



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

**ANALISIS PERBANDINGAN FREE/ OPEN SOURCE ERP
(FOS ERP) DARI ASPEK ARSITEKTUR, FUNGSIONALITAS,
DAN KOMUNITAS**

***ANALYSIS OF COMPARISON ON FREE/ OPEN SOURCE
ERP (FOS ERP) FROM ASPECT OF ARCHITECTURE,
FUNCTIONALITY, AND COMMUNITY***

**TESAR AKRAM PRATAMA
NRP 5213 100 097**

**Dosen Pembimbing
Mahendrawathi ER, S.T., M.Sc., Ph.D**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

ANALISIS PERBANDINGAN FREE/ OPEN SOURCE ERP (FOS ERP) DARI ASPEK ARSITEKTUR, FUNGSIONALITAS, DAN KOMUNITAS

TESAR AKRAM PRATAMA
NRP 5213 100 097

Dosen Pembimbing
Mahendrawathi ER, S.T., M.Sc., Ph.D

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS141501

***ANALYSIS OF COMPARISON ON FREE/ OPEN SOURCE
ERP (FOS ERP) FROM ASPECT OF ARCHITECTURE,
FUNCTIONALITY, AND COMMUNITY***

TESAR AKRAM PRATAMA
NRP 5213 100 097

Supervisor
Mahendrawathi ER, S.T., M.Sc., Ph.D

INFORMATION SYSTEMS DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2016

LEMBAR PENGESAHAN

ANALISIS PERBANDINGAN FREE/ OPEN SOURCE ERP (FOS ERP) DARI ASPEK ARSITEKTUR, FUNGSIONALITAS, DAN KOMUNITAS

TUGAS AKHIR

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

Oleh :

TESAR AKRAM PRATAMA
NRP 5213 100 097

Surabaya, Desember 2016

KETUA
JURUSAN SISTEM INFORMASI



Dr. Ir. Aris Fahyanto, M.Kom
NIP 19650310 199102 1 001

LEMBAR PERSETUJUAN

ANALISIS PERBANDINGAN FREE/ OPEN SOURCE ERP (FOS ERP) DARI ASPEK ARSITEKTUR, FUNGSIONALITAS, DAN KOMUNITAS

TUGAS AKHIR

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

Oleh :


TESAR AKRAM PRATAMA
NRP 5213 100 097

Disetujui Tim Penguji : Tanggal Ujian : 9 Januari 2017
Periode Wisuda : Maret 2017

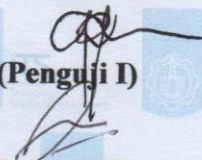
Mahendrawati Er., S.T., M.Sc., Ph.D


(Pembimbing I)

Arif Wibisono, S.Kom., M.Sc


(Penguji I)

Andre Parvian Aristio, S.Kom., M.Sc


(Penguji II)

**ANALISIS PERBANDINGAN *FREE/ OPEN SOURCE*
ERP (FOS ERP) DARI ASPEK ARSITEKTUR,
FUNGSIONALITAS, DAN KOMUNITAS**

Nama Mahasiswa : Tesar Akram Pratama
NRP : 5213 100 097
Jurusan : Sistem Informasi FTIF-ITS
Pembimbing 1 : Mahendrawathi Er., S.T, M.Sc, Ph.D

ABSTRAK

ERP sebagai bagian dari enterprise system yang mengintegrasikan antar data mempunyai peran penting dalam pertukaran informasi antar departemen dalam organisasi. ERP yang berperan menyediakan sistem informasi terintegrasi dalam organisasi ini, kini tidak hanya dibutuhkan oleh perusahaan-perusahaan besar, tetapi juga perusahaan berskala menengah dan kecil. Namun mahalnnya biaya implementasi ERP menjadi masalah utama bagi kebanyakan perusahaan, utamanya mereka yang beroperasi dengan ekonomi terbatas. Oleh karena itu, banyak vendor yang telah mengembangkan ERP dengan biaya lisensi gratis. ERP ini dikenal dengan nama Free/ Open Source ERP (FOS ERP). FOS ERP merupakan jenis ERP yang lebih baik untuk diterapkan bagi perusahaan dengan finansial terbatas. Namun, perkembangan FOS ERP yang pesat menyebabkan pilihan FOS ERP menjadi begitu banyak dan beragam. Perusahaan belum tentu dapat memilih ERP yang tepat sesuai dengan kondisi dan kebutuhannya masing-masing. Oleh sebab itu, diperlukan analisis perbandingan antar FOS ERP sebagai acuan yang komprehensif untuk menentukan ERP yang sesuai pada tingkat perusahaan.

Dalam tugas akhir ini, dilakukan perbandingan antar berbagai FOS ERP dengan kriteria pembanding berdasarkan arsitektur penyusun, fungsionalitas, serta komunitas. Perbandingan untuk aspek arsitektur meninjau pada struktur, kelengkapan, dan

gambaran arsitektur penyusun sistem ERP. Pada aspek fungsionalitas, fokus perbandingan hanya pada modul sales, accounting, dan purchase. Sedangkan untuk aspek komunitas, perbandingan adalah berdasarkan penilaian keaktifan komunitas yang ditinjau dari dua indikator, yakni jumlah anggota dan jumlah knowledge shared. Seluruh kriteria perbandingan tersebut memiliki indikator penilaiannya masing-masing guna mengetahui FOS ERP yang sesuai untuk digunakan perusahaan berdasarkan tiap aspek tersebut.

Setelah melalui hasil penilaian dan perbandingan, maka diperoleh hasil bahwa perbandingan antar sistem FOS ERP dapat memberikan rekomendasi FOS ERP yang sesuai di tingkat perusahaan. Perbandingan yang diberikan adalah berdasarkan beberapa kriteria/ aspek perbandingan. Tiap aspek memiliki rekomendasi satu atau beberapa FOS ERP yang sesuai untuk diterapkan perusahaan. Jadi, hasil penilaian ini dapat menjadi acuan perusahaan mengenai rencana implementasinya terhadap ERP, dimana FOS ERP yang dipilih tersebut bergantung pada kriteria dan penilaian tertentu yang difokuskan oleh tiap perusahaan.

Kata kunci: ERP, FOS ERP, Arsitektur, Fungsionalitas, Komunitas

ANALYSIS OF COMPARISON ON FREE/ OPEN SOURCE ERP (FOS ERP) FROM ASPECTS OF ARCHITECTURE, FUNCTIONALITY, AND COMMUNITY

Nama Mahasiswa : Tesar Akram Pratama
NRP : 5213 100 097
Jurusan : Sistem Informasi FTIF-ITS
Pembimbing 1 : Mahendrawathi Er., S.T, M.Sc, Ph.D

ABSTRACT

ERP as part of the enterprise system which integrates between data has an important role in the exchange of information between departments within the organization. This ERP roles in providing integrated information system in organization, now is not only required by large enterprises, but also medium and small enterprises. But the high cost of ERP implementation is a major problem for most enterprises, especially those operating with limited economic. Therefore, many ERP vendors have developed a free license ERP. This ERP is known as Free/ Open Source ERP (FOS ERP). FOS ERP is a better type of ERP to be applied to enterprises with limited financial. However, the rapid development of FOS ERP led to selection of FOS ERP into a vast and varied. Enterprises can't choose the right ERP yet in accordance with their conditions and needs. Therefore, it is necessary to analyze the comparison between FOS ERP as a comprehensive reference for determining the appropriate ERP at the enterprises level.

In this thesis, comparisons between various ERP FOS is developed with comparative criteria based on architecture, functionality, and community. Comparison on architecture aspects is reviewed to structure, completeness, and picture of ERP system composer architectural. In the aspect of functionality, comparative focus only on sales, accounting, and purchase module. As for community aspect, comparison is

based on an assessment of community activity in terms of the two indicators, namely total of members and total of knowledged shared. All the comparative criteria have each indicator of assessment to determine the appropriate FOS ERP for enterprises use based on each of these perspectives.

After going through the assessment and comparison, then obtained results that comparison between FOS ERP systems can provide recommendations of appropriate FOS ERP at the enterprise level. That given comparison is based on multiple comparative criterias/ perspectives. Each perspective has one or several FOS ERP that appropriate to be applied in enterprises. Therefore, the results of this assessment can be a reference for enterprises on ERP implementation plan, which is the selected FOS ERP depends on the specific assessments and criterias that are focused by each enterprises.

Keywords: ERP, FOS ERP, Architecture, Functionality, Community

KATA PENGANTAR

Alhamdulillah atas karunia, rahmat, barakah, dan jalan yang telah diberikan Allah SWT selama ini sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

ANALISIS PERBANDINGAN *FREE/ OPEN SOURCE* ERP (FOS ERP) DARI ASPEK ARSITEKTUR, FUNGSIONALITAS, DAN KOMUNITAS

Terima kasih atas pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik materi maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Bapak Dr. Ir. Aris Tjahyanto, M.Kom selaku Ketua Jurusan Sistem Informasi ITS Surabaya
2. Ibu Mahendrawathi ER, S.T., M.Sc., Ph.D selaku dosen pembimbing yang meluangkan waktu, memberikan ilmu, petunjuk, dan motivasi untuk kelancaran tugas akhir ini.
3. Bapak Arif Wibisono, S.Kom., M.Sc. dan Bapak Andre Parvian Aristio, S.Kom., M.Sc. selaku dosen penguji yang telah memberikan masukan untuk perbaikan tugas akhir ini.
4. Orang tua penulis, Dr. Ir. Teguh Hari Santoso, M.P dan Ir. Oktarina, M.P yang telah mendokan dan mendukung dalam pengerjaan tugas akhir ini
5. Saudara kandung penulis, Kharisma Nabil Santosa yang turut mendokan dan mendukung penyelesaian tugas akhir
6. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang sangat berharga bagi penulis.
7. Intan Zhofir Asyur Lazuardi yang telah mendukung dan membantu menyelesaikan penulisan ini.
8. Rekan-rekan BELTRANIS yang telah berjuang bersama dalam menjalani perkuliahan di Jurusan Sistem Informasi ITS.
9. Berbagai pihak yang membantu dalam penyusunan Tugas Akhir ini dan belum dapat disebutkan satu per satu dengan dukungan, semangat, dan kebersamaan.

Penyusunan laporan ini masih jauh dari sempurna, untuk itu saya menerima adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat pembaca.

Surabaya, Januari 2017
Penulis,

(Tesar Akram Pratama)

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
LEMBAR PERSETUJUAN.....	ix
ABSTRAK.....	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah.....	4
1.3 Batasan masalah	5
1.4 Tujuan	5
1.5 Manfaat	5
1.6 Relevansi.....	6
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian terkait.....	7
2.2 Dasar teori	9
2.2.1 ERP	9
2.2.2 <i>Free/ Open Source</i> ERP	13
2.2.3 <i>Five Tier</i> TOGAF 9.1	15
2.2.4 Kriteria Perbandingan.....	17
BAB III METODOLOGI PENELITIAN	21
3.1 Tahapan Metodologi Penelitian	21
3.2 Penjabaran Metodologi Penelitian	23
3.2.1 Studi Literatur	23
3.2.2 Perancangan Metode Penelitian.....	23
3.2.3 Perbandingan pada Aspek Arsitektur.....	24
3.2.4 Instalasi FOS ERP.....	24
3.2.5 Perbandingan pada Aspek Fungsionalitas.....	24
3.2.6 Perbandingan pada Aspek Komunitas	25
3.2.7 Analisis Hasil dan Pembahasan	25
3.2.8 Penarikan Kesimpulan dan Rekomendasi.....	25
3.3 Rangkuman Metodologi Penelitian	26
BAB IV PERANCANGAN	31
4.1 Metode Pemilihan	31

4.1.1 <i>Hit on Google Search</i>	31
4.1.2 Google Trends	33
4.2 Metode Perbandingan	37
4.2.1 Jurnal Penelitian	37
4.2.2 Situs Developer	38
4.2.3 Dokumentasi Online	38
BAB V IMPLEMENTASI	41
5.1 Aspek Arsitektur.....	41
5.1.1 Odoo.....	41
5.1.2 Openbravo	45
5.1.3 ADempiere	50
5.1.4 Dolibarr	53
5.1.5 ERPNext.....	56
5.1.6 WebERP	60
5.1.7 ERP5	63
5.1.8 Opentaps.....	65
5.1.9 Apache OFBiz	69
5.1.10 xTuple Postbooks	71
5.2 Aspek Fungsionalitas.....	74
5.2.1 Odoo.....	74
5.2.2 Openbravo	85
5.2.3 ADempiere	99
5.2.4 Dolibarr	108
5.2.5 ERPNext.....	117
5.2.6 WebERP	127
5.2.7 ERP5	135
5.2.8 Opentaps.....	142
5.2.9 Apache OFBiz	152
5.2.10 xTuple Postbooks	163
5.3 Aspek Komunitas	172
BAB VI HASIL DAN PEMBAHASAN	175
6.1 Perbandingan Aspek Arsitektur	175
6.1.1 Data Hasil Perbandingan Arsitektur	175
6.1.2 Pembahasan Perbandingan Arsitektur	178
6.2 Perbandingan Aspek Fungsionalitas	187
6.2.1 Data Hasil Perbandingan Fungsionalitas	188
6.2.2 Pembahasan Perbandingan Fungsionalitas	195

6.3 Perbandingan Aspek Komunitas.....	200
6.3.1 Data Hasil Perbandingan Komunitas	200
6.3.2 Pembahasan Perbandingan Komunitas	205
BAB VII KESIMPULAN DAN SARAN.....	209
7.1 Kesimpulan	209
7.2 Saran	210
DAFTAR PUSTAKA	211
BIODATA PENULIS	215

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 1.1 Kerangka Kerja Riset di Laboratorium Sistem Enterprise	6
Gambar 2.1 ERP Logical Architecture	12
Gambar 2.2 Two-Tier Architecture	12
Gambar 2.3 Three-Tier Architecture	13
Gambar 2.4 Five Tier Architecture.....	16
Gambar 4.1 Diagram 1 Perbandingan FOS ERP menurut Google Trends	34
Gambar 4.2 Diagram 2 Perbandingan FOS ERP menurut Google Trends	35
Gambar 4.3 Diagram 3 Perbandingan FOS ERP menurut Google Trends	35
Gambar 4.4 Diagram 4 Perbandingan FOS ERP menurut Google Trends	36
Gambar 5.1 Arsitektur Odoo	42
Gambar 5.2 Arsitektur Openbravo	46
Gambar 5.3 Komponen Modul Openbravo	48
Gambar 5.4 Arsitektur ADempiere	50
Gambar 5.5 Arsitektur Dolibarr	54
Gambar 5.6 Arsitektur ERPNext.....	58
Gambar 5.7 Arsitektur WebERP	61
Gambar 5.8 Arsitektur ERP5.....	63
Gambar 5.9 Arsitektur Opentaps	66
Gambar 5.10 Arsitektur Apache OFBiz	70
Gambar 5.11 Arsitektur xTuple Postbooks.....	72
Gambar 5.12 Tampilan Halaman Utama Odoo	74
Gambar 5.13 Contoh Tampilan Sales Odoo	75
Gambar 5.14 Contoh Tampilan Purchase Odoo	79
Gambar 5.15 Contoh Tampilan Accounting Odoo	82
Gambar 5.16 Tampilan Halaman Utama Openbravo.....	85
Gambar 5.17 Contoh Tampilan Sales Openbravo	86
Gambar 5.18 Contoh Tampilan Purchase Openbravo	91
Gambar 5.19 Contoh Tampilan Accounting Openbravo	97
Gambar 5.20 Tampilan Halaman Utama ADempiere.....	99
Gambar 5.21 Contoh Tampilan Sales ADempiere.....	100
Gambar 5.22 Contoh Tampilan Purchase ADempiere.....	104

Gambar 5.23 Contoh Tampilan Accounting ADempiere.....	107
Gambar 5.24 Tampilan Halaman Utama Dolibarr.....	109
Gambar 5.25 Contoh Tampilan Sales Dolibarr.....	109
Gambar 5.26 Contoh Tampilan Purchase Dolibarr.....	112
Gambar 5.27 Contoh Tampilan Accounting Dolibarr.....	115
Gambar 5.28 Tampilan Halaman Utama ERPNext	117
Gambar 5.29 Contoh Tampilan Sales ERPNext	117
Gambar 5.30 Contoh Tampilan Purchase ERPNext	121
Gambar 5.31 Contoh Tampilan Accounting ERPNext	124
Gambar 5.32 Tampilan Halaman Utama WebERP	127
Gambar 5.33 Contoh Tampilan Sales WebERP	127
Gambar 5.34 Contoh Tampilan Purchase WebERP	130
Gambar 5.35 Contoh Tampilan Accounting WebERP	133
Gambar 5.36 Tampilan Halaman Utama ERP5	135
Gambar 5.37 Contoh Tampilan Sales ERP5.....	135
Gambar 5.38 Contoh Tampilan Purchase ERP5	138
Gambar 5.39 Contoh Tampilan Accounting ERP5.....	141
Gambar 5.40 Tampilan Halaman Utama Opentaps	142
Gambar 5.41 Contoh Tampilan Sales Opentaps	143
Gambar 5.42 Contoh Tampilan Purchase Opentaps	147
Gambar 5.43 Contoh Tampilan Accounting Opentaps	150
Gambar 5.44 Tampilan Halaman Utama Apache OFBiz.....	152
Gambar 5.45 Contoh Tampilan Sales Apache OFBiz	153
Gambar 5.46 Contoh Tampilan Purchase Apache OFBiz....	157
Gambar 5.47 Contoh Tampilan Accounting Apache OFBiz	162
Gambar 5.48 Tampilan Halaman Utama Postbooks	164
Gambar 5.49 Contoh Tampilan Sales Postbooks.....	165
Gambar 5.50 Contoh Tampilan Purchase Postbooks.....	167
Gambar 5.51 Contoh Tampilan Accounting Postbooks.....	170
Gambar 6.1 Diagram 1 Perbandingan FOS ERP menurut Google Trends.....	204
Gambar 6.2 Diagram 2 Perbandingan FOS ERP menurut Google Trends.....	204
Gambar 6.3 Diagram 3 Perbandingan FOS ERP menurut Google Trends.....	205

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	7
Tabel 2.2 Kriteria dan Penilaian FOS ERP.....	18
Tabel 4.1 Perbandingan Hasil Peringkat FOS ERP	37
Tabel 4.2. Situs Developer FOS ERP.....	38
Tabel 4.3 Dokumentasi Online.....	38
Tabel 6.1 Data Hasil Perbandingan Arsitektur	176
Tabel 6.2 Peringkat berdasarkan Bahasa Pemrograman	182
Tabel 6.3 Peringkat berdasarkan Jenis Database	184
Tabel 6.4 Peringkat berdasarkan Platform Aplikasi	185
Tabel 6.5 Peringkat berdasarkan Sistem Operasi	186
Tabel 6.6 Perbandingan Fitur Utama pada Modul Sales.....	189
Tabel 6.7 Perbandingan Fitur Tambahan pada Modul Sales	190
Tabel 6.8 Perbandingan Fitur Utama pada Modul Purchase	191
Tabel 6.9 Perbandingan Fitur Tambahan pada Modul Purchase	192
Tabel 6.10 Perbandingan Fitur Utama pada Modul Accounting	193
Tabel 6.11 Perbandingan Fitur Tambahan pada Modul Accoun- ting	194
Tabel 6.12 Data Hasil Perbandingan Komunitas Pengguna Aktif	201
Tabel 6.13 Peringkat Perbandingan Komunitas Pengguna Aktif	206
Tabel 6.14 Peringkat Perbandingan Komunitas Pengguna Pasif	208

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

1.1 Latar belakang

Enterprise Resource Planning (ERP) merupakan *enterprise system* yang spesifik untuk mengintegrasikan antar data dan mampu mendukung seluruh fungsi departemen dalam suatu organisasi secara komprehensif [1]. ERP menjadi sistem informasi terintegrasi yang dapat menyediakan berbagai kebutuhan terkait informasi secara spesifik untuk masing-masing departemen pada suatu perusahaan. Implementasi ERP mampu menyeleraskan seluruh sistem di dalam suatu perusahaan menjadi satu sistem yang saling terintegrasi pada satu *database*. Mekanisme tersebut memudahkan beberapa departemen untuk saling berbagi data dan melakukan komunikasi [2]. Sistem ERP ini sendiri dikembangkan oleh berbagai vendor dengan karakteristik dan spesifikasinya masing-masing. Setiap perusahaan yang hendak menerapkan ERP harus mampu menyesuaikan antara kebutuhan perusahaannya dengan vendor ERP yang tepat dan daftar modul yang perlu diterapkannya. Kebutuhan mengenai sistem informasi yang terintegrasi ini telah semakin meluas manfaat dan penggunaannya. ERP tidak lagi hanya dibutuhkan oleh perusahaan-perusahaan besar, namun juga perusahaan skala menengah dan kecil. ERP telah tumbuh dari yang mulanya hanya suatu inovasi atau teknologi baru, kini menjadi

kebutuhan di berbagai kalangan perusahaan beserta tiap fungsi departemen yang dilingkupinya.

Namun, terlepas dari potensi yang ditawarkan, salah satu penghalang atau kendala utama yang dihadapi kebanyakan perusahaan dalam implementasi sistem ERP ialah pada besarnya Total Cost of Ownership (TCO)/ biaya kepemilikan. Salah satu vendor ERP yang termuka di dunia, SAP, tercatat memiliki kalkulasi biaya sebesar £100000 atau sekitar 1,8 Miliar rupiah untuk implementasi pada produk SAP Business One, produk SAP khusus perusahaan skala menengah dan kecil [3]. Produk tersebut sudah merupakan paket produk paling murah yang ditawarkan SAP untuk dapat diterapkan bagi perusahaan. Walaupun suatu perusahaan mampu menyanggupi besarnya biaya tersebut, mereka juga perlu menyesuaikan diri terhadap sumber daya yang harus dimiliki serta waktu implementasi yang panjang. Masalah tersebut semakin dipersulit dengan peluang gagalnya penerapan ERP ketika sedang dijalankan akibat ketidakcocokan dengan proses bisnis dan kurangnya dukungan dari sumber daya yang dimiliki.

Oleh sebab itu, kini telah banyak dikembangkan *Free/Open Source* (FOS) ERP oleh berbagai vendor. FOS ERP ini menawarkan sistem ERP yang tidak memerlukan biaya lisensi. FOS ERP semakin berkembang dan menjadi daya tarik tersendiri bagi beberapa perusahaan, khususnya bagi mereka yang masih beroperasi dengan ekonomi terbatas. Alasan utamanya tentu pada finansial, mengingat tidak perlunya biaya lisensi untuk implementasi. Alasan lainnya ialah fasilitas *source code* yang dibagikan dengan gratis. Hal ini memungkinkan setiap perusahaan dapat mengembangkan dan melakukan kustomisasi ERP sesuai dengan kebutuhan dan kondisinya masing-masing [4]. FOS ERP memungkinkan perusahaan untuk melakukan uji coba implementasi ERP sehingga mereka dapat memilih sistem ERP yang paling sesuai dengan kebutuhan dan proses bisnis masing-masing. Selain itu, sistem ERP memungkinkan untuk beradaptasi dengan tiap perusahaan secara lebih khusus karena fasilitas kustomisasinya. Namun

permasalahan berikutnya ialah menentukan FOS ERP yang paling tepat untuk digunakan oleh perusahaan.

Terdapat beberapa penelitian sebelumnya yang membahas perbandingan antar FOS ERP. Thomas Herzog [5] dalam tesisnya mengemukakan mengenai perbandingan antar tujuh *Open Source* ERP (*SQL Ledger, LX Office, TinyERP, GNUe, ERP5, Opentaps, Compiere*) berdasarkan 5 kriteria (*Functionality, Flexibility, Support, Contiunity, Maturity*) dengan targetnya pada perusahaan skala menengah dan kecil di Uni Eropa dimana tujuannya untuk memberikan rekomendasi FOS ERP yang paling sesuai berdasarkan fokus kriterianya masing-masing. Penelitian ini dilanjutkan oleh Vittorio Gianni Fougatsaro [6] yang pada tesisnya tidak hanya memberikan wawasan lebih mendalam mengenai macam FOS ERP, namun juga melakukan studi empiris untuk membandingkan antar FOS ERP terbaik dan memberikan rekomendasi melalui sudut pandang fungsionalitas dan tipe industri, serta ukuran organisasi. Penelitian-penelitian berikutnya lebih mengerucut ke aspek/ kriteria yang lebih khusus, seperti analisis teoritis dalam pemilihan FOS ERP [7] dan perbandingan antara SAP dengan salah satu FOS ERP terkenal, *OpenERP* [8].

Perkembangan FOS ERP ini sangat pesat. Hingga kini, telah banyak FOS ERP yang berkembang dengan arsitektur yang berbeda-beda sehingga menyebabkan penelitian terkait sebelumnya menjadi kurang valid. Selain itu, beberapa FOS ERP tersebut sudah berkembang dan berganti nama atau bahkan proyeknya telah *discontinue*. Oleh karena itu, tujuan tugas akhir ini dibuat adalah melakukan perbandingan antar FOS ERP dengan periode penelitian yang lebih baru agar dapat menentukan FOS ERP yang paling sesuai untuk digunakan oleh perusahaan di era sekarang. Dalam penelitian ini, dilakukan perbandingan tiap FOS ERP dari berbagai vendor dengan terlebih dahulu meneliti pada arsitektur penyusunnya. Arsitektur penyusun disini dinilai penting karena untuk dapat meneliti tentang ERP, maka perlu diketahui terlebih dahulu bagaimana susunan rangkaian yang menjadi dasar berjalannya sistem ERP. Melalui gambaran arsitektur yang diperoleh,

kemudian dilakukan perbandingan terhadap fungsionalitas yang dimiliki masing-masing FOS ERP. Fungsionalitas ini lebih spesifik untuk membahas mengenai modul-modul utama yang sering dibutuhkan perusahaan dalam menjalankan proses bisnisnya, yakni modul *sales*, *accounting*, dan *purchase*. Jadi dari hasil tersebut akan diketahui sejauh mana FOS ERP mampu menunjang proses bisnis yang berjalan pada perusahaan. Selain dari faktor-faktor tersebut, FOS ERP juga perlu ditinjau dari aspek eksternalnya, yakni bagaimana dukungan terhadap FOS ERP itu sendiri. Dukungan yang dimaksud adalah kontribusi dari pihak komunitas, sekumpulan *user* yang menggunakan FOS ERP yang sama dan saling berhubungan dalam satu portal. Maka dari itu, keaktifan dari pihak komunitas juga menjadi penilaian penting untuk dapat mengetahui sejauh mana para pengguna FOS ERP tersebut telah bereksplorasi dan saling bertukar informasi. Melalui berbagai penilaian tersebut, maka dapat diperoleh rekomendasi FOS ERP yang cocok dan sesuai untuk diterapkan di tingkat perusahaan. Rekomendasi FOS ERP ini juga berguna untuk menjadi objek penelitian lebih mendalam di tingkat pendidikan.

1.2 Rumusan masalah

Merujuk pada latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Apa saja alternatif FOS ERP yang populer sehingga layak untuk dibandingkan?
2. Bagaimana rekomendasi FOS ERP pada aspek arsitektur yang sesuai untuk diterapkan di tingkat perusahaan?
3. Bagaimana rekomendasi FOS ERP pada aspek fungsionalitas yang sesuai untuk diterapkan di tingkat perusahaan?
4. Bagaimana rekomendasi FOS ERP pada aspek komunitas yang sesuai untuk diterapkan di tingkat perusahaan?

1.3 Batasan masalah

Dari permasalahan yang disebutkan di atas, batasan masalah dalam tugas akhir ini adalah:

1. Perbandingan antar FOS ERP yang dilakukan adalah pada segi arsitektur, fungsionalitas (modul *sales*, *accounting*, dan *purchase*), serta keaktifan komunitas
2. Penelitian yang dilakukan hanya berfokus untuk membandingkan antar FOS ERP tanpa meninjau pada suatu objek/ pihak tertentu
3. Jumlah FOS ERP yang diteliti adalah sebanyak 10 macam
4. Penelitian tidak bertujuan memilih FOS ERP terbaik, namun berupa rekomendasi FOS ERP yang sesuai untuk diterapkan di tingkat perusahaan.

1.4 Tujuan

Berdasarkan hasil perumusan masalah dan batasan masalah yang telah disebutkan sebelumnya, maka tujuan yang dicapai dari tugas akhir ini adalah sebagai berikut:

1. Mengetahui macam alternatif FOS ERP yang populer sehingga layak untuk dibandingkan
2. Memberikan rekomendasi FOS ERP yang sesuai untuk diterapkan di tingkat perusahaan pada aspek arsitektur
3. Memberikan rekomendasi FOS ERP yang sesuai untuk diterapkan di tingkat perusahaan pada aspek fungsionalitas
4. Memberikan rekomendasi FOS ERP yang sesuai untuk diterapkan di tingkat perusahaan pada aspek komunitas

1.5 Manfaat

Manfaat yang diharapkan dapat diperoleh dari tugas akhir ini adalah sebagai berikut:

Bagi Perusahaan

1. Memfasilitasi perusahaan yang hendak menerapkan FOS ERP agar dapat mengetahui perbandingan masing-masing FOS ERP

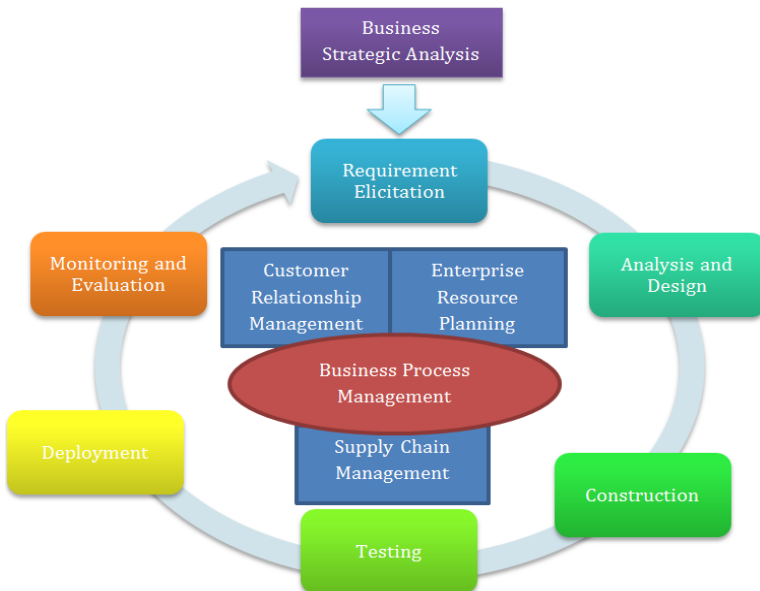
2. Memfasilitasi perusahaan yang hendak menerapkan FOS ERP agar dapat mengetahui FOS ERP yang paling sesuai untuk diimplementasikan

Bagi Institusi Pendidikan

Memberikan rekomendasi FOS ERP yang paling sesuai untuk menjadi bahan pembelajaran dan penelitian

1.6 Relevansi

Tugas akhir ini berkaitan dengan mata kuliah Perencanaan Sumber Daya Perusahaan dan Integrasi Aplikasi Korporasi. Tugas akhir ini termasuk dalam penelitian di bidang Laboratorium Sistem Enterprise, dimana pengerjaannya mengikuti kerangka kerja riset di Laboratorium Sistem Enterprise yang dijelaskan pada gambar 1.1 seperti di bawah ini:



Gambar 1.1 Kerangka Kerja Riset di Laboratorium Sistem Enterprise

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir.

2.1 Penelitian terkait

Penelitian yang dijadikan acuan dalam pengerjaan tugas akhir ini terdapat pada Tabel 2.1

Tabel 2.1 Penelitian Terkait

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
<i>A Comparison of Open Source ERP Systems</i>	Menentukan FOS ERP Package dan kriteria pembandingan, meneliti tiap FOS ERP berdasarkan tiap kriteria, melakukan survei pada perusahaan menengah dan kecil, merekomendasikan pilihan ERP berdasarkan hasil perbandingan	Thomas Herzog	<ul style="list-style-type: none"> • Perbandingan 7 <i>Open Source ERP (SQL Ledger, LX Office, TinyERP, GNUe, ERP5, Opentaps, Compiere)</i> berdasarkan 5 kriteria (<i>Functionality, Flexibility, Support, Contiunity, Maturity</i>) • Deskripsi masing-masing <i>Open Source ERP</i> disertai keuntungan dan kerugian yang dimiliki • Hasil perbandingan tiap kriteria untuk masing-masing <i>Open Source ERP</i> • Rekomendasi pilihan ERP sesuai fokus kriterianya masing-masing

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
<i>A Study of Open Source ERP Systems</i>	<i>Literature Review</i> , identifikasi kebutuhan organisasi, menentukan model evaluasi sistem ERP	Vittorio Gianni Fougats aro	<ul style="list-style-type: none"> • Pilihan <i>Open Source</i> ERP dengan keunggulan dan kelemahan yang dimiliki • Studi Empiris mengenai fitur dan dimensi tiap <i>Open Source</i> ERP • Rekomendasi <i>Open Source</i> ERP yang dibedakan berdasarkan sudut pandang fungsionalitas dan tipe industri, serta ukuran organisasi
<i>A Theoretical Analysis Of Key Points when Choosing Open Source ERP Systems</i>	<i>Literature Review</i> , model kualitatif dan <i>explanatory</i> , karakterisasi dan analisis kriteria perbandingan <i>Open Source</i> ERP	Fernando Gustavo dos Santos Gripe dan Ildeberto Aparecido Rodello	<ul style="list-style-type: none"> • Kriteria penting dalam membandingkan dan menentukan antar <i>Open Source</i> ERP <i>Systems</i> • Analisis dan manfaat penggunaan <i>Open Source</i> ERP
<i>OpenERP evaluation with SAP as reference</i>	<i>Information gathering</i> , hasil pengalaman implementasi, perbandingan dengan nilai SAP, penarikan kesimpulan	Yves Delsart dan Christelle Van Nieuwenhuysen	<ul style="list-style-type: none"> • Dimensi penting sebagai acuan perbandingan antar ERP • Hasil perbandingan <i>OpenERP</i> dengan SAP di tiap dimensi • Tingkat <i>maturity</i> <i>OpenERP</i> sebagai

Judul Penelitian	Metode	Penulis	Hasil yang Didapatkan
			solusi implementasi ERP
<i>A Comparative Analysis and Evaluation of Open Source ERP Systems</i>	<i>Goal selection, Alternatives, Criteria, Sub-criteria, Hierarchy, Preferences, Calculation of weights, Consistency test</i>	Saleh M. Al-Saleem	<ul style="list-style-type: none"> • Prioritas dan bobot masing-masing kriteria (<i>Functionality, Flexibility, Support, Continuity, Maturity</i>) dalam penilaian <i>Open Source ERP</i> • Perbandingan <i>Open Source ERP (SQL Ledger, TinyERP, ERP5)</i> berdasarkan tiap kriteria • Pilihan <i>Open Source ERP</i> terbaik yang mencakup seluruh kriteria

2.2 Dasar teori

Pada bab ini dijelaskan mengenai teori – teori yang digunakan dalam pengerjaan tugas akhir ini, meliputi ERP, arsitektur ERP dan *software*, modul ERP, FOS ERP, komunitas FOS ERP, dan kriteria pembandingan antar FOS ERP.

2.2.1 ERP

Menurut Motiwalla & Thompson [1], *Enterprise Resource Planning (ERP)* merupakan bagian *enterprise system* yang spesifik berperan sebagai sistem informasi yang dapat mengintegrasikan dan mengotomatisasi proses bisnis suatu perusahaan/ organisasi melalui serangkaian data dan informasi dari masing-masing departemen/ divisi. ERP mampu mengakomodasi berbagai kebutuhan sistem informasi secara spesifik untuk departemen-departemen yang berbeda pada suatu perusahaan. Melalui kemampuan tersebut, maka ERP

dapat memberikan manfaat berupa efektifitas dan ketepatan proses bisnis serta efisiensi pada sumber daya dan pengeluaran biaya. Penggunaan ERP menjadikan semua sistem di dalam suatu perusahaan menjadi satu sistem yang terintegrasi dengan satu *database* sehingga beberapa departemen menjadi lebih mudah dalam berbagi data, dan lebih mudah pula dalam melakukan komunikasi.

Setiap ERP memiliki struktur mengenai arsitektur penyusunnya. Arsitektur implementasi ERP merupakan *blueprint* dari sistem ERP yang sebenarnya dan menerjemahkan strategi implementasi *high-level* ERP ke dalam aliran informasi yang saling terhubung antar organisasi [1]. Arsitektur ERP membantu memudahkan perancangan sistem ERP dalam organisasi. Terdapat beberapa macam arsitektur sistem ERP yang telah berkembang di era sekarang ini, yakni antara lain:

1. Logical Architecture

Arsitektur ini lebih berfokus pada dukungan terhadap kebutuhan *end users*. *Logical Architecture* terdiri atas *database schemas* berupa entitas dan *relationships* pada *lowest tier* atau tingkat pertama. Kemudian diikuti dengan *core business processes* dan *business logic* pada *second tier* atau tingkat kedua. Tingkat teratas atau *third tier* berupa detail aplikasi yang mendukung berbagai fungsi bisnis di dalam sistem ERP itu sendiri. *End users* disini tidak berinteraksi dengan pada *first* dan *second tier*. Mereka hanya perlu mengakses pada *third tier*, yakni aplikasi *client-user interface*, dimana ia menyediakan *end users* berupa akses aplikasi fungsional yang meliputi setiap fitur dan modul yang dicakup oleh sistem ERP tersebut. Gambaran lebih jelas terkait struktur *Logical Architecture* pada sistem ERP dapat dilihat pada gambar 2.1.

