; hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *understandability*, *maintainability*, dan *replaceability* sebesar 1,82% berdasarkan kedalaman sebuah *class*.
 - Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik NOC yang semula 0,12 menjadi 0,09; hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *maintainability*, dan

replaceability sebesar 25% berdasarkan jumlah *subclass (child class)* setiap *class*.

- Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik CBO yang semula 7,52 menjadi 6,27; hal ini menunjukkan kenaikan atau perbaikan *efficiency* dan *replaceability* sebesar 16,62% berdasarkan jumlah *Coupling* setiap *class*.
- Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik RFC yang semula 11,58 menjadi 10,46; hal ini menunjukkan kenaikan atau perbaikan *understandability*, *maintainability*, dan *replaceability* sebesar 9,67% berdasarkan jumlah *Message* setiap *class*.
- Berdasarkan perhitungan Metrik Chidamber dan Kemerer, diperoleh nilai metrik LCOM yang semula 20,2 menjadi 32,35, hal ini menunjukkan kenaikan atau perbaikan *efficiency*, *maintainability*, dan *replaceability* sebesar 60,15% berdasarkan jumlah *cohesion* setiap *class*.

2.2. Saran

Tugas akhir ini tidak menyertakan implementasi dari rekomendasi yang diusulkan. Tugas akhir ini dapat menjadi referensi untuk mengukur dan membandingkan kualitas perangkat lunak yang mempunyai fungsi sama. Metrik yang digunakan dalam penelitian ini tidak mengukur semua karakteristik kualitas pada ISO/IEC 9126-1, perlu adanya perpaduan Metrik Chidamber dan Kemerer dengan metrik lainnya agar dapat mengukur semua karakteristik ISO/IEC 9126-1.

DAFTAR PUSTAKA

- [1] Roni Yunis, Arwin Halim, "Studi Empiris Hubungan Metrik Kohesi Dengan Kecenderungan Kesalahan Pada Aplikasi Berorientasi Objek," *ISSN VOL 15, NO 1*, vol. 15, no. ISSN. 1412-0100, p. 2, 2014.
- [2] N. Schneidewind, *Body of Knowledge for Software Quality Measuremen*, 2002.
- [3] T. K. a. N. Y.Liu, *Evolutionary Optimization of Software Quality Modeling with Multiple*, 2010.
- [4] K. K. a. Y. Gu  h  neuc, *On Issues with Software Quality Models*, 2008.
- [5] L. R. Ana Paula L'udtke Ferreira, "A Graph-based Semantics For Object-oriented Programming Constructs," *Science Direct*, p. 1, 2005.
- [6] R. S. Wahono, "Teknik Pengukuran Kualitas Perangkat Lunak," *SF*, p. 1, 2006.
- [7] M. E. Khan, "Different Forms of Software Testing Techniques for Finding Error," 2010.
- [8] J. P. S. K. R. Yeresime Suresh, "Effectiveness of software metrics for object-oriented system," *ScienceDirect*, p. 1, 2012.
- [9] S. C. & C. Kemerer, *MOOSE Metrics for Object Oriented Software Engineering*, 1993.
- [10] F. B. e. Abreu, *Design Metric for Object Oriented Software Systems*, 2009.
- [11] L. H. Rosenberg & L. E. Hyatt, *Software Quality Metrics for Object Oriented Enviroments*, Unisys Technology Conference, 2003.
- [12] J. L. & M. Hagglund, *Maintainability Metrics for Object Oriented Systems*, 2004.
- [13] M. B. & A. Deursen, *Predicting Class Testability using Object Oriented Metrics*, 2004.
- [14] S. C. & C. Kemerer, *A Metrics Suite for Object Oriented Design*, MIT Sloan Scholl of Managemnet, 2009.

- [15] A. R. M, "Konsep Pemograman Berorientasi Objek (OOP)," 2008.
- [16] Rosenberg, Linda H., "Metrics for Object Oriented Environments," *EFAITP/AIE Third Annual Software Metrics Conference*, 1999.
- [17] azkohaq, "Object Oriented Programming," 2013.
- [18] Julio, "Pemograman Berorientasi Objek," 2011.
- [19] Jain, Er. V.K., *The Complete Guide to java*, First Edition, 2001.
- [20] Patrick Naughton & Herbert Schildt, "java : The complete reference," McGraw-Hill Professional, UK, 2008.
- [21] Hariyanto, Bambang, Ir., MT., *Esensi-esensi Bahasa Pemrograman*, Bandung: Informatika, 2005.
- [22] W. R. D. Septian, "Analisis Perbandingan Framework PHP Berdasarkan MOOSE CK dan Properti Kualitas Disain Menggunakan Metode AHP," 2010.
- [23] S. R. Chidamber & C. F. Kemerer, "Towards a Metrics Suite for Object Oriented Design," *OOPSLA Conference*, pp. 197-221, 1991.
- [24] Houari A. Sahraoui, Robert Godin, Thierry Miceli, "Can Metrics Help Bridging the Gap Between the Improvement of OO Design Quality and Its Automation," [Online]. Available: <http://www.iro.umontreal.ca/~sahraouh/papers/ICSM00.pdf>.
- [25] Victor Basili, Lionel Briand and Walcelio Melo., "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Transactions on Software Engineering*, vol. 22, 1996.
- [26] Shyam R. Chidamber, Chris F. Kemerer. A, "Metrics suite for Object Oriented design," *M.I.T. Sloan School of Management*, pp. 53-315, 1993.
- [27] P. S. Stoecklin, "One hour presentation to inform you of new techniques and practices in software development," Florida State University – Computer Science, Panama, 2003.
- [28] Daniela Glasberg, Khaled El Emam, Walcelio Melo, Nazim Madhavji, "Validating Object-Oriented Design Metrics on a CommercialJava Application," *iit-iti.nrc*, 2000.
- [29] Khaled El Emam, Walcelio Melo, Javam, C. Machado., "The prediction of faulty classes using object-oriented design metrics," *The Journal of Systems and Software*, vol. 56, p. 2001,

2001.

- [30] Chidamber, S., Darcy, D., Kemerer, C., "Managerial use of Metrics for Object Oriented Software," *IEEE Transaction on Software Engineering*, Vols. 24, no. 8., no. an Exploratory Analysis, pp. 629-639, 1998.
- [31] Arti Chhikara and R.S.Chhillar, "Analyzing the Complexity of Java Programs using Object Oriented Software Metrics," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 1, No 3, p. 1, 2012.
- [32] Ho-Won Jung and Seung-Gweon Kim, Chang-Shin Chung, "Measuring Software Product Quality: A Survey of ISO/IEC 9126," *IEEE SOFTWARE*, 2004.
- [33] ISO/IEC, "Information technology — Software," ISO/IEC 2000, Geneva, 2000.
- [34] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides,, "Design Patterns: Elements of Reusable Object-Oriented Software," 1995.
- [35] S. M. Imam Cholissodin, "Design Pattrens Introduction," Universitas Brawijaya, Malang, 2006.
- [36] Kusnawi, "Penerapan Design Pattrens untuk perancangan berbasiskan objek oriented".
- [37] Ali Hanzala Khan, Ivan Porres, "Consistency of UML class, object and statechart diagrams," *Journal of Visual and Computing SciencDirect*, no. 26(2015) 42-65, 2014.
- [38] J. Schmuller, Sams Teach Yourself UML in 24 Hours, Complete Starter Kit (3rd Edition), 2004.
- [39] P. Coad, and E. Yourdon, Object-Oriented Design, Englewood Cliffs, NJ, ISBN: 0-13-630070-7: Yourdon Press, 1991.
- [40] S. Chidamber S, and C. Kemerer, "Towards a Metrics Suite for Object Oriented Design," *Object Oriented Programming Systems, Languages and Applications (OOPSLA)*, vol. 10, pp. 197-211, 1991.
- [41] Grady, R.B., Practical Software Metrics for Project Management and Process Improvement, Vols. ISBN: 0-13-720384-5, Englewood Cliffs, NJ: Prentice Hall ISBN: 0-13-720384-5, 1992.
- [42] Rosenberg, Linda H, Lawrence E. Hyatt, Software Quality

Metrics for Object Oriented Environments, 2003.

- [43] E. H. & P. Mursanto, "Peningkatan Software Aplikasi Berdasarkan Properti Kualitas Disain dan Metrics for Object Oriented Software Menggunakan Analytic Hierarchy," Universitas Indonesia, Jakarta, 2010.
- [44] Thomas L. Saaty, *Theory and Applications of the Analytic Network Process: Decision Making with Benefits, Opportunities, Costs, and Risks*, RWS Publications, 2005.
- [45] S. Brinkkemper, "Method Engineering: Engineering of Information Systems Development Methods and Tools," *Information and Software Technology*, vol. 38, pp. 275-280, 1996.
- [46] M. Dickinson, A. Thornton and S. Graves, "Technology Portfolio Management: Optimizing Interdependent Projects Over Multiple Time Periods," *IEEE Transactions on Engineering Management*, vol. 48, no. 4, pp. 518-527, 2001.
- [47] I. Stamelos and L. Angelis, "Managing Uncertainty in Project Portfolio Cost Estimation," *Information & Software Technology*, vol. 43, no. 13, pp. 759-768, 2001.
- [48] M. H. M. L. Q. i. W. B. B. Ye Yang, "Phase Distribution of Software Development Effort," USA, 2008.
- [49] D. Braha and O. Maimon, "The Measurement of a Design Structural and Functional Complexity," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, p. 527, 1998.