2. Physical Architecture

Arsitektur ini lebih berfokus pada efisiensi sumber daya sistem. Sumber daya yang dimaksud adalah seperti biaya, waktu respon, jumlah perangkat, dan lain sebagainya. Arsitektur semacam ini lebih memungkinkan sistem secara keseluruhan untuk lebih *scalable* dan mengurangi sumber

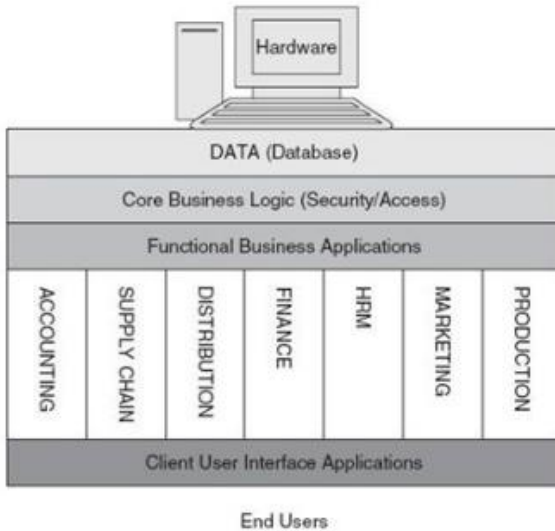
daya yang dibutuhkan. *Physical Architecture* terbagi atas 2 macam, yakni *two-tier architecture* dan *three-tier architecture*. Pada *three-tier architecture*, sistem ERP terbagi atas 3 layer/ tingkatan, yakni *data tier*, *application tier*, dan *presentation/ Web tier*. Sedangkan pada *two-tier architecture*, sistem hanya terbagi atas 2 tingkatan, yakni *data/ application tier* (gabungan antara *data tier* dan *application tier*) dan *presentation/ Web tier*. Gambar 2.2 menampilkan gambaran mengenai *two-tier architecture* dan gambar 2.3 adalah gambaran *three-tier architecture*. Pada tingkatan terbawah, *data tier*, berfokus pada struktur seluruh data organisasional dan hubungannya dengan sistem internal dan eksternal. *Data tier* bertanggungjawab pada manajemen data, dimana ia menyediakan penyimpanan utama untuk seluruh data yang dibagikan antar modul-modul fungsional dan menjaga integritas data yang ditransfer ke dan dari *clients* maupun *servers*. *Application tier* merupakan tingkatan dimana data dimasukkan dan dibagikan antar komponen sistem. Pada tingkat ini mencakup komponen untuk menerapkan *business logic* pada modul-modul fungsionalitas. Jadi *application tier* menjadi jembatan antara *database* pada *data tier* dengan *client applications* pada *Web tier*. Sedangkan *presentation/ Web tier* memungkinkan pengguna untuk mengakses dan menganalisis informasi melalui *desktop application* atau *Web browser*. *Presentation/ Web tier* mencakup *servers* sebagai portal untuk *client/ pengguna* berinteraksi pada aplikasi. Tingkatan ini mencakup aplikasi *graphical user interface* (GUI) guna melakukan *data input*, *information request*, serta *data presented*.

ERP terdiri dari bermacam – macam modul yang disediakan untuk berbagai kebutuhan dalam suatu perusahaan [9]. Modul tersebut terbagi atas 3 jenis, yakni antara lain:

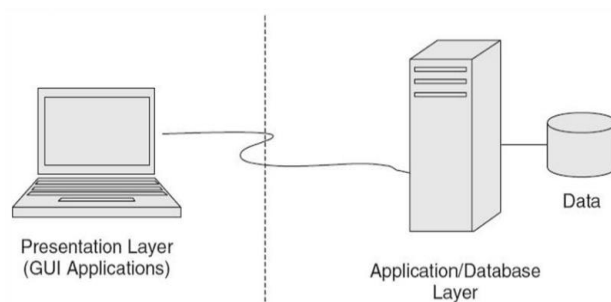
1 Modul Operasi

Modul operasi memiliki beberapa modul bawaan yang menjelaskan mengenai proses bisnis utama dari organisasi atau perusahaan yang secara umum terdiri atas *Sales and*

Distribution, Materials Management, dan Production Planning and Control. Beberapa modul lain merupakan modul pelengkap yang lebih mendetail seperti *General Logistics, Logistics Execution, Quality Management, Plant Maintenance, Customer Service, Project System, dan Environment Management*



Gambar 2.1 ERP Logical Architecture

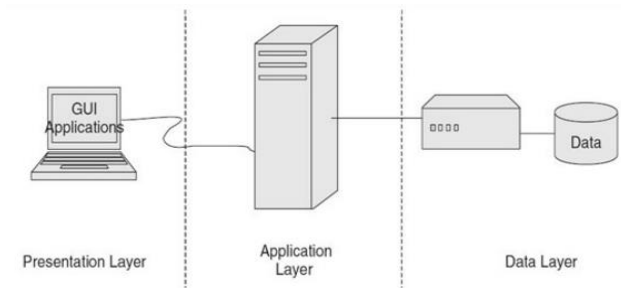


Gambar 2.2 Two-Tier Architecture

2 Modul Finansial dan Akunting

Modul finansial dan akuntansi berfokus pada pencatatan keuangan perusahaan dengan beberapa modul bawaaannya

yakni *Financial Accounting, Controlling, Investment Management, General Accounting, Treasury, Enterprise Controlling*



Gambar 2.3 Three-Tier Architecture

3 Modul Sumber Daya Manusia

Sedangkan modul sumber daya manusia membicarakan dan mendefinisikan tentang manajemen SDM organisasi yang terdiri atas modul *Personnel Management, Personnel Time Management, Payroll, serta Training and Event Management, Organizational Management, dan Travel Management*

2.2.2 Free/ Open Source ERP

Free/ Open Source ERP (FOS ERP) merupakan sistem ERP yang dirancang dan disediakan tanpa harus membutuhkan biaya lisensi. Sistem ERP semacam ini dapat diunduh, dipasang, dan digunakan secara gratis [5]. Selain itu, FOS ERP ini juga mempublikasikan kode program pembangunnya sehingga memungkinkan berbagai organisasi yang menerapkannya dapat mengembangkan lebih lanjut serta melakukan kustomisasi sesuai dengan kebutuhan organisasinya masing-masing. Keuntungan dan kemudahan tersebut menjadi daya tarik FOS ERP sendiri dan sekaligus alasan organisasi lebih memilih FOS ERP daripada ERP yang berbayar. Hingga kini telah terdapat banyak vendor dan developer yang merancang dan mengembangkan FOS ERP. Hal ini menyebabkan jumlah FOS ERP semakin meningkat dan beragam. Sudah banyak juga FOS

ERP yang terkenal dan digunakan di kalangan organisasi. Namun kenyataannya, setiap FOS ERP ini tidak sepenuhnya gratis. Pengembangan dengan fungsionalitas yang lebih kompleks dan cakupan organisasi lebih luas membutuhkan biaya untuk pembaruan dan peralihan versi ERP.

FOS ERP ini sendiri tumbuh dan berkembang tidak hanya karena usaha dari vendor atau tim developer sistemnya saja, namun juga kontribusi dari pihak komunitas. Komunitas disini merupakan orang-orang yang menggunakan suatu FOS ERP yang sama dan kemudian saling terhubung antar satu sama lain dalam suatu portal. Portal yang mempersatukan antar pengguna FOS ERP tersebut dapat berupa forum diskusi yang telah disediakan vendor FOS ERP dalam situsnya sendiri, maupun forum dalam situs lain atau *social media*. Keberadaan komunitas tersebut tidak hanya sekedar untuk menunjukkan eksistensi FOS ERP di mata pengguna, namun juga wadah untuk saling bertukar informasi seputar FOS ERP itu sendiri. Informasi tersebut dapat berupa pertanyaan, keluhan, masalah, kritik dan saran, jawaban, maupun solusi terkait penggunaan FOS ERP. Oleh sebab itu, keberadaan komunitas juga dinilai penting untuk turut memajukan pengembangan FOS ERP. Komunitas yang baik dapat ditinjau dari 5 indikator, yakni:

1. Jumlah anggota
2. Jumlah *knowledge shared*
3. Tanggal pos terakhir
4. Tanggal rilis terakhir
5. Rata-rata waktu rilis

Jumlah anggota menunjukkan jumlah pengguna yang menggunakan FOS ERP tersebut dan merasa nyaman di dalamnya sehingga mereka bersedia untuk bergabung dan berdedikasi terhadap kemajuan FOS ERP. Jumlah *knowledge shared* merupakan jumlah informasi dalam bentuk apapun yang telah dipublikasikan dan diperbincangkan pada portal komunitas FOS ERP tersebut. *Knowledge shared* adalah informasi-informasi yang berguna dan penting mengenai FOS ERP itu sendiri dan menjadi pengetahuan di mata developer dan pengguna. Tanggal pos terakhir adalah tanggal suatu pos/

thread yang akan menjadi *knowledge shared* tersebut diposkan dalam forum komunitas. Tanggal rilis terakhir ialah tanggal sistem aplikasi dari FOS ERP untuk versi terbarunya dikeluarkan. Sedangkan rata-rata waktu rilis adalah rata-rata jarak waktu antar rilis/ keluarnya tiap versi dari aplikasi FOS ERP.

Komunitas yang mampu mencapai kelima indikator tersebut secara maksimal maka dikatakan sebagai komunitas yang aktif. Sehingga keaktifan komunitas menjadi salah satu pengukuran dalam menilai seberapa baik suatu FOS ERP terkait kesesuaiannya untuk diterapkan di tingkat perusahaan.

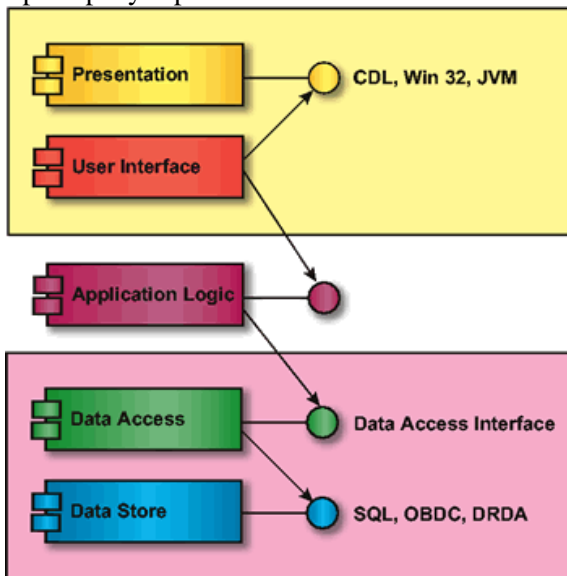
Selain dari 5 indikator tersebut, FOS ERP dapat pula ditinjau dari sisi pengguna yang tidak tergabung dalam komunitas. Pengguna FOS ERP tidak hanya mencakup komunitasnya saja, tapi juga para pengguna pasif dan awam. Pengguna semacam ini menggunakan dan mencari informasi terkait FOS ERP tidak terbatas pada portal komunitas resminya saja, namun juga berbagai sumber informasi yang memperbincangkan FOS ERP tersebut. Para pengguna semacam ini dapat diukur tingkat keaktifannya dari jumlah pencarian dan pembicaraan FOS ERP di dunia *internet*. Perbincangan FOS ERP dari berbagai portal, situs, media sosial, maupun lingkungan lainnya di *internet*, akan terekam oleh *search engine*. Selanjutnya hasil pencarian informasi-informasi tersebut dari *search engine*, dapat dikalkulasi dan dianalisis untuk mengetahui seberapa banyak FOS ERP tersebut diperbincangkan. Semakin banyak perbincangan seputar FOS ERP tersebut, maka semakin banyak pula pengguna pasif dan awamnya di lingkungan *internet*. FOS ERP dengan jumlah pengguna yang banyak dapat mengindikasikan bahwa ia banyak diimplementasikan sehingga layak digunakan oleh perusahaan.

2.2.3 *Five Tier* TOGAF 9.1

Dalam melakukan perbandingan antar FOS ERP pada aspek arsitektur, dibutuhkan acuan struktur arsitektur yang lebih kompleks dari arsitektur 3 *tier* sistem ERP yang dijelaskan

oleh Motiwalla. Acuan yang lebih mendalam adalah pada *5 tier software architecture* [10]

Dalam mencapai fleksibilitas yang maksimal, *software* harus memanfaatkan arsitektur *5 tier*, dimana merupakan perluasan dari arsitektur *3 tier*. Arsitektur ini bertujuan untuk memberikan pemisahan yang jelas dari tiga bidang fungsional utama dari arsitektur (gambar 2.4). Hal ini dikarenakan adanya komponen *client* dan server, baik pada *user interface* (UI), maupun pada penyimpanan data.



Gambar 2.4 Five Tier Architecture

Tiap tingkatan pada arsitektur *3 tier*, didetailkan komponennya lebih mendalam. Pada *presentation tier*, dibedakan menjadi 2 macam komponen, yakni *presentation* dan *user interface*. *Presentation* ini lebih berfokus hanya pada tampilan UI, dimana *user* dapat berinteraksi (memasukkan *input* dan memperoleh *output*) dengan sistem aplikasi melalui bagian ini. Tingkat ini lebih umum disebut sebagai *client tier*. Sedangkan tingkat/ komponen *user interface* bertujuan untuk menampilkan, menyembunyikan, dan mengubah tiap

fungsionalitas yang perlu disajikan pada UI untuk diketahui oleh *user*. Tingkat ini lebih dikenal sebagai *presentation tier*.

Tingkat kedua yang menjadi penghubung antara *presentation* dan *data tier* adalah *application tier*. Tingkat ini sebenarnya dibedakan menjadi 2 komponen, yakni *service* dan *application logic*. *Service* ini bertujuan untuk pemilihan jalur lintasan jaringan agar *client tier* dapat memperoleh akses terhadap data sistem. Ia juga membuat dan mengelola *session* untuk tiap *request* yang diteruskan pada *application logic*. Sedangkan *application logic* mengelola berbagai logika dan fungsionalitas bisnis. Tiap *request* yang telah diteruskan oleh *service* akan diterjemahkan untuk mengetahui modul, fitur, dan data apa saja yang dibutuhkan untuk memenuhi *request* tersebut.

Tingkat ketiga sendiri, yakni *data tier*, dibagi menjadi *data access* dan *data store*. Tiap kebutuhan data yang telah didefinisikan oleh *application logic* akan dikelola lebih lanjut oleh *data access*. Ia yang bertanggung jawab dalam melakukan akses data ke dalam *database* di *data store*. Komunikasi yang dilakukan untuk mengambil data ini adalah melalui *query*. Tiap *query* tersebut yang digunakan untuk mengambil data dari *data store*. *Data store* sendiri tugasnya hanya sebagai repositori seluruh data sistem. Berbagai data sistem tersimpan di dalam *data store* dalam bentuk *database*.

Jadi pada arsitektur 5 *tier* ini, terdapat 6 komponen utama yang menjadi acuan perbandingan FOS ERP dalam aspek arsitektur. Komponen tersebut adalah *client*, *presentation*, *service*, *application logic*, *data access*, dan *data store*.

2.2.4 Kriteria Pembandingan

Kriteria pembandingan merupakan kriteria yang dijadikan sebagai perspektif atau sudut pandang dalam penilaian FOS ERP. Tiap kriteria pembandingan mencakup beberapa poin penilaian FOS ERP. Jadi, nilai akhir untuk tiap kriteria pembandingan diperoleh melalui gabungan hasil nilai dari seluruh penilaian di kriteria tersebut. Berikut tabel 2.2 menjelaskan

uraian mengenai kriteria beserta masing-masing penilaiannya yang diperoleh dari penelitian-penelitian sebelumnya:

Tabel 2.2 Kriteria dan Penilaian FOS ERP

Kriteria	Penilaian	Sumber Penelitian
Functional Fit	Mencakup modul dasar dalam proses bisnis (<i>Sales, Purchase/ Material, Production</i>)	[8], [11]
	Modul dan manajemen terkait <i>inventory</i> dan <i>warehouse</i>	[5], [6], [8]
	Kemampuan manajemen terkait CRM dan SCM	[8]
	Manajemen terkait akuntansi dan detail pencatatan transaksi	[5], [6], [12]
	Menyediakan fitur analitis untuk <i>Business Intelligence</i>	[6], [8]
Technical Fit	Kemudahan prasyarat dan proses instalasi	[13], [14], [8]
	Kemampuan berjalan secara <i>multiplatform</i>	[5], [8], [7]
	Kemampuan manajemen <i>multi users</i> dan <i>multi roles</i>	[5], [6], [8], [7]
	Dukungan terhadap berbagai database (<i>multi-database</i>)	[5], [8], [7]
	Kebutuhan terhadap dukungan <i>environment</i> pada perangkat untuk mendukung sistem	[5], [7]
Ergonomics and Ease of Use	Kemudahan dalam pemahaman <i>User Interface</i>	[5], [6], [8], [7]
	Tampilan <i>User Interface</i> yang fleksibel dan saling terintegrasi pada tiap platform	[5], [6], [8], [7]
	Kemampuan dalam meningkatkan <i>user-adoption</i>	[6]
	Ketersediaan <i>user guide</i> dan dokumentasi penggunaan sistem/aplikasi	[8]
	Pemenuhan prinsip <i>Human-Computer Interaction</i> (HCI)	[15]

Kriteria	Penilaian	Sumber Penelitian
Flexibility and Customization	Kemampuan beradaptasi pada berbagai bidang perusahaan	[14], [16]
	Kesesuaian terhadap proses bisnis perusahaan secara khusus	[16], [11], [17]
	Ketersediaan fitur pengembangan dan kustomisasi sistem	[5], [6], [8], [7]
	Ketersediaan dan kemudahan dalam mengedit <i>source code</i> pemrograman	[5], [6], [8], [7]
	Fleksibilitas terhadap pertumbuhan perusahaan dan perubahan arus pasar	[5], [6], [7]
Support and Continuity	Adanya <i>Online</i> dan <i>Offline Help and Support</i> dari pihak developer	[5], [6], [7]
	Dukungan dari pihak komunitas	[5], [6], [7]
	Dokumentasi terkait informasi dan solusi seputar sistem/ aplikasi	[5], [7]
	Jumlah penelitian terkait sistem dan perusahaan yang berhasil implementasi	[5], [7]
	Frekuensi <i>update</i> sistem/ aplikasi	[5], [8], [7]

Pada penelitian ini, kriteria pembandingan yang digunakan sesuai dengan tujuan awal penelitian yaitu pada arsitektur (berfokus pada struktur, kelengkapan, dan gambaran), fungsionalitas (khususnya modul *sales*, *accounting*, dan *purchase*), dan keaktifan komunitas (ditinjau dari jumlah anggota dan jumlah *knowledge shared*). Jadi apabila dihubungkan dengan kriteria-kriteria pada penelitian sebelumnya, penelitian ini lebih berfokus pada kriteria *functional fit* serta *flexibility and customization*.

(Halaman ini sengaja dikosongkan)

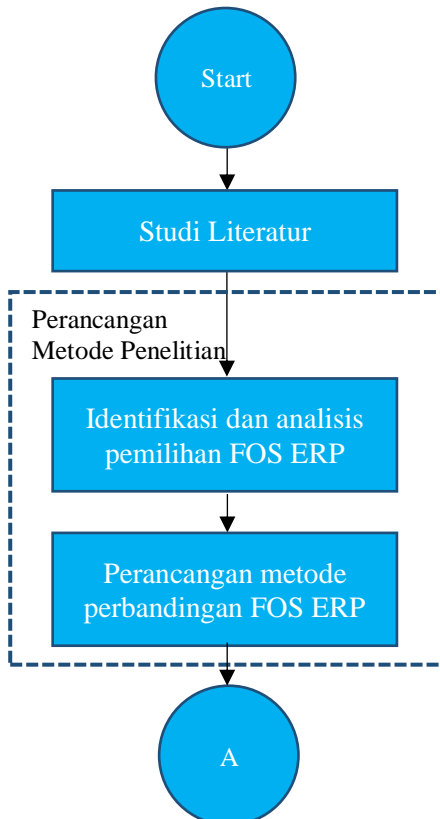
BAB III

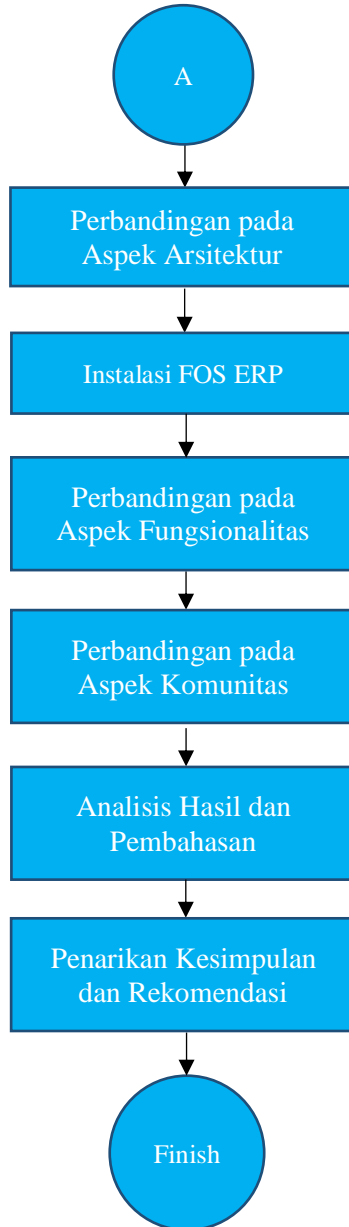
METODOLOGI PENELITIAN

Pada bab metode penelitian akan dijelaskan mengenai tahapan – tahapan apa saja yang dilakukan dalam pengerjaan tugas akhir ini beserta deskripsi dan penjelasan tiap tahapan tersebut. Lalu disertakan jadwal pengerjaan tiap tahapan.

3.1 Tahapan Metodologi Penelitian

Pada sub bab ini akan menjelaskan mengenai metodologi dalam pelaksanaan tugas akhir. Metodologi ini dapat dilihat pada gambar 3.1





3.2 Penjabaran Metodologi Penelitian

Penjabaran terkait metodologi penelitian berisi tahapan-tahapan yang dilaksanakan dalam penelitian tugas akhir ini

3.2.1 Studi Literatur

Tahapan awal ialah studi literatur mengenai *Enterprise Resource Planning (ERP)* dan *Free/ Open Source (FOS) ERP*. Pembelajaran ini berasal dari berbagai *paper*, jurnal, dan *e-book* yang membahas tentang ERP dan/ atau FOS ERP. Hasil dari studi literatur ini ialah pemahaman mengenai definisi dan teori terkait ERP dan FOS ERP, definisi mengenai arsitektur ERP, modul yang dicakup dalam ERP, serta pembahasan mengenai komunitas dalam FOS ERP.

3.2.2 Perancangan Metode Penelitian

Dalam melakukan perancangan penelitian, ditentukan metode untuk penelitian FOS ERP. Metode ini terbagi atas 2 tahapan

3.2.2.1 Metode Pemilihan FOS ERP

Metode ini bertujuan untuk menentukan FOS ERP yang hendak diteliti dan digunakan dalam tugas akhir ini. Aspek perbandingan FOS ERP adalah berdasarkan arsitektur penyusun, fungsionalitas (mencakup modul *sales*, *accounting*, dan *purchase*), dan keaktifan dari komunitas. FOS ERP yang dipilih dibatasi untuk sepuluh macam dan merupakan FOS ERP yang baik, layak, dan telah banyak digunakan oleh perusahaan. *Input* yang diperoleh adalah berupa teori terkait ERP dan FOS ERP serta pemilihan FOS ERP melalui 2 macam cara, yakni *hit on Google Search* dan analisis *Google Trends*. Melalui seluruh *input* tersebut baru selanjutnya dapat dijalankan identifikasi dan analisis FOS ERP paling baik dan sesuai untuk diteliti. Hasil dari tahapan ini berupa sepuluh FOS ERP yang terpilih untuk kemudian dijadikan bahan penelitian dan perbandingan

3.2.2.2 Metode Perbandingan FOS ERP

Kesepuluh FOS ERP yang telah terpilih berikutnya akan diteliti untuk dibandingkan. Oleh sebab itu diperlukan metode dalam perbandingan antar FOS ERP tersebut. Selain daftar FOS ERP

yang terpilih, *input* juga berasal dari deskripsi dan *review* tiap FOS ERP serta pendaftaran berbagai sumber penelitian perbandingan FOS ERP. Jadi hasil *output* yang diperoleh berupa daftar sumber penelitian yang digunakan untuk perbandingan FOS ERP

3.2.3 Perbandingan pada Aspek Arsitektur

Tahapan selanjutnya adalah pada implementasi perbandingan FOS ERP. Implementasi perbandingan yang pertama adalah pada aspek arsitektur. Jadi *input* yang diperlukan dalam tahapan ini adalah definisi terkait arsitektur ERP yang diperoleh dari tahap studi literatur, serta berbagai dokumentasi terkait arsitektur tiap FOS ERP yang diteliti. Hasil implementasi perbandingan FOS ERP pada aspek arsitektur ini adalah berupa gambaran arsitektur masing-masing FOS ERP beserta penjabaran penjelasan tiap bagian dalam arsitektur tersebut

3.2.4 Instalasi FOS ERP

Berikutnya adalah tahapan dalam instalasi FOS ERP. Kesepuluh FOS ERP yang telah terpilih kemudian dipasang ke dalam perangkat *laptop* atau PC. Sebelum instalasi ini, diperlukan pengunduhan *installer* untuk masing-masing FOS ERP yang digunakan. Selain itu, proses instalasi untuk setiap FOS ERP tentu berbeda. Masing-masing FOS ERP memerlukan kriteria dan persyaratannya tersendiri untuk dapat terpasang ke dalam perangkat. Instalasi ini diperlukan guna menjalankan uji coba FOS ERP untuk perbandingan pada aspek fungsionalitas. Hasil yang diperoleh tentu berupa FOS ERP yang telah terpasang dalam perangkat

3.2.5 Perbandingan pada Aspek Fungsionalitas

Setelah instalasi selesai dan seluruh FOS ERP telah terpasang dalam perangkat, maka dilakukan implementasi perbandingan FOS ERP pada aspek fungsionalitas. Implementasi ini juga memerlukan *input* dari tahap studi literatur terkait modul yang dicakup ERP. Uji coba FOS ERP ini difokuskan pada 3 modul utama dalam ERP, yakni *sales*, *purchase*, dan *accounting*. Uji

coba ini meliputi setiap fitur yang dimiliki masing-masing FOS ERP pada ketiga modul tersebut. Jadi, tahapan ini merupakan tahapan yang menyangkut proses operasional penggunaan FOS ERP pada faktor internal. Seluruh hasil uji coba tersebut kemudian dijabarkan dan dijelaskan sejumlah modul dan FOS ERP yang diteliti guna menghasilkan *output* dari tahapan ini.

3.2.6 Perbandingan pada Aspek Komunitas

Implementasi perbandingan FOS ERP yang terakhir adalah pada aspek komunitas. *Input* tahapan ini diperoleh dari penjabaran dan penjelasan situs developer dan komunitas FOS ERP pada tahap studi literatur serta situs/ portal yang digunakan tiap FOS ERP. Implementasi dalam perbandingan di aspek ini adalah melalui dengan cara membandingkan komunitas FOS ERP, baik pada pengguna aktif dan pengguna pasif. Jadi, tahapan ini merupakan tahapan mengenai faktor eksternal pada tiap FOS ERP. Hasil yang diperoleh dari tahapan ini adalah perbandingan komunitas tiap FOS ERP untuk masing-masing jenis pengguna dengan pengukurannya masing-masing.

3.2.7 Analisis Hasil dan Pembahasan

Setelah seluruh aspek perbandingan FOS ERP diimplementasikan, maka tahapan berikutnya adalah melakukan analisis terhadap hasil datanya dan menjabarkan pembahasan analisis tersebut. Dalam tahapan ini, seluruh hasil implementasi perbandingan kesepuluh FOS ERP akan digabungkan dan dibedakan berdasarkan macam aspeknya, yakni arsitektur, fungsionalitas, dan komunitas. Data hasil penggabungan perbandingan tersebut yang menjadi dasar dalam analisis dan pembahasan perbandingan FOS ERP. Oleh sebab itu, *output* yang diperoleh juga merupakan hasil pembahasan terkait perbandingan FOS ERP di tiap aspek

3.2.8 Penarikan Kesimpulan dan Rekomendasi

Tahapan terakhir dalam tugas akhir ini ialah penarikan kesimpulan dan rekomendasi. Dalam pelaksanaan tahapan ini, diperlukan hasil pembahasan FOS ERP di tiap aspek, dimana

telah dijalankan di tahap sebelumnya. Melalui masukan tersebut, maka kemudian disimpulkan pilihan FOS ERP yang sesuai bagi perusahaan untuk tiap aspeknya. Pilihan FOS ERP ini bukan berupa FOS ERP yang paling tepat untuk seluruh aspeknya. Namun pilihannya disajikan dalam bentuk rekomendasi FOS ERP dengan pertimbangannya masing-masing di tiap aspek. Jadi, rekomendasi yang diberikan tidak selalu menunjukkan 1 atau beberapa FOS ERP untuk tiap aspek.

3.3 Rangkuman Metodologi Penelitian

Rangkuman metodologi berisikan metodologi yang dilakukan dalam penelitian ini, dimulai dari rangkaian aktivitas, tujuan, *input*, *output*, dan metode yang digunakan seperti yang terdapat pada tabel 3.

Tabel 3.1 Rangkuman Metodologi Penelitian

Aktivitas	Tujuan	<i>Input</i>	<i>Output</i>	Metode
Studi Literatur ERP dan FOS ERP	Mempelajari definisi ERP dan FOS ERP beserta aspek arsitektur, fungsionalitas, dan komunitas yang terkait di dalamnya	Literatur <i>paper</i> , jurnal, dan <i>e-book</i> terkait ERP dan FOS ERP	<ul style="list-style-type: none"> • Teori dan pemahaman dalam ERP dan FOS ERP • Definisi arsitektur ERP • Modul dalam ERP • Penjelasan komunitas FOS ERP 	Pembelajaran
Identifikasi dan analisis pemilihan FOS ERP	Mengidentifikasi dan memilih 10 FOS ERP untuk penelitian	<ul style="list-style-type: none"> • Teori dan pemahaman dalam ERP dan FOS ERP • Pemilihan FOS ERP melalui <i>hit on Google Search</i> • Pemilihan FOS ERP melalui analisis <i>Google Trends</i> 	FOS ERP yang terpilih untuk penelitian	Observasi dan analisis
Perancangan metode perbandingan FOS ERP	Menentukan metode perbandingan antar FOS ERP	<ul style="list-style-type: none"> • FOS ERP yang terpilih untuk penelitian • Deskripsi dan <i>review</i> tiap FOS ERP • Pendaftaran sumber penelitian perbandingan 	Sumber penelitian perbandingan FOS ERP	Pembelajaran dan observasi

Aktivitas	Tujuan	<i>Input</i>	<i>Output</i>	Metode
Implementasi perbandingan FOS ERP pada aspek arsitektur	Merancang gambaran dan penjelasan arsitektur tiap FOS ERP	<ul style="list-style-type: none"> • Definisi arsitektur ERP • Dokumentasi arsitektur tiap FOS ERP 	<ul style="list-style-type: none"> • Gambaran arsitektur FOS ERP • Penjabaran penjelasan tiap bagian arsitektur FOS ERP 	Perancangan dan pembelajaran
Instalasi FOS ERP	Memasang FOS ERP ke dalam perangkat	<ul style="list-style-type: none"> • FOS ERP yang terpilih untuk penelitian • <i>Installer</i> dan kebutuhan instalasi tiap FOS ERP 	Penjabaran penjelasan tiap bagian FOS ERP pada modul <i>sales</i> , <i>purchase</i> , dan <i>accounting</i>	Instalasi aplikasi
Implementasi perbandingan FOS ERP pada aspek fungsionalitas	Menjelaskan fungsionalitas tiap FOS ERP pada modul <i>sales</i> , <i>purchase</i> , dan <i>accounting</i>	<ul style="list-style-type: none"> • Modul dalam ERP • FOS ERP terpasang dalam perangkat 	Penjabaran penjelasan tiap bagian FOS ERP pada modul <i>sales</i> , <i>purchase</i> , dan <i>accounting</i>	Uji coba

Aktivitas	Tujuan	<i>Input</i>	<i>Output</i>	Metode
Implementasi perbandingan FOS ERP pada aspek komunitas	Membandingkan komunitas FOS ERP, baik pada pengguna aktif maupun pasif	<ul style="list-style-type: none"> • Penjelasan komunitas FOS ERP • Forum/ portal komunitas tiap FOS ERP 	<ul style="list-style-type: none"> • Jumlah <i>user</i> aktif tiap FOS ERP • Jumlah <i>knowledge shared</i> tiap FOS ERP 	Observasi
Analisis hasil perbandingan FOS ERP	Membandingkan dan menilai tiap FOS ERP berdasarkan seluruh aspek perbandingan	Data hasil perbandingan FOS ERP pada tiap aspek	Hasil pembahasan FOS ERP pada tiap aspek	Perbandingan dan penilaian
Penentuan rekomendasi FOS ERP yang sesuai bagi perusahaan	Menentukan rekomendasi FOS ERP yang sesuai bagi perusahaan untuk tiap aspek yang dibandingkan	Hasil pembahasan FOS ERP pada tiap aspek	Kesimpulan dan rekomendasi FOS ERP sesuai bagi perusahaan	Rangkuman

(Halaman ini sengaja dikosongkan)

BAB IV

PERANCANGAN

Pada bab ini dijelaskan perancangan awal yang diperlukan sebelum melakukan penelitian tugas akhir. Bab ini mencakup metode-metode dalam persiapan penelitian tugas akhir disertai penjelasannya

4.1 Metode Pemilihan

Metode pemilihan merupakan cara dalam mencari berbagai FOS ERP yang berpotensi menjadi objek dalam penelitian tugas akhir. Seluruh FOS ERP yang terdaftar nantinya harus dipilih dan dieliminasi untuk menjadi 10 macam FOS ERP. Dalam metode ini terdapat 2 macam cara untuk mendapatkan 10 macam FOS ERP yang paling layak sebagai objek penelitian. Cara tersebut ialah menghitung jumlah *hit* FOS ERP pada Google Search Engine dan menganalisis melalui Google Trends

4.1.1 *Hit on Google Search*

Cara ini ditempuh dengan cara pencarian macam FOS ERP melalui dokumentasi *online* yang disediakan oleh situs-situs direktori aplikasi. Situs direktori yang digunakan sebagai acuan dalam mencari macam-macam aplikasi sistem FOS ERP adalah sebagai berikut:

- a) <https://sourceforge.net/>
- b) <https://www.getapp.com/>
- c) <http://www.softwareinsider.com/>
- d) <https://www.g2crowd.com/>
- e) <https://www.cloudswave.com/>
- f) <http://www.capterra.com/>
- g) <http://www.axzm.com/>

Melalui seluruh FOS ERP yang telah didaftarkan tersebut, maka diperiksa mengenai keaktifannya hingga kini. Terdapat 14 FOS ERP yang masih aktif hingga kini dan dapat digunakan secara umum oleh *end user*. Berikut daftar yang diperoleh:

- a) ADempiere
- b) Odoo
- c) Openbravo
- d) ERPNext
- e) Opentaps
- f) Dolibarr
- g) WebERP
- h) xTuple Postbooks
- i) Vienna Advantage
- j) CK-ERP
- k) Apache OFBiz
- l) LedgerSMB
- m) MixERP
- n) ERP5

Seluruh daftar hasil FOS ERP tersebut perlu dieliminasi menjadi 10 FOS ERP. Cara pertama adalah dengan menghitung jumlah *hit* FOS ERP pada Google Search Engine. Cara penghitungannya adalah dengan melihat jumlah hasil pencarian dari Google Search Engine terkait *keyword* pada tiap FOS ERP. Berikut jumlah hasil pencarian tiap FOS ERP pada Google Search Engine:

- a) ADempiere : ± 2.570.000
- b) Odoo : ± 11.400.000
- c) Openbravo : ± 3.920.000
- d) ERPNext : ± 136.000
- e) Opentaps : ± 79.600
- f) Dolibarr : ± 579.000
- g) WebERP : ± 273.000
- h) xTuple Postbooks : ± 508.000
- i) Vienna Advantage : ± 21.500
- j) CK-ERP : ± 6.750
- k) Apache OFBiz : ± 121.000
- l) LedgerSMB : ± 31.500
- m) MixERP : ± 73.400
- n) ERP5 : ± 706.000

Hasil dari jumlah pencarian FOS ERP pada Google Search Engine ini diperingkatkan dari hasil pencarian terbanyak hingga

terendah. Berikut hasil peringkat FOS ERP berdasarkan jumlah pencarian pada Google Search Engine:

1. Odoo
2. Openbravo
3. ADempiere
4. ERP5
5. Dolibarr
6. xTuple Postbooks
7. WebERP
8. ERPNext
9. Apache OFBiz
10. Opentaps
11. MixERP
12. LedgerSMB
13. Vienna Advantage
14. CK-ERP

Dari hasil peringkat empat belas FOS ERP tersebut, maka 10 FOS ERP yang dipilih melalui cara ini ialah Odoo, ADempiere, Openbravo, ERP5, Dolibarr, xTuple Postbooks, WebERP, ERPNext, Apache OFBiz, dan Opentaps. Sedangkan FOS ERP yang dieliminasi atau dibuang adalah MixERP, LedgerSMB, Vienna Advantage, dan CK-ERP.