LAMPIRAN A – Method dan Atribut

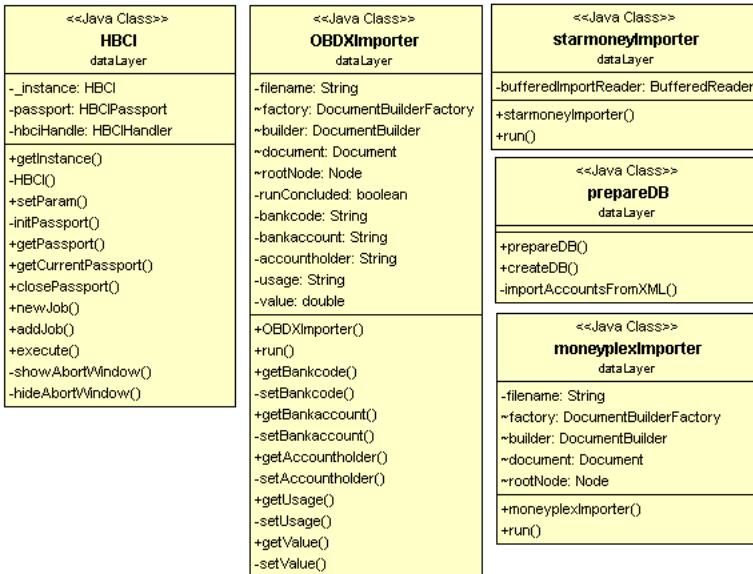
Lampiran ini berisi *method* dan atribut dari masing-masing *class*. *Method* dan atribut *package* dataLayer part 1 terdapat pada Gambar B. 1 *Method* dan Atribut *Package* dataLayer part 1.

<pre><<Java Class>> DB dataLayer -con: Connection -DBVersionString: String -DBVersionDouble: double +DB() +getConnection() +setConnection() +getIdentifierQuote() +getDatabaseOffset() +getLastInsertID() +update() +updateRaw() +insert() +insertRaw() +deleteLogical() +deletePhysical() +getDBVersionString() +getDBVersionDouble()</pre>	<pre><<Java Class>> fileImporters dataLayer +fileImporters() +importKTimeTracker() +importTaskCoach() <<Java Class>> fileUtils dataLayer +CSV_SEMICOLON_PATTERN: String -csvREsemicolon: Pattern +CSV_COMMA_PATTERN: String -csvREcomma: Pattern +fileUtils() +readFileAsString() +parseCSVLineSemicolon() +parseCSVLineComma()</pre>	<pre><<Java Class>> gnuPG dataLayer -kGnuPGCommand: String -tmpFile: File -gpg_exitCode: int -gpg_result: String -gpg_err: String +sign() +clearSign() +signAndEncrypt() +encrypt() +decrypt() +getResult() +getErrorString() +getExitCode() -runGnuPG() -createTempFile() +gnuPG()</pre>
<pre><<Java Class>> ProcessStreamReader dataLayer ~stream: StringBuffer ~in: InputStreamReader ~BUFFER_SIZE: int ~ProcessStreamReader() +run() ~getString()</pre>	<pre><<Java Class>> invalidImportFormatException dataLayer -serialVersionUID: long +invalidImportFormatException()</pre>	<pre><<Java Class>> Messages dataLayer -BUNDLE_NAME: String -RESOURCE_BUNDLE: ResourceBundle -Messages() +getString()</pre>
	<pre><<Java Class>> HBCLImporter dataLayer -start: Date -end: Date +HBCLImporter() +run()</pre>	<pre><<Java Class>> prematureException dataLayer -serialVersionUID: long +prematureException()</pre>

Gambar B. 1 *Method* dan Atribut *Package* dataLayer part 1

A-2

Method dan atribut *package* dataLayer part 2 terdapat pada Gambar B. 2 Method dan Atribut Package dataLayer part 2.



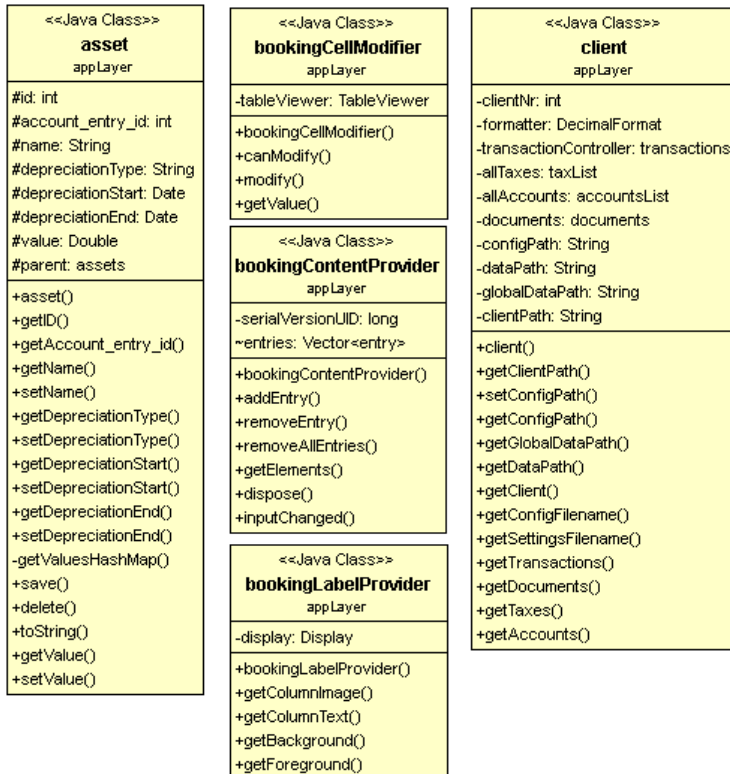
Gambar B. 2 Method dan Atribut Package dataLayer part 2

Method dan atribut package appLayer part 1 terdapat pada Gambar B. 3 Method dan Atribut Package appLayer part 1.

<pre><<Java Class>> account appLayer</pre>	<pre><<Java Class>> typeNotFoundException appLayer</pre> <pre>-serialVersionUID: long +typeNotFoundException()</pre>	<pre><<Java Class>> accountsList appLayer</pre>
<pre>-serialVersionUID: long ~id: int ~code: String ~description: String ~indexInStringArray: int ~subAccounts: int ~type: int ~refersTo: Integer ~entries: Vector<entry> ~openingBalance: Double -parent: accountsList ~accountTypes: String[] ~subAccountTypes: String[]</pre>	<pre><<Java Class>> accountException appLayer</pre> <pre>-serialVersionUID: long +accountException()</pre>	<pre>-serialVersionUID: long ~accountsStringArray: String[] ~accounts: Vector<account> ~journal: Vector<entry> ~periodStart: Date ~periodEnd: Date ~periodDefined: boolean ~addEmptyAccount: boolean ~accountsPredefined: int</pre>
<pre>+account() +isEmpty() +account() +removeAllEntries() +update() +save() +delete() +getTypeIDForString() +setType() +getType() +getTypeString() +setType() +setIndexInStringArray() +getAsString() +getDescription() +getID() +getIndexInStringArray() +addEntry() +getEntries() +hasEntries() +getCoder() +toString() +getTypes() +getSubAccountTypesCoder() +setReferTo() +getSubAccountsSQLString() +getSubAccountsTypeIDForString() +getSubAccounts() +getOpeningBalance() +getBalance() +setOpeningBalance()</pre>	<pre><<Java Class>> AccountNotFoundException appLayer</pre> <pre>-serialVersionUID: long +AccountNotFoundException()</pre>	<pre>+accountsList() +getAccountsFromDatabase() +addPleaseSelectAccount() +addEmptyAccount() ~fillFromDB() +signalChange() +getNumAccounts() +setPeriod() +isEmpty() +getEntriesFromDatabase() ~updateWithAccountTotal() +getCompletePeriodStart() +getCompletePeriodEnd() +getYearsCovered() +getDefaultAccount() +getBankAccount() +getReceivablesAccount() +getPersonalDrawAccount() +getLiabilitiesAccount() +getRevenuesAccount() +getStandardVATCreditAccount() +getStandardVATDebitAccount() +getStringArray() +getAccountForListID() +getAccountForID() +getAccountForCode() +getAccounts() +getJournal() +getImportEntries() +getEntryForImportID()</pre>
	<pre><<Java Class>> application appLayer</pre> <pre>-versionString: String ~versionDouble: double ~OBDXversionString: String ~OBDXversionDouble: double +versionHistory: String</pre> <pre>+application() +getVersionString() +getVersionDouble() +getOBDXVersionString() +getOBDXVersionDouble() +getAppName()</pre>	
	<pre><<Java Class>> assets appLayer</pre> <pre>~assets: Vector<asset> ~emptyAsset: boolean</pre> <pre>+assets() +getAssetsFromDB() +addEmptyAsset() +getAssets() +signalChange()</pre>	

Gambar B. 3 Method dan Atribut Package appLayer part 1

Method dan atribut *package* appLayer part 2 terdapat pada Gambar B. 4 *Method* dan Atribut *Package* appLayer part 2.



Gambar B. 4 *Method* dan Atribut *Package* appLayer part 2

Method dan atribut *package* appLayer part 3 terdapat pada Gambar B. 5 *Method* dan Atribut *Package* appLayer part 3.

<<Java Class>> configs appLayer	
<pre> ~factory: DocumentBuilderFactory ~builder: DocumentBuilder ~configurationDocument: Document ~settingsDocument: Document -printerName: String -OoOoPathName: String -WinstonPathName: String -GPGPathName: String -databaseName: String -databaseDriverName: String -databaseDriverFileName: String -databasePassword: String -databaseServer: String -databaseUser: String -databaseType: String -printPoweredBy: Boolean -startWithCommunity: Boolean -useExternalDB: Boolean -organisationName: String -taxID: String -stateListIDX: Integer -taxOfficeListIDX: Integer -VATPeriod: String -officeApplication: IOfficeApplication -officeApplicationPath: String -senderEmail: String -SMTPServer: String -SMTPUsername: String -SMTPPassword: String -useSMTPAuth: Boolean -defaultFont: Font -accountCode: String -bankName: String -holderName: String -bankCode: String -ctAPI: String -useCardReaderPINpad: Boolean </pre>	<pre> +configs() +getDefaultFont() +readConfiguration() +readSettings() -readPrinterName() -readChipcardSettings() -readClientSettings() -readDatabaseCredentials() -readMiscSettings() +getPrinterName() +getOoOoPath() +getGPGPath() +getWinstonPath() +getDatabaseName() +getDatabaseDriverName() +getDatabaseDriverFileName() +getDatabasePassword() +getDatabaseServer() +getDatabaseUser() +getDatabaseType() +getOrganisationName() +getStateListIDX() +getTaxOfficeListIDX() +getTaxID() +getVATPeriod() +getSenderEmail() +getSMTPServer() +getSMTPUsername() +getSMTPPassword() +getAccountCode() +getBankName() +getHolderName() +getBankCode() +getBankCode() +shallUseSMTPAuth() +getctAPI() </pre>
	<pre> +shallUseCardReaderPINpad() +getOfficeApplication() +disposeOfficeApplication() +shallPrintPoweredBy() +startWithCommunity() +shallUseExternalDB() +setNodeAttribute() +setPrinterName() +setDatabaseName() +setDatabaseDriverName() +setDatabaseDriverFileName() +setDatabasePassword() +setDatabaseServer() +setDatabaseUser() +setDatabaseType() +setOoOoPath() +setWinstonPath() +setGPGPath() +setPrintPoweredBy() +setStartWithCommunity() +setUseExternalDB() +setOrganizationName() +setTaxID() +setStateListIDX() +setTaxOfficeListIDX() +setVATPeriod() +setEmailSender() +setSMTPServer() +setSMTPUsername() +setSMTPPassword() +setUseSMTPAuth() +setctAPI() +setUseCardReaderPINpad() +setAccountCode() +setBankName() +setHolderName() +setBankCode() +writeConfiguration() +writeSettings() +getSampleValue() </pre>

Gambar B. 5 Method dan Atribut Package appLayer part 3

Method dan atribut *package* appLayer part 4 terdapat pada Gambar B. 6 *Method* dan Atribut *Package* appLayer part 4.

<pre> <<Java Class>> contact appLayer ~name: String ~co: String ~street: String ~zip: String ~location: String ~country: String ~email: String ~phone: String ~fax: String ~VATID: String ~bankaccount: String ~bankcode: String ~accountholder: String ~id: int ~role: int ~paymentmethod: int #parent: contacts </pre>	<pre> <<Java Class>> contacts appLayer #emptyContact: boolean #contactsList: Vector<contact> ~roles: String[] ~paymentMethods: String[] +contacts() +getRoles() +getRoleIDForString() +getPaymentMethods() +getPaymentIDForString() +getContactsFromDB() +addEmptyContact() +signalChange() +getContacts() </pre>	<pre> <<Java Class>> document appLayer -importFilename: String -parent: documents -id: int -entry_id: int -date_entry: Date -date_document: Date -number: String -md5: String -subject: String -value: String -isLinked: boolean +document() +getSubject() +setSubject() +getValue() +setValue() +setImportFilename() +getImportFilename() +save() +setID() +getID() +setIsLinked() +isLinked() +setDateEntry() +getDateEntry() +setDateDocument() +getDateDocument() +setNumber() +getNumber() +setSourceMD5() +getSourceMD5() +setEntryID() +getEntryID() +getValueForColumnName() +getChangeValueForColumnName() +setChangeValueForColumnName() </pre>
<pre> +contact() +contact() ~contact() +getNumber() +bindToID() +getName() +getStreet() +getZIP() +getLocation() +getCountry() +getCO() +getEmail() +getPhone() +getFax() +getVATID() +getID() +getRole() +setName() +setStreet() +setZIP() +setLocation() +setCountry() +setCO() +setEmail() +setPhone() +setFax() +setRole() +setVATID() +toString() +delete() +save() +getBankAccount() +setBankaccount() +getBankCode() +setBankcode() +getAccountholder() +setAccountholder() +getPaymentmethod() +setPaymentmethod() </pre>	<pre> <<Java Class>> contextComposite appLayer -currentHelpURL: String +contextComposite() +getCurrentHelpURL() +setCurrentHelpURL() </pre>	<pre> +document() +getSubject() +setSubject() +getValue() +setValue() +setImportFilename() +getImportFilename() +save() +setID() +getID() +setIsLinked() +isLinked() +setDateEntry() +getDateEntry() +setDateDocument() +getDateDocument() +setNumber() +getNumber() +setSourceMD5() +getSourceMD5() +setEntryID() +getEntryID() +getValueForColumnName() +getChangeValueForColumnName() +setChangeValueForColumnName() </pre>
<pre> +contact() +contact() ~contact() +getNumber() +bindToID() +getName() +getStreet() +getZIP() +getLocation() +getCountry() +getCO() +getEmail() +getPhone() +getFax() +getVATID() +getID() +getRole() +setName() +setStreet() +setZIP() +setLocation() +setCountry() +setCO() +setEmail() +setPhone() +setFax() +setRole() +setVATID() +toString() +delete() +save() +getBankAccount() +setBankaccount() +getBankCode() +setBankcode() +getAccountholder() +setAccountholder() +getPaymentmethod() +setPaymentmethod() </pre>	<pre> <<Java Class>> documents appLayer -allDocuments: Vector<document> +documents() +getNewDocument() +getAsArray() +getDocumentsFromDatabase() +getElements() +dispose() +inputChanged() +getDocumentForNumber() +getUnlinkedNumbers() </pre>	<pre> <<Java Class>> undefinedNumberException appLayer -serialVersionUID: long +undefinedNumberException() </pre>
	<pre> <<Java Class>> elementNotFoundException appLayer -serialVersionUID: long +elementNotFoundException() </pre>	<pre> <<Java Class>> dateColumn appLayer -serialVersionUID: long +dateColumn() +dateColumn() +getEditorType() +setValue() +getStringValue() +getValue() +assignValueFromResultSet() </pre>
<pre> +contact() +contact() ~contact() +getNumber() +bindToID() +getName() +getStreet() +getZIP() +getLocation() +getCountry() +getCO() +getEmail() +getPhone() +getFax() +getVATID() +getID() +getRole() +setName() +setStreet() +setZIP() +setLocation() +setCountry() +setCO() +setEmail() +setPhone() +setFax() +setRole() +setVATID() +toString() +delete() +save() +getBankAccount() +setBankaccount() +getBankCode() +setBankcode() +getAccountholder() +setAccountholder() +getPaymentmethod() +setPaymentmethod() </pre>	<pre> <<Java Class>> referenceColumn appLayer -serialVersionUID: long ~value: String +referenceColumn() +getEditorType() +assignValueFromResultSet() +getStringValue() +setValue() +getValue() </pre>	<pre> +dateColumn() +dateColumn() +getEditorType() +setValue() +getStringValue() +getValue() +assignValueFromResultSet() </pre>