4.1.2 Google Trends

Analisis melalui Google Trends ialah cara dalam membandingkan jumlah pencarian tiap *keyword* di Google Search Engine terkait FOS ERP yang hendak diteliti dengan penggunaan berbagai pengukuran dalam Google Trends. Pengukuran yang digunakan antara lain adalah rentang waktu, area pencarian, metode penelusuran, dan kategori pencarian. Cara ini ditempuh untuk memperkuat cara pertama yang telah dilakukan sebelumnya. Karena perbandingan dalam Google Trends hanya dibatasi untuk 5 macam *keyword* saja, maka 14 FOS ERP ini dibedakan berdasarkan beberapa kategori yang disesuaikan dari hasil peringkat pada cara pertama. Berikut detail untuk masing-masing pengukuran yang digunakan:

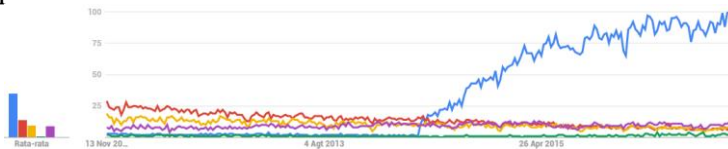
- a) Area pencarian : Seluruh dunia

- b) Rentang waktu : 5 tahun terakhir (2011-2016)
- c) Kategori pencarian : Semua kategori
- d) Metode penelusuran : Penelusuran *web*

Kategori FOS ERP yang pertama dibandingkan ialah pada 5 FOS ERP yang memiliki jumlah *hit* pada Google Search terbanyak, yakni:

1. Odoo : bertanda biru
2. Openbravo : bertanda merah
3. ADempiere : bertanda kuning
4. ERP5 : bertanda hijau
5. Dolibarr : bertanda ungu

Berikut gambar 4.1 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori pertama



Gambar 4.1 Diagram 1 Perbandingan FOS ERP menurut Google Trends

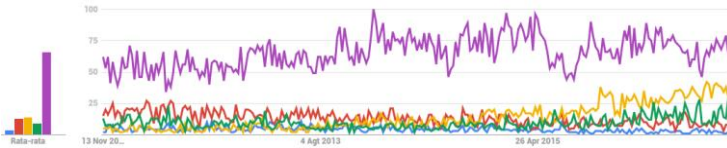
Dari hasil perbandingan tersebut, diambil 3 FOS ERP dengan rata-rata perbandingan tertinggi. Jadi FOS ERP yang terpilih adalah:

1. Odoo
2. Openbravo
3. ADempiere

FOS ERP yang tidak terpilih akan dibandingkan dengan daftar FOS ERP berikutnya. Jadi, kategori FOS ERP kedua yang dibandingkan adalah:

1. ERP5 : bertanda hijau
2. Dolibarr : bertanda ungu
3. xTuple Postbooks : bertanda biru
4. WebERP : bertanda merah
5. ERPNext : bertanda kuning

Berikut gambar 4.2 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori kedua



Gambar 4.2 Diagram 2 Perbandingan FOS ERP menurut Google Trends

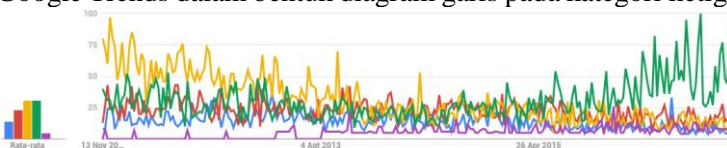
Dari hasil perbandingan tersebut, 3 FOS ERP yang terpilih adalah:

1. Dolibarr
2. ERPNext
3. WebERP

FOS ERP yang tidak terpilih akan dibandingkan dengan daftar FOS ERP berikutnya. Jadi, kategori FOS ERP ketiga yang dibandingkan adalah:

1. ERP5 : bertanda hijau
2. xTuple Postbooks : bertanda biru
3. Apache OFBiz : bertanda merah
4. Opentaps : bertanda kuning
5. MixERP : bertanda ungu

Berikut gambar 4.3 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori ketiga



Gambar 4.3 Diagram 3 Perbandingan FOS ERP menurut Google Trends

Dari hasil perbandingan tersebut, 3 FOS ERP yang terpilih adalah:

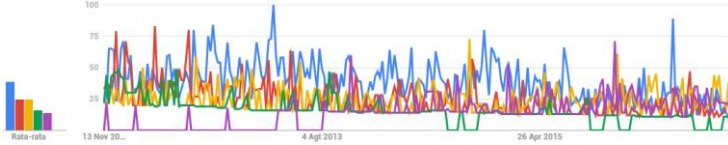
1. ERP5
2. Opentaps
3. Apache OFBiz

FOS ERP yang tidak terpilih akan dibandingkan dengan daftar FOS ERP berikutnya. Jadi, kategori FOS ERP keempat yang dibandingkan adalah:

1. xTuple Postbooks : bertanda biru

2. MixERP : bertanda ungu
3. LedgerSMB : bertanda merah
4. Vienna Advantage : bertanda kuning
5. CK-ERP : bertanda hijau

Berikut gambar 4.4 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori keempat



Gambar 4.4 Diagram 4 Perbandingan FOS ERP menurut Google Trends

Karena telah terdapat 9 FOS ERP yang terpilih (tiap kategori 3 FOS ERP), maka pada perbandingan terakhir ini hanya diambil 1 FOS ERP. Selain untuk menggenapkan jumlah FOS ERP menjadi 10, pemilihan hanya 1 FOS ERP ini juga karena perbandingan keempat ini ialah perbandingan FOS ERP yang paling sedikit jumlah *hit* pada Google Search dan rata-rata penelusurannya melalui Google Trends. Jadi 1 FOS ERP terakhir yang terpilih adalah xTuple Postbooks. Berikut hasil peringkat FOS ERP berdasarkan rata-rata penelusuran pada Google Trends:

1. Odoo
2. Openbravo
3. ADempiere
4. Dolibarr
5. ERPNext
6. WebERP
7. ERP5
8. Opentaps
9. Apache OFBiz
10. xTuple Postbooks

Hasil peringkat pada cara kedua dibandingkan dengan hasil peringkat pada cara pertama. Berikut tabel 4.1 yang memaparkan perbandingan hasil peringkat pada cara pertama dan kedua

Tabel 4.1 Perbandingan Hasil Peringkat FOS ERP

Peringkat	Metode Pemilihan	
	Hit on Google Search	Google Trends
1	Odoo	Odoo
2	Openbravo	Openbravo
3	ADempiere	ADempiere
4	ERP5	Dolibarr
5	Dolibarr	ERPNext
6	xTuple Postbooks	WebERP
7	WebERP	ERP5
8	ERPNext	Opentaps
9	Apache OFBiz	Apache OFBiz
10	Opentaps	xTuple Postbooks
11	MixERP	LedgerSMB
12	LedgerSMB	Vienna Advantage
13	Vienna Advantage	CK-ERP
14	CK-ERP	MixERP

Melalui perbandingan tersebut, dapat disimpulkan bahwa daftar 10 FOS ERP teratas pada cara pertama dan kedua adalah sama. FOS ERP yang tereliminasi pada cara pertama dan kedua tetap sama, yakni MixERP, Ledger SMB, Vienna Advantage, dan CK-ERP. Jadi 10 FOS ERP yang terpilih adalah sesuai dengan hasil peringkat pada kedua cara pemilihan

4.2 Metode Perbandingan

Metode yang dilakukan dalam perbandingan FOS ERP di penelitian ini mengacu pada 3 sumber, yakni sebagai berikut:

4.2.1 Jurnal Penelitian

Sumber ini diperoleh dari penelitian terdahulu yang membahas dan membandingkan FOS ERP, baik berupa buku, *e-book*, maupun *paper*. Jurnal penelitian yang digunakan adalah penelitian resmi yang hasilnya valid dan dapat dipertanggungjawabkan. Daftar jurnal penelitian yang dalam tugas akhir ini sudah dijelaskan sebelumnya pada tabel 2.1 mengenai penelitian terkait dalam tinjauan pustaka.

4.2.2 Situs Developer

Sumber ini didapat dalam bentuk pedoman dan bantuan yang diberikan vendor/ developer FOS ERP terkait. Pedoman dan bantuan tersebut tentunya yang dapat dipahami dan digunakan oleh orang awam/ calon pengguna FOS ERP. Berikut tabel 4.2 yang menjelaskan mengenai *link* situs developer untuk masing-masing FOS ERP:

Tabel 4.2. Situs Developer FOS ERP

No	Nama FOS ERP	Link Situs Developer
1	Odoo	https://www.odoo.com/
2	Openbravo	http://www.openbravo.com/
3	ADempiere	http://adempiere.net/web/guest/welcome
4	Dolibarr	https://www.dolibarr.org/
5	ERPNext	https://erpnext.com/
6	WebERP	http://www.weberp.org/
7	ERP5	https://www.erp5.com/
8	Opentaps	http://www.opentaps.org/
9	Apache OFBiz	https://ofbiz.apache.org/
10	xTuple Postbooks	https://xtuple.com/products/postbooks

4.2.3 Dokumentasi *Online*

Sumber ini ialah berupa pembahasan perbandingan FOS ERP yang dijelaskan di situs-situs web. Situs web yang digunakan dalam perbandingan FOS ERP adalah situs direktori dan perbandingan aplikasi yang sudah terpercaya dan memiliki jumlah *user* yang banyak. Berikut tabel 4.3 menjelaskan terkait daftar situs web yang menjadi sumber dokumentasi *online* dalam membandingkan antar FOS ERP:

Tabel 4.3 Dokumentasi Online

No	Nama Situs Web	Link Situs Web
1	GetApp	https://www.getapp.com
2	Software Insider	http://www.softwareinsider.com/
3	G2Crowd	https://www.g2crowd.com/
4	Capterra	http://www.capterra.com/

No	Nama Situs Web	Link Situs Web
5	Cloudswave	https://www.cloudswave.com
6	AXZM	http://www.axzm.com/

Ketiga sumber tersebut (jurnal penelitian, situs developer, dan dokumentasi *online*) merupakan dasar metode utama yang digunakan dalam penelitian perbandingan FOS ERP ini.

(Halaman ini sengaja dikosongkan)

BAB V

IMPLEMENTASI

Pada bab implementasi ini dijelaskan hasil implementasi yang telah dilakukan dalam tugas akhir terkait perbandingan FOS ERP pada tiap aspek yang diteliti, yakni arsitektur, fungsionalitas (modul *sales*, *purchase*, dan *accounting*), serta komunitas

5.1 Aspek Arsitektur

Pada aspek ini dijelaskan implementasi perbandingan FOS ERP berupa gambaran arsitektur masing-masing FOS ERP beserta penjabaran penjelasan tiap bagian dalam arsitektur tersebut

5.1.1 Odoo

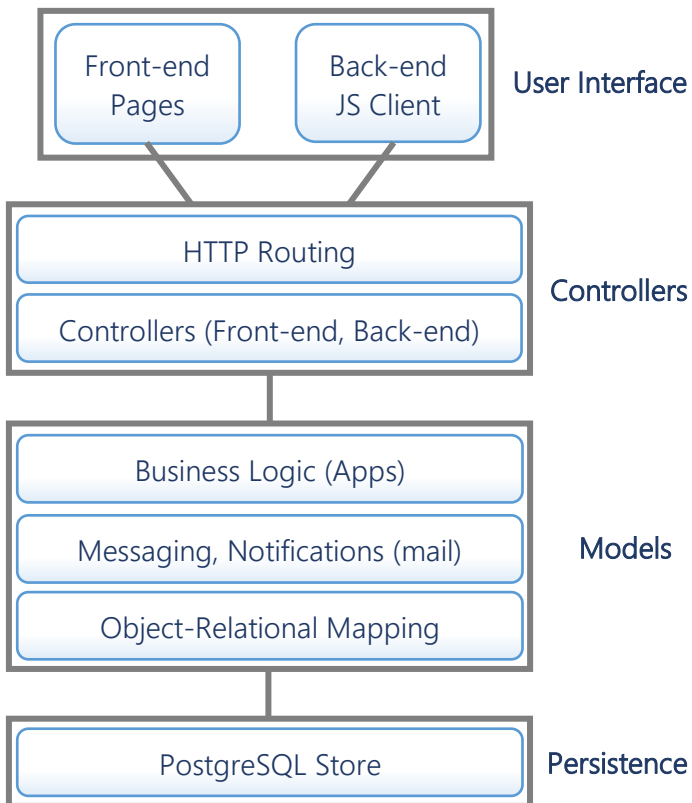
Pada gambar 5.1 menjelaskan mengenai arsitektur Odoo. Odoo mempunyai arsitektur *4-tier*. Arsitektur yang dibangun ini berawal dari MVC (*Model-View-Control*) architecture. MVC architecture adalah bentuk arsitektur yang digunakan dalam *software engineering*, dimana ia memisahkan antara data penyimpan (*model*), akses data dan *business logic* (*control*), dan tampilan *user interface* (*view*) [18].

Namun seiring dengan kebutuhannya dan perkembangan *software engineering*, Odoo menyendirikan tingkat untuk bagian database pada *layer persistence*. Jadi *layer* untuk arsitektur Odoo adalah sebagai berikut:

1. User Interface

Layer yang menampilkan *user interface* (UI) Odoo pada *user*. *User interface* ini terbagi atas 2 macam. **Front-end pages** merupakan halaman depan Odoo yang digunakan *user* untuk melihat, mengubah, atau memasukkan input pada tampilan Odoo. Odoo memiliki 3 macam tampilan *front-end*, yakni, *web application*, *online application*, dan *mobile application*. Seluruh tampilan *front-end* tersebut dapat saling terintegrasi dalam 1 sistem aplikasi Odoo.

Sedangkan pada **Back-end** atau halaman belakang untuk menampilkan atau mengubah fungsionalitasnya adalah menggunakan JS (JavaScript) Client. Penggunaan JavaScript ini adalah karena sifatnya yang fleksibel terhadap berbagai platform sehingga mudah diaplikasikan pada Odoo yang mengusung *multi-platform*, dan juga bersifat dinamis sehingga memudahkan dalam perubahan atau penyisipan kode *script* dan pengembangan kustomisasi tampilan.



Gambar 5.1 Arsitektur Odoo

2. Controllers

Controllers merupakan *layer* yang berfungsi untuk menghubungkan komunikasi antara *user interface* dengan *model*. *Controllers* berguna dalam menerjemahkan keinginan input *user* yang dimasukkan melalui UI, dan meminta akses data pada *model* agar dapat menyediakan data yang perlu untuk ditampilkan *user interface*. *Controllers* Odoo berawal dari **HTTP Routing** yang berguna dalam proses pemilihan jalur lintasan jaringan via HTTP. Dalam menjalankan fungsi tersebut digunakan XML-RPC (*eXtensible Markup Language – Remote Procedure Call*), yakni protokol *prosedur call* (pemanggilan prosedur) secara *remote* terkait *request* data yang bekerja di lingkungan internet [19]. Jadi melalui XML-RPC dua sistem yang terpisah dan berbeda platform serta lingkungan (*user interface* menggunakan JavaScript dan *model* menggunakan python) bisa saling berkomunikasi lewat sarana file XML. XML adalah bahasa pemrograman untuk proses *encoding* atau menerjemahkan *request user* dari UI untuk meminta akses data pada *model* dan mengambil data pada *persistence*. HTTP disini menjadi media untuk *transport* data yang diminta dari UI dalam bentuk JavaScript dan pada akhirnya diberikan kembali sesuai *request* (permintaan) UI. Selanjutnya, *request* data tersebut dilanjutkan pada bagian **Controllers** yang menangani sekaligus membedakan permintaan untuk *front-end* dan *back-end page*, yakni WSGI (*Web Server Gateway Interface*). WSGI merupakan *interface* universal yang menghubungkan antara *web servers* dan *web applications* pada bahasa Python [20]. Jadi, *request* data yang sebelumnya berformat Javascript, kini telah diterjemahkan dan disesuaikan dengan bahasa Python melalui WSGI sehingga dapat diteruskan pada *layer* berikutnya, *Model*.

3. Models

Models adalah *layer* yang fungsi utamanya adalah untuk pemodelan data sesuai *request* data yang diterima untuk mengambil data-data yang dibutuhkan pada *database* di

layer persistence. Tahapan awalnya adalah pada **Business logic** yang berfungsi untuk menentukan data-data apa saja yang dibutuhkan untuk memenuhi *request* data serta jenis modul dan fitur mana yang disesuaikan untuk menampilkan data tersebut. Kemudian tahapan berikutnya adalah menentukan skema **Messaging and Notification**. Tahapan ini ialah pemberitahuan pada *layer* bawah terkait akses permintaan data dan pada *layer* atas terkait hasil pengambilan data. Terakhir ialah pada komponen penting dalam komunikasi terhadap *database* yakni **ORM** (*Object Relational Mapping*). ORM menyediakan fungsionalitas dan *interface* guna berkomunikasi dari *application server* ke *database server*. ORM melakukan *building query* terkait data-data yang dibutuhkan dari *database*, mengelola tabel dan detail datanya yang telah diperoleh dari *database*, serta berbagai sifat RDBMS (*Relational Database Management System*) yang harus dipenuhi data.

4. Persistence

Layer terakhir adalah *persistence*, yakni *layer* yang hanya berisi dan menyimpan data-data yang tetap. *Layer* ini lebih berfungsi sebagai repositori untuk menampung seluruh data-data dalam sistem, tanpa melakukan modifikasi atau manipulasi data. Oleh karena itu, isi dalam *layer* ini hanyalah *database* utama yang digunakan oleh Odoo, yaitu **PostgreSQL**. PostgreSQL *database* menyimpan seluruh data sistem, data aplikasi, dan berbagai elemen konfigurasi sistem. Berbagai *query* yang digunakan untuk mengakses dan mengambil data dilakukan melalui PostgreSQL. Dalam versi terbaru Odoo 10, *database* yang disediakan melalui PostgreSQL sudah mendukung penyimpanan secara *cloud service* sehingga *database server* tidak lagi terbatas pada satu perangkat saja.

Bahasa : JavaScript, Python

Database : PostgreSQL

OS : Linux, Mac OS X, Unix, Windows, Android, iOS

5.1.2 Openbravo

Gambar 5.2 merupakan penjelasan arsitektur Openbravo. Openbravo memiliki arsitektur 3-tier, yakni terdiri atas *User Interface*, *Application Server*, dan *Database* [21]. Berikut penjelasan untuk masing-masing *tier* dalam arsitektur tersebut:

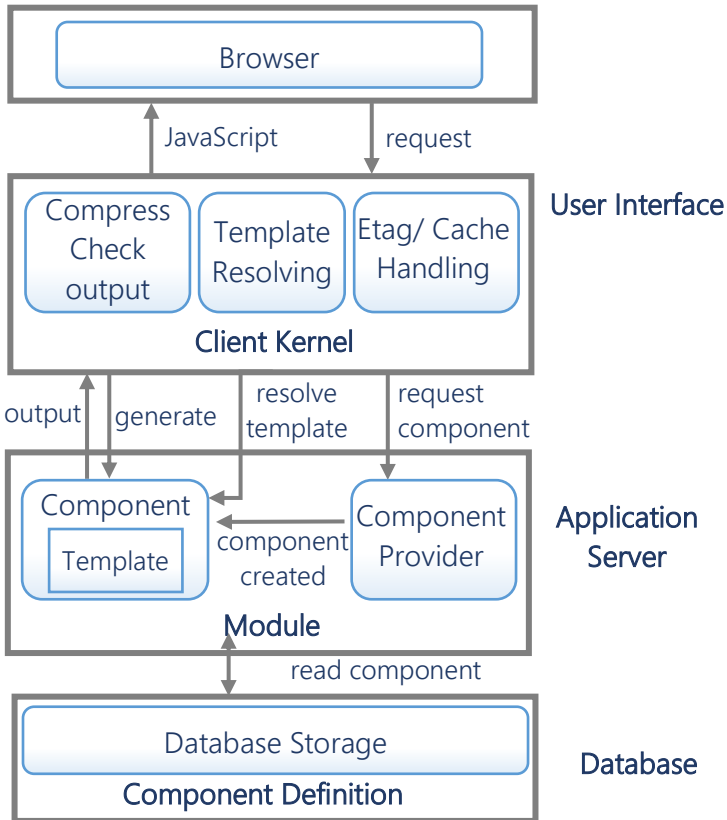
1. User Interface

Openbravo memiliki dua tingkatan dalam pemrosesan UI. Untuk berinteraksi langsung dengan *user*, ia menggunakan **Browser** sebagai media guna menampilkan UI. UI yang ditampilkan ini menggunakan bahasa pemrograman JavaScript. Fungsi dari *browser* adalah untuk memasukkan *input* berupa *request* Component yang diinginkan *user* serta menampilkan hasil *output* tersebut pada *user*. Setiap kali *request* yang diberikan akan diteruskan pada tingkatan berikutnya, yakni **Client Kernel**. Client Kernel bertugas mencari modul yang bertanggung jawab dalam menangani *request* Component. Dalam Client Kernel ini terdapat 3 komponen yakni antara lain:

a. Etag/ Cache Handling

Etag merupakan identifikasi khusus yang diberikan oleh *server* pada *database* kepada obyek yang akan disampaikan ke pada client/ UI. Obyek data dalam Openbravo ini dinamakan Component. Prinsipnya bahwa untuk setiap Component di server, memiliki Etag yang berbeda-beda dan berubah setiap kali Component di dalam server tersebut dimodifikasi. Jadi apabila terdapat suatu *request* Component yang dikirimkan dari *browser*, maka *request* tersebut akan diteruskan pada Client Kernel. Client Kernel tersebut akan memeriksa pada *database server*. Apabila suatu *request* tersebut memiliki Component dengan Etag yang sama, artinya Component tersebut tidak mengalami perubahan dari *request* sebelumnya sehingga Client Kernel dapat mengirimkan Component yang telah tersimpan pada cache pada *browser*. Namun jika Etag tersebut berbeda, maka Component telah mengalami modifikasi pada *database*

server sehingga diperlukan pengambilan Component dari *server* sebelum ditampilkan pada Browser.



Gambar 5.2 Arsitektur Openbravo

b. Template Resolving

Template resolving ini merupakan area yang berfungsi sebagai penyedia Template dalam memenuhi kebutuhan Component. Tiap Component yang berada di bagian Module pada Application Server, membutuhkan Template untuk dapat direpresentasikan pada bagian Client/ UI. Oleh sebab itu, tugas dari Template Resolving menyimpan kumpulan Template tersebut untuk dapat

disalurkan pada bagian Module dan ‘membungkus’ Component yang diberikan. Tampilan yang diberikan oleh Component dapat dibentuk dengan menggunakan Template atau dieksekusi menggunakan Javacode untuk menghasilkan JavaScript yang ditampilkan pada Browser

c. Compress Check Output

Pada bagian **Compress Check Output** ini, Client Kernel memproses hasil/ *output* berupa dua hal, yakni memeriksa *syntax* dari hasil dan memkompresnya. Hasil disini maksudnya berupa JavaScript yang dihasilkan oleh Template Resolving. Pemeriksaan *syntax* bertujuan untuk menghindari adanya kesalahan atau *error* kode pada saat halaman JavaScript tersebut ditampilkan. Sedangkan pengompresan bertujuan untuk menurunkan *size* halaman sehingga pemrosesan halaman untuk ditampilkan dapat jauh lebih cepat. Proses pemeriksaan *syntax* ini dilakukan oleh Jslint, sedangkan proses pengompresannya menggunakan JSmin.

2. Application Server

Terdapat tiga komponen penting yang terdapat dalam *Application Server* ini, antara lain yakni:

a. Component

Seluruh *request* yang dikirimkan dari Browser diterjemahkan di bagian *Application Server*. *Request* tersebut diimplementasikan dalam bentuk **Component**. Component ini yang menjalankan penggunaan *navigation bar*, *grids*, *form*, *selector*, maupun *layout* secara keseluruhan. Component ini diterapkan dalam modul. Modul inilah yang memiliki kebebasan menyeluruh terkait bagaimana menjalankan component. Tiap Component yang hendak diteruskan ke bagian Client Kernel dikemas dalam Template agar mempermudah dan mempercepat pemrosesan hasilnya untuk ditampilkan pada Browser. Apabila Template tersebut tersedia di bagian Template Resolving, maka Template akan diberikan pada bagian Component untuk pengemasan data. Namun jika tidak terdapat Template, maka tampilan

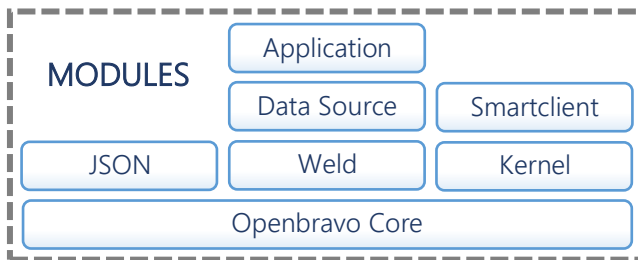
akan dieksekusi menggunakan Javacode untuk menghasilkan JavaScript. Hasil dari Component baru selanjutnya diteruskan ke bagian Compress Check Output

b. Component Provider

Component di dalam modul dikelola oleh **Component Provider**, dimana tugasnya yaitu bertanggung jawab untuk membuat Component dan menyediakannya pada bagian Client Kernel. Selain itu ia juga berfungsi untuk mencatat dan menjaga konten tampilan yang statis dari modul kepada Client Kernel. Setiap modul perlu mengimplementasikan Component Provider.

c. Module

Module merupakan bagian yang melingkupi dan mengontrol Component dan Component Provider. Module membaca definisi Component yang diinginkan dari *request user* melalui tabel-tabel atau sumber lain yang tersimpan dalam *database*. Pada gambar 5.3, dipaparkan beberapa Module yang diterapkan dalam Openbravo yakni:



Gambar 5.3 Komponen Modul Openbravo

- JSON. Module yang menyediakan JSON REST web service guna melakukan komunikasi data antara *client* dengan *server*
- Weld. Module yang menyediakan *dependency injection* (konsep dalam memberikan/ menginjeksi suatu kelas di dalam *source code* ke kelas yang lain yang merupakan dependensinya/ kelas yang membutuhkannya) dan manajemen/ pengelolaan terhadap Component

- Kernel. Modul yang mengurus tugas infrastruktur sebagai pemrosesan *request*, *event handling*, serta tindakan kompresi dan *caching*
- DataSource. Modul yang menggunakan modul JSON dan menyediakan fungsionalitas *request* data pada tingkat yang lebih tinggi (*higher level*) untuk membangun *query* dan tindakan terkait data
- SmartClient. Modul yang menyediakan *library* untuk UI pada SmartClient (Area implementasi RIA/ Rich Internet Application Framework untuk menggabungkan kelebihan-kelebihan dari aplikasi berbasis web dan desktop dalam menyajikan UI visual yang cantik dan interaktif
- Application. Modul yang berisi implementasi terkait *navigation bar*, *grid and forms*, serta aplikasi lain yang berorientasi *client* dan *server*

Keseluruhan modul ini merupakan Openbravo Core yakni otak pemrosesan utama dalam sistem Openbravo.

3. Database

Tier terakhir adalah **Database**. Pada *database* ini tersimpan berbagai Component beserta definisinya, seperti tabel, kolom, baris, dan detail isi. Openbravo lebih memilih menyebut tiap datanya sebagai suatu Component. Component di dalam *database* ini yang nantinya akan diakses oleh *Application server*, tepatnya pada bagian Modul untuk dibaca dan diambil bagian Component yang dibutuhkan untuk dapat memenuhi *request* yang dikirimkan oleh *user*. Jadi, fungsi utama *database* ini hanya untuk menyimpan dan menampung seluruh Component dalam *database server*. *Database* yang dapat didukung oleh sistem Openbravo lebih beragam jenisnya, yakni Apache Derby, MySQL, PostgreSQL, Oracle dan HSQLDB (Hyper SQL Database). Tiap jenis *database* tersebut memiliki metode instalasinya masing-masing dengan disesuaikan pada sistem operasi yang digunakan *user*.

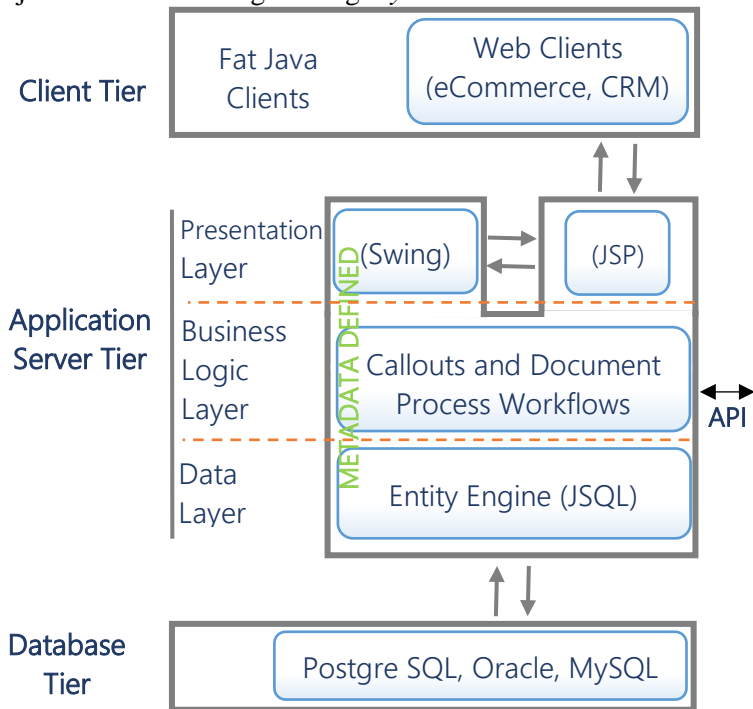
Bahasa : Java, JavaScript

Database : Apache Derby, MySQL, PostgreSQL, Oracle, HSQLDB

OS : Linux, Mac OS X, Unix, Windows

5.1.3 ADempiere

Gambar 5.4 menjelaskan mengenai arsitektur ADempiere. FOS ERP ini terdiri atas arsitektur 3-tier yakni terdiri atas *Client Tier*, *Application Server Tier*, dan *Database Tier* [22]. Berikut penjelasan untuk masing-masing *layer/ tier* dalam arsitektur:



Gambar 5.4 Arsitektur ADempiere

1. Client Tier

Pada *layer Client* yang digunakan untuk interaksi aplikasi dengan *user*, ADempiere dapat diakses, baik melalui *web application*, *desktop application*, maupun *mobile application*. Sisi Client diakses menggunakan **Web Client**. Web Client ini merupakan UI yang bertujuan untuk menampilkan tiap fitur (seperti contohnya eCommerce dan CRM) dalam bentuk aplikasi *web* dengan memanfaatkan media berupa *web browser*. ADempiere menerapkan sistem **Java fat clients** karena ia menyediakan banyak fungsionalitas atau fitur-fitur terkait implementasi ERP dan CRM. Maka dengan kata lain, ia menerapkan sistem *thick client-thin server* sehingga ia menampung dan mengelola banyak client dalam pemrosesannya serta server hanya untuk pengelolaan data.

2. Application Server Tier

Application Server merupakan *server* yang menjadi penghubung komunikasi antara *Client Tier* dan *Database Tier*. *Server* ini dibagi kembali menjadi 3 bagian *layer*, yakni sebagai berikut:

- a. *Presentation layer* untuk menyajikan format data sesuai kebutuhan *user* dan meneruskannya pada Web Client untuk ditampilkan.

Pada *Presentation layer*, terdapat **Java Swing** dan **JSP** (*Java Server Pages*). Java Swing adalah *librari* Java yang digunakan untuk membangun *Graphical User Interface* (GUI) dengan platform Java. Java Swing memungkinkan rancang bangun UI secara *cross platform* (platform yang berbeda) atau *multi platform*. Java Swing ini berkomunikasi langsung dengan JSP. JSP adalah pemrograman *web* dengan bahasa Java yang juga berjalan pada platform Java. Melalui bagian ini, dilakukan pemrograman untuk mengontrol presentasi isi atau tampilan *web* dengan bahasa Java yang nantinya akan ditampilkan pada Web Client. JSP ini memungkinkan halaman *web* dapat dibangun secara dinamis berdasarkan HTML atau XML. Jadi Java Swing berperan dalam menyajikan tampilan dalam bentuk GUI Java, sedangkan

JSP menyajikannya dalam bentuk tampilan *web* untuk diteruskan pada Web Client.

- b. *Business Logic layer* untuk menerjemahkan *request* data dari user dan memahami bagian-bagian data mana yang perlu diakses dan yang perlu disembunyikan.

Dalam *layer* ini dilakukan tahap **Callouts and Document Process Workflow**. Tahapan ini adalah alur pemrosesan yang mencakup dua hal, yakni dokumen dan pemanggilan fungsi. Jadi berbagai *request* data yang diperoleh dari *layer* sebelumnya akan memicu pemanggilan fungsi-fungsi Java terkait berbagai modul, fitur, dan data yang perlu ditampilkan sesuai dengan *request* yang diminta. Alur pemrosesan ini juga mencakup dokumen yang dibutuhkan untuk memenuhi *request* tersebut. Pada *layer* ini pun turut berhubungan langsung dengan API (*Application Programming Interface*) yang dihubungkan dengan sistem aplikasi ADempiere. Terdapat beberapa API yang dapat dihubungkan dengan ADempiere. API ini biasanya juga digunakan pada *data layer* guna mengakses data yang tersimpan dalam *database*

- c. *Data layer* untuk berkomunikasi dengan *database* guna mengakses dan mengambil data dari *database* melalui pembuatan *query*.

Pada *layer* ini digunakan **Entity Engine** dari JSQL. Entity Engine merupakan sekumpulan alat dan pola yang digunakan untuk memodelkan dan mengelola entitas suatu data. Entitas itu sendiri didefinisikan oleh sekumpulan *field* dan *relation* pada entitas lainnya. Entity Engine ini bertujuan untuk memudahkan penggunaan entitas data yang dibangun berdasarkan data-data dalam *database*. Jadi perannya disini adalah untuk berkomunikasi dengan *database* yang berada di *Database Tier* dengan cara mengakses data-data yang telah dipanggil oleh *business logic layer* untuk kemudian dibangun entitasnya. Melalui data-data yang telah diambil dari *database* dan kemudian dibangun entitasnya, maka diteruskan ke *layer* atasnya untuk dapat memenuhi

request data. Entity Engine yang digunakan ADempiere adalah JSQL atau java.sql [23]. Perannya adalah untuk menyediakan API guna mengakses dan memproses data yang tersimpan dalam *database*. API yang digunakan oleh JSQL ialah JDBC (*Java Database Connectivity*). JDBC ini bertugas untuk membangun *SQL Statement* guna mengambil data dari *database* serta melakukan tindakan *read* dan *write* pada data.

3. Database Tier

Pada *tier* terbawah ini, fungsinya hanya fokus untuk menyimpan dan menampung seluruh data-data sistem dalam *database server*. ADempiere ini dapat bekerja dengan beberapa macam *database*, yakni **PostgreSQL**, **Oracle**, dan **MySQL** karena ia menggunakan bahasa pemrograman yang telah banyak diaplikasikan secara umum, yakni Java. Java dapat berhubungan dengan tiap jenis *database* tersebut. Tiap *user* yang menggunakan ADempiere dapat memilih salah satu *database* tersebut sesuai kebutuhan dan keunggulan yang dimiliki. Perbedaannya hanya pada cara instalasi dan koneksi pada *application server* ADempiere. *Database* inilah yang menjadi sumber data untuk menyediakan berbagai data yang dibutuhkan oleh *user* dengan cara pengaksesan data via *SQL Statement* yang dibangun pada *data layer*

Bahasa : Java

Database : PostgreSQL, Oracle, MySQL

OS : Linux, Mac OS X, Unix, Windows), Android

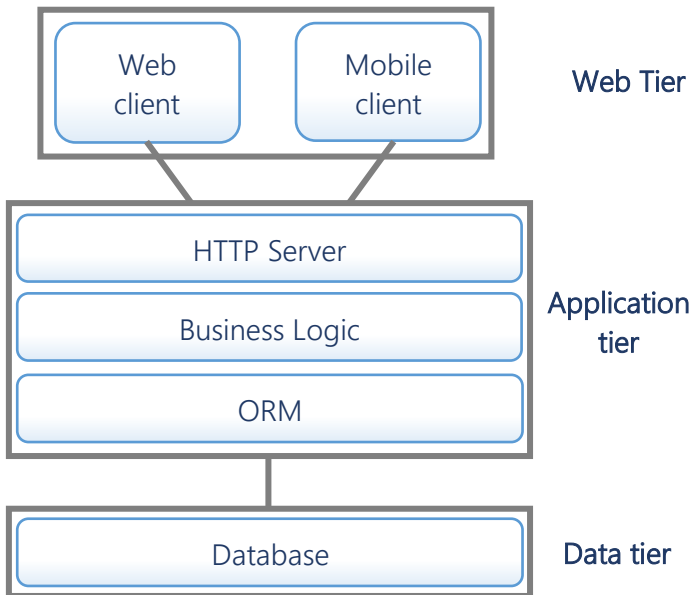
5.1.4 Dolibarr

Pada gambar 5.5 menjelaskan mengenai arsitektur Dolibarr. Dolibarr mempunyai arsitektur *3-tier* yang terbagi atas *Web tier*, *Application tier*, dan *Data tier*. Berikut penjelasan untuk masing-masing *tier* yang ada di dalam arsitektur Dolibarr:

1. Web tier

Dolibarr merupakan *web-based application*. Jadi aplikasi ini dapat diakses di tempat manapun, hanya memerlukan koneksi *Internet* dan *Web browser*. UI yang digunakan pada

Web tier ini berupa **Web clients**, dimana *user* dapat mengakses aplikasi melalui *Web browser* yang ia gunakan. Tiap UI yang ditampilkan pada *Web browser* adalah hasil *generate* dari bagian *web server* (Apache), karena UI dan sistem aplikasinya secara keseluruhan ditulis murni dengan bahasa pemrograman PHP [24]. Selain berupa aplikasi *web*, Dolibarr juga menyediakan **Mobile clients**, yakni tampilan berupa aplikasi *mobile* yang dapat diakses melalui perangkat Android, namun dengan nama myDoli. myDoli merupakan *front-end client* untuk aplikasi Dolibarr, dimana ia bukan merupakan *stand-alone program*. myDoli hanya menyediakan *front-end* untuk aplikasi Dolibarr secara *online*. Jadi tiap tampilan dan fitur di dalamnya berintegrasi langsung dengan sistem yang disematkan pada aplikasi *web*. Dolibarr ini sendiri dapat diakses melalui *mobile application* baik secara gratis (*limited-feature*) dan berbayar (*full-feature*)



Gambar 5.5 Arsitektur Dolibarr

2. Application tier

Pada bagian **Application tier**, terjadi komunikasi antara *Web tier* dengan *Data tier* untuk menerjemahkan keinginan *user* yang ia inputkan melalui UI, agar sistem dapat menyediakan fungsionalitas (modul dan fitur) beserta data yang tersimpan pada *database* di *Data tier*. Oleh sebab itu, pada tahapan ini diawali dengan pemrosesan pada bagian **HTTP server** yang berguna sebagai *web server* untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP. Dolibarr menggunakan Apache sebagai *web server* karena kemudahannya dalam berintegrasi dengan bahasa pemrograman PHP. Tiap tampilan yang diberikan pada UI adalah hasil pemrosesan oleh Apache. Tahapan berikutnya ialah **Business logic** untuk menerjemahkan logika bisnis dari *request* data sehingga dapat menyesuaikan dengan modul yang perlu diakses. Modul yang disediakan Dolibarr terbagi atas 3 macam yang disesuaikan dengan sistem operasi yang digunakan *user*, yakni LAMP (Linux, Apache, MySQL, dan PHP), WAMP (Windows, Apache, MySQL, dan PHP), dan MAMP (Mac OS X, Apache, MySQL, dan PHP) [25]. Tiap modul tersebut berisi modul terkait ERP dan CRM yang sama, pembedanya hanya pada sistem operasi yang mendasarinya. Dasar dalam pembangunan aplikasi Dolibarr ini adalah pada 4 pondasi, yakni sistem operasi (Windows/ Linux/ Mac OS X), Apache sebagai *web server*, MySQL sebagai *database*, dan PHP sebagai *programming language*. Tiap modul yang perlu digunakan untuk memenuhi *request* data didefinisikan pada Business logic. Setelah itu, dibutuhkan akses data guna mengambil data yang diperlukan dari dalam *database*. Oleh sebab itu tahapan berikutnya ialah pada ORM. ORM yang digunakan Dolibarr menerapkan mode koneksi Active Record, dimana tiap *class* merepresentasikan tiap tabel dengan saling terhubung melalui metode CRUD (Create, Read, Update, Delete). *Class* merupakan *record* dari tabel dan hanya berisi kode untuk mengakses tiap *line* dan *field* pada tabel.

3. Data tier

Data tier pada sistem aplikasi hanya berupa *database* yang berfungsi sebagai *repository* untuk menampung tiap data-data dalam sistem, tanpa melakukan modifikasi atau manipulasi data di dalamnya. Dalam versi standarnya, Dolibarr menggunakan MySQL sebagai *database* [24]. Namun seiring dengan perkembangannya, kini Dolibarr juga dapat berinteraksi dengan *database* dari PostgreSQL dan MariaDB. Tiap instalasi aplikasi harus menyelesaikan instalasi terhadap *database* yang digunakan untuk mendefinisikan bagian *Data tier*. Namun, Dolibarr juga menawarkan *auto-installer* berupa paket untuk seluruh kebutuhan instalasi aplikasi yang mencakup Apache, MySQL, dan PHP. Paket tersebut terbagi atas 3 macam, yakni DoliWamp (*auto-intaller* untuk *user* Windows), DoliDeb (*auto-intaller* untuk *user* Linux Debian/ Ubuntu), dan DoliRpm (*auto-intaller* untuk *user* Linux RedHat/ Fedora/ OpenSuse/ Mandriva/ Mageia).

Bahasa : PHP

Database : MySQL, PostgreSQL, MariaDB

OS : Linux, Mac OS X , Windows

5.1.5 ERPNext

Gambar 5.6 merupakan penjelasan arsitektur ERPNext. ERPNext mempunyai arsitektur *3-tier* yang menerapkan prinsip MVC dengan disertai *tools* untuk pemodelan metadata sehingga mampu menambah fleksibilitas bagi *user* untuk dapat menyesuaikan aplikasi dengan tujuan atau kebutuhannya sendiri tanpa perlu pemrograman. Tingkatan arsitekturnya terdiri atas *Front-End*, *Application Server*, dan *Database* [26]. Berikut penjelasan untuk masing-masing tingkatan tersebut:

1. Front-End

User akan berkomunikasi dengan sistem via HTTP yang langsung berhubungan dengan bagian **Front-End**. UI berupa Front-End ini dapat diakses via *web browser* maupun *mobile*. Bagian ini berfungsi sebagai Load Balancer, dimana

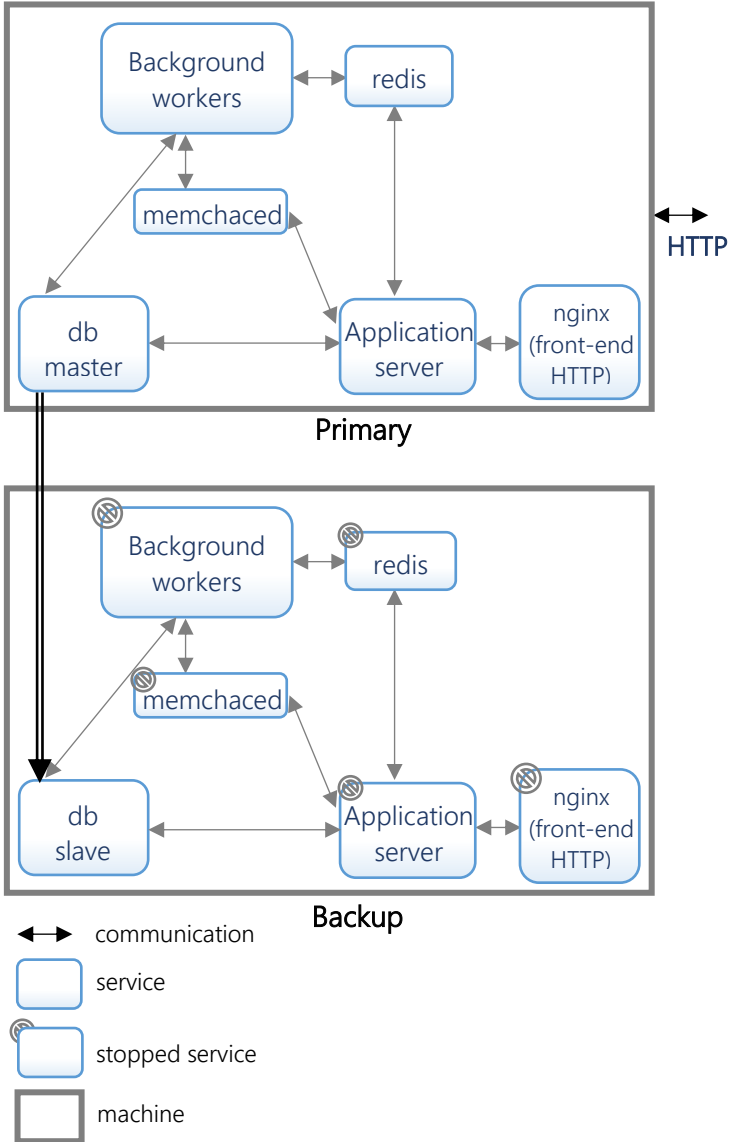
tugasnya adalah mendistribusikan *request* HTTP yang masuk. Load Balancer merupakan penyeimbang beban yang masuk dalam tiap *request* agar server nantinya tidak memproses beban terlalu besar/ berat sehingga pemrosesan dapat memiliki performa, kemudahan, dan kecepatan yang lebih baik. Tujuannya adalah menghindari adanya *single point of failure* (kesalahan pada satu titik/ bagian) yang menyebabkan kehilangan data atau terhentinya layanan. Pada Front-End ini, ERPNext menggunakan web server Nginx. Nginx ini bekerja dengan prinsip *event-based*, dimana tiap *thread* (beban/ *load*) yang masuk akan dicatat untuk kemudian diproses terkait *request*-nya, dan baru dipanggil kembali ketika *request* tersebut selesai penuh. Oleh sebab itu Nginx mampu memanfaatkan seminimal mungkin *thread* untuk memproses *request* dari *user*, sehingga akhirnya memori yang terpakai oleh Nginx menjadi minimal. Karena memori yang dipakai sangat kecil, maka hasilnya server menjadi ringan dan lebih responsif (memiliki respon sangat cepat).

2. Application Server

Pada tingkatan ini, dilakukan pemrosesan *request* HTTP oleh **Application Server**. Di dalamnya terjadi pembuatan *query* untuk mengakses data, baik yang masih tersimpan pada *cache*, atau berada di *database*. Application Server merupakan bagian terpenting dalam memproses *request* HTTP agar dapat sesuai/ memenuhi keinginan *user*. Dalam pemrosesannya tersebut, terdapat tiga komponen yang memberikan dukungan, yakni antara lain :

a. Background Workers

Background Workers berfungsi dalam mengeksekusi penjadwalan *jobs*. *Jobs* yang dimaksud disini adalah *thread* yang harus diproses dan dipenuhi oleh Application Server. Tujuannya adalah untuk menentukan urutan dan waktu tiap *thread* agar tidak terjadi *bottle neck* atau kesalahan dalam pemrosesan antar *thread* pada Application Server



Gambar 5.6 Arsitektur ERPNext

b. Memcached

Memcached bertugas dalam aktivitas *caching*. Performa dalam Memcached ini bergantung pada desain dan konfigurasi seluruh *nodes* Memcached pada Application Server. Tugasnya adalah pada penyimpanan tiap data yang sering atau terakhir digunakan ke dalam memori. Tujuannya untuk mempercepat proses akses data pada Application Server dalam rangka memenuhi *request* dari *user* sehingga tidak memerlukan pencarian di dalam *database*

c. Redis

Redis berperan sebagai *broker/* penjabar tugas/ *jobs* bagi Background Workers. Jadi, tiap *thread* yang masuk dan diproses oleh Application Server perlu diidentifikasi dan ditentukan daftarnya. Ini merupakan tugas dari Redis, sebelum akhirnya diteruskan kepada Background Workers untuk ditentukan jadwal urutan dan waktu tiap *thread* tersebut.

3. Database

Database di sistem aplikasi ERPNext ini terdiri atas berbagai *nodes* data. Ia memberikan layanan transparansi bagi Application Server agar kebutuhan akses data dapat dilakukan dengan mudah. Database ERPNext menerapkan *multi node failure*, yakni apabila terjadi kesalahan atau kegagalan pada suatu *node*, maka terdapat alternatif *node* (*backup nodes*) yang dapat menggantikannya. Tujuannya adalah agar *availability* (ketersediaan) layanan dapat terus terjaga serta menghindari terjadinya kehilangan data.

Untuk menerapkan konsep *multi node failure*, maka ERPNext menggunakan dua macam mesin. Mesin pertama ialah layanan utama sistem yang dijalankan dalam aplikasi ERPNext. Sedangkan mesin kedua berperan sebagai layanan *backup*. Layanan ini mencakup seluruh bagian dan komponen seperti pada layanan utama. Namun tiap bagian atau komponen tersebut memiliki *stopped service* yang bertujuan agar tiap proses yang telah disimpan pada mesin *backup* akan tetap tersedia di dalamnya dan hanya digunakan

ketika terjadinya *failure* pada suatu *point* atau *nodes*. Jadi fungsi mesin *backup* disini adalah sebagai penyedia alternatif layanan ketika terjadi permasalahan pada mesin utama. Proses dalam mengakses mesin *backup* ini berawal ketika terjadinya suatu *failure*, maka *database master* (*database* utama) akan mengkomunikasikannya pada *database slave* (*database backup*). Selanjutnya *database slave* ini yang akan meneruskan permintaan akses tersebut pada bagian atau komponen yang dibutuhkan untuk memberikan alternatif *point* atau *nodes* kepada bagian atau komponen dalam mesin utama yang mengalami *failure*.

Bahasa : JavaScript, Python

Database : MySQL, MariaDB

OS : Linux (Ubuntu dan CentOS), Android

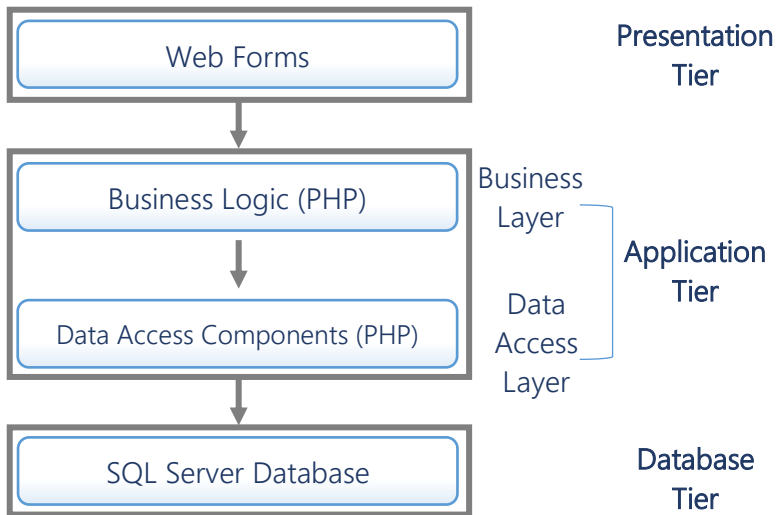
5.1.6 WebERP

Gambar 5.7 menjelaskan mengenai arsitektur WebERP. WebERP mempunyai arsitektur 3-tier yang sederhana [27].

Terdapat 3 macam *tier* yang tersusun dalam arsitektur ini, yakni *Presentation Tier*, *Application Tier*, *Database Tier*. Penjelasan untuk masing-masing *layer* tersebut adalah sebagai berikut:

1. Presentation Tier

Pada *tier* teratas ini (**Presentation Layer**), terdapat UI dari WebERP sendiri yang berguna dalam interaksi *user* terhadap sistem. Tiap *input* yang hendak dimasukkan oleh *user* dan *output* yang hendak ditampilkan kepadanya adalah melalui UI ini. UI yang digunakan oleh WebERP adalah dalam bentuk Web Forms. Dalam *layer* ini juga berisi ASP.Net web server controls, Validation controls and ObjectDataSource controls. ASP.Net sendiri atau Active Server Pages .NET merupakan kumpulan teknologi dalam Framework .NET untuk membangun aplikasi web secara dinamis dengan disertai layanan *web XML*. ASP.Net berfungsi sebagai bahasa pemrograman pada *client-side* di bagian Web Forms. Halaman ASP.Net dijalankan di *web server* untuk kemudian dibuatkan halaman yang dikirim pada Web Forms sebagai UI.



Gambar 5.7 Arsitektur WebERP

2. Application Tier

Tier ini terbagi menjadi 2 macam *layer* guna pemrosesan logika dari *request user* dan kemudian diterjemahkan menjadi akses data pada *database*. *Layer* pada *application tier* antara lain adalah:

a. Business Layer

Layer berikutnya ini berupa **Business Layer** yang berisi kelas-kelas untuk mengimplementasikan logika bisnis. Jadi tiap *request* yang dikirimkan *user* melalui Web Forms pada *Presentation Layer* akan diterjemahkan maksud dan kebutuhannya ke dalam bahasa pemrograman yang digunakan oleh sistem/ *application server*, yakni bahasa PHP. Implementasi logika bisnis ini bertujuan untuk mendefinisikan tiap kebutuhan untuk memenuhi *request user*. Jadi dengan begitu, *request user* akan terpetakan ke dalam bahasa PHP dan dapat dianalisis terkait data-data apa saja yang dibutuhkan.

b. Data Access Layer

Setelah tiap data didefinisikan, maka selanjutnya tugas sistem akan dilimpahkan pada **Data Access Layer**. *Layer*

ini berisi logika akses data yang dihasilkan dari tiap komponen. Bahasa yang digunakan pada *layer* ini masih tetap PHP sehingga hasil terjemahan *request user* sudah sesuai untuk langsung digunakan. Tiap data yang didefinisikan *Business Layer* akan dibuatkan aksesnya terhadap *database* untuk memenuhi kebutuhan terkait data tersebut. Dalam melakukan akses tersebut, tentunya dibutuhkan *building query* untuk berinteraksi dengan *database*. Selain itu, pada *layer* ini juga memungkinkan adanya penambahan berbagai aplikasi .Net dari platform .Net yang bertujuan untuk memudahkan kustomisasi sistem aplikasi agar sesuai dengan kebutuhan *user*, tanpa diperlukan *editing source code* yang kompleks

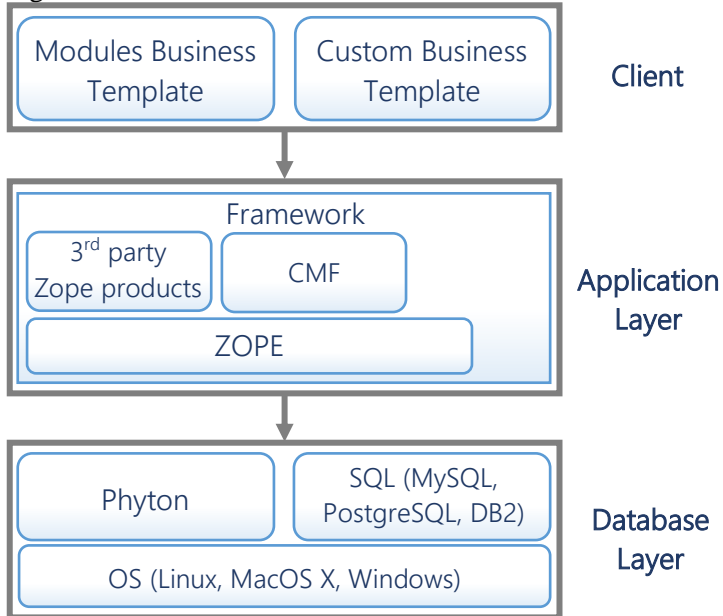
3. Database Tier

Tier yang terakhir tentu saja ialah **Database Layer**, dimana ia berfungsi sebagai *SQL Server database*. Di dalam *tier* ini tersimpan *database*, tabel, dan *stored procedures* yang berguna untuk menyediakan kebutuhan-kebutuhan data dalam menyelesaikan *request user*. Fungsi utama pada *Database Tier* adalah untuk menyimpan dan menampung seluruh data dalam *database server*. *Database* yang dapat didukung oleh WebERP sendiri ialah MySQL dan MariaDB. Sebelumnya, aplikasi ini juga menyediakan pilihan untuk *database* PostgreSQL. Hanya saja karena banyaknya pemakai *database* MySQL dan pertanyaan yang muncul seputar *database* tersebut, maka WebERP memutuskan untuk menghentikan dukungannya terhadap PostgreSQL. *Database* yang digunakan harus memiliki fitur *innodb transaction table*. InnoDB merupakan jenis *storage engine* yang dimiliki MySQL dengan tujuan untuk menyimpan data yang sangat besar.

Bahasa : ASP.Net dan PHP
 Database : MySQL dan MariaDB
 OS : Linux dan Windows

5.1.7 ERP5

Pada gambar 5.8 menjelaskan mengenai arsitektur ERP5. ERP5 mempunyai arsitektur yang sama dengan kebanyakan FOS ERP lainnya, yakni *third-tier architecture* [28]. Tingkatan ini dimulai dari *database layer*, *application*, dan yang paling atas ialah *client*. Penjelasan untuk masing-masing *tier* tersebut adalah sebagai berikut:



Gambar 5.8 Arsitektur ERP5

1. Client

Pada tingkat **Client** ini, tersedia UI sebagai penghubung interaksi antara *user* dengan sistem aplikasi. UI yang disediakan oleh ERP5 adalah berupa aplikasi web sehingga diperlukan Browser untuk menjalankannya. UI ini berbentuk sekumpulan *business template* yang berbeda-beda sesuai modul/ fitur yang tersedia. Untuk Modules Business Template, terdapat beberapa modul utama ERP5 seperti ERP5 Core, Accounting, PDM (*Physical Data Model*), MRP (*Material Requirement Planning*), SCM (*Supply Chain*

Management), dan CRM (*Customer Relationship Management*). Terdapat pula *business template* untuk fitur-fitur khusus, yakni ERP5 Apparel dan ERP5 Banking. Selain dari seluruh *business template* tersebut, ERP5 juga memungkinkan pembuatan *business template* baru yang disesuaikan dengan kebutuhan *user* atau disebut sebagai Custom Business Template. Tiap *user* perlu menyesuaikan kebutuhan akses *input* dan *output* mengenai sistem ERP dengan *business template* yang sesuai.

2. Application Layer

Pada *layer* ini *engine* utama yang berperan ialah Framework. **Application Layer** dijalankan berdasarkan ERP5 Framework yang terdiri atas beberapa komponen penting. Pertama ialah Zope, *web application server* untuk *open source*, dimana *source code* utamanya ditulis dalam bahasa Python. Zope memiliki fitur *transactional database* objek yang dapat menyimpan tidak hanya konten dan data kustom, namun juga *template-template dynamic HTML*, *script*, *search engine*, dan koneksi serta kode *database* relasional. Zope menyediakan seluruh fungsionalitas pada *application server*, mulai dari *authentication*, *transactions*, *data persistence*, hingga *component management*. Komponen Zope yang digunakan oleh ERP5 mencakup CMF (*Content Management Framework*), kerangka kerja sederhana yang menyediakan konsep inti terkait implemmentasi dokumen pada ERP5. Jadi, CMF disini berfungsi dalam manajemen dokumen. Selain itu, terdapat banyak komponen *third party* lainnya dari produk Zope yang digunakan, seperti penggunaan Formulatur dan Forms.

3. Database Layer

Layer terbawah merupakan **Database Layer**. *Layer* ini mencakup *database* beserta spesifikasi sistem operasi dan bahasa pemrograman yang digunakan. ERP5 dapat berjalan di berbagai sistem operasi (*cross-platform*), baik pada Linux, Mac OS X, Unix, maupun Windows. Python menjadi bahasa yang membangun sistem aplikasi dengan alasan karena penggunaannya yang mudah dipahami oleh *user* pemula

terkait *script application* serta memungkinkan developer ahli untuk mendesain arsitektur *dynamic software* yang kompleks. Tiap *request SQL* pada ERP5 berjalan sederhana karena tidak diperlukannya penggunaan fitur kompleks pada *database*. ERP5 menyatukan konsep antara ODB (*Object Database*) dengan RDB (*Relational Database*) untuk penyimpanan dan *indexing* pada data guna menghasilkan *data model* dan performa *query* yang fleksibel dan maksimal. ERP5 dapat bersesuaian dengan beberapa jenis *database*, yakni PostgreSQL, MySQL, dan DB2. Namun lebih direkomendasikan penggunaan *database* pada MySQL karena performa yang tinggi meski menagani *data sets* yang besar.

Bahasa : Python

Database : PostgreSQL, MySQL, DB2

OS : Linux, Mac OS X, Unix, Windows

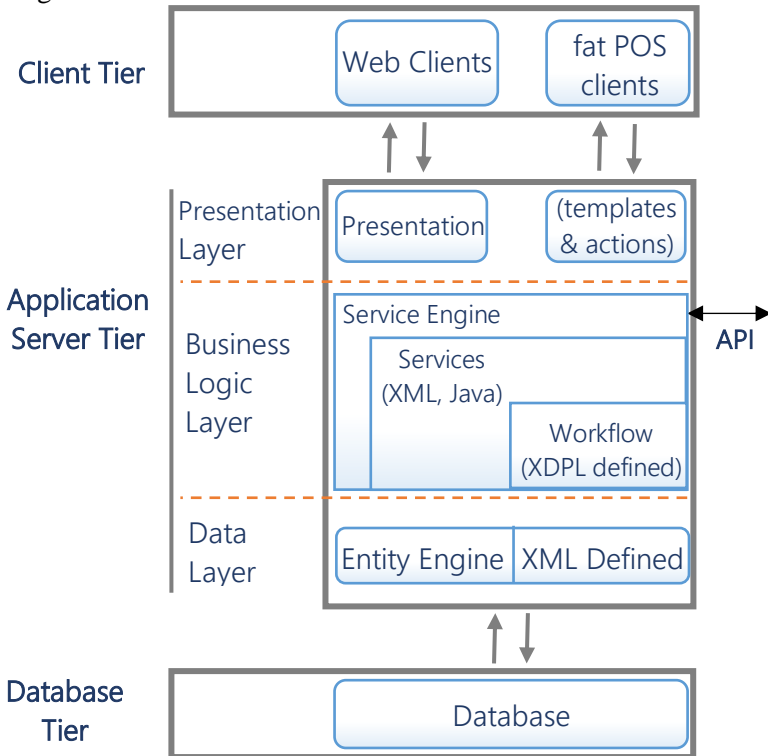
5.1.8 Opentaps

Gambar 5.9 adalah penjelasan arsitektur Opentaps. Opentaps memiliki arsitektur *3-tier* yang mencakup *Client Tier*, *Application Server Tier*, dan *Database Tier* [29]. Berikut penjelasan untuk tiap *tier* arsitektur:

1. Client Tier

Tier yang langsung berhubungan dengan *user* adalah **Client Tier**. Pada *tier* ini, terdapat dua macam UI yang dapat digunakan *user* untuk berkomunikasi terhadap sistem, yakni **Web clients** dan **Fat POS clients**. Web clients merupakan UI utama yang dapat digunakan *user* untuk berinteraksi dengan sistem dalam Opentaps serta mengakses berbagai fitur dan modul yang dimilikinya. Opentaps menyajikan UI dengan berbasis aplikasi web dan aplikasi *mobile*. Hanya saja pada aplikasi *mobile* fokusnya lebih kepada aktivitas CRM saja, dimana aplikasinya dapat diintegrasikan dengan aplikasi web lain, yakni Twitter dan Gmail. Pada Twitter, *user* dapat memonitor *tweet* dari orang-orang di kontaknya atau di grupnya, menganalisis *tweet* untuk mengidentifikasi *rending topics*, mengirim dan membalas *tweet*, serta melihat

jumlah *following* dan *follower* dari orang-orang di kontakannya. Sedangkan pada Gmail, *user* dapat melihat aktivitas dari orang-orang di kontakannya, termasuk berbagai catatan, diskusi, dan *tweet*. UI yang kedua adalah Fat POS clients. UI ini lebih diperuntukkan untuk kegiatan transaksi *retail*, dimana fungsinya adalah untuk mencatat waktu dan tempat ketika suatu barang masuk atau keluar dari toko *retailer*. Biasanya, POS clients ini berupa program aplikasi bagi kasir.



Gambar 5.9 Arsitektur Opentaps

2. Application Server Tier

Pada *tier* kedua ini terbagi lagi menjadi 3 bagian/ *layer*. **Applicaiton Server Tier** terdiri atas 3 macam *layer*, yakni sebagai berikut:

a. Presentation layer

Layer ini lebih bertujuan untuk menghubungkan Client Tier dengan Business Logic Layer. Meneruskan *request* data yang dikirimkan *user* agar pemrosesannya dapat sesuai untuk *layer* berikutnya. Untuk UI utama, terdapat *layer* **Presentation** yang bertugas dalam meneruskan *request* data Web client, baik melalui aplikasi web ataupun *mobile*, kepada Service Engine. Ia juga yang nantinya menyesuaikan tampilan data yang diperoleh dari Service Engine untuk agar sesuai model tampilan di Web Client. Sedangkan untuk UI Fat POS clients, *layer* ini berupa **template and actions**. Maksudnya disini, ia memberikan *template* tampilan untuk mengemas data-data yang berasal dari Service Engine agar sesuai tampilan UI untuk POS client. Selain itu juga tiap data yang masuk dari POS client akan diidentifikasi dan didaftarkan agar sesuai dengan *template* sehingga memudahkan Service Engine dalam menerjemahkan tiap data tersebut.

b. Business Logic layer

Pada *layer* ini terjadi penerjemahan logika bisnis antara *request* dari *user* dengan akses data yang dibutuhkan. Business Logic Layer menyesuaikan *request* data dengan kebutuhan modul, fitur, dan data dari sistem. Aktor utama pada *layer* ini ialah **Service Engine**. Ia merupakan mesin yang memproses tiap *request* agar dapat diidentifikasi terkait modul, fitur, dan data yang dibutuhkan. Opentaps menggunakan Service Engine yang bernama POJO (*Plain Old Java Object*), dimana ia didesain untuk dapat melakukan *mount* (penggabungan) Java *service objects* dengan Service Engine milik Opentaps [30]. Dalam penggunaan POJO ini, diperlukan deklarasi terlebih dahulu mengenai *service* yang digunakan. Oleh sebab itu, dalam Service Engine ini, terjadi deklarasi **Service** dengan menggunakan file XML dan Java. Jadi tiap layanan (fungsionalitas) yang dimiliki Opentaps diidentifikasi untuk dapat diketahui mengenai fitur dan modul yang dapat digunakan dalam memenuhi *request*

data. Dalam Service Engine juga terjadi penentuan **Workflow** atau alur kerja dalam format XDPL (XML Programming Defined Language) agar dapat diidentifikasi waktu dan jadwal untuk pemenuhan tiap *request*. Service Engine Opentaps yang terbaru telah memungkinkan penggunaan API. Melalui adanya API ini, Opentaps dapat dikembangkan lebih lanjut oleh tiap developer sesuai kebutuhannya masing-masing. Selain itu juga karena Opentaps terus berusaha mengembangkan aplikasinya agar dapat bersinkronasi lebih kompleks antara aplikasi web dan *mobile*, maka API ini menjadi salah satu alternatif solusi untuk memudahkan integrasi antar platform.

c. Data layer

Data *layer* ini terdiri atas **Entity Engine** yang berfungsi dalam mengelola keperluan akses data terhadap *database* melalui model data yang didefinisikan dalam bentuk file XML. Jadi pada *layer* ini juga diperlukan **XML defined**, dimana nantinya ia yang akan berperan sebagai *interface* dengan *database*. Tiap *request* data yang telah diterjemahkan terkait kebutuhan fungsionalitas (modul dan fitur) dan datanya, diteruskan pada Data *layer* guna mengakses dan mengambil data yang diperlukan dari dalam *database*.

3. Database Tier

Pada **Database tier** ini berfungsi dalam menyimpan tiap data dan dokumen beserta relasinya. Data *layer* yang digunakan Opentaps mampu mengakses dan mencari data dari berbagai sumber, baik itu dari *database*, dokumen, maupun melalui web dengan penggunaan API. Kemampuan tersebut dapat dicapai karena ia menerapkan Domain Driven Architecture untuk membuat *interface* dengan berbagai sumber data tersebut. Domain Driven Architecture sendiri adalah konsep terkait penggabungan antar domain dalam satu sistem aplikasi. Domain yang dimaksud adalah domain untuk tiap jenis data. Melalui identifikasi domain ini, maka Data *layer* dapat mengetahui dimana sumber data yang harus diakses

atau dicari untuk dapat memenuhi kebutuhan data. Mengenai jenis *database* sendiri, Opentaps pada mulanya mampu berinteraksi dengan 3 macam *database*, yakni MySQL, PostgreSQL, dan maxDB. Namun seiring dengan perkembangan *database* sendiri, kini Opentaps lebih memfokuskan instalasi pada *database* yang populer atau umum digunakan *user*, yaitu MySQL dan PostgreSQL.

Bahasa : Java

Database : MySQL, PostgreSQL

OS : Linux, Mac OS X, Unix, Windows

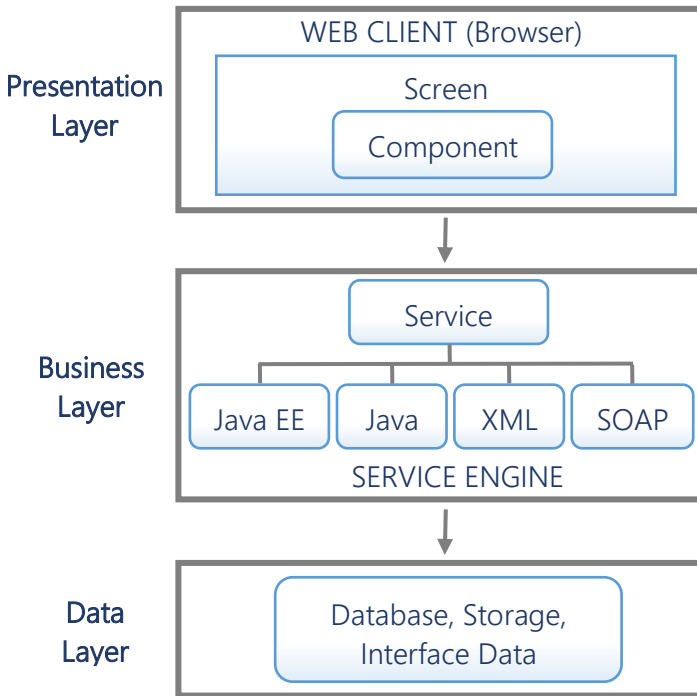
5.1.9 Apache OFBiz

Gambar 5.10 menjelaskan mengenai arsitektur Apache OFBiz. Apache OFBiz memiliki arsitektur 3-tier, yakni terdiri atas *Presentation Layer*, *Bussiness Layer*, dan *Data Layer* [31]. Berikut penjelasan untuk masing-masing *layer* dalam arsitektur tersebut:

1. Presentation Layer

Pada **Presentation Layer**, Apache OFBiz menggunakan konsep ‘screens’ untuk merepresentasikan setiap halaman UI pada aplikasinya. Jadi, tiap halaman tersebut bertindak sebagai suatu *screen/* layar. Suatu halaman UI yang dieksekusi melalui Browser dalam bentuk Web client, terdiri atas beberapa komponen. Suatu komponen tersebut dapat berupa *header*, *footer*, isi data, fitur, dan lain sebagainya. Ketika satu halaman dieksekusi, maka seluruh komponen akan digabungkan bersama-sama sesuai dengan ketentuan yang telah didefinisikan pada bagian *screen definition*. Format dari komponen itu sendiri dapat bermacam-macam, baik berupa Java Server Pages (JSP), halaman FTL yang dibangun melalui FreeMarker Template, maupun Widget untuk *Menu* dan *Forms*. Halaman FTL (FreeMarker Template) merupakan halaman *output* yang dihasilkan dari *library* Java bernama FreeMarker [32]. FreeMarker sendiri memiliki fungsi yang hampir sama dengan JSP, yakni untuk membentuk dan menghasilkan *output* teks dalam format

halaman *web* HTML. Sedangkan untuk *Widget*, ia adalah teknologi spesifik yang memang disediakan Apache OFBiz.



Gambar 5.10 Arsitektur Apache OFBiz

2. Business Layer

Application Layer atau **Business Layer** merupakan *layer* yang mendefinisikan layanan yang disediakan bagi *user*. Layanan ini dapat berupa beberapa jenis, baik itu Java methods, SOAP, *simple services*, maupun *workflows*. Tiap layanan pada Apache OFBiz ditangani oleh *Service Engine*. *Service Engine* ini bertanggung jawab terhadap tiap *invocation* (permintaan), transaksi dan keamanan. Apache OFBiz menggunakan teknologi *open-source* yang terstruktur dan terstandarisasi seperti Java, Java EE, XML dan SOAP. Meskipun Apache OFBiz dibangun dengan menggunakan konsep pada Java EE, namun konsep tersebut

diimplementasikan dengan cara yang berbeda-beda. Hal ini dapat dikarenakan Apache OFBiz telah dirancang sebelum banyak pembaruan dalam Java EE. Selain itu juga dapat disebabkan dari faktor developer atau *user*, dimana mereka menginginkan implementasi sistem yang tidak hanya terikat pada konsep Java EE saja.

3. Data Layer

Layer paling bawah ialah **Data Layer**, dimana tugasnya untuk bertanggung jawab terhadap akses *database*, *storage* (penyimpanan data), maupun penyediaan *interface* data bagi *Business Layer*. Data yang diakses sistem tidak berdasarkan konsep Object Oriented, tetapi dengan cara relasional. Jadi tiap entitas yang direpresentasikan sebagai baris dalam *database* tersebut, disediakan bagi *Business Layer* sebagai suatu *generic value*. Artinya, hasil yang diberikan pada *Business Layer* tidak berupa teks, melainkan sekumpulan entitas yang terbagi ke dalam beberap *fields* dan diakses atau dibedakan berdasarkan *fields name* (nama kolom). Apache OFBiz secara *default* telah menyediakan Apache Derby sebagai bagian dari *database* Java yang dimiliki produk IBM. Namun penggunaannya hanya bagus untuk evaluasi, *development* dan *testing* sistem. Apabila *user* membutuhkan *server* dengan kinerja *database* yang lebih baik, maka diperlukan instalasi terhadap jenis *database* lain. Apache OFBiz mampu mendukung berbagai jenis *database*, seperti PostgreSQL, MySQL, Oracle, Microsoft SQL Server, Sybase dan DB2. Tetapi *database* yang paling populer digunakan oleh *user* untuk aplikasi ini ialah PostgreSQL dan MySQL

Bahasa : FreeMarker, Java

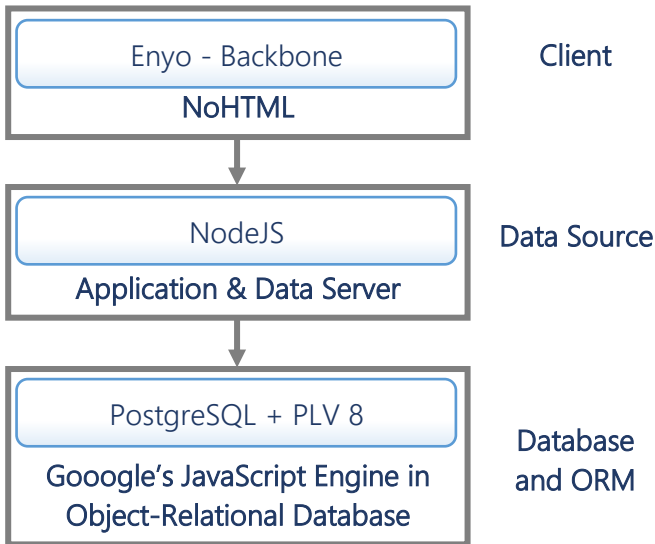
Database: PostgreSQL, MySQL, Oracle, Microsoft SQL Server, Sybase, DB2

OS : Linux, Mac OS X, Unix, Windows

5.1.10 xTuple Postbooks

Pada gambar 5.11 menjelaskan mengenai arsitektur xTuple Postbooks. Arsitektur dari Postbooks merupakan *third-tier*

architecture dengan terdiri atas *Client*, *Data Source*, dan *Database and ORM* [33]. Postbooks menjunjung tinggi bahasa JavaScript untuk dapat digunakan pada seluruh *tier*. Tujuan utamanya agar developer tidak perlu repot mempelajari seluruh bahasa yang digunakan sistem, cukup satu bahasa saja agar ia dapat memahami sistem secara menyeluruh serta lebih fokus terhadap masalah bisnis yang hendak dipenuhi dan diselesaikan oleh perusahaan penggunanya. Penjelasan untuk masing-masing *tier* tersebut adalah sebagai berikut:



Gambar 5.11 Arsitektur xTuple Postbooks

1. Client

Pada bagian **Client**, Postbooks menggunakan gabungan antara Enyo dan Backbone. Enyo merupakan *framework* untuk membangun aplikasi HTML5 yang bekerja pada semua perangkat. Hal ini mendasari tujuan Postbooks untuk dapat membangun UI yang dapat berjalan bersamaan baik pada aplikasi *desktop* maupun *mobile*. Hanya saja pada aplikasi *mobile* fokusnya lebih kepada kontak hubungan dengan sesama *user* dan pelanggan saja. Cara kerjanya memungkinkan *user* untuk membangun UI tanpa

memerlukan penulisan kode HTML. Struktur aplikasi Enyo dibangun dalam hirarki *object model* sehingga tiap komponen yang penting dapat digunakan kembali untuk UI lainnya. Jadi tiap UI baru yang dibangun dapat disesuaikan untuk membuat modul baru dengan lebih efisien. Postbooks menamai sistem UI mereka sebagai Enyo-x. Sedangkan Backbone adalah sistem model yang ditulis dalam JavaScript. Apabila Enyo memungkinkan *user* untuk menentukan tampilan UI, maka Backbone menyediakan *layer* dimana *user* dapat menangani logika bisnis yang berjalan di balik layar UI. Sistem Backbone ini disebut Backbone-x, sistem untuk menangani kebutuhan desain khusus dan memulai *library* untuk *core model* guna mendukung PostBooks.

2. Data Source

Aktor yang paling berperan dalam **Data Source** adalah NodeJS. V8 adalah *engine* JavaScript yang dimiliki Google sebagai daya bagi *browser* Chrome. Namun karena sifat kekuatan dan portabilitasnya, kini V8 digunakan pada banyak *browser*. NodeJS adalah platform untuk membangun aplikasi jaringan berbasis *web* dengan menggunakan JavaScript. Penggunaannya lebih dikhususkan untuk lingkungan *cloud server*. Postbooks menggunakannya sebagai dasar untuk layanan *desktop*, baik dalam melayani sistem, maupun menangani *request* data. NodeJS kini terus berkembang dan terkenal dalam kalangan komunitas developer dan menjadi salah satu bidang *open source* dengan kemampuan lebih cepat daripada teknologi sebelumnya. Oleh sebab itu, Postbooks menggunakannya sebagai *application server* dan *data server*.

3. Database and ORM

Pada **tier Database and ORM**, terdapat *database* dan PLV8. PLV8 merupakan integrasi dari JavaScript ke dalam *database* PostgreSQL (jenis *database* yang digunakan Postbooks), dimana ia memungkinkan *user* untuk menulis tiap fungsi *database* dengan menggunakan JavaScript. Postbooks memanfaatkan teknologi tersebut untuk

membangun sistem *embedded* ORM bagi PostgreSQL. Hal ini memungkinkan PostgreSQL untuk dapat berinteraksi dengan JavaScript dalam bentuk JSON. memetakan ORM sendiri di JSON. Postbooks yang merupakan bagian dari xTuple memang mengedepankan bahasa pemrograman JavaScript sebagai bahasa utama dalam sistemnya, sehingga implementasinya tidak hanya pada UI di *Client tier* saja, namun juga digunakan pada *application tier (Data Source)* dan *data tier (Database and ORM)*.