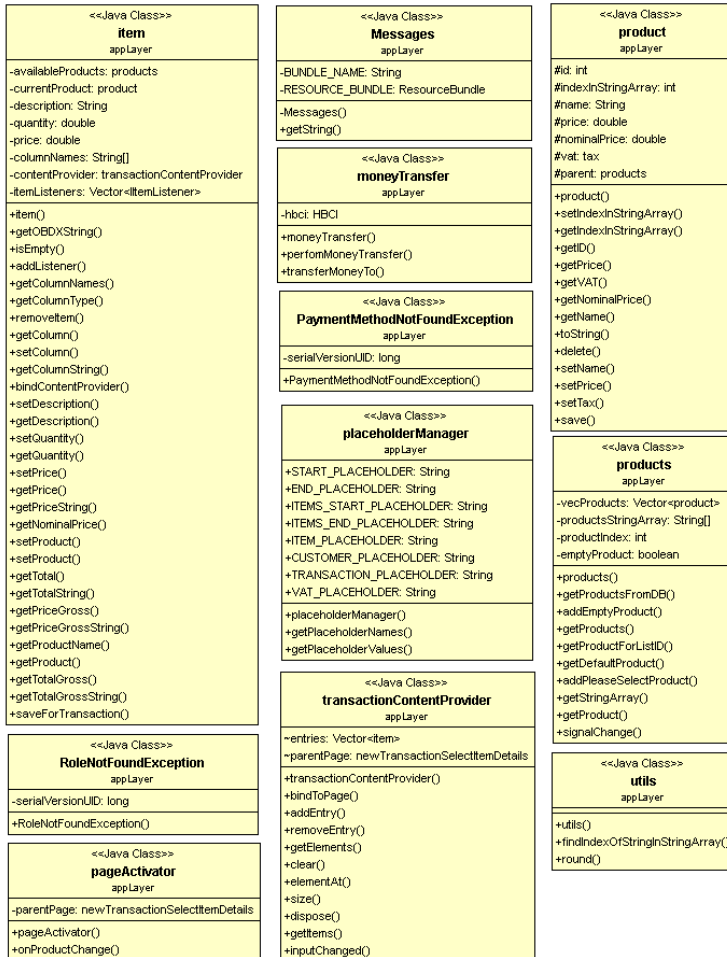
Gambar B. 6 Method dan Atribut Package appLayer part 4

Method dan atribut package appLayer part 5 terdapat pada Gambar B. 7 Method dan Atribut Package appLayer part 5.

<pre> <<Java Class>> columnNotFoundException appLayer -serialVersionUID: long +columnNotFoundException() </pre>	<pre> <<Java Class>> entry appLayer -serialVersionUID: long -Id: int +numberUndefined: int -number: int -import_item_id: int -contentProvider: bookingContentProvider -referredTransactionId: Integer -columns: Vector<column> -autoUpdateDBonEntryChange: boolean +entry() +entry() +entry() +entry() +prepare() +setAutoUpdate() -reNumberIndices() -getColumnsUpdateStr() -getColumnsInsertStrDefinitions() -getColumnsInsertStrValues() -getColumnName() +getColumnForIndex() +setImportID() +getImportID() +assignBasicValues() +assignBasicValuesFromResultSet() +removeColumn() +hasColumn() +getID() +isCreateSide() +setVAT() +toString() +isEmpty() +getColumnNames() +getColumnType() +getColumn() +getColumnAbsoluteValue() +getColumnString() +setColumn() +getAccountAsString() +getContraAccountAsString() +getCreditAccount() +getDebitAccount() +setCreditAccount() +setDebitAccount() +getValue() +setValue() +getDate() +getReference() +getDescription() +getComment() +setComment() +getVAT() +bindContentProvider() +update() +insert() +book() +bind() +setReferToTransaction() +setDate() +getReferToTransaction() +removeFromImport() +delete() +clone() +compareTo() +getNumber() +setNumber() </pre>	<pre> <<Java Class>> accountColumn appLayer -serialVersionUID: long -dbNameSubAccounts: String +accountColumn() +getEditorType() +getAccount() +assignValueFromResultSet() +getValue() +getStringValue() +setStringValue() +setDBUpdateStr() +getDBInsertStrDefinition() +getDBInsertStrValue() </pre>
<pre> <<Java Class>> taxColumn appLayer -serialVersionUID: long +taxColumn() +taxColumn() +getEditorType() +setValue() +getStringValue() +getValue() +assignValueFromResultSet() +getTaxValue() </pre>		<pre> <<Java Class>> entryNotInThisAccountException appLayer -serialVersionUID: long +entryNotInThisAccountException() </pre>
<pre> <<Java Class>> column appLayer -serialVersionUID: long #dbName: String #name: String -index: int #value: Object +column() +column() +setDbName() +getStringValue() +getName() +setName() +getIndex() +setIndex() +getEditorType() +getValue() +setValue() +getDBUpdateStr() +assignValueFromResultSet() +getDBInsertStrDefinition() +getDBInsertStrValue() </pre>		<pre> <<Java Interface>> IcontentProvider appLayer +getCurrentHelpURL() +setCurrentHelpURL() </pre>
<pre> <<Java Class>> doubleColumn appLayer -serialVersionUID: long +doubleColumn() +doubleColumn() +getEditorType() +setValue() +assignValueFromResultSet() +setDoubleValue() +getStringValue() +getValue() +getDoubleValue() </pre>		<pre> <<Java Interface>> ItemListener appLayer +onProductChange() </pre>
<pre> <<Java Class>> itemTextList appLayer -shell: Shell -document: ITextDocument -theTransaction: transaction -placeHolderManager: placeholderManager -placeHolderFields: ITextField[] +itemTextList() +process() +getItemTemplate() +insertItem() </pre>		<pre> <<Java Class>> Items appLayer -selectedItemIndex: int -table Viewer: Table Viewer +Items() +setSelectedItemIndex() +setCurrentDescription() +setCurrentQuantity() +setCurrentPrice() +bindTableViewer() +loadFromDB() +emptyAllItems() </pre>
		<pre> <<Java Class>> itemTableList appLayer -shell: Shell -document: ITextDocument -theTransaction: transaction -placeHolderManager: placeholderManager -placeHolderFields: ITextField[] +itemTableList() +process() +processSingleTable() +updateFormulas() </pre>

Gambar B. 7 Method dan Atribut Package appLayer part 5

Method dan atribut *package* appLayer part 6 terdapat pada Gambar B. 8 Method dan Atribut *Package* appLayer part 6.



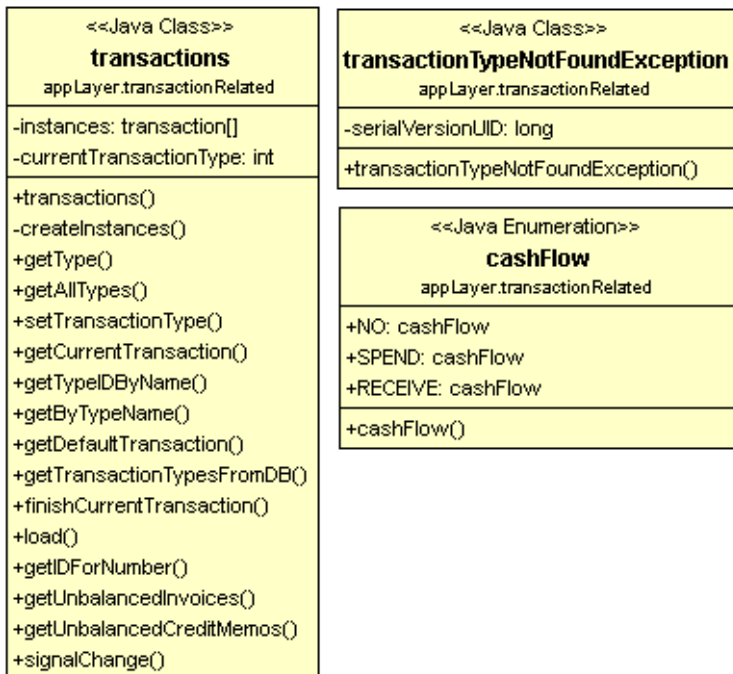
Gambar B. 8 Method dan Atribut *Package* appLayer part 6

Method dan atribut *package* `appLayer.taxRelated` terdapat pada Gambar B. 9 *Method* dan Atribut *Package* `appLayer.taxRelated`.

<pre><<Java Class>> state appLayer.taxRelated -name: String -taxCodeFormat: String -taxCodes: Vector<String> ~state() ~addTaxCode() +getOrderNumber() -getTaxIDWithoutFirstSlice() +getName() +getTaxCodeFormat()</pre>	<pre><<Java Class>> tax appLayer.taxRelated -serialVersionUID: long -description: String -value: double -IDinList: int -ID: int -creditAccount: account -debitAccount: account -creditTaxField: int -debitTaxField: int -selectedCreditAccount: boolean ~parent: taxList +tax() +bindToList() +getDescription() +getDinList() +getID() +setDescription() +getValue() +getFactor() +getCreditAccount() +getDebitAccount() +setCreditAccount() +setDebitAccount() +setValue() +getSelectedAccount() +isOnCreditSide() +getCreditTaxField() +getDebitTaxField() +setCreditTaxField() +setDebitTaxField() +setSelectedAccountCredit() +toString() +clone() +delete() +save()</pre>	<pre><<Java Class>> taxList appLayer.taxRelated -serialVersionUID: long -VATs: Vector<tax> -emptyName: String -creditDebitEnabled: boolean #addEmptyTax: boolean +taxList() +getTaxesFromDatabase() +getVATAtListIndex() +setEmptyElementName() +enableCreditDebit() +getVATs() +addEmptyTax() +getVATArray() +getVATByID() +getDefault() +getStringArray() +signalChange()</pre>
<pre><<Java Class>> states appLayer.taxRelated -states: Vector<state> +states() +getStates()</pre>		
<pre><<Java Class>> IRSoffices appLayer.taxRelated -offices: IRSoffice[] +IRSoffices() +getIRSoffices()</pre>		
<pre><<Java Class>> IRSoffice appLayer.taxRelated -id: String -name: String +IRSoffice() +getName() +getID()</pre>		<pre><<Java Class>> taxNotFoundException appLayer.taxRelated -serialVersionUID: long +taxNotFoundException()</pre>

Gambar B. 9 *Method* dan Atribut *Package* `appLayer.taxRelated`

Method dan atribut *package* appLayer.transactionRelated part 1 terdapat pada Gambar B. 10 *Method* dan Atribut *Package* appLayer.transactionRelated part 1.



Gambar B. 10 *Method* dan Atribut *Package* appLayer.transactionRelated part 1

Method dan atribut *package* `appLayer.transactionRelated` part 2 terdapat pada Gambar B. 11 *Method* dan Atribut *Package* `appLayer.transactionRelated part 2`.

<<Java Class>> transaction appLayer.transactionRelated	
<pre> #allItems: Items ~insertedTransactionID: int ~recipient: contact ~number: String ~numericNumber: int ~precedingTransactionID: int ~dueDate: Date ~issueDate: Date ~remarks: String ~filenameODT: String ~filenamePDF: String ~filenameOBDX: String ~filenameRelativeODT: String ~filenameRelativePDF: String ~filenameRelativeOBDX: String ~parent: transactions ~typeID: int ~typeIndex: int ~typePeriod: int ~typeName: String ~typePrefix: String ~typeFormat: String +transaction() +getParent() +bindToType() +getType() +getCashFlow() +getTransactionName() +book() +getNumWorkflowSteps() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getToDoItems() +toString() ~getTotal() +setIssueDate() </pre>	<pre> -recalcDueDate() +getTotalGross() ~emptyItems() +setReferTo() +cloneFrom() +getTotalString() +getTotalGrossString() +getNewTransactionNumber() ~getTwoDigitNumStrFromInt() ~applyFormatToNumericNumber() +getVatTotal() +getVatList() +save() +getID() +getItems() +setTransactionAttributes() +loadItemsFromDB() +addItem() +getRecipient() +getAllProductsAndItems() +getIssueDate() +getDueDate() +bindToCustomer() +getRefersTo() +getNumber() +getUnchangedNumber() +getFilenameODT() +getFilenamePDF() +getFilenamePDFGPG() +getFilenameOBDX() +getRemarks() +getPrecedingTransactionID() +setRemarks() +getTypeID() +getTypeIndex() +getTypePeriod() +getTypeName() +getTypePrefix() +getTypeFormat() +setTypeIndex() +setTypePeriod() +setTypeName() +setTypePrefix() +setTypeFormat() +saveType() </pre>

Gambar B. 11 *Method* dan Atribut *Package* `appLayer.transactionRelated part 2`

Method dan atribut *package* appLayer.transactionRelated part 3 terdapat pada Gambar B. 12 Method dan Atribut Package appLayer.transactionRelated part 3.

<pre> <<Java Class>> cancelation appLayer.transactionRelated +cancelation() +book() -getOriginalTransactionsEntry() +getTransactionName() +getType() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>	<pre> <<Java Class>> creditmemo appLayer.transactionRelated ~pendingCreditAccount: account ~pendingDebitAccount: account ~finishCreditAccount: account ~finishDebitAccount: account +creditmemo() +book() +getType() +getTransactionName() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>	<pre> <<Java Class>> personalDrawing appLayer.transactionRelated ~pendingCreditAccount: account ~pendingDebitAccount: account ~finishCreditAccount: account ~finishDebitAccount: account +personalDrawing() +book() +getType() +getTransactionName() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>
<pre> <<Java Class>> invoice appLayer.transactionRelated ~pendingCreditAccount: account ~pendingDebitAccount: account ~finishCreditAccount: account ~finishDebitAccount: account +invoice() +getType() +book() +getTransactionName() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>	<pre> <<Java Class>> offer appLayer.transactionRelated +offer() +book() +getTransactionName() +getType() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>	<pre> <<Java Class>> reminder appLayer.transactionRelated +reminder() +book() +getTransactionName() +getType() +getCreditAccountForWorkflowStep() +getDebitAccountForWorkflowStep() +getNumWorkflowSteps() +getTodoItems() +getCashFlow() </pre>

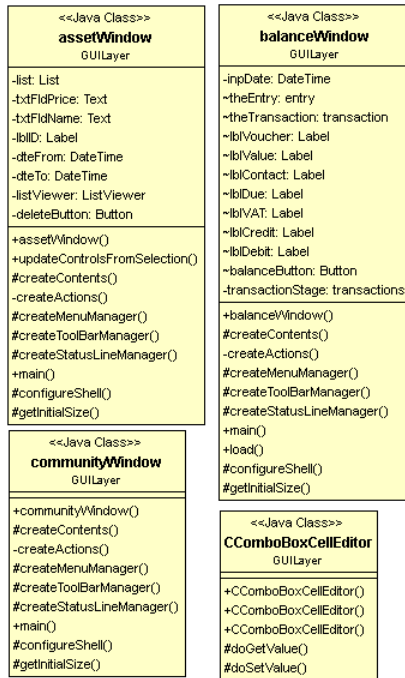
Gambar B. 12 Method dan Atribut Package appLayer.transactionRelated part 3

Method dan atribut package GUILayer part 1 terdapat pada Gambar B. 13 Method dan Atribut Package GUILayer part .

<pre> <<Java Class>> aboutWindow GUILayer -FreeSoftwareStyledText: StyledText -somethingStyledText: StyledText -versionHistoryStyledText: StyledText #imageX: int #imageY: int -handCursorActive: boolean -defaultCursor: Cursor -handCursor: Cursor +aboutWindow() +inWhichClickableArea() #createContents() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> accountDefinitionWindow GUILayer -cmbSubaccounts: Combo -cmbType: Combo -btFldDescription: Text -btFldCode: Text -lblID: Label -deleteButton: Button -listViewer: ListView +accountDefinitionWindow() #createContents() +updateControlsFromSelection() -createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> accountingEditWindow GUILayer #tableViewer: TableView -allEntryList: entryList #contentProvider: bookingContentProvider -table: Table +accountingEditWindow() +refreshAccountingEntries() #createContents() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>
<pre> <<Java Class>> ContentProvider GUILayer +ContentProvider() +getElements() +getAllAccounts() +dispose() +inputChanged() </pre>	<pre> <<Java Class>> ContentProvider GUILayer #allAssets: assets +ContentProvider() +getElements() +dispose() +inputChanged() </pre>	<pre> <<Java Class>> accountSelectionChangedListener GUILayer #parent: accountDefinitionWindow +accountSelectionChangedListener() +selectionChanged() </pre>
<pre> <<Java Class>> entryList GUILayer +entryList() </pre>	<pre> <<Java Class>> assetSelectionChangedListener GUILayer #parent: assetWindow +assetSelectionChangedListener() +selectionChanged() </pre>	<pre> <<Java Class>> browserWindow GUILayer -uri: String +browserWindow() #createContents() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>

Gambar B. 13 Method dan Atribut Package GUILayer part 1

Method dan atribut package GUILayer part 2 terdapat pada Gambar B. 14 Method dan Atribut Package GUILayer part 2.



Gambar B. 14 Method dan Atribut Package GUILayer part 2

Method dan atribut *package* GUILayer part 3 terdapat pada Gambar B. 15 Method dan Atribut *Package* GUILayer part 3.