Bahasa : JavaScript

Database : PostgreSQL

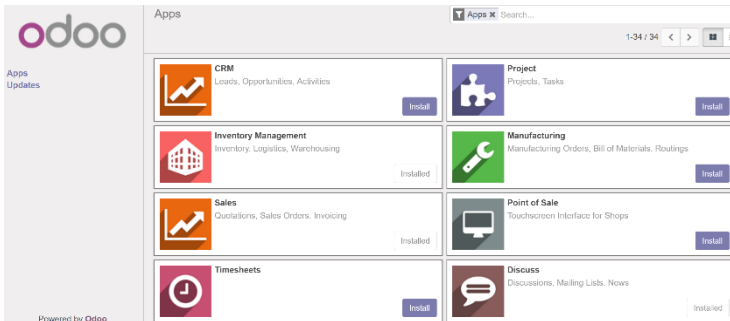
OS : Linux, Mac OS X, Windows

5.2 Aspek Fungsionalitas

Aspek ini menjelaskan implementasi perbandingan FOS ERP pada aspek fungsionalitas modul *sales*, *purchase*, dan *accounting*

5.2.1 Odoo

Berikut gambar 5.12 menunjukkan tampilan halaman utama pada Odoo

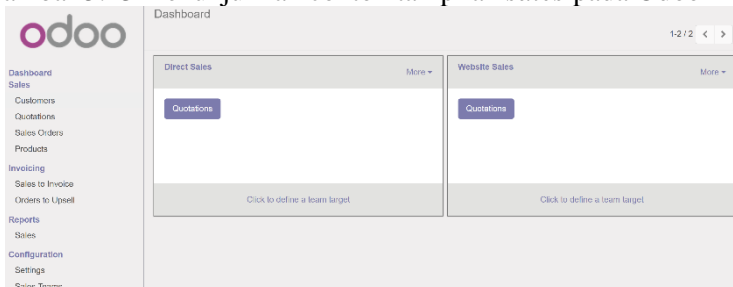


Gambar 5.12 Tampilan Halaman Utama Odoo

5.2.1.1 Sales

Modul *sales* tersedia pada Odoo untuk mengelola seluruh aktivitas terkait proses penjualan yang dilakukan perusahaan kepada pelanggan. Modul ini terdiri atas beberapa fitur yang

merepresentasikan tiap aktivitas dalam *sales management*. Gambar 5.13 menunjukkan contoh tampilan *sales* pada Odoo



Gambar 5.13 Contoh Tampilan Sales Odoo

1. Quotation

Quotation diartikan sebagai dokumen atau formulir penawaran penjualan dari perusahaan untuk mencatat berbagai jenis barang dapat ditawarkan dan dijual kepada pelanggan. Dalam pembuatan *quotation* ini, daftar pelanggan perlu didefinisikan terlebih dahulu ke dalam sistem. Beberapa data penting pada saat pendefinisian pelanggan adalah sebagai berikut:

- a) Jenis pelanggan (individu atau perusahaan)
- b) Nama pelanggan
- c) Detail alamat pelanggan
- d) Detail kontak pelanggan

Data kolom-kolom lain dapat diisi atau dikosongi (opsional). Pada pembuatan *quotation* sendiri, terdapat data-data lagi yang harus diisi, antara lain:

- a) Nama pelanggan (berdasarkan daftar pelanggan)
- b) Tanggal mulai order
- c) Tanggal akhir order
- d) Aturan pembayaran
- e) Detail produk (nama produk, deskripsi, kuantitas, harga, pajak, total biaya)

Perusahaan dapat mencetak *quotation* atau mengirimkannya melalui *email* kepada pelanggan. Apabila *quotation* ini telah

dinyatakan selesai (*save*), maka *quotation* tersebut resmi dikirimkan/ diberikan kepada pelanggan.

2. *Sales Order*

Sales order adalah dokumen yang digunakan untuk mengelola penjualan terhadap pemesanan yang dilakukan oleh pelanggan. Pembuatan *sales order* ini harus didahului dengan pembuatan *quotation* terlebih dahulu. Pada daftar *quotation* yang telah dikirimkan, dipilih ‘Confirm Sale’ untuk meneruskan *quotation* menjadi *sales order*. Data *sales order* akan dibuat berdasarkan data pada *quotation*.

Pada fitur *sales order* ini juga memungkinkan pemantauan terhadap status pengiriman barang. Terdapat beberapa status pada barang yang hendak dikirim, yakni:

a) *Draft*

Detail pengiriman belum ditentukan pada order

b) *Waiting Availability*

Pengiriman belum siap dilakukan

c) *Partially Available*

Pengiriman dikirimkan sebagian/ secara terpisah

d) *Available*

Pengiriman telah tersedia dan siap dilakukan pada order

e) *Done*

Pengiriman telah dilakukan

Pengiriman berstatus *available* dapat difinalisasi dengan pilihan ‘Validate’ untuk menyatakan bahwa pengiriman telah dilakukan sehingga status pengiriman barang menjadi *done*. Pengiriman barang yang telah dilakukan (berstatus *done*) dapat pula dikembalikan pada perusahaan sehingga statusnya akan kembali seperti sebelumnya. Ketika pengiriman ulang dilakukan, maka dokumen pengiriman baru akan dibuat dan validasi pengiriman kembali diperlukan untuk menyatakan pengiriman yang baru telah dilakukan.

3. *Invoice*

Pembuatan *invoice* merupakan kelanjutan dari *sales order* sehingga datanya juga dibuat berdasarkan *sales order*. Barang yang telah dikirimkan kepada pelanggan harus

dibayarkan kepada perusahaan. Perusahaan menggunakan *invoice* sebagai formulir penagihan biaya pada pelanggan. Terdapat beberapa macam *invoice* yang dapat dibuat, yakni:

1. Invoiceable lines
Invoice langsung dibayar secara lunas
2. Invoiceable lines (deduct down payments)
Pembayaran *invoice* untuk sisa tagihan yang belum terbayar dari uang muka
3. Down payment (percentage)
Invoice dibayarkan dengan uang muka (*down payment*). Pembayaran uang muka ini dalam bentuk persentase. Pada pembayaran ini juga harus diisikan *income account* untuk mendefinisikan akun yang menampung uang muka tersebut serta pilihan pajak jika diperlukan
4. Down payment (fixed amount)
Invoice dibayarkan dengan uang muka (*down payment*) namun dalam jumlah biaya yang eksak. Pada pembayaran ini juga harus diisikan *income account* dan pilihan pajak jika diperlukan

Invoice yang telah dibuat akan berstatus *draft* selama pembayaran belum dilakukan. Jika pembayaran telah dilakukan, maka *invoice* dapat divalidasi (*validate*) untuk mengisi detail pembayaran. Pada tahapan ini, *invoice* berstatus *open*. Pada pembayaran, perlu diisikan data tambahan, yakni:

- a) Jurnal/ metode pembayaran
- b) Total uang yang dibayarkan
- c) Tanggal pembayaran
- d) Nomor memo

Jika pembayaran telah dilunasi sesuai dengan total biaya pada *invoice*, maka status *invoice* tersebut akan berganti menjadi *paid*. Pelanggan dapat pula meminta *refund* atau pengembalian dana atas pembayaran yang dilakukan pada pilihan 'Refund Invoice'. Terdapat beberapa data yang harus diisi, yakni metode, alasan, dan tanggal *refund*. Ketika

refund dilakukan, maka *invoice* baru akan dibuat dan statusnya akan kembali sebagai *draft*. Validasi *invoice* kembali diperlukan untuk menyatakan pembayaran yang baru telah dilakukan.

4. Reports

Pada modul *sales* juga terdapat fitur untuk melihat laporan terkait catatan penjualan yang telah dilakukan. Laporan ini dapat dilihat dalam bentuk tabel maupun diagram. Pada laporan ini dapat ditinjau hasil penjualan perusahaan berdasarkan beberapa pengukuran, yakni:

- a) *Gross weight* (berat kotor)
- b) *of Lines* (jumlah *line* produk/ barang)
- c) *of Qty* (jumlah barang)
- d) *Untaxed Total* (total biaya tanpa pajak)
- e) *Qty to Invoice* (jumlah barang yang belum dibuatkan *invoice*)
- f) *Qty Invoiced* (jumlah barang yang sudah dibuatkan *invoice*)
- g) *Volume* (total volume barang)
- h) *Qty Delivered* (jumlah barang yang sudah dikirimkan)
- i) *Total* (total biaya)
- j) *Count* (jumlah penjualan)

Laporan ini akan terhubung dengan modul *accounting* untuk pencatatan seluruh transaksi perusahaan.

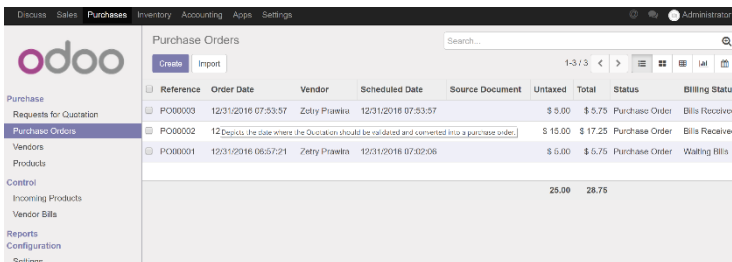
5.2.1.2 *Purchase*

Modul *purchase* tersedia pada Odoo untuk mengelola seluruh aktivitas terkait proses pembelian dan pengadaan material/ barang perusahaan kepada vendor. Modul ini terdiri atas beberapa fitur yang merepresentasikan tiap aktivitas dalam *purchase management*. Berikut gambar 5.14 menunjukkan contoh tampilan halaman *purchase* pada Odoo

1. Request for Quotation

Request for Quotation (RFQ) diartikan sebagai dokumen atau formulir permintaan pembelian barang dari perusahaan kepada vendor. Dalam pembuatan RFQ ini, daftar vendor

perlu didefinisikan terlebih dahulu ke dalam sistem. Beberapa data penting pada saat pendefinisian vendor adalah sebagai berikut:



Reference	Order Date	Vendor	Scheduled Date	Source Document	Untaxed	Total	Status	Billing Status
PO00003	12/31/2016 07:33:57	Zetzy Prawita	12/31/2016 07:33:57		\$ 5.00	\$ 5.75	Purchase Order	Bills Received
PO00002	12/31/2016 08:57:21	Zetzy Prawita	12/31/2016 07:33:57		\$ 15.00	\$ 17.25	Purchase Order	Bills Received
PO00001	12/31/2016 08:57:21	Zetzy Prawita	12/31/2016 07:33:57		\$ 5.00	\$ 5.75	Purchase Order	Waiting Bills
					25.00	28.75		

Gambar 5.14 Contoh Tampilan *Purchase Odoo*

- Jenis vendor (individu atau perusahaan)
- Nama pelanggan
- Detail alamat pelanggan
- Detail kontak pelanggan

Data kolom-kolom lain dapat diisi atau dikosongi (opsional). Pada pembuatan RFQ sendiri, terdapat data-data lagi yang harus diisi, antara lain:

- Nama vendor (berdasarkan daftar pelanggan)
- Nomor referensi vendor
- Tanggal order
- Detail produk (nama produk, deskripsi, tanggal kedatangan, kuantitas, harga, pajak, total biaya)

Perusahaan dapat mencetak RFQ atau mengirimkannya melalui *email* kepada vendor. Apabila RFQ ini telah dinyatakan selesai (*save*), maka RFQ tersebut resmi dikirimkan/ diberikan kepada vendor.

2. *Purchase Order*

Purchase order adalah dokumen yang digunakan untuk mengelola pembelian untuk pengadaan material/ barang kepada vendor. Pembuatan *purchase order* ini harus didahului dengan pembuatan RFQ terlebih dahulu. Pada daftar RFQ yang telah dikirimkan, dipilih ‘Confirm Order’

untuk meneruskan RFQ menjadi *purchase order*. Data *purchase order* akan dibuat berdasarkan data pada RFQ. Sama halnya dengan fitur *sales order* pada modul *sales*, fitur *purchase order* juga memungkinkan pemantauan terhadap status pengiriman barang. Terdapat beberapa status pada barang yang hendak dikirim, yakni:

a) *Draft*

Detail pengiriman belum ditentukan pada order

b) *Waiting Availability*

Pengiriman belum siap dilakukan

c) *Partially Available*

Pengiriman dikirimkan sebagian/ secara terpisah

d) *Available*

Pengiriman telah tersedia dan siap dilakukan pada order

e) *Done*

Pengiriman telah dilakukan

Pengiriman berstatus *available* dapat difinalisasi dengan pilihan ‘Validate’ untuk menyatakan bahwa pengiriman telah dilakukan sehingga status pengiriman barang menjadi *done*. Pengiriman barang yang telah dilakukan (berstatus *done*) dapat pula dikembalikan pada vendor sehingga statusnya akan kembali seperti sebelumnya. Ketika pengiriman ulang dilakukan, maka dokumen pengiriman baru akan dibuat dan validasi pengiriman kembali diperlukan untuk menyatakan pengiriman yang baru telah dilakukan. Jika perusahaan telah menerima pesannya, terdapat pilihan ‘Receive Products’ untuk menentukan jumlah barang yang telah diperoleh

3. Vendor Bills

Pembuatan *Vendor bills* merupakan kelanjutan dari *purchase order* sehingga datanya juga dibuat berdasarkan *purchase order*. Barang yang telah dikirimkan kepada perusahaan harus dibayarkan kepada pelanggan. *Vendor bills* digunakan sebagai formulir penagihan biaya pada perusahaan. Terdapat beberapa data dalam *bill* yang harus diisi, yakni:

- a) Nama vendor
- b) Nomor referensi vendor
- c) *Purchase order* yang dibayar
- d) Referensi dokumen
- e) Tanggal tagihan
- f) Batas tanggal tagihan
- g) Detail produk dan total pembayaran (otomatis sesuai data dari *purchase order* yang dipilih)

Bill yang telah dibuat akan berstatus *draft* selama pembayaran belum dilakukan. Jika pembayaran telah dilakukan, maka *bill* dapat divalidasi (*validate*) untuk mengisi detail pembayaran. Pada tahapan ini, *bill* berstatus *open*. Pada pembayaran, perlu diisikan data tambahan, yakni:

- a) Jurnal/ metode pembayaran
- b) Total uang yang dibayarkan
- c) Tanggal pembayaran
- d) Nomor memo

Jika pembayaran telah dilunasi sesuai dengan total biaya pada *bill*, maka status *bill* tersebut akan berganti menjadi *paid*. Perusahaan dapat pula meminta *refund* atau pengembalian dana atas pembayaran yang dilakukan pada pilihan 'Ask to Refund'. Terdapat beberapa data yang harus diisi, yakni metode, alasan, dan tanggal *refund*. Ketika *refund* dilakukan, maka *bill* baru akan dibuat dan statusnya akan kembali sebagai *draft*. Validasi *bill* kembali diperlukan untuk menyatakan pembayaran yang baru telah dilakukan.

4. Reports

Pada modul *purchase* juga terdapat fitur untuk melihat laporan terkait catatan pembelian yang telah dilakukan. Laporan ini dapat dilihat dalam bentuk tabel maupun diagram. Pada laporan ini dapat ditinjau hasil pengadaan perusahaan berdasarkan beberapa pengukuran, yakni:

- a) *Gross weight* (berat kotor)
- b) *Days to Deliver* (jumlah hari pengiriman)
- c) *Days to Valuate* (jumlah hari validasi)

- d) *Products Value* (nilai produk)
- e) *Purchase-Standard Price* (biaya standar)
- f) *Average Price* (rata-rata biaya)
- g) *Volume* (total volume barang)
- h) *of Lines* (jumlah *line* produk/ barang)
- i) *Product Quantity* (jumlah barang)
- j) *Total* (total biaya)
- k) *Count* (jumlah pembelian)

Laporan ini akan terhubung dengan modul *accounting* untuk mencatatkan seluruh transaksi perusahaan.

5.2.1.3 Accounting

Modul *accounting* pada Odoo saling berhubungan dengan modul-modul lain yang dapat menyebabkan terjadi pencatatan akuntansi, utamanya pada modul *sales* dan *purchase*. Pada modul ini, terdapat fitur untuk *sales* sendiri yang mencakup *invoice* pada pelanggan, pembayaran penjualan, daftar pelanggan, dan daftar produk yang dijual. Terdapat pula fitur untuk *purchase* sendiri yang mencakup *bill* pada vendor, pembayaran pembelian, daftar vendor, dan produk yang dibeli. Selain fitur yang terhubung langsung dengan modul, ada pula fitur-fitur yang dikhususkan untuk pencatatan akuntansi perusahaan secara keseluruhan.

Gambar 5.15 adalah contoh tampilan *accounting* pada Odoo

Code	Name	Type
100000	Fixed Asset Account	Fixed Assets
101000	Current Assets	Current Assets
101110	Stock Valuation Account	Current Assets
101120	Stock Interim Account (Received)	Current Assets
101130	Stock Interim Account (Delivered)	Current Assets
101200	Account Receivable	Receivable
101300	Tax Paid	Current Assets
101401	Bank	Bank and Cash
101501	Cash	Bank and Cash
101600	Opening Income Account	Other Income
101700	Liquidity Transfers	Current Assets
102000	Non-current assets	Non-current Assets

Gambar 5.15 Contoh Tampilan Accounting Odoo

1. Journal Entries

Jurnal akuntansi yang digunakan untuk mencatat transaksi yang terjadi dalam perusahaan, baik berupa transaksi penjualan maupun pembelian. Jurnal ini dibuat secara otomatis ketika adanya suatu dokumen *invoice*. Jurnal ini juga dapat dibuat secara manual dengan memasukkan tiap data terkait catatan transaksi dan akuntansi yang dibutuhkan. Data-dalam jurnal ini antara lain:

- a) Nama jurnal
- b) Tanggal pencatatan jurnal
- c) Referensi jurnal (opsional)
- d) *Item* jurnal (akun yang digunakan, *partner* terkait, label, debit, dan kredit)

Tiap transaksi yang dicatat terdiri atas bagian debit dan kredit, dimana total dari keduanya harus sama/ seimbang. Jurnal ini digunakan juga sebagai dasar dalam pencatatan buku kas

2. Chart of Accounts

Fitur ini menyimpan berbagai daftar rekening dalam bentuk daftar yang terstruktur, dimana terdiri dari rangkaian kode, nama, dan jenis akun. Daftar rekening ini berguna untuk memudahkan penyusunan dan pembacaan tiap akun dalam akuntansi

3. Manual Payment and Invoice Matching

Fitur untuk melakukan penyesuaian pada pembayaran dan *invoice* yang bermasalah atau memiliki kondisi lain. *Invoice* dan pembayaran tersebut akan diperbaiki agar dapat dicatat akuntansinya ke dalam buku kas secara tepat

4. Manual Tax Adjustments

Fitur untuk melakukan penyesuaian pada aturan perpajakan yang bermasalah atau memiliki kondisi tertentu. Aturan pajak tersebut harus diperbaiki, termasuk dalam akun debit dan kreditnya agar tidak terjadi kekeliruan pencatatan akuntansi ke dalam buku kas

5. Business Intelligence

Fitur ini fungsinya sama seperti fitur *Reports* pada *sales* dan *purchase*. Ia berguna untuk melihat laporan terkait catatan penjualan atau pembelian dalam bentuk tabel atau diagram. Laporan ini dapat ditinjau hasil penjualan atau pembelannya melalui beberapa pengukuran

6. PDF Reports

Berbagai laporan akuntansi dalam kurun waktu setahun atau beberapa tahun dapat diakses melalui fitur ini. Fitur ini menghasilkan laporan akuntansi akhir dalam bentuk PDF untuk diunduh dan dapat dicetak secara *offline*. Berbagai laporan yang dicakup antara lain:

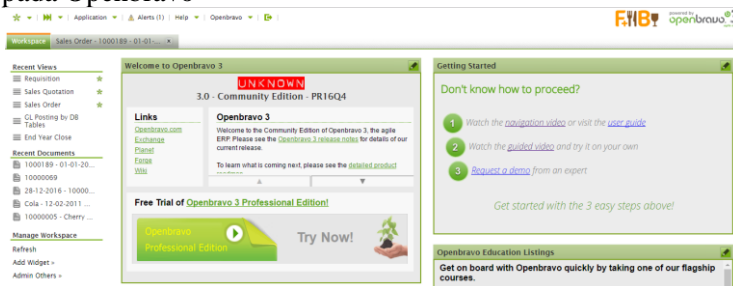
- *Sales/ Purchase Journal*
Jurnal akuntansi terkait seluruh catatan transaksi *sales* atau *purchase* perusahaan
- *Partner Ledger*
Buku kas layaknya *General Ledger* namun dikhususkan dan dibedakan berdasarkan tiap *partner* perusahaan yang melakukan transaksi dan menyebabkan perubahan keuangan perusahaan
- *General Ledger*
Buku kas besar yang digunakan untuk pencatatan seluruh aliran kas masuk/ piutang (debit) dan keluar/ hutang (kredit) beserta pembayaran yang dilakukan
- *Trial Balance*
Neraca saldo/ neraca sisa berupa daftar yang memuat seluruh akun dengan saldo totalnya yang bersumber dari buku kas besar perusahaan
- *Balance Sheet*
Neraca keuangan perusahaan yang menampilkan pengeluaran (akun hutang) dan pendapatan (akun piutang) perusahaan dengan disertai total keuntungan/ kerugian yang didapat
- *Profit and Loss*

Pencatatan seluruh keuntungan yang diperoleh dan kerugian yang dikeluarkan perusahaan

- Aged Partner Balance
Laporan akuntansi dengan fungsi yang sama seperti *Partner Ledger*, namun menyediakan catatan yang lebih detail untuk tiap rentang waktu yang diinginkan
- Financial Report
Laporan finansial lain yang tidak termasuk laporan-laporan di atas dan dapat dibuat sendiri sesuai keinginan perusahaan

5.2.2 Openbravo

Berikut gambar 5.16 menunjukkan tampilan halaman utama pada Openbravo



Gambar 5.16 Tampilan Halaman Utama Openbravo

5.2.2.1 Sales

Sales Management pada Openbravo berhubungan dengan seluruh kegiatan yang berhubungan dengan proses penjualan (*sales*) kepada pelanggan dan pelaporan yang dibutuhkan. Berikut gambar 5.17 menunjukkan contoh tampilan halaman *sales* pada Openbravo. Fitur dan langkah dalam proses bisnis pada *Sales Management* Openbravo adalah sebagai berikut:

1. Sales Quotation

Sales quotation merupakan formulir penawaran penjualan yang berfungsi untuk mencatat berbagai jenis barang yang ingin dan dapat dijual kepada pelanggan. Pada fitur ini

Openbravo memberikan pilihan bahwa perusahaan dapat membuat *quotation* kepada pelanggan ataupun tidak. Apabila perusahaan ingin membuatnya, maka terdapat beberapa keterangan yang harus ia isi, antara lain yaitu:



Gambar 5.17 Contoh Tampilan Sales Openbravo

Daftar Quotation

- a) Nama perusahaan
- b) Jenis dokumen transaksi
- c) Nomor dokumen
- d) Tanggal *quotation* dikeluarkan
- e) Batas tanggal *quotation* valid (opsional)
- f) Nama pelanggan/ *partner* bisnis
- g) Alamat *partner*
- h) Daftar harga
- i) Metode pembayaran
- j) Aturan pembayaran
- k) Lokasi *warehouse*
- l) Alasan penolakan *quotation* (opsional)

Daftar Line (Produk)

- a) Nomor *line*
- b) Jenis produk
- c) Jumlah pemesanan
- d) Satuan produk
- e) Harga produk
- f) Total harga
- g) Pajak (opsional)

Setelah seluruh bagian tersebut terisi, maka sistem akan memeriksa apakah *quotation* tersebut diterima atau ditolak. Jika tidak ada kondisi penolakan *quotation* yang terjadi, maka *quotation* disetujui untuk dibuat. Selanjutnya, perusahaan dapat membuat *sales order* dengan berdasarkan *quotation* tersebut.

2. *Sales Order*

Sales order sendiri merupakan dokumen/ formulir yang digunakan untuk mengelola penjualan dari suatu pemesanan. *Sales order* pada Openbravo dibuat oleh perusahaan berdasarkan *purchase order* yang diberikan oleh pelanggan. *Sales order* ini bisa dapat dibuat berdasarkan 2 cara, yakni: Generate dari *Quotation*

Sales order dibuat melalui *sales quotation* yang telah dihasilkan sebelumnya. Tiap *quotation* yang telah selesai dibuat akan selalu diteruskan ke daftar *sales order* untuk ditinjau lebih lanjut sebagai pembayaran

Buat *Sales Order* Baru

Sales order ini dapat dibuat langsung oleh perusahaan sendiri. Berikut data yang harus diisikan dalam membuat *sales order* baru:

- a) Nama perusahaan
- b) Jenis dokumen transaksi
- c) Nomor dokumen
- d) Tanggal order
- e) Jadwal tanggal pengiriman
- f) Nama pelanggan/ *partner* bisnis
- g) Alamat *partner*
- h) Daftar harga
- i) Metode pembayaran
- j) Aturan pembayaran
- k) Lokasi *warehouse*
- l) Aturan *invoice*

Apabila tidak terdapat kondisi penolakan yang dipenuhi, maka *sales order* dapat dibuat atau di-*book* untuk masuk dalam proses pembayaran. Pada proses pembayaran, seluruh

data dari *sales order* akan disalin pada dokumen *payment* (pembayaran). Terdapat pula beberapa tambahan data yang terdiri atas:

- a) Nomor dokumen *payment*
- b) Nomor referensi (opsional)
- c) Tanggal pembayaran
- d) Tujuan deposit
- e) Jenis transaksi

3. Good Shipment

Proses ini terjadi selama persiapan hingga pelaksanaan pengiriman barang kepada pelanggan. *Good shipment* menyangkut pembuatan dan pengelolaan pengiriman barang ke pelanggan yang ditempuh dengan cara pencatatan dokumen terkait pengiriman tersebut.

Good shipment ini dibuat langsung oleh perusahaan sendiri berdasarkan daftar barang dan stok yang ia miliki. Berikut data yang harus diisikan dalam membuat dokumen pengiriman barang baru:

- a) Nama perusahaan
- b) Jenis dokumen transaksi
- c) Nomor dokumen
- d) Nama dan jenis *warehouse*
- e) Nama pelanggan/ *partner* bisnis
- f) Alamat *partner*
- g) Tanggal pengiriman
- h) Lokasi pengiriman (opsional)
- i) Deskripsi (opsional)

Dalam pembuatan *good shipment* yang baru, diperlukan pula pengisian daftar *line* produk baru.

Tiap daftar *good shipment* yang telah dikelola dan ditinjau dalam seluruh detailnya dapat difinalisasi (*complete*). Apabila telah diselesaikan atau difinalisasi, artinya barang telah siap dikirimkan pada pelanggan (*post*).

4. Return from Customer

Dokumen ini bertujuan untuk memenuhi proses dalam pengembalian barang yang telah diterima pelanggan kepada perusahaan kembali. Pengembalian ini dapat diakibatkan berbagai hal, antar lain adalah *cancelled order*, *damaged on arrival*, *incorrect amount*, *incorrect order*, dan *spoiled product*. Beberapa data yang perlu diisi dalam dokumen ini adalah:

- a) Nama perusahaan
- b) Jenis dokumen transaksi
- c) Nomor dokumen
- d) Tanggal *order*
- e) Nama pelanggan/ *partner* bisnis
- f) Alamat *partner*
- g) Nama dan jenis *warehouse*
- h) Aturan *invoice*
- i) Aturan pembayaran
- j) Metode pembayaran (opsional)
- k) Daftar harga
- l) Alasan pengembalian (opsional)
- m) Deskripsi (opsional)

Melalui dokumen pengembalian yang diperoleh dari pelanggan ini, perusahaan memiliki 2 pilihan. Pertama, ia dapat memilih 'Pick/Edit Line' untuk mengubah dan memasukkan *line* produk berdasarkan hasil barang yang telah dikirimkan. Kedua adalah 'Book' untuk menyetujui dan menerima pengembalian barang tersebut.

5. Return Material Receipt

Dokumen ini sangat berhubungan dengan proses *return from customer*. Pada tahapan ini, perusahaan menerima kembali barang hasil pengembalian dari pelanggan. Perusahaan juga meninjau kembali barang yang ia terima kembali tersebut. Kondisi barangnya dibedakan menjadi *damaged on arrival*, *incorrect product*, dan *in good condition*.

Detail data dalam dokumen ini kurang lebih sama dengan dokumen pada *return from customer*. Penyesuaian hanya

pada jenis dokumen, nomor dokume, dan tanggal penerimaan pengembalian barang.

6. Create Shipment from Orders

Proses ini juga dapat ditempuh dengan tujuan yang sama dengan *good shipment*, yakni pengelolaan pengiriman barang kepada pelanggan. Perbedaannya, dokumen ini dibuat melalui order yang telah dihasilkan dari dokumen *sales order*. Tiap order yang telah selesai dibuat pada tahapan *sales order*, akan selalu diteruskan ke daftar *shipment* ini untuk ditinjau dan diproses lebih lanjut. Apabila order tersebut telah diproses, maka finalisasi pengiriman barang pada pelanggan dijalankan.

7. Sales Invoice

Dokumen ini ialah digunakan dalam pengolaan *invoice* bagi pelanggan. Seluruh daftar *invoice* yang telah dibuat akan ditampilkan pada dokumen ini, baik *invoice* yang dibuat secara manual, maupun dari dokumen *sales order*. Tiap daftar *invoice* tersebut dapat ditinjau terkait detail seluruh bagian datanya. Perusahaan yang meninjau dapat mengambil 3 macam tindakan, yakni 'Post' untuk meneruskan hasil *invoice* pada *ledger* (buku kas), 'Reactivate' untuk menunda atau membatalkan *invoice*, dan 'Add Payment' untuk mengelola pembayaran terhadap *invoice* tersebut. Selain itu, *invoice* tersebut juga dapat dicetak atau dikirimkan melalui *email*

8. Create Invoice from Orders

Proses ini dilakukan dengan tujuan untuk membuat dokumen *invoice*. Namun dokumen ini dibuat melalui order yang telah dihasilkan dari dokumen *good shipment*. Tiap order yang telah selesai dikirimkan kepada pelanggan pada tahapan *good shipment*, akan kemudian diteruskan ke daftar *create invoice from orders*. Perusahaan dapat meninjau dan memproses daftar order lebih lanjut. Apabila order tersebut telah diproses, maka *generate invoice* telah dilakukan dan dimasukkan ke dalam daftar *invoice* yang ada di dokumen *sales invoice*.

9. Generate Invoice

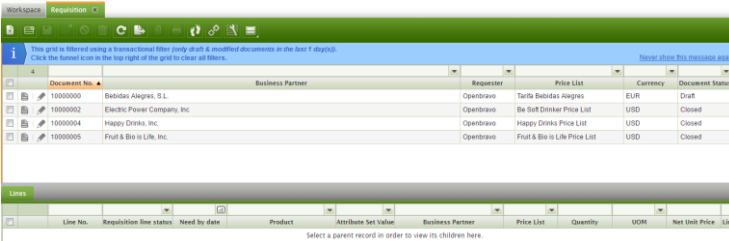
Tahapan ini ialah pembuatan *invoice* untuk dimasukkan dalam dokumen *sales invoice*. *Invoice* yang dibuat ini tidak berarti dapat dibuat secara manual, namun harus berdasarkan daftar *sales order* yang telah ada. Pembuatan *invoice* ini ialah pada *sales order* yang masih tertunda (*pending*) untuk ditagih. *Sales order* yang dapat dibuat *invoice* ialah order yang telah memenuhi kriteria sebagai *sales invoice*. Data yang diisikan untuk *generate invoice* ini antara lain adalah:

- a) Tanggal *invoice*
- b) Perusahaan yang membuat *invoice*
- c) Keterangan *child organization*
- d) Pilihan *sales order*
- e) Nama pelanggan/ *partner* bisnis
- f) Tanggal pengiriman *invoice*

Hasil *invoice* yang dibuat ini akan diteruskan pada daftar di dokumen *sales order*.

5.2.2.2 Purchase

Pada Openbravo *Purchase Management* disebut sebagai *Procurement Management*. Modul ini berkaitan dengan seluruh aktivitas dalam pembayaran barang dan jasa yang berasal dari *supplier* serta berbagai pelaporan yang dibutuhkan. Gambar 5.18 adalah contoh tampilan tampilan *purchase* pada Openbravo.



The screenshot shows the Openbravo Requisition interface. At the top, there is a navigation bar with 'Application', 'Alerts (1)', 'Help', and 'Openbravo'. Below the navigation bar, there is a table with the following columns: Document No., Business Partner, Requester, Price List, Currency, and Document Status. The table contains four rows of data:

Document No.	Business Partner	Requester	Price List	Currency	Document Status
1000000	Bebidas Alegres, S.L.	Openbravo	Tarta Bebidas Alegres	EUR	Draft
1000002	Electric Power Company, Inc.	Openbravo	Be Soft Drinkers Price List	USD	Closed
1000004	Happy Drinks, Inc.	Openbravo	Happy Drinks Price List	USD	Closed
1000005	Fruit & Bio to Life, Inc.	Openbravo	Fruit & Bio to Life Price List	USD	Closed

Below the table, there is a 'Lines' section with a table that has columns: Line No., Requisition line status, Need by date, Product, Attribute Set Value, Business Partner, Price List, Quantity, UOM, Net Unit Price, and Lin. The table is currently empty, and a message below it says 'Select a parent record in order to view its children here.'

Gambar 5.18 Contoh Tampilan *Purchase* Openbravo

1. Requisition

Requisition merupakan formulir permintaan secara formal yang dibutuhkan perusahaan. Tiap *requisition* ini berisi *item*

atau barang-barang yang dibutuhkan disertai vendor yang diinginkan dan daftar harganya. Data yang diisikan dalam dokumen *requisition* ini ialah:

Daftar Requisition

- a) Nama perusahaan
- b) Nomor dokumen
- c) Nama vendor/ *partner* bisnis
- d) *User* pemohon
- e) Daftar harga (opsional)
- f) Deskripsi (opsional)

Daftar Line (Produk)

- a) Nomor *line*
- b) Tanggal produk dibutuhkan
- c) Jenis produk
- d) Jumlah pemesanan
- e) Satuan produk
- f) Harga produk (opsional)
- g) Total harga (opsional)
- h) Nama vendor/ *partner* bisnis (opsional)
- i) Daftar harga (opsional)
- j) Diskon (opsional)
- k) Deskripsi (opsional)

Setelah seluruh bagian tersebut terisi, maka *requisition* tersebut akan diberikan kepada *procurement team* untuk dikelola dan ditinjau lebih lanjut.

2. Manage Requisitions

Tahapan ini merupakan ini dimaksudkan untuk digunakan oleh *procurement team* perusahaan agar memperoleh gambaran mengenai keseluruhan barang yang dibutuhkan. Setelah seluruh *requisition* dibuat, tim ini yang akan memutuskan bagaimana kondisi *requisition* yang dibuat oleh anggota atau departemen lain dari perusahaan tersebut.

Procurement team dapat mengubah kembali data *requisition* yang telah dibuat. Apabila telah selesai meninjaunya, maka ia dapat menyelesaikan *requisition* tersebut sehingga statu

draft akan berganti menjadi *complete*. Kemudian terdapat 2 macam pilihan, yakni ‘Create *Purchase Order*’ untuk menyetujui *requisition* dan meneruskannya sebagai *purchase order* (PO) serta ‘Close’ untuk menolak *requisition* dan membatalkan permintaan tersebut.

3. Requisition to Order

Tahapan ini sama halnya dengan tahapan akhir pada ‘Manage Requisition’, dimana tiap *requisition* yang telah diselesaikan dan berstatus *complete* akan masuk pada daftar ini. Tindakan yang dapat dilakukan juga sama, *procurement team* dapat menolak *requisition* dengan memilih ‘remove’ atau menyetujuinya dengan pilihan ‘Add’. Apabila disetujui dan hendak diteruskan sebagai PO, maka terdapat pilihan ‘Create’ untuk membuat PO berdasarkan data pada *requisition*.

4. Purchase Order

Purchase order merupakan dokumen yang akan diberikan kepada vendor. *Procurement team* dapat membuat dan mengelola order yang dibuat untuk dikirim ke pemasok/*supplier* eksternal. Dokumen ini digunakan untuk mendaftarkan produk atau jasa yang akan dibeli oleh perusahaan. Setelah dokumen diselesaikan (*booked*), maka akan dikirim ke *supplier* melalui surat atau *email* dan dapat dibayarkan DP jika diperlukan. Selain pilihan *book*, *procurement team* dapat pula memilih *process* jika order tersebut masih dapat diproses namun belum difinalisasi karena akan perubahan atau *void* untuk membatalkan order karena tidak dibutuhkan lagi.

Apabila dokumen order difinalisasi, maka akan masuk pada pembayaran. Berikut data pembayaran yang harus diisikan:

- a) Nomor dokumen pembayaran
- b) Nomor referensi
- c) Metode pembayaran
- d) Tanggal pembayaran
- e) Sumber dana pembayaran

5. Good Receipt

Dokumen *good receipt* digunakan untuk pencatatan barang pesanan perusahaan yang telah masuk. Secara fisik, *warehouse team* adalah penanggungjawab dalam penerimaan barang dan penataannya ke dalam *warehouse*. Dalam pencatatannya, terdapat beberapa data yang perlu diisikan, yakni antara lain:

- a) Nama organisasi
- b) Nomor dokumen
- c) Jenis dokumen
- d) Nama vendor/ *partner* bisnis
- e) Alamat *partner*
- f) Lokasi *warehouse*
- g) Tanggal perpindahan barang
- h) Tanggal akuntansi barang
- i) *Purchase order* (opsional/ jika *good receipt* dari *purchase order* yang telah dibuat)
- j) Referensi order (opsional)
- k) Deskripsi (opsional)

Apabila daftar pencatatan tersebut telah terisi, maka dilanjutkan dengan pengisian data *line* produk untuk detail produk yang masuk. Dokumen *good receipt* yang telah difinalisasi akan menyebabkan kuantitas barang pada *warehouse* menjadi bertambah

6. Pending Good Receipt

Tahapan ini ditempuh untuk mengelola *purchase order* yang belum terselesaikan pencatatan barangnya. Jika belum dicatat pada dokumen *good receipt*, maka *purchase order* tersebut akan masuk pada daftar ini. Apabila order tersebut telah masuk, maka cukup memasukkan tanggal kedatangan barang dan lokasi *warehouse*. Setelah itu dokumen dapat diproses dan diteruskan untuk menambah kuantitas barang yang telah masuk tersebut.