<pre> <<Java Class>> testCase GUILayer -testDescription: String #config/Window: config/Window -tr: testResult +copyFile() +getResult() +testCase() ~doTest() #execute() +getDescription() </pre>	<pre> <<Java Class>> configWindow GUILayer -txtFldGPGPath: Text -txtFldCTAPI: Text -txtFldWinstonPath: Text -txtFldOOoPath: Text -istPrinters: List -txtFlddatabasetype: Text -txtFlddatabasedriverfilename: Text -txtFlddatabasedriver: Text -txtFlddatabaspassword: Text -txtFlddatabasemailer: Text -txtFlddatabasename: Text -txtFlddatabaserver: Text -chkUseExternalDB: Button -chkPowered: Button -chkUseChipcardPinpad: Button -group: Group -istStatus: List -tabFolder: TabFolder -testCases: Vector<testCase> -waitForCorrectConfigBeforeMain/Window: boolean +configWindow() #createContents() +checkOOo() -getOOoMajorVersion() +test() -testConnect() +testAndShow() -selectValuesFromConfigFile() +connectDatabase() -createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() +show() #getInitialSize() </pre>	<pre> <<Java Class>> testResult GUILayer -resultString: String -result: Boolean -isFatal: boolean +testResult() +getResult() +getResultString() +setFatal() +getFatal() </pre>
<pre> <<Java Class>> templateCheck GUILayer +templateCheck() ~doTest() </pre>		<pre> <<Java Class>> mainDirTestCase GUILayer +mainDirTestCase() ~doTest() </pre>
<pre> <<Java Class>> HSQLOLockCheck GUILayer +HSQLOLockCheck() ~doTest() </pre>		<pre> <<Java Class>> OOoCheck GUILayer +OOoCheck() ~doTest() </pre>
<pre> <<Java Class>> databaseConnectivityTest GUILayer +databaseConnectivityTest() ~doTest() </pre>		<pre> <<Java Class>> databaseDriverNameTestCase GUILayer +databaseDriverNameTestCase() ~doTest() </pre>
<pre> <<Java Class>> standardPositiveTestResult GUILayer +standardPositiveTestResult() </pre>		<pre> <<Java Class>> tableCheck GUILayer +tableCheck() ~doTest() </pre>

Gambar B. 15 Method dan Atribut *Package* GUILayer part 3

Method dan atribut package GUILayer part 4 terdapat pada Gambar B. 13 Method dan Atribut Package GUILayer part .

<pre> <<Java Class>> contactsWindow GUILayer -cmbPayment: Combo -txtFldAccountHolder: Text -txtFldBankCode: Text -txtFldBankAccount: Text -txtFldVATID: Text -txtFldFax: Text -txtFldPhone: Text -listViewer: ListViewer -txtFldEmail: Text -cmbRole: Combo -txtFldCountry: Text -txtFldLocation: Text -txtFldZip: Text -txtFldStreet: Text -txtFldCO: Text -txtFldName: Text -bankCodeLabel: Label -bankAccountLabel: Label -lblID: Label -deleteButton: Button -notYetAssignedText: String -customerIDs: Vector<Integer> +contactsWindow() #createContents() #checkBankData() +updateControlsFromSelection() -createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> ContentProvider GUILayer #allContacts: contacts +ContentProvider() +getElements() +dispose() +inputChanged() </pre>	<pre> <<Java Class>> designerWindow GUILayer ~vecPrefix: Vector<String> -transactionPrefix: String -saveButton: Button -officeFrame1: JFrame -textDocument1: ITextDocument ~officeApplication: IOfficeApplication -cmbTemplate: Combo +designerWindow() +getTextDocument() -loadTemplates() +getTransactionPrefix() #createContents() +close() -createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>
	<pre> <<Java Class>> contactSelectionChangedListener GUILayer #parent: contactsWindow +contactSelectionChangedListener() +selectionChanged() </pre>	<pre> <<Java Class>> documentCellModifier GUILayer -documentCellModifier() +canModify() +getValue() +modify() </pre>
	<pre> <<Java Class>> selectionAdapterWithParent GUILayer -parent: designerWindow +selectionAdapterWithParent() +widgetSelected() </pre>	<pre> <<Java Class>> documentLabelProvider GUILayer -captionCodes: String[] -documentLabelProvider() +getColumnImage() +getColumnText() +getColumnNames() <<Java Class>> testTest GUILayer +testTest() ~doTest() </pre>
	<pre> <<Java Class>> documentsList GUILayer +documentsList() </pre>	

Gambar B. 16 Method dan Atribut Package GUILayer part 4

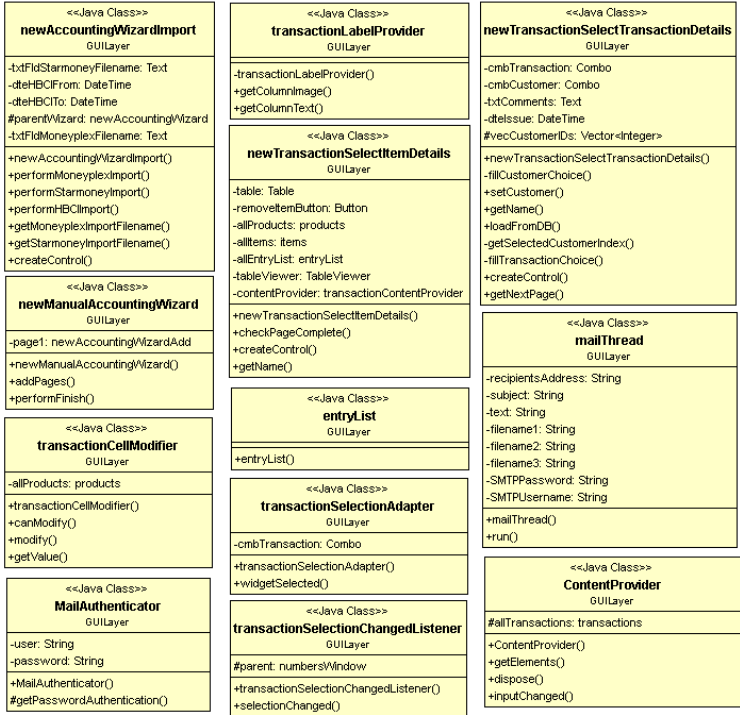
Method dan atribut *package* GUILayer part 5 terdapat pada Gambar B. 17 Method dan Atribut Package GUILayer part 5.

<pre> <<Java Class>> documentsWindow GUILayer -table_1: Table +documentsWindow() #createContents() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> MainWindow GUILayer -importDocumentsAction: Action -importOBDXAction: Action -projectHomePageAction: Action -sourceforgePageAction: Action -importTimetrackerAction: Action -exportVatAnnouncementAction: Action -settingsAction: Action -importFromBankAction: Action -accountDefinitionAction: Action -accountsAction: Action -txtFldOoOoPath: Text -usageStatsID: int -todoWindowOpen: boolean -configurationAction: Action -numbersformatsAction: Action -quitAction: Action -taxesAction: Action -aboutAction: Action -templateDesignerAction: Action -todoButton: Button +MainWindow() #setConfigPath() +getVersionHistory() #createContents() +getTxtFldOoOoPath() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() +updateNrTodoItems() #configureShell() #getInitialSize() +signalTodoWindowOpen() +isTodoWindowOpen() </pre>	<pre> <<Java Class>> Messages GUILayer -BUNDLE_NAME: String -RESOURCE_BUNDLE: ResourceBundle -Message() +getString() <<Java Class>> entryList GUILayer +entryList() <<Java Class>> newAccountingWizard GUILayer -page1: newAccountingWizardImport -page2: newAccountingWizardAdd +newAccountingWizard() +addPages() +performImport() +performFinish() <<Java Class>> newAccountingWizardAdd GUILayer -text: Text -parentWizard: newAccountingWizard #table: Table #tableViewer: TableViewler #contentProvider: bookingContentPro... -allEntryList: entryList -taxes: taxList -accounts: accountsList +newAccountingWizardAdd() +setVisible() +createControl() +getEntryList() +performBookings() </pre>
<pre> <<Java Class>> entryDetailWindow GUILayer #balanceButton: Button #oi: OBDXImporter +entryDetailWindow() #createContents() #createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>		
<pre> <<Java Class>> HBCICallbackDialog GUILayer +HBCICallbackDialog() +callback() </pre>		

Gambar B. 17 Method dan Atribut Package GUILayer part 5

A-18

Method dan atribut package GUILayer part 6 terdapat pada Gambar B. 18 Method dan Atribut Package GUILayer part 6.



Gambar B. 18 Method dan Atribut Package GUILayer part 6

Method dan atribut package GUILayer part 7 terdapat pada Gambar B. 19 Method dan Atribut Package GUILayer part 7.