7. Purchase Invoice

Purchase invoice adalah dokumen untuk mendaftar dan mengelola *invoice* dari vendor. Dokumen *purchase invoice*

berisikan pernyataan barang atau jasa yang disediakan oleh vendor/ *supplier* berupa kuantitas dan harganya. Tiap daftar *invoice* tersebut dapat ditinjau terkait detail seluruh bagian datanya. Perusahaan yang meninjau dapat mengambil 3 macam tindakan, yakni 'Post' untuk meneruskan hasil *invoice* pada *ledger* (buku kas), 'Reactivate' untuk menunda atau membatalkan *invoice*, dan 'Add Payment' untuk mengelola pembayaran terhadap *invoice* tersebut. Selain itu, *invoice* tersebut juga dapat dicetak atau dikirimkan melalui *email*

8. Matched Invoice

Tahapan ini digunakan untuk meninjau penyesuaian antara inventori barang dengan catatan akuntansi barang tersebut. Tiap kuantitas barang yang masuk harus sesuai dengan jumlah pengeluaran yang akan dibayarkan pada perusahaan kepada vendor/ *supplier*. Apabila telah sesuai antara kedua bagian tersebut, artinya pembelian barang tidak mengalami masalah dan dapat dilanjutkan (*post*) pada tahapan pembayaran

9. Return to Vendor

Dokumen ini bertujuan untuk memenuhi proses dalam pengembalian barang yang telah diterima perusahaan kepada vendor kembali. Pengembalian ini dapat diakibatkan berbagai hal, antar lain adalah *cancelled order*, *damaged on arrival*, *incorrect amount*, *incorrect order*, dan *spoiled product*. Beberapa data yang perlu diisi dalam dokumen ini adalah:

- a) Nama perusahaan
- b) Jenis dokumen transaksi
- c) Nomor dokumen
- d) Tanggal *order*
- e) Nama pelanggan/ *partner* bisnis
- f) Alamat *partner*
- g) Nama dan jenis *warehouse*
- h) Aturan pembayaran
- i) Metode pembayaran (opsional)

j) Nomor referensi pengembalian (opsional)

k) Alasan pengembalian (opsional)

Melalui dokumen pengembalian yang dibuat ini, perusahaan memiliki 2 pilihan. Pertama, ia dapat memilih ‘Pick/Edit Line’ untuk mengubah dan memasukkan line produk atas barang yang akan diubah atau ditambahkan. Kedua adalah ‘Book’ untuk menyetujui dan melakukan finalisasi atas pengembalian barang kepada vendor

10. Return to Vendor Shipment

Tahapan ini merupakan kelanjutan dari ‘Return to Vendor’, yakni dilakukan saat pengiriman barang kembali pada vendor. Pada tahapan ini berpengaruh pula pada kondisi kuantitas barang pada *warehouse*. Saat pengiriman barang dilakukan, inventori barang tentu akan berkurang. Data yang ada di dalam dokumen ini antara lain:

a) Nama perusahaan

b) Jenis dokumen transaksi

c) Nomor dokumen

d) Nama pelanggan/ *partner* bisnis

e) Alamat *partner*

f) Tanggal perpindahan barang

g) Tanggal akuntansi barang

h) Nama dan jenis *warehouse*

i) Nomor referensi pengembalian (opsional)

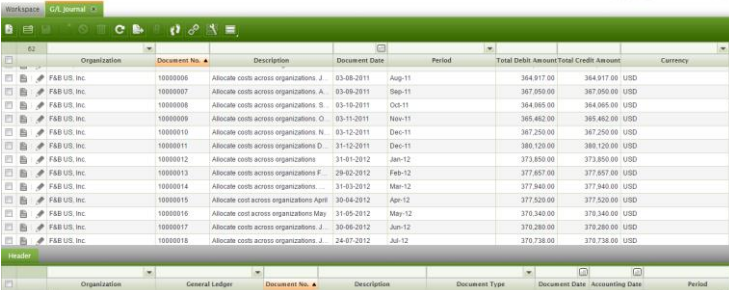
j) Dekripsi (opsional)

11. Landed Cost

Dokumen ini berguna untuk mengalokasikan biaya tambahan seperti, transportasi, asuransi, atau pekerjaan dalam penerimaan barang. Hal ini dibutuhkan karena biaya barang tambahan termasuk dalam tahapan penerimaan barang dan perlu disesuaikan dalam pencatatan akuntansinya. Pencatatan biaya ini tidak wajib ada, hanya diperlukan jika memang biaya tambahan timbul pada saat penerimaan barang.

5.2.2.3 Accounting

Akuntansi pada Openbavo termasuk dalam Financial Management. Pencatatan akuntansi ini tidak terdaftar tiap dokumen bagiannya secara terurut. Akuntansi yang disediakan adalah dalam bentuk macam-macam dokumen terkait pencatatan keuangan. Berikut gambar 5.19 menunjukkan contoh tampilan halaman *accounting* pada Openbravo.



Organization	Document No.	Description	Document Date	Period	Total Debit Amount	Total Credit Amount	Currency
FAB US, Inc.	10000006	Allocate costs across organizations J.	03-08-2011	Aug-11	364,917.00	364,917.00	USD
FAB US, Inc.	10000007	Allocate costs across organizations A.	03-08-2011	Sep-11	367,050.00	367,050.00	USD
FAB US, Inc.	10000008	Allocate costs across organizations B.	03-10-2011	Oct-11	364,066.00	364,066.00	USD
FAB US, Inc.	10000009	Allocate costs across organizations O.	03-11-2011	Nov-11	365,462.00	365,462.00	USD
FAB US, Inc.	10000010	Allocate costs across organizations H.	03-12-2011	Dec-11	367,250.00	367,250.00	USD
FAB US, Inc.	10000011	Allocate costs across organizations D.	31-12-2011	Dec-11	368,120.00	368,120.00	USD
FAB US, Inc.	10000012	Allocate costs across organizations G.	31-01-2012	Jan-12	373,850.00	373,850.00	USD
FAB US, Inc.	10000013	Allocate costs across organizations F.	29-02-2012	Feb-12	377,857.00	377,857.00	USD
FAB US, Inc.	10000014	Allocate costs across organizations.	31-03-2012	Mar-12	377,846.00	377,846.00	USD
FAB US, Inc.	10000015	Allocate costs across organizations April	30-04-2012	Apr-12	373,536.00	373,536.00	USD
FAB US, Inc.	10000016	Allocate costs across organizations May	31-05-2012	May-12	379,340.00	379,340.00	USD
FAB US, Inc.	10000017	Allocate costs across organizations J.	30-06-2012	Jun-12	370,280.00	370,280.00	USD
FAB US, Inc.	10000018	Allocate costs across organizations J.	24-07-2012	Jul-12	370,738.00	370,738.00	USD

Gambar 5.19 Contoh Tampilan *Accounting* Openbravo

1. Simple G/L Journal

Dokumen ini merupakan bentuk ringkasan yang lebih sederhana dari buku kas besar (*G/L journal*). Melalui dokumen ini, perusahaan dapat langsung memantau pada pencatatan akuntansi yang belum terselesaikan atau membutuhkan tindak lanjut. Dokumen ini juga memungkinkan pembuatan catatan akuntansi secara manual, dimana data-datanya terdiri atas:

- Nama perusahaan
- Nomor dokumen
- Deskripsi (opsional)
- Tanggal dokumen (opsional)
- Tanggal akuntansi (opsional)
- Periode (opsional)
- Mata uang (opsional)

2. G/L Journal

G/L (General Ledger) journal merupakan buku kas besar yang digunakan untuk pencatatan seluruh aliran kas masuk dan keluar beserta pembayaran yang dilakukan. Seluruh

pencatatan dari modul-modul lain yang berhubungan dengan *invoice*, pembayaran, dan kondisi-kondisi yang menyebabkan perubahan keuangan perusahaan akan tercatat secara menyeluruh pada dokumen ini. Selain pencatatan secara otomatis melalui tahapan yang sudah dilakukan modul-modul lain, dimungkinkan pula pencatatan manual pada dokumen ini. Data yang perlu dimasukkan antara lain:

- a) Nama perusahaan
- b) Nomor dokumen
- c) Deskripsi (opsional)
- d) Tanggal dokumen (opsional)
- e) Tanggal akuntansi (opsional)
- f) Periode (opsional)
- g) Mata uang (opsional)

Data yang dimasukkan secara manual pada dokumen ini akan diproses secara *batch* atau keseluruhan sebagai catatan dalam buku kas besar perusahaan pada tahun tersebut.

3. End Year Close

Dokumen ini bertujuan sebagai rangkuman pencatatan akuntansi akhir tahun perusahaan. Pencatatan dokumen ini akan menyebabkan seluruh akuntansi dalam satu tahun ditutup/ diselesaikan. Catatan akuntansi yang belum terselesaikan dapat dilanjutkan pada tahun berikutnya dengan tambahan pengecualian tertentu. Tiap catatan yang telah ditutup berarti telah selesai dan tidak akan dibahas lagi pada buku kas besar di awal tahun.

Dokumen ini yang menjadi dasar dalam laporan tahunan perusahaan. Terdapat 3 jenis laporan yang dapat dihasilkan dari dokumen ini. Pertama ialah 'Profit and Loss Closing' untuk laporan keuntungan dan kerugian yang dialami perusahaan selama setahun. Kedua, 'Balance Sheet Closing' untuk laporan neraca keuangan akhir tahun perusahaan (debit dan kredit). Ketiga adalah 'Balance Sheet Opening' untuk membuat laporan neraca keuangan perusahaan di awal tahun berdasarkan kondisi keuangan di akhir tahun sebelumnya

4. GL Posting by DB Tables

GL Posting by DB Tables merupakan tahapan untuk memasukkan catatan akuntansi pada buku kas besar ke dalam *database* yang tersedia pada Openbravo. Catatan yang dimasukkan ini dapat dibedakan berdasarkan beberapa kriteria yang dimasukkan sebelum *entry* data ke *database*, yakni antar lain:

- a) Nama perusahaan
- b) Jenis tabel
- c) Tanggal awal
- d) Tanggal akhir

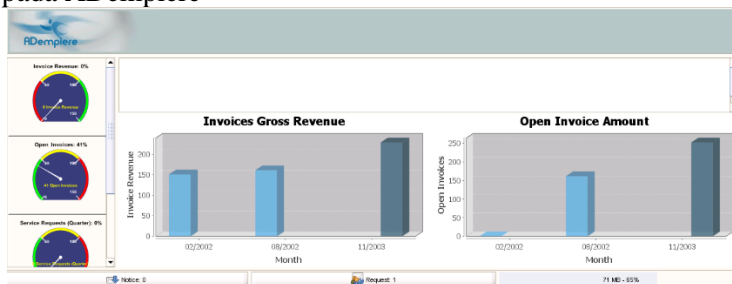
5. Budget

Dokumen *budget* digunakan untuk membuat dan mengelola *budget* perusahaan dalam rangka tujuan khusus/ tertentu. Dokumen ini dibuat di awal tahun dengan data-datanya antar lain:

- a) Nama perusahaan
- b) Nama perencanaan *budget*
- c) Tahun perencanaan
- d) Deskripsi (opsional)

5.2.3 ADempiere

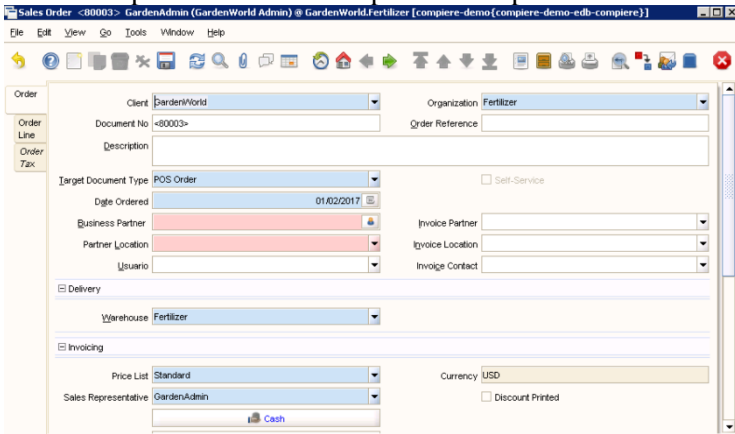
Berikut gambar 5.20 menunjukkan tampilan halaman utama pada ADempiere



Gambar 5.20 Tampilan Halaman Utama ADempiere

5.2.3.1 Sales

Modul *sales* dalam ADempiere tersedia pada menu/ modul bernama *Quote-to-Invoice*. Modul ini mengelola seluruh aktivitas penjualan dari mulai pembutan *quotation/ quote* hingga menghasilkan *invoice*. Gambar 5.21 menunjukkan contoh tampilan halaman *sales* pada ADempiere.



Gambar 5.21 Contoh Tampilan Sales ADempiere

1. Sales and Marketing

Fitur ini merupakan fitur opsional/ tambahan dalam modul *sales* ADempiere, dimana fungsinya adalah melakukan perencanaan *sales* dan marketing layaknya pada CRM. Fitur ini mencakup aktivitas-aktivitas antara lain:

a) Sales Setup

Pendefinisian alur kerja (*workflow*) dalam *sales*, yakni antara *business partner*, *user*, *sales region*, *commission*, *marketing channel*, hingga *marketing campaign*

b) Sales Region

Penentuan tiap area penjualan yang dicakup perusahaan. Tiap *sales region* harus ditentukan nama tempatnya, deskripsi, dan penanggungjawab (*sales representative*)

c) Commission

Komisi yang dimaksud adalah bayaran yang diberikan perusahaan bagi *sales representative* atau *business*

partner yang menjalankan penjualan kepada pelanggan. Jadi didefinisikan jumlah bayarannya, penerimanya, metode pembayaran, serta waktu dan frekuensi pembayaran

d) Commission Run

Setelah komisi didefinisikan, maka aktivitas ini ialah menjalankan pengelolaan komisi. Aktivitas ini membutuhkan pencatatan dokumen berupa data-data dari *commission* dengan tambahan rentang waktu, total biaya, dan detail *invoice*

e) Commission Run Detail

Dokumen yang telah dibuat dapat diidentifikasi dan dicari pada bagian ini. Daftar dokumen komisi yang telah dibuat juga dapat diunduh sebagai pdf ataupun dicetak *offline*

f) Marketing Channel

Pendefinisan saluran/ metode *marketing* perusahaan. Data yang perlu diisikan yakni nama *channel*, deskripsi, status, dan warna pencetakan *marketing*

g) Marketing Campaign

Pendefinisan aktivitas *marketing* sesuai dengan metode yang telah ditentukan pada *marketing channel*. Data yang diperlukan ialah nama *campaign*, deskripsi, nama *channel*, tanggal mulai dan akhir, serta jumlah biaya

2. Sales Order

Dalam pembuatan *sales order*, terdapat berbagai detail aktivitas yang dicakup di dalamnya. Aktivitas tersebut antara lain ialah:

a) Quotation

Pembuatan *quotation* ditujukan pada target pelanggan yang akan melakukan pembelian pada perusahaan. Detail data di dalamnya mencakup informasi umum *quotation*, detail barang yang ditawarkan, serta biaya yang diperlukan

b) New Sales Order

Setiap pembuatan *sales order* baru, maka terdapat berbagai data yang harus diisikan dari mulai informasi umum order, detail barang yang dipesan, pajak dan biaya order, *warehouse* pengiriman, metode penagihan (*invoicing*), referensi tambahan, serta detail status order

c) Order List

Dokumen order yang telah dibuat dapat diidentifikasi dan dicari. Pencarian daftar dokumen juga dapat melalui berbagai cara, baik melalui *order detail* (detail data pada *sales order*), *open order* (order yang masih belum ditangani), maupun *order transaction* (detail rentang waktu transaksi order). Daftar dokumen order yang telah dibuat juga dapat diunduh sebagai pdf ataupun dicetak *offline*

d) Order Setting

Terdapat berbagai pengaturan *sales order* yang dapat dilakukan, dari mulai konversi *quotation* menjadi order, perubahan harga order, pembuatan *sales order* dari PO, perubahan status order, maupun pemrosesan beberapa order dalam satu waktu

e) Subscription

Subscription merupakan pengelolaan terkait aktivitas berlangganan dari *customer* bagi perusahaan. Pengelolaan ini digunakan untuk menentukan metode berlangganan dan melihat daftar *subscription*

3. Shipment

Pengiriman order yang dilakukan perusahaan dapat ditempuh dengan 2 cara, yakni manual dan otomatis. Manual apabila dokumen pengiriman dibuat baru tanpa adanya referensi dokumen *sales order*. Otomatis jika dokumen diperoleh dari *sales order* yang diteruskan. Selain data-data dari *sales order*, dilengkapi pula tambahan data detail pengiriman, yakni:

a) Detail *warehouse* (nama, aturan, *sales representative*, dan prioritas)

- b) Metode pengiriman (saluran dan biaya pengiriman)
- c) Status pengiriman (perpindahan barang, status dokumen, dan tanggal barang sampai)

Dokumen yang telah dibuat dapat diidentifikasi dan dicari pada shipment details. Daftar dokumen pengiriman juga dapat diunduh sebagai pdf ataupun dicetak offline, baik sebagai arsip perusahaan maupun untuk diberikan pada pelanggan

4. *Sales Invoice*

Pembuatan dokumen tagihan/ invoice atas barang yang telah dibeli pelanggan juga dapat ditempuh dengan 2 cara, yakni manual dan otomatis. Manual apabila dokumen invoice dibuat baru tanpa adanya referensi dokumen sebelumnya. Otomatis jika dokumen diperoleh dari *sales order* yang diteruskan. Selain data-data dari *sales order*, dilengkapi pula tambahan data detail penagihan, yakni:

- a) Status *invoice* (total tagihan dan status dokumen)
- b) Penjadwalan pembayaran (metode pembayaran, batas waktu tanggal, diskon dan pajak)

Dokumen yang telah dibuat dapat diidentifikasi dan dicari pada print invoice. Daftar dokumen invoice juga dapat diunduh sebagai pdf ataupun dicetak offline, baik sebagai arsip perusahaan maupun untuk diberikan pada pelanggan. Dalam pembayarannya sendiri, dokumen tersebut tidak termasuk dalam modul Quote-to-Invoice, namun ditangani pada modul Open Items.

5.2.3.2 *Purchase*

Modul *purchase* dalam ADempiere tersedia pada menu/ modul bernama Requisition-to-Invoice. Modul ini mengelola seluruh aktivitas pembelian dari mulai pembuatan RFQ hingga menghasilkan invoice. Berikut gambar 5.22 menunjukkan contoh tampilan halaman *purchase* pada ADempiere.

Gambar 5.22 Contoh Tampilan *Purchase* ADempiere

1. RFQ

RFQ atau Request for Quotation merupakan dokumen yang diajukan perusahaan untuk meminta formulir quotation dari vendor. Data-data yang perlu diisikan antara lain ialah:

- a) Informasi umum dokumen
- b) Detail vendor/ *business partner* yang digunakan
- c) Detail produk yang diinginkan

Dokumen RFQ yang telah dibuat dapat diidentifikasi dan dicari. Pencarian daftar dokumen dapat dicari baik berdasarkan RFQ yang direspon (RFQ Response), maupun yang tidak (RFQ Unanswered). Daftar dokumen RFQ juga dapat diunduh sebagai pdf ataupun dicetak offline untuk diberikan pada vendor

2. Requisition

Quotation yang telah dikirimkan oleh vendor dapat ditindaklanjuti oleh perusahaan. Dokumen requisition merupakan dokumen permintaan order barang kepada vendor. Berbeda dengan RFQ, dokumen ini ditujukan hanya pada satu vendor yang disepakati perusahaan. Data-data yang perlu diisikan antara lain ialah:

- a) Informasi umum dokumen
- b) Detail produk yang diinginkan

c) Status dokumen

Dokumen requisition yang telah dibuat dapat diidentifikasi dan dicari pada open requisition. Daftar dokumen requisition juga dapat diunduh sebagai pdf ataupun dicetak offline untuk diberikan pada vendor

3. *Purchase Order*

Pembuatan dokumen PO yang dilakukan perusahaan dapat ditempuh dengan 2 cara, yakni manual dan otomatis. Manual apabila dokumen PO dibuat baru tanpa adanya referensi dokumen requisition. Otomatis jika dokumen diperoleh dari requisition yang diteruskan. Selain data-data dari requisition, dilengkapi pula tambahan data detail pengiriman, yakni:

- a) Pajak dan biaya order
- b) *Warehouse* penerimaan
- c) Metode pengiriman
- d) Metode penagihan (*invoicing*)
- e) Referensi tambahan (opsional)

Daftar dokumen PO yang telah dibuat dapat diunduh sebagai pdf ataupun dicetak offline untuk diberikan pada vendor

4. *Material Receipt*

Material barang yang telah dikirimkan oleh vendor akan masuk dalam inventori perusahaan. Penerimaan material ini harus dicatat dalam dokumen material receipt. Data yang dicatat antara lain ialah:

- a) Informasi umum dokumen
- b) Detail material barang yang diterima
- c) Detail *warehouse* (nama, aturan, agen perusahaan, dan prioritas)
- d) Status penerimaan (perpindahan barang, status dokumen, dan tanggal barang sampai)

Dokumen yang telah dibuat dapat diidentifikasi dan dicari pada material receipt details. Daftar dokumen penerimaan juga dapat diunduh sebagai pdf ataupun dicetak offline, baik sebagai arsip perusahaan maupun untuk diberikan pada vendor terkait konfirmasi order yang telah didapat

5. Vendor Invoice

Pembuatan dokumen tagihan/ invoice atas barang yang telah diterima perusahaan dapat ditempuh dengan 2 cara, yakni manual dan otomatis. Manual apabila dokumen invoice dibuat baru tanpa adanya referensi dokumen sebelumnya. Otomatis jika dokumen diperoleh dari *purchase* order dan material receipt yang diteruskan. Selain data-data dari referensi dokumen, dilengkapi pula tambahan data detail penagihan, yakni:

- a) Status *invoice* (total tagihan dan status dokumen)
- b) Penjadwalan pembayaran (metode pembayaran, batas waktu tanggal, diskon dan pajak)

Dokumen invoice juga dapat dikelola untuk pembayaran beberapa invoice dalam satu waktu, yakni pada invoice batch. Daftar dokumen invoice juga dapat diunduh sebagai pdf ataupun dicetak offline, baik sebagai arsip perusahaan maupun untuk diberikan pada vendor

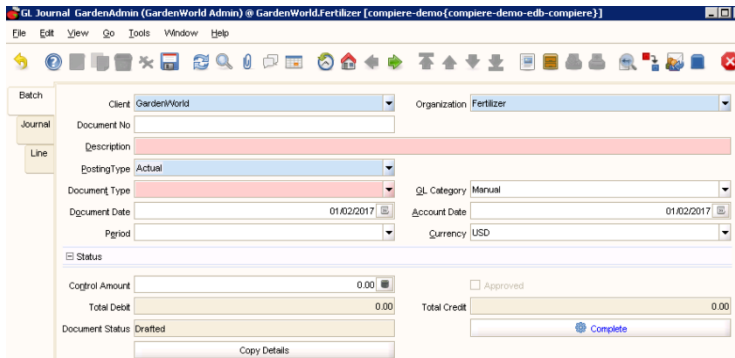
Dalam pembayarannya sendiri, dokumen tersebut tidak termasuk dalam modul Requisition-to-Invoice, namun ditangani pada modul Open Items.

6. Matching

Fitur ini digunakan untuk menyesuaikan dokumen-dokumen yang masih belum bersesuaian. Jadi antara dokumen *purchase* order, penerimaan barang (material receipt), maupun penagihannya (invoice) harus saling bersesuaian, jelas, dan runut alur kerjanya. Apabila terjadi permasalahan pada salah satu atau beberapa dokumen, maka fitur ini yang digunakan untuk mengatasi dan meluruskannya

5.2.3.3 Accounting

Modul *accounting* pada ADempiere termasuk dalam modul *Performance Analysis*. Aktivitas akuntansi perusahaan dalam modul ini ditangani oleh submodul *Financial Reporting*. Berikut gambar 5.23 menunjukkan contoh tampilan halaman *accounting* pada ADempiere.



Gambar 5.23 Contoh Tampilan Accounting ADEMPIERE

1. Financial Report

Laporan finansial ini merupakan gabungan dari seluruh laporan akuntansi perusahaan. Tiap laporan seperti Balance Sheet, Income Statement, dan Statement of Cash Flow dikelola dan ditinjau dari laporan ini. Tujuannya adalah untuk memudahkan pemantauan status dari tiap laporan akuntansi perusahaan

2. Statement of Accunt

Laporan akun dapat diidentifikasi dan dikelola melalui fitur ini. Fitur ini digunakan untuk mencari dan menghasilkan dokumen terkait budget, actual, commitment, resevation, dan statistical akuntansi perusahaan

3. Trial Balance

Laporan trial balance merupakan neraca saldo perusahaan. Laporan ini adalah daftar yang memuat seluruh akun dengan saldo totalnya yang bersumber dari buku kas besar (General Ledger) perusahaan. Fitur ini digunakan untuk mencari dan menghasilkan laporan neraca saldo perusahaan.

4. GL Journal

Buku kas besar yang digunakan untuk pencatatan seluruh aliran kas masuk/ piutang (debit) dan keluar/ hutang (kredit) beserta pembayaran yang dilakukan. Fitur ini digunakan untuk menghasilkan laporan buku kas perusahaan yang berasal dari berbagai catatan/ jurnal akuntansi dalam rentang waktu tertentu

5. Cash Journal

Jurnal kas adalah jurnal yang dibuat atau digunakan untuk mencatat semua transaksi perusahaan. Jurnal ini tidak termasuk pada submodul Financial Reporting, melainkan masuk pada modul Open Items. Jurnal ini mencakup 2 macam pencatatan, yakni:

a) *Accounts Receivable*

Pencatatan penerimaan pendapatan/ penghasilan perusahaan. Pencatatan transaksinya berhubungan dengan dokumen-dokumen pada modul *sales (Quote-to-Invoice)*

b) *Accounts Payable*

Pencatatan pengeluaran biaya perusahaan. Pencatatan transaksinya berhubungan dengan dokumen-dokumen pada modul *purchase (Requisition-to-Invoice)*

6. GL Budget

Fitur ini digunakan untuk membuat dokumen perencanaan budgeting perusahaan. Fitur ini terdapat pada modul Performance Analysis namun masuk pada submodul *Accounting Rules*. Dokumen budget yang telah dibuat dapat dipantau pada fitur Budget Control

5.2.4 Dolibarr

Berikut gambar 5.24 menunjukkan tampilan halaman utama pada Dolibarr

The screenshot shows the Dolibarr 'Modules setup' page. At the top, there is a navigation bar with 'Home', 'Third parties', 'Products', 'Commercial', 'Financial', and 'Bank/Cash' tabs. The main content area is titled 'Modules setup' and includes a search bar and a table of available modules. The table lists modules such as 'Human Resource Management (HR)' and 'Customer Relation Management (CRM)', each with a description, version number (4.0.3), and a 'Setup' button. The interface also shows a sidebar with various navigation options like 'Dashboard', 'Setup', 'Company/Foundation', 'Modules', 'Menus', 'Display', 'Translation', 'Widgets', 'Alerts', 'Security', 'Limits and accuracy', 'PDF', 'E-mails', 'SMS', 'Dictionaries', and 'Other setup'.

Available modules	More modules...
Keyword: <input type="text"/> Origin: <input type="text"/> Status: <input type="text"/> Refresh <input type="button"/> Reset <input type="button"/>	
Human Resource Management (HR) Setup	
Users & groups	Users and groups management 4.0.3 <input type="checkbox"/> Requirec ##
Members	Foundation members management 4.0.3 <input type="checkbox"/> ##
Leave Requests management	Declare and follow employees leaves requests 4.0.3 <input type="checkbox"/> ##
Expense reports	Management and claim expense reports (transportation, meal, ...) 4.0.3 <input type="checkbox"/> ##
Salarios	Management of employees salaries and payments 4.0.3 <input type="checkbox"/> ##
Customer Relation Management (CRM) Setup	
Third parties	Companies and contact management (customers, prospects...) 4.0.3 <input type="checkbox"/> ##

Gambar 5.24 Tampilan Halaman Utama Dolibarr

5.2.4.1 Sales

Modul *sales* tidak secara eksplisit disebutkan di Dolibarr. Modul ini termasuk dalam modul CRM, dimana membutuhkan fitur Third Parties dan Commercial untuk mengaksesnya. Fitur Third Parties digunakan untuk membuat daftar pelanggan, sedangkan fitur Commercial untuk membuat order pelanggan. Gambar 5.25 ialah contoh tampilan halaman *sales* Dolibarr.

The screenshot shows the Dolibarr interface for 'List of orders (1)'. It features a search bar, a sidebar with 'Customer orders' and 'New order' options, and a main table with the following data:

Ref. -	Ref. order for customer	Third party	City	Zip Code	Order date	Planned date of delivery	Amount (net of tax)	Status	Billed
CO1012-0001	213	Maspion	Surabaya	60111	12/29/2016	12/23/2016	300.000.00	Delivered	No
Total							300.000.00		

Gambar 5.25 Contoh Tampilan Sales Dolibarr

1. Create Order

Dalam membuat order pelanggan, daftar pelanggan harus sudah didefinisikan pada fitur Third Parties. Bagian subfitur Create Order terdapat pada fitur Commercial yang terhubung dengan daftar pelanggan di fitur Third Parties. Data-data yang perlu diisi dalam membuat order baru adalah sebagai berikut:

- a) Nomor referensi
- b) Nama pelanggan
- c) Tanggal order
- d) Tanggal perencanaan pengiriman
- e) Aturan pembayaran
- f) Metode pembayaran
- g) Tanggal waktu pembayaran
- h) Metode pengiriman
- i) Sumber order
- j) Catatan (opsional)

Setiap order yang telah selesai dibuat akan diteruskan untuk mengisi detail produk pada bagian new line.

2. New Line

Bagian subfitur ini tergabung dengan create order pada subfitur Order card. Tiap line produk yang hendak dibuat, dibedakan atas 2 macam, yakni produk jadi (barang/ jasa) dan produk pre-order. Pilihan tersebut diikuti dengan pengisian beberapa data, yakni:

- a) Nama produk
- b) Pajak (tergantung ketersediaan aturan pajak yang didukung Dolibarr)
- c) Harga produk
- d) Jumlah pembelian
- e) Diskon

Tiap produk yang telah diisikan dan ditambahkan, akan dikalkulasi total biayanya. Jika seluruh detail order, baik informasi umumnya maupun produknya telah diisikan, maka finalisasi order dilakukan dengan pemilihan 'Validate'. Order yang telah selesai dibuat dapat dikirimkan melalui email kepada pelanggan atau dicetak secara offline.

3. Shipment

Dalam pengiriman barang, data yang diperoleh dari subfitur Order card akan disalin pada subfitur ini. Untuk pengirimannya, diperlukan pendefinisian warehouse yang berperan sebagai lokasi pengiriman. Berikut data yang perlu diisikan dalam membuat warehouse:

- a) Nomor referensi
- b) Nama lokasi
- c) Deskripsi (opsional)
- d) Alamat, kode pos, kota
- e) Negara
- f) Status

Jika telah terdapat lokasi *warehouse* untuk pengiriman produk, maka pengiriman dapat dilakukan. Terdapat pula data tambahan yang perlu diisikan sehubungan dengan pengiriman order tersebut, seperti berat, tinggi, lebar, dan *tracking number*. Perusahaan juga dapat menentukan sendiri jumlah barang yang dapat dikirimkan (tidak harus sesuai

dengan jumlah barang yang dipesan). Apabila telah selesai, maka finalisasi pengiriman dilakukan dengan pemilihan 'Validate'. Pengiriman yang telah selesai dibuat dapat dikirimkan melalui *email* kepada pelanggan atau dicetak secara *offline*.

4. Invoice

Barang yang telah dikirimkan kepada pelanggan akan ditindaklanjuti oleh perusahaan terkait penagihan pembayarannya. Kegiatan tersebut dikelola pada subfitur invoice. Data dalam subfitur ini mencakup:

- a) Nama pelanggan
- b) Metode *invoice*
- c) Diskon yang diterapkan
- d) Tanggal *invoice*
- e) Metode pembayaran
- f) Akun bank
- g) Catatan (opsional)
- h) Total biaya tagihan

Apabila telah selesai, maka finalisasi invoice dilakukan dengan pemilihan 'Validate'. Invoice yang telah selesai dibuat dapat dikirimkan melalui email kepada pelanggan atau dicetak secara offline.

5. Payment

Aktivitas atau tahapan terakhir ialah pada pembayaran. Subfitur ini dapat dijalankan ketika invoice telah selesai dibuat. Data dalam pembayaran ini dibuat berdasarkan data pada invoice. Perbedaannya hanya pada tanggal invoice yang diganti dengan tanggal pembayaran. Apabila metode pembayaran yang digunakan melibatkan penggunaan pihak ketiga pembayaran (transfer Bank, cek, kartu kredit), maka diperlukan data tambahan untuk nomor, transmitter, dan nama bank.

Perusahaan juga dapat memasukkan sendiri jumlah pembayaran yang diberikan pelanggan (tidak harus sesuai dengan total biaya pada invoice). Pembayaran yang dilakukan diakhiri dengan pemilihan 'Pay'. Namun

pembayaran tersebut tidak harus dibayarkan secara penuh. Jika seluruh total biaya telah dibayarkan, finalisasi pembayaran dilakukan dengan pemilihan 'Validate'. Pengiriman yang telah selesai dibuat dapat dikirimkan melalui email kepada pelanggan atau dicetak secara offline.

5.2.4.2 Purchase

Modul *purchase* juga sama seperti modul *sales* tidak tersedia langsung pada Dolibar. Modul ini termasuk dalam modul (Supplier Relationship Management) SRM, dimana membutuhkan fitur Third Parties dan Commercial untuk mengaksesnya. Fungsi fitur Third Parties dan Commercial penjelasannya sama seperti pada bagian *sales*. Gambar 5.26 adalah contoh tampilan halaman *purchase* pada Dolibarr. Gambar 5.26 ialah contoh tampilan *accounting* pada Dolibarr.

The image shows the Dolibarr 'New order' interface. At the top, there is a navigation bar with icons for Home, Third parties, Products, Commercial, Financial, and Bank/Cash. Below this is the Dolibarr logo and the title 'New order'. On the left, there is a sidebar menu with sections for 'Customer orders' (New order, List, Statistics) and 'Suppliers orders' (New order, List, Draft, Validated, Approved, Ordered, Partially received, Everything received, Canceled, Refused, Statistics). The main area contains a form with the following fields: 'Ref.' (set to 'Draft'), 'Supplier' (a dropdown menu), 'Ref. supplier' (a text input field), 'Payment terms' (set to 'Due Upon Receipt'), 'Payment type' (a dropdown menu), 'Planned date of delivery' (a date picker), 'Note (public)' (a text area), and 'Note (private)' (a text area). At the bottom of the form, there are two buttons: 'Create draft' and 'Cancel'.

Gambar 5.26 Contoh Tampilan *Purchase* Dolibarr

1. Create Order

Dalam membuat order supplier, daftar supplier harus sudah didefinisikan pada fitur Third Parties. Bagian subfitur ini terdapat pada fitur Commercial yang terhubung dengan daftar pelanggan di fitur Third Parties. Data-data yang perlu diisi dalam membuat order baru adalah sebagai berikut:

- a) Nomor referensi
- b) Nama *supplier*
- c) Aturan pembayaran
- d) Metode pembayaran

- e) Tenggat waktu pembayaran
- f) Catatan (opsional)

Setiap order yang telah selesai dibuat akan diteruskan untuk mengisikan detail produk pada bagian new line.

2. New Line

Bagian subfitur ini tergabung dengan create order pada subfitur Order card. Tiap line produk yang hendak dibuat, dibedakan atas 2 macam, yakni produk jadi (barang/ jasa) dan produk pre-order. Pilihan tersebut diikuti dengan pengisian beberapa data, yakni:

- a) Nama produk
- b) Pajak (tergantung ketersediaan aturan pajak yang didukung Dolibarr)
- c) Harga produk
- d) Jumlah pembelian
- e) Diskon

Tiap produk yang telah diisikan dan ditambahkan, akan dikalkulasi total biayanya. Jika seluruh detail order, baik informasi umumnya maupun produknya telah diisikan, maka finalisasi order dilakukan dengan pemilihan 'Validate and Approve'. Order yang telah selesai dibuat dapat dikirimkan melalui email kepada supplier atau dicetak secara offline. Order yang telah divalidasi harus didetailkan kembali terkait tanggal dan metode ordernya pada tabel order.

3. Receivement

Barang pesanan perusahaan yang telah sampai dari supplier, perlu divalidasi kedatangannya. Tidak terdapat fitur khusus untuk ini. Hanya saja pada Order card, ketika order telah dibuat, maka pada tabel order akan diganti dengan tabel receive. Tabel ini membutuhkan isian data tanggal kedatangan barang dan status kedatangan barang. Macam status tersebut antara lain:

- a) Total
Barang pesanan datang keseluruhan
- b) *Partial*
Barang pesanan datang sebagian

c) *Never Received*

Barang pesanan tidak diterima

d) *Canceled*

Barang pesanan dibatalkan

4. Invoice

Barang yang telah dipesan kepada supplier akan ditagih pembayarannya melalui invoice untuk perusahaan. Kegiatan tersebut dikelola pada subfitur Invoice. Data dalam subfitur ini mencakup:

- a) Referensi *invoice*
- b) Nama *supplier*
- c) Nomor referensi *supplier*
- d) Metode *invoice*
- e) Label *invoice*
- f) Tanggal *invoice*
- g) Batas waktu pembayaran
- h) Metode pembayaran
- i) Akun bank
- j) Catatan (opsional)
- k) Total biaya tagihan

Apabila telah selesai, maka finalisasi invoice dilakukan dengan pemilihan 'Validate'. Invoice yang telah selesai dibuat dapat dikirimkan melalui email kepada supplier atau dicetak secara offline.

5. Payment

Aktivitas atau tahapan terakhir ialah pada pembayaran. Subfitur ini dapat dijalankan ketika invoice telah selesai dibuat. Data dalam pembayaran ini dibuat berdasarkan data pada invoice. Perbedaannya hanya pada tanggal invoice yang diganti dengan tanggal pembayaran. Apabila metode pembayaran yang digunakan melibatkan penggunaan pihak ketiga pembayaran (transfer Bank, cek, kartu kredit), maka diperlukan data tambahan untuk nomor, transmitter, dan nama bank.

Perusahaan juga dapat memasukkan sendiri jumlah pembayaran yang dibayarkan pada supplier (tidak harus sesuai dengan total biaya pada invoice). Pembayaran yang dilakukan diakhiri dengan pemilihan 'Pay'. Namun pembayaran tersebut tidak harus dibayarkan secara penuh. Jika seluruh total biaya telah dibayarkan, finalisasi pembayaran dilakukan dengan pemilihan 'Validate'. Pengiriman yang telah selesai dibuat dapat dikirimkan melalui email kepada supplier atau dicetak secara offline.

5.2.4.3 Accounting

Modul *accounting* ini pada Dolibarr termasuk dalam modul finance. Dalam fitur ini dikelola seluruh daftar data invoice dan pembayaran, baik yang terkait dengan pelanggan maupun supplier. Selain invoice dan pembayaran, terdapat pula subfitur Reporting. Penggunaan untuk fitur *accounting* juga perlu menyertakan fitur Bank/Cash. Gambar 5.27 menunjukkan contoh tampilan halaman *accounting* pada Dolibarr.

The screenshot shows the Dolibarr Accounting Reporting interface. The top navigation bar includes Home, Third parties, Products, Commercial, Financial, and Bank/Cash. The left sidebar lists various modules like Customers invoices, Suppliers invoices, and Reporting. The main content area is titled 'Report' and contains a form for configuring a report. The report name is 'Balance of income and expenses, annual summary'. The calculation mode is 'Mode Incomes-Expenses said cash accounting'. The report period is '2013 - 2016'. The build on date is 'Dec 30, 2016'. Below the form is a table with columns for years (2013, 2014, 2015, 2016) and sub-columns for Expense and Income for each month (January to June).

	2013		2014		2015		2016	
Month	Expense	Income	Expense	Income	Expense	Income	Expense	Income
January								
February								
March								
April								
May								
June								

Gambar 5.27 Contoh Tampilan Accounting Dolibarr

1. Reporting

Subfitur ini berfungsi layaknya buku kas besar, dimana ia mencatat seluruh akuntansi terkait finansial perusahaan. Subfitur ini terbagi ke dalam beberapa bagian, yakni:

a) Income/Expense

Neraca keuangan terkait debit dan kredit perusahaan. Laporan ini tersedia dalam rentang waktu 4 tahun yang dibedakan berdasarkan bulan.

b) Turnover

Pendapatan/ penghasilan yang diperoleh perusahaan (*sales turnover*). Laporan ini tersedia dalam rentang waktu 4 tahun yang dibedakan berdasarkan bulan

c) *Sales Journal*

Jurnal untuk mencatat transaksi penjualan barang secara kredit kepada pelanggan. Laporan ini dapat diatur rentang waktunya dan terdiri atas data tanggal, dokumen akuntansi, jenis akun, serta jumlah debit dan kreditnya.

d) *Purchases Journal*

Jurnal untuk mencatat transaksi pembelian barang secara kredit kepada *supplier*. Laporan ini dapat diatur rentang waktunya dan terdiri atas data tanggal, dokumen akuntansi, jenis akun, serta jumlah debit dan kreditnya.

2. Bank/Cash

Fitur ini bertujuan sebagai repositori data terkait akun finansial yang digunakan perusahaan. Dalam pembuatan akunnya, terdapat beberapa data yang wajib diisikan, yaitu:

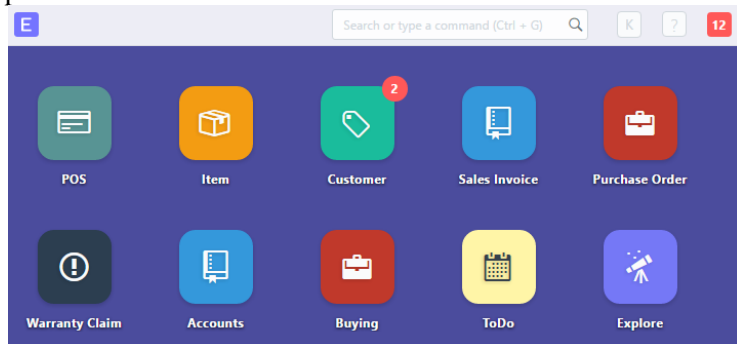
- a) Nomor referensi
- b) Label bank/ *cash*
- c) Jenis akun
- d) Mata uang
- e) Status akun
- f) Jumlah saldo awal
- g) Tanggal pembuatan akun

Data lain pada pembuatan akun ini bersifat opsional sehingga pengisiannya disesuaikan dengan kelengkapan akun finansial perusahaan. Akun finansial ini yang nantinya

digunakan dalam setiap aktivitas yang dapat mempengaruhi perubahan keuangan perusahaan.

5.2.5 ERPNext

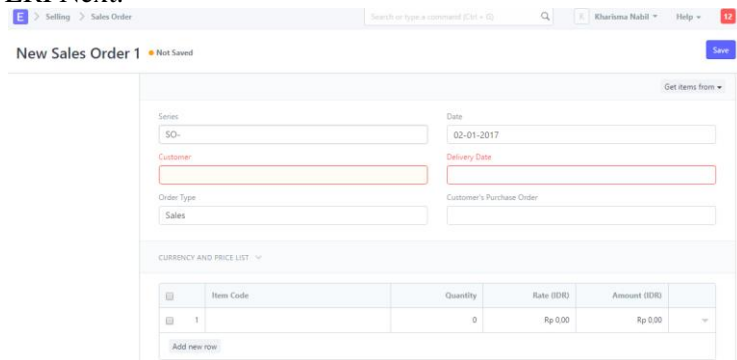
Berikut gambar 5.28 menunjukkan tampilan halaman utama pada ERPNext



Gambar 5.28 Tampilan Halaman Utama ERPNext

5.2.5.1 Sales

Pada modul ini, ERPNext menggunakan istilah ‘Selling’. Modul yang digunakan untuk menjalankan aktivitas penjualan perusahaan ini, dibagi ke dalam beberapa fitur. Berikut gambar 5.29 menunjukkan contoh tampilan halaman *sales* pada ERPNext.



Gambar 5.29 Contoh Tampilan Sales ERPNext

1. Quotation

Quotation yang digunakan untuk memberikan penawaran bagi pelanggan mencakup beberapa data yang perlu diisikan oleh perusahaan. Data tersebut antara lain:

- a) Tujuan *quotation*
- b) Nama pelanggan
- c) Nama perusahaan
- d) Tanggal *quotation*
- e) Jenis order
- f) Detail produk (nama, jumlah, harga, total biaya)
- g) Pajak

2. Sales Order

Fitur ini dikhususkan untuk membuat *sales order* yang berasal dari quotation atas pemesanan yang diajukan pelanggan kepada perusahaan. Dalam pembuatan *sales order* ini, data-data yang diisikan antara lain:

- a) Kode order
- b) Data yang terotomatisasi dari *quotation* (nama pelanggan, perusahaan, tanggal pemesanan, jenis order, detail barang)
- c) Tanggal pengiriman barang
- d) Kategori daftar harga
- e) Lokasi *warehouse* untuk pengiriman

Apabila *sales order* ini telah dinyatakan selesai (submit) maka order tersebut resmi dilakukan dan dipenuhi perusahaan. Perusahaan dapat mencetak *sales order* atau mengirimkannya melalui email kepada pelanggan

3. Make Delivery

Tahapan ini bertujuan untuk pengiriman barang kepada pelanggan. Pengisian dokumen ini dilakukan sebelum pengiriman barang. Data yang perlu diisikan dalam dokumen ini ialah:

- a) Kode pengiriman
- b) Data yang terotomatisasi dari *sales order* (nama pelanggan, perusahaan, tanggal pengiriman, jenis order, detail barang, kategori daftar harga)

c) Alamat pengiriman

Setelah dokumen pengiriman ini difinalisasi (save), maka perusahaan siap untuk mengirimkan barang sesuai pesanan pelanggan

4. Stock Ledger

Pencatatan terkait stok inventori dibutuhkan setelah pengiriman dilakukan. Hal ini dikarenakan adanya perpindahan barang dari inventori gudang sehingga stok barang akan berkurang. Data detail barang dalam catatan stok inventori ini antara lain:

- a) Tanggal perubahan barang
- b) Jenis barang
- c) Nama barang
- d) Kategori/ grup barang
- e) Brand (opsional)
- f) Deskripsi barang (opsional)
- g) Lokasi *warehouse*
- h) Satuan barang
- i) Jumlah barang
- j) *Balance/* neraca barang

5. *Sales Invoice*

Setelah pengiriman barang dilakukan dan barang diterima pelanggan, maka perusahaan berhak melakukan penagihan. Penagihan ini dibuat berdasarkan dokumen *sales order*. Data dalam invoice ini adalah:

- a) Kode *invoice*
- b) Data yang terotomatisasi dari *sales order* (nama pelanggan, perusahaan, tanggal pengiriman, detail barang, kategori daftar harga)
- c) Metode pembayaran
- d) Batas tanggal pembayaran

Setelah dokumen invoice ini difinalisasi (save), maka perusahaan resmi untuk menagih pelanggan terhadap barang pesanan yang telah mereka kirimkan. Pencatatan pada invoice ini juga akan masuk dalam catatan akuntansi terkait

akun pendapatan (income account) dan sumber biaya (cost center). Perusahaan dapat mencetak invoice atau mengirimkannya melalui email kepada pelanggan

6. *Accounting Ledger*

Tahapan ini berhubungan dengan buku kas besar (General Ledger), dimana perusahaan dapat melihat pencatatan hutang (kredit) dan piutang (debit) terkait order dan invoice yang telah ia lakukan.

7. *Payment Entry*

Pembayaran yang dilakukan oleh pelanggan kepada perusahaan akan dikelola pada dokumen ini. Dokumen ini akan mengacu pada invoice dan General Ledger yang telah dibuat sebelumnya. Data-data yang tertera adalah sebagai berikut:

- a) Tipe *voucher*
- b) Kode pembayaran
- c) Tanggal pembayaran dilakukan
- d) Data yang terotomatisasi dari *invoice* dan *General Ledger* (akun hutang dan piutang)
- e) Nomor referensi
- f) Tanggal referensi

Setelah dokumen pembayaran ini difinalisasi (save), maka perusahaan telah menyatakan bahwa urusan pembayaran pelanggan telah selesai. Status pembayaran ini akan diperbarui pada *sales invoice*, sehingga dari status 'draft' akan diganti menjadi 'paid'. Status tagihan dan pengiriman akan diperbarui pada *sales order* ('draft' jadi 'completed')

5.2.5.2 *Purchase*

Pada modul ini, ERPNext menggunakan istilah 'Buying'. Modul yang digunakan untuk menjalankan aktivitas pembelian perusahaan ini, dibagi ke dalam beberapa fitur. Berikut gambar 5.30 menunjukkan contoh tampilan halaman *purchase* pada ERPNext.

The screenshot shows the 'New Purchase Order' form in ERPNext. The form is titled 'New Purchase Order 1' and is marked as 'Not Saved'. It includes a search bar at the top with the text 'Search or type a command (Ctrl + Q)'. The user's name 'Kharisma Nabli' and a help icon are visible in the top right corner. The form fields are as follows:

- Series: PO-
- Date: 02-01-2017
- Supplier: (empty field)
- Supply Raw Materials: No

Below the form fields are sections for 'ADDRESS AND CONTACT' and 'CURRENCY AND PRICE LIST'. At the bottom, there is a table with the following data:

Item Code	Quantity	Rate (IDR)	Amount (IDR)
1	0	Rp 0.00	Rp 0.00

An 'Add new row' button is located below the table.

Gambar 5.30 Contoh Tampilan *Purchase* ERPNext

1. *Purchase Order*

Dokumen ini digunakan untuk mengajukan order pembelian material perusahaan kepada vendor. Dalam pembuatan *purchase order* ini, perusahaan dapat memasukkan secara manual ataupun melalui material request atau supplier quotation. Apabila diotomatisasi melalui material request, maka data tersebut akan disesuaikan dengan kebutuhan material perusahaan yang sudah habis atau waktunya dibeli kembali untuk dimasukkan pada inventori. Apabila diotomatisasi melalui supplier quotation, maka data akan disesuaikan dengan quotation yang diajukan supplier kepada perusahaan. Data-data yang ada dalam dokumen ini mencakup:

- a) Kode order
- b) Tanggal order
- c) Nama vendor/ *supplier*
- d) Nama perusahaan
- e) Pilihan *raw material* (opsional)
- f) *Sales order* (opsional)
- g) Kategori daftar harga
- h) Detail produk (nama, jumlah, harga, total biaya)

Setelah dokumen pembayaran ini difinalisasi (save), maka perusahaan telah menyatakan bahwa ia melakukan order

barang kepada vendor. Perusahaan dapat mencetak *purchase order* atau mengirimkannya melalui email kepada vendor

2. *Purchase Receipt*

Dokumen ini ditujukan untuk mencatat material yang telah diterima oleh perusahaan. Setelah material yang dikirimkan vendor diterima, maka perusahaan dapat melakukan pencatatan *purchase receipt* yang terdiri atas:

- a) Kode penerimaan
- b) Tanggal penerimaan barang
- c) Nama *vendor*
- d) Waktu penerimaan
- e) Kategori daftar harga
- f) Detail produk
- g) Jumlah barang yang diterima
- h) Jumlah barang yang disetujui untuk diterima

Setelah dokumen penerimaan ini difinalisasi (save), maka perusahaan menyatakan telah menerima material yang dikirimkan vendor

3. *Stock Ledger*

Pencatatan terkait stok inventori dibutuhkan setelah penerimaan dilakukan. Hal ini dikarenakan adanya perpindahan barang ke dalam inventori gudang sehingga stok barang akan bertambah. Data detail barang dalam catatan stok inventori ini sama halnya dengan tahapan 'Stock Ledger' pada modul *Sales* karena merupakan dokumen yang sama

4. *Accounting Ledger*

Tahapan ini berhubungan dengan buku kas besar (General Ledger), dimana perusahaan dapat melihat pencatatan hutang (kredit) dan piutang (debit) terkait order dan penerimaan material yang telah ia lakukan.

5. *Purchase Invoice*

Setelah penerimaan barang diterima, maka perusahaan wajib membayara penagihan. Penagihan ini dibuat berdasarkan dokumen *purchase order*. Data dalam invoice ini adalah:

- a) Kode *invoice*

- b) Data yang terotomatisasi dari *purchase order* (nama *supplier*, perusahaan, tanggal order, detail barang, kategori daftar harga)
- c) Nomor *invoice supplier*
- d) Tanggal *invoice supplier*

Setelah dokumen invoice ini difinalisasi (save), maka perusahaan dinyatakan telah membayar pada vendor.. Pencatatan pada invoice ini juga akan masuk dalam catatan akuntansi dalam General Ledger terkait akun piutang (accounts payable)

6. Payment Entry

Pembayaran yang dilakukan oleh perusahaan kepada vendor akan dikelola pada dokumen ini. Dokumen ini akan mengacu pada invoice dan General Ledger yang telah dibuat sebelumnya. Data-data yang tertera adalah sebagai berikut:

- a) Tipe *voucher*
- b) Kode pembayaran
- c) Tanggal pembayaran dilakukan
- d) Data yang terotomatisasi dari *invoice* dan *General Ledger* (akun hutang dan piutang)
- e) Nomor referensi
- f) Tanggal referensi

Setelah dokumen pembayaran ini difinalisasi (save), maka perusahaan telah menyatakan bahwa urusan pembayaran kepada vendor telah selesai. Status pembayaran ini akan diperbarui pada *purchase invoice*, sehingga dari status 'draft' akan diganti menjadi 'paid'. Status tagihan dan penerimaan akan diperbarui pada *purchase order* ('draft' menjadi 'completed')

5.2.5.3 Accounting

Modul akuntansi dalam ERPNext dikenal sebagai 'Accounts'. Modul ini mencakup catatan akuntansi yang terjadi selama proses bisnis perusahaan yang terdiri dari dokumen dan laporan. Gambar 5.31 ialah contoh tampilan *accounting* ERPNext.

The screenshot shows the 'New Journal Entry' form in Accounting ERPNext. At the top, there is a navigation bar with 'Accounts > Journal Entry', a search bar, and user information 'Kharisma Nabil' with a 'Help' button. The main title is 'New Journal Entry 1' with a 'Not Saved' indicator and a 'Save' button. The form is divided into several sections:

- Entry Type:** A dropdown menu set to 'Journal Entry'.
- Posting Date:** A date field set to '02-01-2017'.
- Series:** A dropdown menu set to 'JV-'.
- Company:** A dropdown menu set to 'PT ABC'.
- Table:** A table with columns: Account, Party, Debit, Credit, and a dropdown arrow. The first row contains the number '1' in the Account column. Below the table are buttons for 'Add multiple rows' and 'Add new row'.
- Reference Number:** An empty text field.
- Reference Date:** An empty date field.
- Multi Currency:** A checkbox that is currently unchecked.

Gambar 5.31 Contoh Tampilan Accounting ERPNext

1. Dokumen Akuntansi

Dokumen ini merupakan catatan akuntansi yang berhubungan dengan modul-modul lain pada saat aktivitas yang mengakibatkan perubahan keuangan perusahaan

- **Journal Entry**
Penambahan jurnal akuntansi yang akan masuk ke dalam buku kas besar (*General Ledger*), baik berupa akun hutang (pengeluaran keuangan) maupun akun piutang (penerimaan keuangan)
- **Sales Invoice**
Dokumen tagihan terkait penjualan barang kepada pelanggan yang harus dibayarkan kepada perusahaan
- **Purchase Invoice**
Dokumen tagihan terkait penerimaan barang dari vendor yang harus dibayarkan oleh perusahaan
- **Customer**
Manajemen dalam pendaftaran dan *tracking* terhadap pelanggan beserta statusnya terhadap perusahaan
- **Supplier**
Manajemen dalam pendaftaran dan *tracking* terhadap vendor/ *supplier* beserta statusnya terhadap perusahaan
- **Chart of Accounts**

Daftar Rekening sebagai daftar yang tersusun dalam struktur tertentu yang terdiri dari unsur-unsur rangkaian kode dan nama akun secara sistematis untuk memudahkan penyusunan dan pembacaan tiap akun dalam akuntansi

2. General Ledger

General Ledger merupakan buku kas besar yang digunakan untuk pencatatan seluruh aliran kas masuk dan keluar beserta pembayaran yang dilakukan. Seluruh pencatatan dari modul-modul lain yang berhubungan dengan invoice, pembayaran, dan kondisi-kondisi yang menyebabkan adanya perubahan keuangan perusahaan akan tercatat secara menyeluruh pada dokumen ini.

3. Trial Balance

Laporan ini merupakan neraca saldo atau disebut juga neraca sisa. Laporan ini adalah daftar yang memuat seluruh akun dengan saldo totalnya yang bersumber dari buku kas besar (General Ledger) perusahaan. Jadi, laporan ini baru dapat dibuat setelah buku kas besar selesai disusun/ ditutup.

4. Gross Profit

Laporan ini merupakan catatan keuntungan yang diperoleh perusahaan. Laporan ini berkaitan erat dengan modul selling, dimana pada akhir penerimaan pencatatan penjualan barang akan dihitung pula keuntungan yang diperoleh perusahaan dari penjualan tersebut. Laporan ini membutuhkan data dari dokumen *sales order* dan *sales invoice* untuk dapat melakukan kalkulasi terkait jumlah keuntungan penjualan perusahaan

5. Accounts Receivable

Pencatatan akun piutang yang diterima oleh perusahaan. Akun ini akan masuk ke dalam bagian debit pada buku kas besar dan neraca keuangan perusahaan. Data dalam akun piutang ini mencakup informasi umum dalam piutang perusahaan dan data dari dokumen payment pada modul Selling

6. Accounts Payable

Pencatatan akun hutang yang dilakukan oleh perusahaan. Akun ini akan masuk ke dalam bagian kredit pada buku kas besar dan neraca keuangan perusahaan. Data dalam akun hutang ini mencakup informasi umum dalam hutang perusahaan dan data dari dokumen payment pada modul Buying

7. Sales Register

Laporan ini merupakan rangkuman untuk mendaftarkan seluruh transaksi penjualan yang dijalankan perusahaan selama setahun.

8. Purchase Register

Laporan ini merupakan rangkuman untuk mendaftarkan seluruh transaksi pembelian yang dijalankan perusahaan selama setahun.

9. Balance Sheets

Laporan berupa neraca keuangan perusahaan yang terdiri atas akun hutang dan piutang. Data untuk akun hutang diperoleh dari laporan accounts payable dan akun piutang didapat dari laporan accounts receivable. Fitur ini juga menyediakan diagram garis untuk menunjukkan kenaikan dan penurunan dalam akun hutang dan piutang yang dialami perusahaan

10. Profit and Loss Statement

Laporan ini mencatat seluruh keuntungan dan kerugian yang dialami perusahaan selama setahun. Sama halnya dengan fitur 'Balance Sheets', fitur ini juga menyediakan diagram garis untuk menunjukkan kenaikan dan penurunan, namun untuk keuntungan dan kerugian yang dialami perusahaan

11. Financial Analytics

Tools yang digunakan untuk analisa keuangan perusahaan untuk tujuan pemantauan finansial. Fitur ini juga menyediakan diagram garis untuk menunjukkan kenaikan dan penurunan dalam pembelian/ pengadaan (*purchase*) dan penjualan (*sales*) yang dialami perusahaan

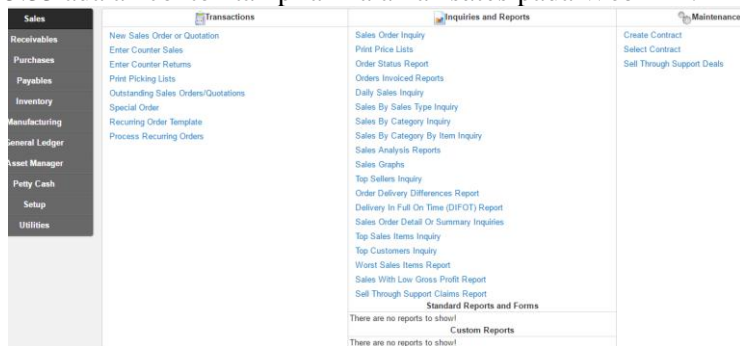
5.2.6 WebERP

Berikut gambar 5.32 menunjukkan tampilan halaman utama pada WebERP

Gambar 5.32 Tampilan Halaman Utama WebERP

5.2.6.1 Sales

Modul *sales* pada WebERP mencakup berbagai aktivitas terkait pengelolaan penjualan perusahaan. Pada tiap modul WebERP selalu terbagi atas 3 bagian utama. Contohnya pada modul *Sales*, terdapat Transactions (aktivitas transaksi penjualan), Inquiries and Reports (formulir dan laporan penjualan), serta Maintenance (dukungan dan perbaikan). Seluruh aktivitas dan fitur utama modul tersedia pada bagian Transactions. Gambar 5.33 adalah contoh tampilan halaman *sales* pada WebERP.



Gambar 5.33 Contoh Tampilan Sales WebERP

1. Quotations

Quotation dapat dibuat perusahaan untuk diajukan pada perusahaan. Pengisian quotation ini sama halnya dengan *sales order*, yakni dengan data-data sebagai berikut:

- a) Nama customer dan *branch*
- b) Detail produk
- c) Harga produk
- d) Diskon

Tiap quotation yang telah dibuat akan menghasilkan file pdf untuk diberikan pada pelanggan, baik secara online melalui email, maupun secara offline dengan pencetakan dokumen

2. *Sales Order*

Pada fitur ini, perusahaan dapat membuat *sales order* bagi pelanggan, baik dari data quotation yang disetujui pelanggan, maupun *sales order* baru tanpa adanya quotation. Dalam membuat dokumen ini, perusahaan harus mendefinisikan terlebih dahulu pelanggannya pada fitur Customers serta produk-produk yang mereka miliki/ jual pada fitur Items. Dokumen dapat dibuat dengan mengacu pada data pelanggan dan data items yang telah masuk pada database. Data yang dimasukkan sama seperti pada quotation namun dengan urutan langkah yang lebih panjang. Urutan langkah pengisian data adalah sebagai berikut:

- a) Pemilihan *customer* dan *branch*
- b) Pemilihan *line* produk yang diorder
- c) Pengisian detail pengiriman
- d) Pengaturan pembayaran

Pada pengaturan pembayaran ini, perusahaan perlu mendetailkannya pada fitur Counter *Sales* dan Returns

3. Counter *Sales* dan Returns

Fitur ini digunakan untuk pengelolaan pembayaran pada penjualan. Counter *Sales* berfungsi untuk penghitungan biaya penjualan secara otomatis. Tiap order yang telah dimasukkan akan dihitung total biaya, pajak, dan metode pembayaran yang digunakan. Fitur ini juga membuat invoice

secara otomatis yang nantinya akan diberikan pada pelanggan.

Counter Returns berfungsi dalam pengelolaan kredit pembayaran oleh pelanggan. Melalui fitur ini, pencatatan status kredit pada tiap pembayarannya akan masuk pada akuntansi secara otomatis.

4. Outstanding *Sales Order*

Fitur ini hanya digunakan pada *sales order* yang masih memungkinkan dalam perubahan kuantitas pemesan produk untuk dikirimkan atau ditagih dalam invoice. Apabila suatu *sales order* perlu diubah pemesanannya, maka kondisi tersebut dapat terpenuhi selama yang barang belum dikirimkan. Fitur ini dapat mengubah jumlah line produk dan detail tagihan pada invoice pelanggan

5. Recurring *Sales Orders*

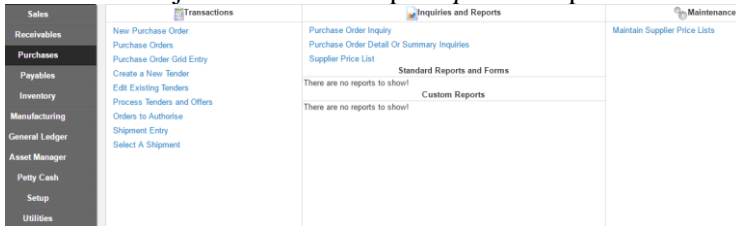
Suatu order dapat diatur agar memungkinkan untuk pengulangan order yang sama beberapa kali. Pilihan dalam pengulangan order ini dapat dilakukan tiap minggu, 2 minggu, bulan, 2 bulan, kuartal, 6 bulan, ataupun tahun. Dalam pembuatan order berulang ini, proses awalnya sama dengan order biasa (pengisian jumlah line produk dan detail pengiriman). Namun pada saat validasi order, terdapat pilihan 'Create Recurring Order' untuk pengaturan order berulang yang mencakup data-data sebagai berikut:

- a) Frekuensi jumlah order berulang
- b) Tanggal awal perulangan order
- c) Tanggal akhir perulangan order
- d) Pilihan pembuatan *invoice* (otomatis jika data *invoice* sama untuk tiap order atau manual jika data *invoice* harus didefinisikan berbeda di tiap order)

Fitur ini memungkinkan pembuatan template order sehingga suatu order baru dapat dibuat berdasarkan template. Pembuatan order yang mengacu pada template dapat diubah hanya pada frekuensi dan rentang waktu perulangan order, tidak pada detail produknya.

5.2.6.2 *Purchase*

Modul *purchase* telah tersedia pada WeERP guna mengorganisir seluruh aktivitas terkait pengadaan material barang pada vendor/ supplier. Sama halnya dengan modul *sales*, terdapat 3 bagian utama. Transactions merupakan bagian untuk mengelola seluruh aktivitas *purchase*. Gambar 5.34 menunjukkan contoh tampilan *purchase* pada WebERP.



Gambar 5.34 Contoh Tampilan *Purchase* WebERP

1. Supplier Tender

Fitur pada modul *purchase* ini memungkinkan perusahaan membuat tender untuk barang-barang yang mereka butuhkan kepada lebih dari supplier. Secara umum, fungsinya sama seperti *purchase requisition*. Supplier yang berminat dapat mengirimkan penawarannya berupa kuantitas barang dan harga yang ditetapkan. Perusahaan dapat memilih penawaran yang ia setuju untuk dilanjutkan menjadi *purchase order*. Data dalam supplier tender ini antara lain:

- a) Informasi umum (tanggal pengiriman, lokasi *warehouse*, detail kontak)
- b) Pilihan *supplier* yang diinginkan
- c) Detail produk (nama dan kuantitas barang)

Seluruh proses tender ini akan berjalan melalui email antara perusahaan dengan supplier. Supplier juga dapat mengirimkan email pribadi untuk memberikan penawaran lain kepada perusahaan. Perusahaan dapat meninjau tender, baik dari hasil tender maupun email pribadi supplier. Tiap tender tersebut akan menghadapi 3 macam tindakan, yakni:

- a) *Accept*

Tender diterima dan dikonversi sebagai *purchase order*

- b) *Reject*

Tender ditolak dan dihapus dari data perusahaan. Informasi penolakan ini akan dikirimkan juga pada *supplier* yang bersangkutan

c) *Defer*

Tender sementara ditunda dan masih dapat digunakan perusahaan untuk penawaran berikutnya

2. *New Purchase Order*

Dalam pembuatan *purchase order* baru, perlu didefinisikan terlebih dahulu *supplier* yang digunakan perusahaan. Pembuatan dokumen ini akan diawali dengan data *supplier* yang dimiliki dan hendak dipesan barangnya. Setelah pemilihan *supplier* yang digunakan, maka berikutnya diisikan data terkait barang yang akan dibeli dan dikirimkan. Data-data tersebut antara lain:

- a) Nama *line* produk
- b) Kategori stok produk
- c) Kuantitas produk
- d) Harga tiap produk
- e) Total biaya

Setelah finalisasi pembuatan *purchase order*, maka dokumen tersebut akan diperiksa (authorized) terkait kelayakannya untuk pengadaan barang pada *supplier*. Sebelum diperiksa maka status *purchase order* adalah pending

3. *Authorize Purchase Order*

Purchase order yang telah dibuat harus diperiksa sebelum dikirimkan pada *supplier*. Pihak pemeriksa ini melihat pada *purchase order* yang masih berstatus pending. *Purchase order* yang diperiksa dapat menyebabkan statusnya berubah menjadi 3 macam, yakni:

- a) *Authorised*
Order disetujui untuk dilakukan
- b) *Cancelled*
Order ditunda pelaksanaannya, namun tidak dihapus dengan tujuan audit dan inspeksi order di kemudian hari
- c) *Rejected*

Order ditolak karena tidak memenuhi *requirement* untuk pengadaan barang

Order yang disetujui dan berstatus *authorised* akan dicetak untuk dikirimkan pada *supplier*. Pada tahapan ini order akan berganti status menjadi *printed*. Order berstatus tersebut sudah dapat dikirimkan pada *supplier*, baik melalui email ataupun pengiriman dokumen offline.

4. Amending *Purchase Order*

Suatu *purchase order* dapat pula diperbaiki atau diubah. Dokumen tersebut dinamakan *outstanding purchase order*. *Purchase order* yang masih berstatus *pending* akan dapat diubah detail produk dan pengirimannya. Sedangkan untuk status *authorised* atau *printed*, perubahan pada *purchase order* akan menyebabkan pemeriksaan ulang (*re-authorised*) sehingga akan kembali berstatus *pending*. *Purchase order* dengan status *cancelled* atau *rejected* akan dikembalikan pada status *pending*. Namun untuk *purchase order* yang telah berstatus *completed*, tidak akan dapat diubah lagi.

5. Shipment

Pengiriman barang yang diberikan *supplier* akan dilakukan ketika *purchase order* telah dikirimkan perusahaan. Pengiriman ini perlu didetailkan data-datanya antara lain:

- a) Metode pengiriman
- b) Jasa pengiriman yang digunakan
- c) Lokasi kedatangan barang
- d) *Expected Time of Arrival (ETA)*/ estimasi waktu kedatangan barang

Pengiriman barang menyebabkan timbulnya *invoice* yang harus dibayarkan oleh perusahaan. Data dalam *invoice* ini dibuat berdasarkan data *purchase order* dan *shipment* dengan disertai beberapa tambahan data yakni total biaya pembayaran (biaya pemesanan ditambah pengiriman barang). Total biaya pembayaran adalah keseluruhan biaya yang harus diberikan perusahaan kepada *supplier* yang menyediakan kebutuhan barang

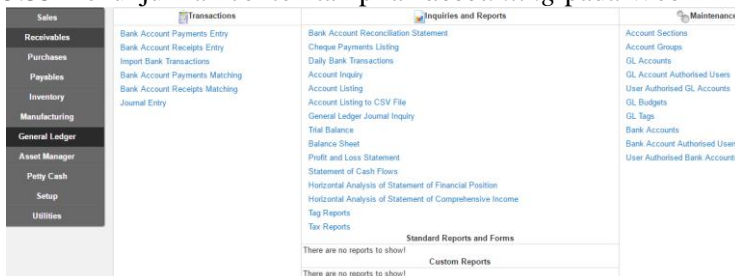
6. Receive *Purchase Order*

Ketika barang dari supplier telah sampai di tangan perusahaan, maka diperlukan pengelolaan barang tersebut ke dalam warehouse. Data utama yang harus dimasukkan adalah jumlah barang yang diterima, karena barang yang dipesan belum tentu sama dengan yang diperoleh (supplier mungkin mengirimkan order dalam beberapa waktu yang berbeda).

Dokumen Goods Received Note akan dihasilkan pada tahapan ini. Dokumen ini berisikan detail barang yang telah diperoleh perusahaan dan ditempatkan pada warehouse. Bagian penerima barang yang bertanggung jawab mengisikan dokumen ini untuk diarsipkan perusahaan dan dikirimkan pada supplier. Jika seluruh barang yang dipesan telah didapat, maka status *purchase order* akan berganti menjadi *completed*.

5.2.6.3 *Accounting*

Modul *accounting* pada WebERP dikelola pada modul General Ledger. Modul ini yang mengintegrasikan seluruh akun hutang dan piutang pada tiap modul beserta stok inventornya. Gambar 5.35 menunjukkan contoh tampilan *accounting* pada WebERP.



Gambar 5.35 Contoh Tampilan *Accounting* WebERP

1. General Ledger Accounts

Buku kas besar merupakan pencatatan utama terkait biaya untuk semua akuntansi. Pencatatan ini dibedakan berdasarkan debit dan kredit. Buku kas besar ini mengintegrasikan seluruh catatan dari akun hutang (accounts

payable), akun piutang (accounts receivable), rekening bank, stok persediaan, biaya pengiriman, dan biaya kontrak biaya. Akun buku kas ini juga dapat membedakan 2 macam laporan, yakni neraca keuangan (balance sheets) dan profit and loss statement. Akun ini juga yang menentukan pencatatan mana yang harus masuk pada neraca saldo (trial balance). Tiap transaksi yang masuk pada buku kas akan disematkan tag untuk memudahkan penelusuran transaksi dalam rangka pembuatan laporan. Tag ini juga berfungsi untuk mengidentifikasi keuntungan dan kerugian yang diperoleh pada tiap transaksi

2. General Ledger Budget

General Ledger budget digunakan untuk membuat dan mengelola budget perusahaan. Informasi terkait budget ini terdiri atas 3 bagian, yakni:

- a) Gambaran finansial tahun lalu dan perbandingannya dengan *budget* yang dimiliki
- b) Tampilan finansial saat ini
- c) Tampilan finansial tahun berikutnya

Pembuatan budget dapat dilakukan pada bagian finansial saat ini dan tahun berikutnya dengan penentuannya untuk budget tiap bulan dan totalnya selama setahun

3. Bank Accounts

Pengaturan terkait akun rekonsiliasi bank pada perusahaan. Akun bank ini yang menjadi sumber keuangan perusahaan. Akun ini dibedakan ke dalam 2 macam, yakni akun pembayaran dan penerimaan. Akun bank pembayaran digunakan untuk segala transaksi yang membutuhkan pengeluaran keuangan perusahaan, termasuk pembayaran pada supplier. Sedangkan akun bank penerimaan adalah untuk transaksi terkait pendapatan perusahaan yang menyebabkan penambahan keuangan. Akun bank ini terintegrasi dengan buku kas besar untuk memberikan posisi finansial perusahaan dan menjadi referensi pembuatan tiap laporan neraca

4. Sales Journal

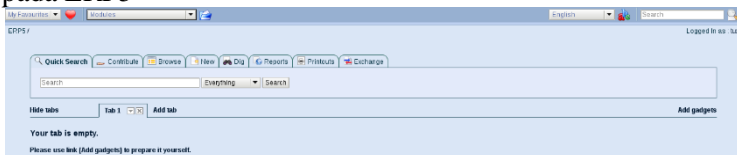
Catatan terkait transaksi penjualan perusahaan. Bagian debit pada jurnal ini ialah akun diskon, debitor, dan debit akun bank. Bagian kredit ialah pada akun *sales* general ledger, pajak, pengiriman ulang, dan kredit akun bank

5. Stock Journal

Catatan terkait perpindahan stok inventori perusahaan. Bagian debit pada jurnal ini ialah biaya pada akun *sales* dan bagian kreditnya ialah akun stok. Jurnal ini juga menjadi referensi perhitungan terkait stok. Cost of goods sold (COGS) ditentukan berdasarkan stok barang yang terjual. Profit and loss statement untuk penyesuaian stok ditentukan berdasarkan catatan kategori stok.

5.2.7 ERP5

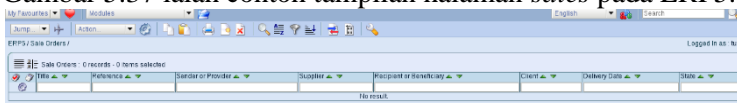
Berikut gambar 5.36 menunjukkan tampilan halaman utama pada ERP5



Gambar 5.36 Tampilan Halaman Utama ERP5

5.2.7.1 Sales

Modul *sales* pada ERP5 dinamai sebagai *sales orders*, dimana seluruh aktivitas dalam penjualan akan dilakukan di dalamnya. Gambar 5.37 ialah contoh tampilan halaman *sales* pada ERP5.



Gambar 5.37 Contoh Tampilan Sales ERP5

1. New Sales Orders

Aktivitas penjualan barang kepada pelanggan dimulai dari pembuat *sales* order baru. Data yang perlu diisi dalam *sales* order baru adalah sebagai berikut:

- a) Nomor dokumen
- b) Perusahaan pengirim
- c) Nama pengirim
- d) Nama penjual
- e) Tingkat prioritas (*urgent*, normal, *urgent*)
- f) Mata uang
- g) *Incoterm* (opsional)
- h) Metode pengiriman (darat, air, udara, surat pos)
- i) Kondisi penjualan
- j) Nomor referensi
- k) Nama pelanggan
- l) Nama penerima
- m) Nama pembeli
- n) Tanggal pengiriman
- o) Tanggal kedatangan
- p) Komentar (opsional)
- q) Deskripsi (opsional)
- r) Status order

Setelah data umum terisi, maka dilanjutkan untuk pengisian detail produknya, yakni antara lain:

- a) Nomor dokumen
- b) Nomor referensi
- c) Nomor *index*
- d) Jenis produk (barang atau jasa)
- e) Nama produk
- f) Nama *supplier*
- g) Tambah kondisi (pajak, diskon)
- h) Kode akun *supplier*
- i) Kuantitas produk
- j) Satuan produk
- k) Harga
- l) Total biaya

Apabila telah selesai, maka finalisasi *sales order* dilakukan dengan pemilihan ‘Confirm Order’. Status order akan berganti dari draft menjadi confirmed.

2. *Sales Picking Lists*

Setelah *sales order* dibuat, maka produk tersebut akan dikirimkan kepada pelanggan. Fitur ini yang mengelola dokumen pengiriman produk. Data dari *sales order* akan dimasukkan dalam data *sales packing*. Tambahan data yang diperlukan adalah sebagai berikut:

- a) Pengirim *invoice*
- b) Penerimaan *invoice*

Apabila telah selesai, maka finalisasi *sales packing* dilakukan dengan pemilihan ‘Ship Packing List’. Status order akan berganti dari confirmed menjadi shipped. Kemudian jika produk yang dikirimkan telah sampai pada pelanggan, maka pada data *sales packing* dipilih ‘Receive Packing List’. Status order akan berganti dari shipped menjadi received

3. *Sales Invoice Transaction*

Setelah produk dikirimkan dan diterima oleh pelanggan, maka perusahaan berhak menagih pembayaran. Fitur ini yang mengelola dokumen invoice untuk penagihan. Data dari *sales packing* akan dimasukkan dalam data invoice. Data invoice harus dimasukkan dalam catatan akuntansi. Jika tidak terdapat data yang harus diubah dari data *sales packing*, maka pada data invoice dipilih ‘Journalise Transaction’. Status order akan berganti menjadi journalised. Pilihan tersebut akan menyebabkan terbentuknya catatan akuntansi penjualan yang dibedakan atas debit dan kredit

Apabila catatan akuntansi telah benar, maka finalisasi akuntansi dilakukan dengan pemilihan ‘Post Transaction to General Ledger’ sehingga data tersebut akan masuk pada buku kas besar (General Ledger). Status order akan berganti dari journalised menjadi posted to general ledger. Akuntansi yang telah selesai dilakukan akan ditutup dengan pemilihan ‘Definitively Close Transaction’. Status order akan berganti dari posted to general ledger menjadi closed. Jika status order

sudah menjadi closed, maka transaksi dianggap selesai dan tidak dapat diubah lagi. Dokumen invoice ini dapat dicetak offline untuk dikirimkan pada pelanggan

4. Payment Transaction

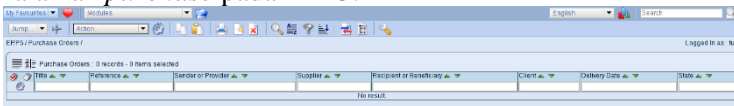
Dalam pencatatan pembayaran yang diberikan pelanggan kepada perusahaan, data order yang telah selesai ditutup transaksinya akan diteruskan pada dokumen pembayaran dengan pemilihan 'Create Related Payment'. Terdapat data awal yang harus diisikan yaitu:

- a) Akun bank
- b) Metode pembayaran
- c) Tanggal pembayaran

Data dari invoice akan dimasukkan dalam data payment dengan disertai tambahan data di atas. Apabila dokumen pembayaran telah benar, maka finalisasi pembayaran dilakukan dengan pemilihan 'Confirm Transaction'. Hasil dari pemilihan tersebut akan menyatakan bahwa pembayaran telah selesai dilakukan oleh pelanggan

5.2.7.2 Purchase

Modul *purchase* pada ERP5 dinamai sebagai *purchase orders*, dimana seluruh aktivitas dalam pembelian akan dilakukan di dalamnya. Berikut gambar 5.38 menunjukkan contoh tampilan halaman *purchase* pada ERP5.