<pre> «Java Class» newTransactionWizard GUILayer -page1: newTransactionSelectTransactionDetails -page2: newTransactionSelectItemDetails -page3: newTransactionWizardPreview -page4: newTransactionWizardPrintAndSave -officeFrame: JFrame -textDocument: TTextDocument +getWizardSelect() +setOfficeFrame() +getOfficeFrame() +setDocument() +getDocument() +newTransactionWizard() +preview() +addPages() +performFinish() </pre>	<pre> «Java Class» newTransactionWizardPreview GUILayer -parentWizard: newTransactionWizard -officeFrame: JFrame +newTransactionWizardPreview() +onActivate() +getName() +createControl() +insertHeader() +process() +setVisible() -appendDocument() -doReplaceItemless() </pre>	<pre> «Java Class» productsWindow GUILayer -listViewer: ListViewer -ombVAT: Combo -bxFidPrice: Text -bxFidName: Text -lblID: Label ~vatIDs: Vector<Integer> ~productIDs: Vector<Integer> -deleteButton: Button +productsWindow() #createContents() +updateControlsFromSelection() -createActions() -getVATIDs() #createMenuBar() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>
<pre> «Java Class» numbersWindow GUILayer -listViewer: ListViewer -bxFidPeriod: Text -bxFidFormat: Text -bxFidIndex: Text -bxFidPrefix: Text -bxFidName: Text -lblID: Label +numbersWindow() #createContents() +updateControlsFromSelection() -createActions() #createMenuBar() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> «Java Class» newTransactionWizardPrintAndSave GUILayer -parentWizard: newTransactionWizard -chkPrint: Button -chkEmail: Button +isPrintChecked() +isEmailChecked() +getName() +newTransactionWizardPrintAndSave() +createControl() +getNextPage() </pre>	<pre> «Java Class» ContentProvider GUILayer #allProducts: products +ContentProvider() +getElement() +dispose() +inputChanged() </pre>
	<pre> «Java Class» passwordDialog GUILayer -RESET_ID: int -usernameField: Text -passwordField: Text -username: String -password: String +passwordDialog() #createDialogArea() +getUsername() +getPassword() #createButtonsForButtonBar() #buttonPressed() </pre>	<pre> «Java Class» productSelectionChangedListener GUILayer #parent: productsWindow +productSelectionChangedListener() +selectionChanged() </pre>

Gambar B. 19 Method dan Atribut Package GUILayers part 7

Method dan atribut package GUILayer part 8 terdapat pada Gambar B. 20 Method dan Atribut Package GUILayer part 8.

<pre><<Java Class>> reportWizard GUILayer #spreadDocument: ISpreadsheetDocument +reportWizard() +addPages() +getSpreadDocument() +isOpenable() +setSpreadDocument() +performFinish()</pre>	<pre><<Java Class>> reportWizardShow GUILayer #items: Vector<Object[]> #parentWizard: reportWizard #spreadsheet1: XSpreadsheet #spreadsheet2: XSpreadsheet #spreadsheet3: XSpreadsheet #spreadsheet4: XSpreadsheet +reportWizardShow() +createControl() -reportJournal() -reportLists() -reportTAccounts() -reportAccountsList() +setVisible() +dispose()</pre>	<pre><<Java Class>> settingsWindow GUILayer -txtHoldername: Text -txtBankaccount: Text -txtBankcode: Text -txtBankname: Text -chkSMTPAuth: Button -txtSMTPpassword: Text -txtSMTPuser: Text -txtSMTPserver: Text -cmbVATperiod: Combo -cmbIRS: Combo -txtOrganizationName: Text -cmbState: Combo -txtTaxID: Text -bankCodeLabel: Label -bankAccountLabel: Label +settingsWindow() -checkBankDetails() #createContents() -createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize()</pre>
<pre><<Java Class>> tAccount GUILayer -accountObject: account -headerLines: Vector<String> -creditLines: Vector<tAccountLine> -debitLines: Vector<tAccountLine> -footerLines: Vector<String> -TOP: int -BOTTOM: int -LEFT: int -RIGHT: int +tAccount() +addHeaderLine() +addFooterLine() +addCreditLine() +addDebitLine() -getMaxLines() +draw() -drawBorder()</pre>	<pre><<Java Class>> tAccountLine GUILayer ~value: String ~reference: String +tAccountLine() +getValue() +getReference()</pre>	<pre><<Java Interface>> tableWindow GUILayer +refreshTable()</pre>
<pre><<Java Class>> reportWizardSelect GUILayer -dteFrom: DateTime -dteTo: DateTime #parentWizard: reportWizard +reportWizardSelect() +createControl() +getNextPage()</pre>	<pre><<Java Class>> warningFocusAdapter GUILayer ~warned: boolean ~sh: Shell +warningFocusAdapter() +focusGained()</pre>	<pre><<Java Class>> ContentProvider GUILayer #allTaxes: taxList +ContentProvider() +getElement() +dispose() +inputChanged()</pre>
	<pre><<Java Class>> taxSelectionChangedListener GUILayer #parent: taxesWindow +taxSelectionChangedListener() +selectionChanged()</pre>	

Gambar B. 20 Method dan Atribut Package GUILayer part 8

Method dan atribut *package* *GUILayer* part 9 terdapat pada Gambar B. 21 *Method* dan Atribut *Package GUILayer part 9*.

<pre> <<Java Class>> taxesWindow GUILayer -:txtFldDebitTax: Text -:txtFldCreditTax: Text -:cmbDebitAccount: Combo -:cmbCreditAccount: Combo -:txtFldValue: Text -:txtFldDescription: Text -:deleteButton: Button -:lblID: Label -:vats: taxList #listViewer: ListViewer -:unknownIDString: String +taxesWindow() #createContents() +updateControlsFromSelection() -:createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> TimeCellEditor GUILayer -:dateTime: DateTime -:defaultStyle: int +TimeCellEditor() #createControl() #doGetValue() #doSetFocus() #doSetValue() ~:applyEditorValueAndDeactivate() #focusLost() #keyReleaseOccured() </pre>	<pre> <<Java Class>> VATannouncementWindow GUILayer -:cmbYear: Combo -:cmbPeriod: Combo +VATannouncementWindow() +isOpenable() #createContents() -:createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() </pre>
<pre> <<Java Class>> timetrackerImportWindow GUILayer -:txtTaskCoachFilename: Text -:txtKTimecoachFilename: Text -:importButton: Button +timetrackerImportWindow() #createContents() -:createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() #configureShell() #getInitialSize() </pre>	<pre> <<Java Class>> todoWindow GUILayer -:table: Table -:tableEditors: Vector<Link> -:shell: Shell +todoWindow() +refreshToDoList() #createContents() -:checkTableForDefaultValue() -:checkTemplates() -:checkUnprocessedOffers() -:checkUnbalancedInvoices() +getShell() -:checkUnbalancedCreditmemos() -:createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() +main() #configureShell() #getInitialSize() +close() </pre>	<pre> <<Java Class>> exportAction GUILayer -:fieldValues: HashMap<Integer,String> -:VATtoPay: Double #parentWindow: VATannouncementWindow +exportAction() -:finishFieldValues() -:getValueForField() +widgetSelected() </pre>
<pre> <<Java Class>> todoBalanceSelectionAdapter GUILayer -:entryID: int -:shell: Shell +todoBalanceSelectionAdapter() +widgetSelected() </pre>	<pre> <<Java Class>> viewerWindow GUILayer -:url: String #textDocument: ITextDocument +viewerWindow() #createContents() -:createActions() #createMenuManager() #createToolBarManager() #createStatusLineManager() #configureShell() +close() #getInitialSize() </pre>	

Gambar B. 21 *Method* dan Atribut *Package GUILayer part 9*

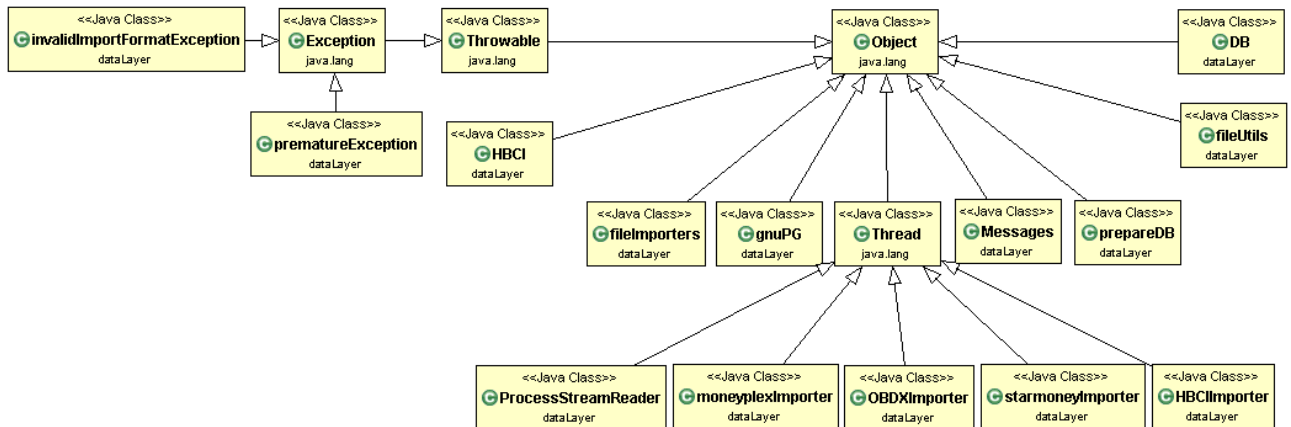
A-22

(Halaman ini sengaja dikosongkan)

LAMPIRAN B – Generaization (Inheritance) Relationship

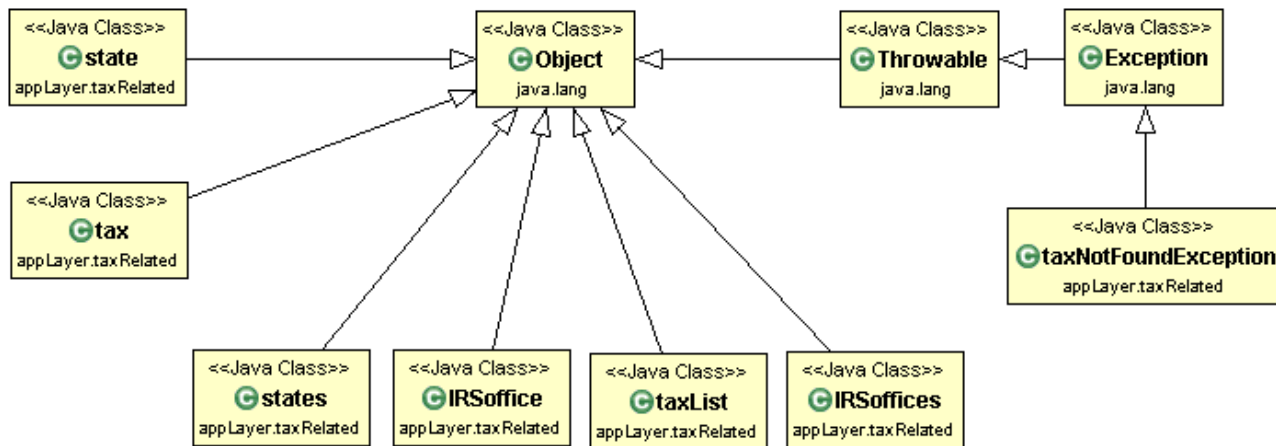
Lampiran ini menjabarkan struktur hirarki masing-masing package.

Inheritance hierarchy package dataLayer terdapat pada Gambar C. 1 *Inheritance Hierarchy Package dataLayer*.



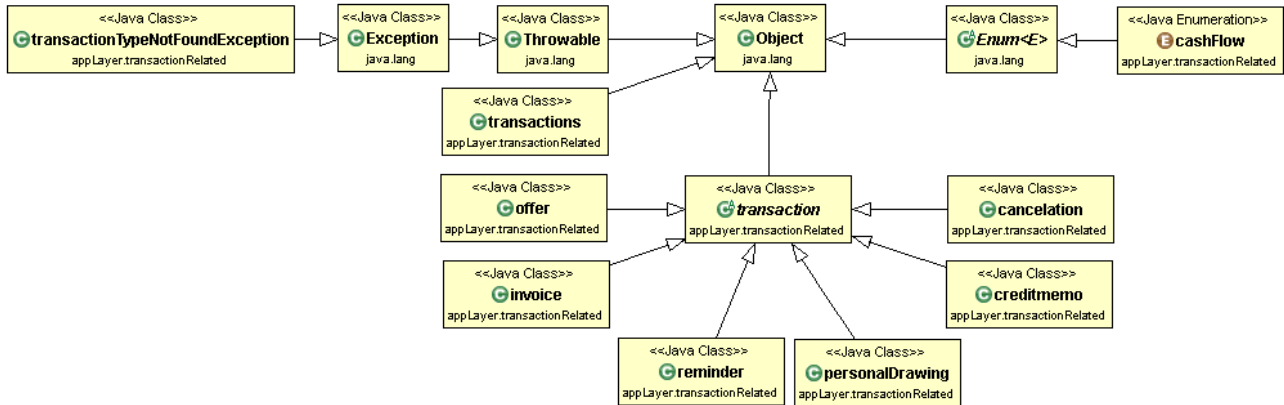
Gambar C. 1 *Inheritance Hierarchy Package dataLayer*

Inheritance hierarchy package appLayer.taxRelated terdapat pada Gambar C. 2 *Inheritance Hierarchy Package appLayer.taxRelated*.



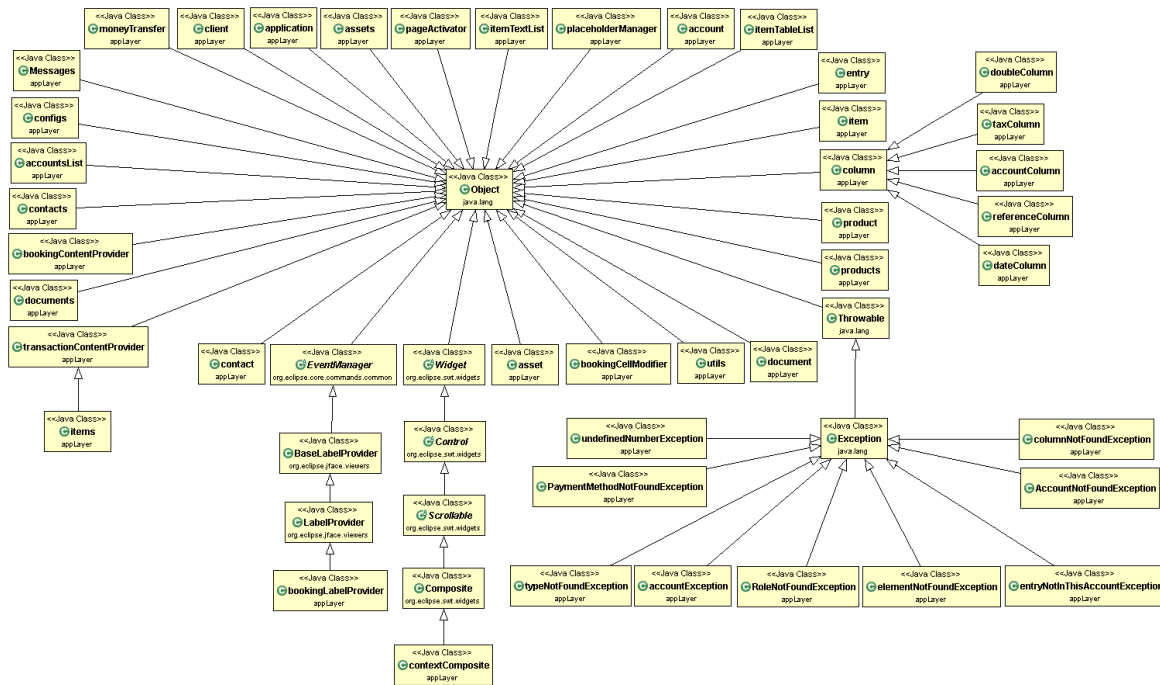
Gambar C. 2 *Inheritance Hierarchy Package appLayer.taxRelated*

Inheritance hierarchy package appLayer.transactionRelated terdapat pada Gambar C. 3 Inheritance Hierarchy Package appLayer.transactionRelated.



Gambar C. 3 Inheritance Hierarchy Package appLayer.transactionRelated

Inheritance hierarchy package appLayer terdapat pada Gambar C. 4 Inheritance hierarchy package appLayer. Inheritance hierarchy package GUILayer terdapat pada Gambar C. 5 Inheritance hierarchy package GUILayer.



Gambar C. 4 Inheritance hierarchy package appLayer

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis bernama lengkap Aula Ayubi. Penulis lahir di Kediri, tanggal 9 November 1992 dan merupakan anak ke empat dari empat bersaudara. Penulis telah menempuh pendidikan formal di SD Negeri Manisrenggo I Kediri, SMP Negeri 1 Kediri, serta SMA Negeri 1 Kediri.

Pada Tahun 2011, penulis diterima di Jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember Surabaya dan tercatat sebagai mahasiswi dengan NRP 5211100111. Selama masa perkuliahan, penulis aktif di bidang akademik dan organisasi. Penulis pernah lolos didanai pada Pekan Keilmiah Mahasiswa DIKTI, 20 Besar gemastik ITB, Juara 3 Kompetisi Aplikasi ITS. Penulis pernah menjadi asisten dosen pada mata kuliah Sistem Operasi dan Rekayasa Perangkat Lunak, penulis juga pernah menjadi Asisten Laboratorium Perencanaan dan Pengembangan Sistem Informasi selama empat semester. Di bidang non-akademik, penulis mengikuti organisasi mahasiswa, UKM Kewirausahaan ITS 2011/2012, HMSI Berdedikasi 2012/2013 dan HMSI Sinergi 2013/2014, serta beberapa kepanitian acara tingkat jurusan, institut, dan nasional. Penulis juga pernah berkesempatan kerja paruh waktu di CV.Dynamic Team Solution 2013-2014. Penulis juga pernah melaksanakan kerja praktik di Mahkamah Konstitusi-Indonesia selama 2 bulan pada tahun 2014.

Di akhir tahun perkuliahannya, penulis memilih bidang minat PPSI dengan topik tugas akhir di bidang pengukuran kualitas perangkat lunak. Untuk keperluan penelitian, dapat menghubungi penulis melalui email: aulaayubi@gmail.com.

(Halaman ini sengaja dikosongkan)