Gambar 5.38 Contoh Tampilan Purchase ERP5

1. New Purchase Orders

Aktivitas pembelian barang kepada supplier dimulai dari pembuat *purchase order* baru. Data yang perlu diisikan dalam *purchase order* baru adalah sebagai berikut:

- a) Nomor dokumen
- b) Perusahaan pembeli
- c) Nama penerima
- d) Nama pembeli

- e) Tingkat prioritas (*urgent*, normal, *urgent*)
- f) Mata uang
- g) *Incoterm* (opsional)
- h) Metode pengiriman (darat, air, udara, surat pos)
- i) Kondisi pembelian
- j) Nomor referensi
- k) Nama *supplier*
- l) Nama pengirim
- m) Nama penjual
- n) Tanggal pengiriman
- o) Tanggal kedatangan
- p) Komentar (opsional)
- q) Deskripsi (opsional)
- r) Status order

Setelah data umum terisi, maka dilanjutkan untuk pengisian detail produknya, yakni antara lain:

- a) Nomor dokumen
- b) Nomor referensi
- c) Nomor *index*
- d) Jenis produk (barang atau jasa)
- e) Nama produk
- f) Nama *supplier*
- g) Tambahan kondisi (pajak, diskon)
- h) Kode akun pembelian
- i) Kuantitas produk
- j) Satuan produk
- k) Harga
- l) Total biaya

Apabila telah selesai, maka finalisasi *purchase* order dilakukan dengan pemilihan 'Confirm Order'. Status order akan berganti dari draft menjadi confirmed.

2. *Purchase Picking Lists*

Setelah *purchase* order dibuat, maka produk tersebut akan dikirimkan kepada perusahaan. Fitur ini yang mengelola

dokumen pengiriman produk. Data dari *purchase* order akan dimasukkan dalam data *purchase* packing.

Apabila telah selesai ditinjau, maka finalisasi *purchase* packing dilakukan dengan pemilihan 'Ship Packing List'. Status order akan berganti dari confirmed menjadi shipped. Kemudian jika produk yang dikirimkan telah sampai pada perusahaan, maka pada data *purchase* packing dipilih 'Receive Packing List'. Status order akan berganti dari shipped menjadi received

3. *Purchase Invoice Transaction*

Setelah produk dikirimkan dan diterima oleh perusahaan, maka supplier berhak menagih pembayaran. Fitur ini yang mengelola dokumen invoice untuk penagihan. Data dari *purchase* packing akan dimasukkan dalam data invoice. Data invoice harus dimasukkan dalam catatan akuntansi. Jika tidak terdapat data yang harus diubah dari data *purchase* packing, maka pada data invoice dipilih 'Journalise Transaction'. Status order akan berganti menjadi journalised. Pilihan tersebut akan menyebabkan terbentuknya catatan akuntansi pembelian yang dibedakan atas debit dan kredit. Apabila catatan akuntansi telah benar, maka finalisasi akuntansi dilakukan dengan pemilihan 'Post Transaction to General Ledger' sehingga data tersebut akan masuk pada buku kas besar (General Ledger). Status order akan berganti dari journalised menjadi posted to general ledger. Akuntansi yang telah selesai dilakukan akan ditutup dengan pemilihan 'Definitively Close Transaction'. Status order akan berganti dari posted to general ledger menjadi closed. Jika status order sudah menjadi closed, maka transaksi dianggap selesai dan tidak dapat diubah lagi. Dokumen invoice ini dapat dicetak offline sebagai arsip perusahaan atau dikirimkan pada supplier

4. *Payment Transaction*

Dalam pencatatan pembayaran yang diberikan perusahaan kepada supplier, data order yang telah selesai ditutup transaksinya akan diteruskan pada dokumen pembayaran

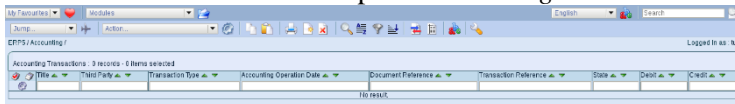
dengan pemilihan ‘Create Related Payment’. Terdapat data awal yang harus diisikan yaitu:

- a) Akun bank
- b) Metode pembayaran
- c) Tanggal pembayaran

Data dari invoice akan dimasukkan dalam data payment dengan disertai tambahan data di atas. Apabila dokumen pembayaran telah benar, maka finalisasi pembayaran dilakukan dengan pemilihan ‘Confirm Transaction’. Hasil dari pemilihan tersebut akan menyatakan bahwa pembayaran telah selesai dilakukan oleh perusahaan.

5.2.7.3 Accounting

Modul *accounting* ERP5 digunakan untuk mencatat akuntansi pada tiap transaksi yang menyebabkan terjadi perubahan keuangan perusahaan. Modul ini dibuat berdasarkan aktivitas dari modul-modul lain yang terlibat dalam perubahan keuangan. Gambar 5.39 ialah contoh tampilan *accounting* ERP5.



Gambar 5.39 Contoh Tampilan Accounting ERP5

1. Accounting Transaction

Fitur ini digunakan untuk melihat daftar catatan akuntansi yang telah dilakukan perusahaan, baik transaksi pembayaran maupun penerimaan. Data akuntansi yang dicakup yakni:

- a) Judul dokumen
- b) Pihak ketiga yang terkait
- c) Jenis transaksi
- d) Tanggal pencatatan
- e) Referensi dokumen
- f) Referensi transaksi
- g) Status dokumen
- h) Jumlah debit
- i) Jumlah kredit

Pada tiap dokumen tersebut dapat diklik untuk dilihat lebih detail catatan akuntansinya

2. Accounting View

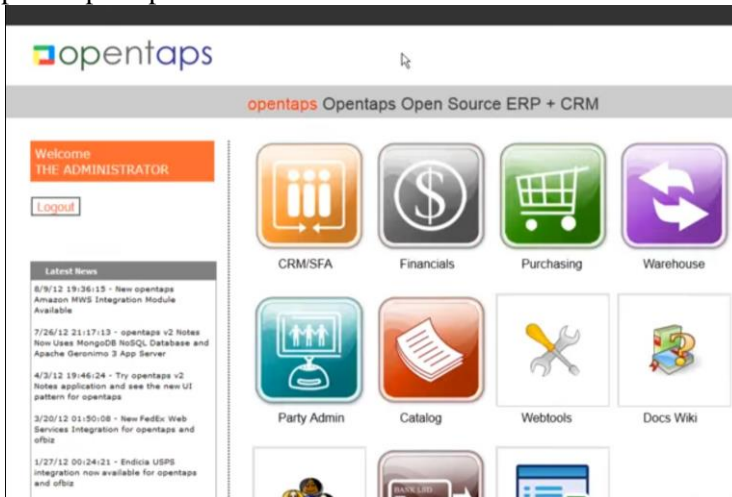
Fitur ini merupakan kelanjutan dari *Accounting* Transaction, dimana ia menampilkan data lengkap pada dokumen yang dituju, seperti data *purchase* order untuk catatan pembelian produk atau data *sales* order untuk catatan penjualan produk. Dalam fitur ini juga terdapat tampilan buku kas besar untuk tiap transaksi yang dipilih. Dalam tampilan ini terdiri atas data-data sebagai berikut:

- a) ID transakssi
- b) Akun-akun yang berpengaruh
- c) Jumlah debit tiap akun
- d) Jumlah kredit tiap akun

Akuntansi ini dapat dicetak untuk menjadi arsip perusahaan offline dalam buku kas.

5.2.8 Opentaps

Berikut gambar 5.40 menunjukkan tampilan halaman utama pada Opentaps



Gambar 5.40 Tampilan Halaman Utama Opentaps

5.2.8.1 Sales

Sales pada Opentaps tergabung dalam modul CRM. Di dalam modul tersebut, terdapat fitur Orders yang berfungsi untuk mengelola aktivitas penjualan perusahaan. Bagian dalam fitur Orders ini memuat beberapa subfitur. Berikut gambar 5.41 menunjukkan contoh tampilan halaman *sales* pada Opentaps.



Gambar 5.41 Contoh Tampilan *Sales* Opentaps

1. Order Settings

Pada subfitur ini, terdapat berbagai pengaturan yang digunakan untuk mengelola order. Data-data yang tertera pada order settings antara lain adalah:

- a) Nama toko/ perusahaan (*store*)
- b) Nama *channel* pada toko
- c) Jenis order
- d) Nama pelanggan
- e) Nomor PO (*Sales Order*)
- f) Kode *tracking*
- g) Penanggungjawab

Apabila pengaturan ini telah disimpan (update), maka pembuatan order berikutnya akan berdasarkan order settings ini

2. Create Order

Tiap order yang akan dipesan oleh pelanggan pada perusahaan, akan dibuat melalui subfitur ini. Data-data yang

perlu diisi dalam membuat suatu order adalah sebagai berikut:

- a) Nomor ID produk
- b) Kuantitas
- c) Tanggal order
- d) Deskripsi (opsional)
- e) Tag departemen
- f) Tag divisi
- g) Tag aktivitas

Dalam pengisian produk yang dipesan pada order, terdapat ID produk yang akan terhubung dengan order items yang mencakup detail produk dalam order. Order yang telah selesai dibuat akan difinalisasi dengan tombol 'Add to Order'

3. Order Items

Produk yang akan dipesan akan didetailkan pada subfitur ini. Dalam detail produk ini mencakup data berupa:

- a) ID produk
- b) Nama produk
- c) Deskripsi produk
- d) Kuantitas
- e) Harga satuan
- f) Pajak/ diskon
- g) Total biaya

Detail produk ini akan terhubung dengan subfitur create order melalui ID produk

4. Order Terms

Detail aturan dalam order terkait penagihan yang diajukan perusahaan dalam bentuk invoice serta metode pembayaran yang dilakukan pelanggan. Pembayaran yang dikelola dalam subfitur ini terbatas dalam bentuk metode yang digunakan dan jumlah pembayaran yang diberikan

5. Shipping Destination

Detail pengiriman produk terkait alamat tujuan pelanggan yang melakukan pembelian. Subfitur ini terhubung dengan

modul shipment untuk pengelolaan aktivitas pengiriman barang secara menyeluruh.

6. Order Summary

Subfitur ini digunakan sebagai rangkuman dari order yang telah dibuat. Tujuannya adalah untuk memudahkan perusahaan dalam memantau tiap order yang telah dibuat

7. My Orders

Subfitur ini bertujuan untuk mengumpulkan seluruh order yang telah dibuat ke dalam satu daftar tabel. Perusahaan dapat melihat seluruh order yang mereka kelola melalui subfitur ini serta meninjau status tiap order. Data yang tertera pada subfitur ini antara lain:

- a) Tanggal order
- b) ID dan nama order
- c) Nomor PO
- d) Nama pelanggan
- e) Status order

8. Find Order

Subfitur ini digunakan untuk mencari suatu atau beberapa order sekaligus berdasarkan filter tertentu. Dalam subfitur ini terdapat beberapa data yang dapat diisikan untuk mencari order sesuai dengan isian data tersebut, yakni antara lain:

- a) ID order
- b) Nomor PO
- c) Nama order
- d) Nama pelanggan
- e) Nama toko/ perusahaan
- f) Status order
- g) Nama produk
- h) Rentang tanggal order
- i) Penanggungjawab

Tiap data tersebut dapat diisikan salah satu untuk mencari order atau beberapa data sekaligus untuk lebih menspesifikkan order yang dicari

9. Detail Order

Subfitur ini mencakup seluruh detail pada suatu order. Terdapat beberapa bagian pada subfitur ini, yakni:

- Order information
Informasi detail order secara umum, dimana terdiri atas tanggal order, status order, nomor PO, *channel* penjualan, dan penanggungjawab
- Payment information
Informasi mengenai detail pembayaran terhadap order yang mencakup metode pembayaran, data *invoice*, pilihan penggunaan kartu kredit, dan total biaya pembayaran
- Contact information
Informasi terkait pelanggan yang melakukan order tersebut. Informasi ini terdiri atas nama pelanggan, alamat *email*, alamat pengiriman, nomor telepon, metode pengiriman, dan tanggal pengiriman

Pada subfitur ini juga terdapat order items untuk detail produk pada order yang dipilih. Detail order ini dapat difinalisasi dengan pilihan persetujuan (*approved*) pada status order. Detail ini juga dapat dikirimkan kepada pelanggan melalui email atau dicetak secara offline

5.2.8.2 *Purchase*

Purchase pada Opentaps tergabung dalam modul *Purchase*. Namun dalam modul ini juga mencakup berbagai aktivitas yang berhubungan dengan pembelian, yakni daftar *supplier*, proses *manufacturing* dan perencanaan, serta pelaporan. Untuk pengelolaan yang berfokus pada aktivitasnya pembelian sendiri, terdapat pada fitur Orders. Berikut gambar 5.42 menunjukkan contoh tampilan halaman *purchase* pada Opentaps.

1. Order Settings

Sama halnya dengan fitur Orders pada *sales*, subfitur ini merupakan pengaturan yang digunakan untuk mengelola order. Data-data yang tertera pada order settings antara lain adalah:

- a) Nama *supplier*
- b) Nama order
- c) Nomor PO (*Purchase Order*)
- d) Aturan order
- e) Tanggal awal pengiriman
- f) Batas tanggal pengiriman

Jika pengaturan telah disimpan (update), maka pembuatan order berikutnya akan berdasarkan order settings ini

The screenshot displays the OpenTaps 'Create Order' form. On the left, there are sections for 'Order Settings' (including Supplier: Demo Supplier, Currency: USD, Total: \$565.00) and 'Show Grid' (with buttons for Open Orders, Find Orders, Reserve Order, Invoice Supplier, and Estimated Delivery Date). The main 'Create Order' section features a 'Product ID' dropdown menu with a scrollable list of products such as '02-1000 Tiny Gizmo', '02-1004 Purple Gizmo', '02-1005 NIT Gizmo', '02-1006 Open Gizmo', '02-1006 1-Open Gizmo (LGPL)', '02-1006 2-Open Gizmo (GPL)', '02-1006 3-Open Gizmo (BSD)', '02-1006 4-Open Gizmo (MIT)', '02-2002 Square Gizmo', '02-2044 Round Gizmo', '02-2005 Purple Gizmo', '02-7000 Massive Gizmo', '02-8044 Big Gizmo', '02-9290 Hi/Mar Gizmo', '02-8400KIT:Auto-Assembled Gizmo Basket', '02-010: Digital Gizmo', and '02-MANUFACTURED: Internally Manufactured Product for Evert Production Runs Testing'. Below this is an 'Order Items' section with fields for 'Product' (02-1000 - Tiny Gizmo), 'Inventory' (ATP = 10, QOH = 73), 'Ship After Date', and 'Ship Before Date'. There are also dropdown menus for 'Division Tag', 'Department Tag', 'Activity Tag', and 'Activity Tag'. At the bottom right, a 'Cart Total' of '\$565.00' is shown.

Gambar 5.42 Contoh Tampilan *Purchase* Opentaps

2. Create Order

Tiap order yang akan dipesan oleh perusahaan pada supplier, akan dibuat melalui subfitur ini. Data-data yang perlu diisi dalam membuat suatu order adalah sebagai berikut:

- a) Nomor ID produk
- b) Kuantitas
- c) Tanggal order
- d) Deskripsi (opsional)
- e) Tag departemen
- f) Tag divisi
- g) Tag aktivitas

Dalam pengisian produk yang dipesan pada order, terdapat ID produk yang akan terhubung dengan order items yang mencakup detail produk dalam order. Order yang telah selesai dibuat difinalisasi dengan tombol 'Add to Order'

3. Order Items

Produk yang akan dipesan akan didetailkan pada subfitur ini. Dalam detail produk ini mencakup data berupa:

- a) ID produk
- b) Nama produk
- c) Deskripsi produk
- d) Kuantitas
- e) Harga satuan
- f) Pajak/ diskon
- g) Total biaya

Detail produk ini akan terhubung dengan subfitur create order melalui ID produk

4. Order Terms

Detail aturan dalam order terkait penagihan yang diberikan supplier dalam bentuk invoice serta metode pembayaran yang dilakukan perusahaan. Pembayaran yang dikelola dalam subfitur ini terbatas dalam bentuk metode yang digunakan dan jumlah pembayaran yang diberikan

5. Shipping Destination

Detail pengiriman produk terkait alamat tujuan perusahaan yang melakukan pembelian. Subfitur ini terhubung dengan modul shipment untuk pengelolaan aktivitas pengiriman barang secara menyeluruh.

6. Order Summary

Subfitur ini digunakan sebagai rangkuman dari order yang telah dibuat. Tujuannya adalah untuk memudahkan perusahaan dalam memantau tiap order yang telah dibuat

7. My Orders

Subfitur ini bertujuan untuk mengumpulkan seluruh order yang telah dibuat ke dalam satu daftar tabel. Perusahaan dapat melihat seluruh order yang mereka kelola melalui subfitur ini serta meninjau status tiap order. Data yang tertera pada subfitur ini antara lain:

- a) Tanggal order
- b) ID dan nama order
- c) Nomor PO

- d) Nama *supplier*
- e) Status order

8. Find Order

Subfitur ini digunakan untuk mencari suatu atau beberapa order sekaligus berdasarkan filter tertentu. Dalam subfitur ini terdapat beberapa data yang dapat diisikan untuk mencari order sesuai dengan isian data tersebut, yakni antara lain:

- a) ID order
- b) Nomor PO
- c) Nama order
- d) Nama *supplier*
- e) Nama toko/ perusahaan
- f) Status order
- g) Nama produk
- h) Rentang tanggal order
- i) Penanggungjawab

Tiap data tersebut dapat diisikan salah satu untuk mencari order atau beberapa data sekaligus untuk lebih menspesifikkan order yang dicari

9. Detail Order

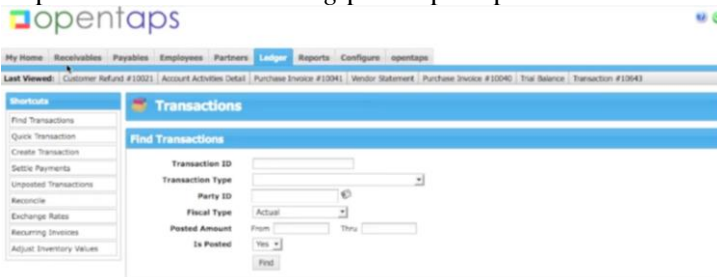
Subfitur ini mencakup seluruh detail pada suatu order. Terdapat beberapa bagian pada subfitur ini, yakni:

- Order information
Informasi detail order secara umum, dimana terdiri atas tanggal order, status order, nomor PO, *channel* pembelian, dan penanggungjawab
- Payment information
Informasi mengenai detail pembayaran terhadap order yang mencakup metode pembayaran, data *invoice*, pilihan penggunaan kartu kredit, dan total biaya pembayaran
- Contact information
Informasi terkait *supplier* yang menjual barang pesanan perusahaan. Informasi ini terdiri atas nama *supplier*, alamat *email*, alamat pengiriman, nomor telepon, metode pengiriman, dan tanggal pengiriman

Pada subfitur ini juga terdapat order items untuk detail produk pada order yang dipilih. Detail order ini dapat difinalisasi dengan pilihan persetujuan (approved) pada status order. Detail ini juga dapat dikirimkan kepada supplier melalui email atau dicetak secara offline

5.2.8.3 Accounting

Opentaps menyediakan sendiri modul *accounting* dengan berbagai fitur yang mencakup pencatatan akuntansi perusahaan secara standar. Berikut gambar 5.43 menunjukkan contoh tampilan halaman *accounting* pada Opentaps.



Gambar 5.43 Contoh Tampilan Accounting Opentaps

1. Receivable

Fitur ini digunakan untuk pencatatan akun piutang yang diterima oleh perusahaan. Akun ini akan masuk ke dalam bagian debit pada buku kas besar (General Ledger) perusahaan. Akun receivable mengelola dokumen-dokumen berupa:

- a) Tagihan penjualan (*sales invoice*)
- b) Pembayaran dari pelanggan (*customer payment*)
- c) Kebijakan dengan pelanggan (*customer agreement*)
- d) Akun kredit pelanggan (*customer credit account*)
- e) Biaya bunga (*finance charge*)
- f) Laporan *receivable*

2. Payables

Fitur ini digunakan untuk pencatatan akun hutang yang dilakukan oleh perusahaan. Akun ini akan masuk ke dalam

bagian kredit pada buku kas besar (General Ledger) perusahaan. Akun payable mengelola dokumen berupa:

- a) Tagihan pembelian pada vendor (*vendor invoice*)
- b) Pembayaran pada vendor (*vendor bill*)
- c) Kebijakan komisi (*commission agreement*)
- d) Laporan *payable*

3. Employee

Fitur ini digunakan untuk pengelolaan gaji pada karyawan. Karena gaji termasuk faktor yang mempengaruhi perubahan keuangan perusahaan, maka pengelolaannya juga masuk pada modul *accounting*. Dalam fitur ini dimuat pengelolaan dalam membuat, mencari, dan menampilkan daftar gaji karyawan

4. Partners

Fitur ini bertujuan untuk mengelola partner bisnis yang berhubungan dengan perusahaan. Fokus pengelolaan yang dicakup adalah pada tagihan dengan partner (*partner invoice*) serta kebijakan dengan partner (*partner agreement*)

5. Ledger

Fitur ini merupakan fitur untuk mengelola pencatatan akuntansi dalam aliran kas perusahaan. Tiap transaksi yang terjadi (penjualan ataupun pembayaran) dimasukkan ke dalam sini. Pencatatan transaksi ini harus diisikan jumlah biayanya, baik pada sisi debit maupun kredit. Jumlah biaya pada kedua sisi harus sama/ seimbang (*balance*) agar dapat diteruskan ke dalam buku kas (*general ledger*)

6. Reports

Fitur ini mencakup berbagai laporan akuntansi dalam kurun waktu tertentu (biasanya tahunan). Laporan yang dicakup kurang lebih sama seperti modul *accounting* ERP secara umum, yakni:

- a) *Accounting*
 - *Balances* pada *GL account*
 - *Transaction summary*
- b) *Inventory*
 - *Inventory Report*

- *Sales, Inventory, Profitability Analysis*
- c) Financial
 - *Trial Balance*
 - *Income Statement*
 - *Balance Sheet*
 - *Cash Flow Statement*
- d) Tax
 - *Tax Summary Report*
 - *Sales Tax Statement*
- e) Analysis
 - *GL Account Activity Analysis*
- f) Budgeting
 - *Budget Plan*

5.2.9 Apache OFBiz

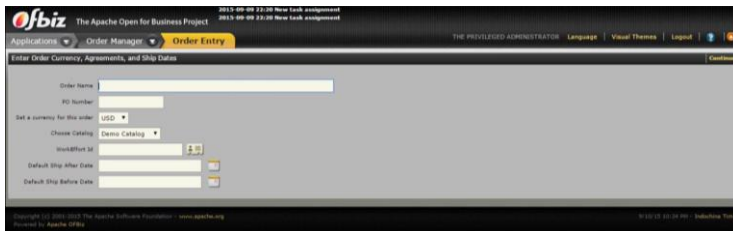
Berikut gambar 5.44 menunjukkan tampilan halaman utama pada Apache OFBiz



Gambar 5.44 Tampilan Halaman Utama Apache OFBiz

5.2.9.1 Sales

Modul *sales* pada Apache OFBiz berada di menu yang sama dengan modul *purchase*, yakni *Order Manager*. Gambar 5.45 menunjukkan contoh tampilan halaman *sales* pada Apache OFBiz. Pada menu ini, fitur-fitur yang termasuk dalam modul *sales* adalah sebagai berikut:



Gambar 5.45 Contoh Tampilan Sales Apache OFBiz

1. Quotes

Fitur ini berfungsi layaknya quotation, yakni melakukan penawaran barang yang diinginkan pelanggan dari perusahaan. Data-data yang perlu diisikan dalam dokumen ini, antara lain:

- a) ID dokumen
- b) Status dokumen
- c) Rentang waktu penawaran produk
- d) Nama perusahaan
- e) Rentang waktu pemenuhan produk
- f) Detail produk yang ditawarkan (nama produk, kuantitas, harga)

Data quotation yang disetujui oleh pelanggan dapat diteruskan menjadi *sales order*

2. Order Entry

Dalam membuat order entry ini dapat dibuat dua macam order, *sales order* dan *purchase order*. Untuk membuat *sales order*, maka data-data yang perlu dimasukkan adalah:

- a) Nama toko/ perusahaan
- b) *Sales channel*
- c) Penanggungjawab
- d) Nama pelanggan

Selanjutnya maka akan masuk pengisian data terkait informasi umum pada *sales order*, yakni:

- a) Nama order
- b) Nomor PO
- c) Mata uang

- d) Katalog yang digunakan
- e) Tanggal awal pengiriman
- f) Batas tanggal pengiriman

Pengisian terakhir adalah pada detail produk yang dipesan pada *sales order*. Data-datanya antara lain adalah:

- a) ID produk
- b) Nama produk
- c) Kuantitas
- d) Harga produk
- e) Tambahan kondisi (pajak/ diskon) (opsional)
- f) Total biaya

Sales order yang telah terisikan seluruh data tersebut dapat difinalisasi dengan pemilihan 'Finalize Order' agar order dapat diteruskan ke tahap selanjutnya

3. Shipment

Dalam pengiriman barang atas *sales order*, dokumen pengiriman akan dibuat berdasarakan *sales order* itu sendiri. Data tambahan yang diperlukan adalah:

- a) Alamat tujuan pengiriman
- b) Pilihan pengiriman berkala/ *partial* (opsional)
- c) Metode pengiriman
- d) Jasa pengiriman
- e) Instruksi tambahan (opsional)

4. Payment

Pembayaran yang dilakukan oleh pelanggan ke perusahaan mencakup pengisian data dokumen pembayaran yang terdiri atas:

- a) ID pembayaran
- b) Metode pembayaran
- c) Total biaya
- d) Pilihan penggunaan kartu kredit (opsional)
- e) Pilihan pembayaran berkala/ *partial* (opsional)

Setelah pembayaran terisi, maka seluruh dokumen diakhiri dengan pemilihan 'Final Order Review'. Tampilan review akan muncul agar perusahaan dapat meninjau pengisian data

dari mulai *sales order* hingga *sales payment*. Proses dalam pembuatan order divalidasi dan difinalisasi dengan pemilihan ‘Create Order’

5. Find Order

Subfitur ini digunakan untuk mencari suatu atau beberapa order sekaligus berdasarkan filter tertentu. Dalam subfitur ini terdapat beberapa data yang dapat diisikan untuk mencari order sesuai dengan isian data tersebut, yakni antara lain:

- a) ID order
- b) Nomor PO
- c) Nama order
- d) Nama pelanggan
- e) Nama toko/ perusahaan
- f) Status order
- g) Nama produk
- h) Rentang tanggal order
- i) Penanggungjawab

Tiap data tersebut dapat diisikan salah satu untuk mencari order atau beberapa data sekaligus untuk lebih menspesifikkan order yang dicari

6. Detail Order

Subfitur ini mencakup seluruh detail pada suatu order. Terdapat beberapa bagian pada subfitur ini, yakni:

- *Sales Order Number*
Informasi detail order secara umum, dimana terdiri atas tanggal order, status order, *channel* penjualan, penanggungjawab, dan pilihan *invoice*
- *Payment information*
Informasi mengenai detail pembayaran terhadap order yang mencakup ID pembayaran, metode pembayaran, tujuan pembayaran, dan total biaya, dan status pembayaran
- *Contact information*

Informasi terkait pelanggan yang melakukan order tersebut. Informasi ini terdiri atas nama pelanggan, alamat *email*, alamat pengiriman, dan nomor telepon

Detail order ini dapat difinalisasi dengan pilihan persetujuan (approved) pada status order. Detail ini juga dapat dicetak secara offline dalam bentuk pdf

7. Returns

Fitur ini digunakan untuk pengelolaan dalam pengembalian produk. Fitur ini dapat digunakan baik untuk *sales* maupun *purchase*. Data yang perlu diisikan antara lain:

- a) ID Header (untuk *sales* pilih Return from Customer)
- b) Status dokumen
- c) Asal pengembalian
- d) Tujuan pengembalian
- e) Metode pembayaran
- f) Akun finansial
- g) Akun *billing*/ tagihan
- h) Tanggal dokumen
- i) *Destination facility* (tujuan tempat inventori produk)
- j) Pilihan metode penerimaan inventori

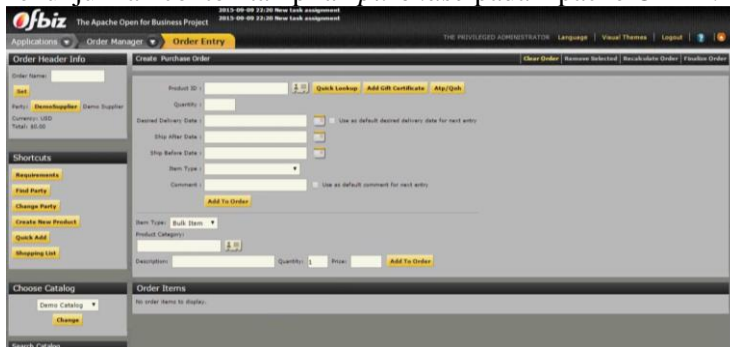
Setelah pengisian informasi umum pada Return Header, maka pengisian selanjutnya adalah pada Return Items. Produk yang hendak dikembalikan didetailkan pada bagian ini dengan mencakup data:

- a) Nama produk
- b) Total biaya produk
- c) Deskripsi
- d) Jumlah yang dikembalikan
- e) Harga produk
- f) Biaya yang dikembalikan
- g) Alasan pengembalian
- h) Jenis pengembalian
- i) Status produk
- j) Penyesuaian biaya produk (pengiriman, diskon)
- k) Biaya lain-lain

Seluruh pengisian dokumen untuk pengembalian produk diakhiri dan difinalisasi dengan pilihan 'Accept Return'. Detail dokumen pengembalian ini dapat ditinjau pada bagian Return History.

5.2.9.2 Purchase

Karena modul *purchase* berada di menu yang sama dengan modul *sales* pada Apache OFBiz, maka beberapa fitur yang dicakup tidak banyak yang berbeda. Fitur yang berbeda hanya aktivitas *purchase* yang kontras dengan *sales*. Gambar 5.46 menunjukkan contoh tampilan *purchase* pada Apache OFBiz.



Gambar 5.46 Contoh Tampilan *Purchase* Apache OFBiz

1. Request

Fitur ini berfungsi layaknya requisition, yakni melakukan permintaan barang yang diinginkan perusahaan kepada vendor/ supplier. Data-data yang perlu diisi dalam dokumen ini, antara lain:

- a) ID dokumen
- b) Status dokumen
- c) Rentang waktu permintaan produk
- d) Nama perusahaan
- e) Rentang waktu pemenuhan produk
- f) Detail produk yang diminta (nama produk, kuantitas, harga)

Data requisition yang dipenuhi oleh supplier dapat diteruskan menjadi *purchase* order apabila terdapat

kesepakatan antara kedua pihak terkait order produk yang diminta

2. Order Entry

Dalam membuat order entry ini dapat dibuat dua macam order, *sales* order dan *purchase* order. Untuk membuat *purchase* order, maka data-data yang perlu dimasukkan adalah:

- a) Nama perusahaan
- b) Penanggungjawab
- c) Nama *supplier*

Selanjutnya maka akan masuk pengisian data terkait informasi umum pada *purchase* order, yakni:

- a) Aturan order (opsional)
- b) ID Order
- c) Nama order
- d) Mata uang
- e) Katalog yang digunakan
- f) Tanggal awal pengiriman
- g) Batas tanggal pengiriman

Pengisian terakhir adalah pada detail produk yang dipesan pada *purchase* order. Data-datanya antara lain adalah:

- a) ID produk
- b) Nama produk
- c) Kuantitas
- d) Tanggal kedatangan produk
- e) Harga produk
- f) Tambahan kondisi (pajak/ diskon) (opsional)
- g) Total biaya

Purchase order yang telah terisikan seluruh data tersebut dapat difinalisasi dengan pemilihan 'Finalize Order'. Jika terdapat aturan/ kebijakan pada *purchase* order, maka diperlukan pengirisan Order Terms. Jika tidak, maka order dapat langsung diteruskan ke tahap selanjutnya

3. Shipment

Dalam pengiriman barang atas *purchase order*, dokumen pengiriman akan dibuat berdasarkan *purchase order* itu sendiri. Data tambahan yang diperlukan adalah:

- a) Alamat tujuan pengiriman
- b) Pilihan pengiriman berkala/ *partial* (opsional)
- c) Metode pengiriman
- d) Jasa pengiriman *third party*

4. Payment

Pembayaran yang dilakukan oleh perusahaan ke supplier mencakup pengisian data dokumen pembayaran yang terdiri atas:

- a) ID pembayaran
- b) Metode pembayaran
- c) Total biaya
- d) Pilihan penggunaan kartu kredit (opsional)
- e) Pilihan pembayaran berkala/ *partial* (opsional)

Setelah pembayaran terisi, maka seluruh dokumen diakhiri dengan pemilihan 'Final Order Review'. Tampilan review akan muncul agar perusahaan dapat meninjau pengisian data dari mulai *purchase order* hingga *purchase payment*. Proses dalam pembuatan order divalidasi dan difinalisasi dengan pemilihan 'Create Order'

5. Find Order

Subfitur ini digunakan untuk mencari suatu atau beberapa order sekaligus berdasarkan filter tertentu. Dalam subfitur ini terdapat beberapa data yang dapat diisikan untuk mencari order sesuai dengan isian data tersebut, yakni antara lain:

- a) ID order
- b) Nama order
- c) Nama *supplier*
- d) Nama perusahaan
- e) Status order
- f) Nama produk
- g) Rentang tanggal order
- h) Penanggungjawab

Tiap data tersebut dapat diisikan salah satu untuk mencari order atau beberapa data sekaligus untuk lebih menspesifikkan order yang dicari

6. Detail Order

Subfitur ini mencakup seluruh detail pada suatu order. Terdapat beberapa bagian pada subfitur ini, yakni:

- *Purchase Order Number*
Informasi detail order secara umum, dimana terdiri atas tanggal order, status order, *channel* pembelian, penanggungjawab, dan pilihan *invoice*
- *Payment information*
Informasi mengenai detail pembayaran terhadap order yang mencakup ID pembayaran, metode pembayaran, tujuan pembayaran, dan total biaya, dan status pembayaran
- *Contact information*
Informasi terkait *supplier* yang menyediakan order tersebut. Informasi ini terdiri atas nama *supplier*, alamat *email*, alamat pengiriman, dan nomor telepon

Detail order ini dapat difinalisasi dengan pilihan persetujuan (approved) pada status order. Detail ini juga dapat dicetak secara offline dalam bentuk pdf

7. Returns

Fitur ini digunakan untuk pengelolaan dalam pengembalian produk. Sama halnya dengan *sales*, fitur juga dapat diterapkan pada *purchase*. Data yang perlu diisikan antara lain:

- a) ID Header (untuk *purchase* pilih Return from Vendor)
- b) Status dokumen
- c) Asal pengembalian
- d) Tujuan pengembalian
- e) Metode pembayaran
- f) Akun finansial
- g) Akun *billing*/ tagihan
- h) Tanggal dokumen

- i) *Destination facility* (tujuan tempat inventori produk)
 - j) Pilihan metode penerimaan inventori
- Setelah pengisian informasi umum pada Return Header, maka pengisian selanjutnya adalah pada Return Items. Produk yang hendak dikembalikan didetailkan pada bagian ini dengan mencakup data:
- a) Nama produk
 - b) Total biaya produk
 - c) Deskripsi
 - d) Jumlah yang dikembalikan
 - e) Harga produk
 - f) Biaya yang dikembalikan
 - g) Alasan pengembalian
 - h) Jenis pengembalian
 - i) Status produk
 - j) Penyesuaian biaya produk (pengiriman, diskon)
 - k) Biaya lain-lain

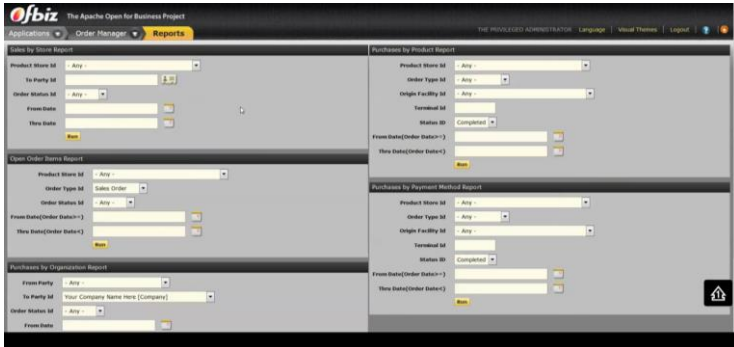
Seluruh pengisian dokumen untuk pengembalian produk diakhiri dan difinalisasi dengan pilihan 'Accept Return'. Detail dokumen pengembalian ini dapat ditinjau pada bagian Return History.

5.2.9.3 *Accounting*

Modul akuntansi pada Apache OFBiz tersedia pada menu sendiri yakni *Accounting*. Modul ini juga berhubungan dengan fitur Report pada menu Order Manager untuk mengintegrasikan antara order yang dihasilkan (baik dari *sales* maupun *purchase*) kepada akuntansi perusahaan. Gambar 5.47 menunjukkan contoh tampilan halaman *accounting* pada Apache OFBiz.

1. AR

AR atau *Account Receivable* merupakan pencatatan akun piutang yang diterima oleh perusahaan. Akun ini akan masuk ke dalam bagian debit pada buku kas besar dan neraca keuangan perusahaan.



Gambar 5.47 Contoh Tampilan *Accounting* Apache OFBiz

2. AP

AP atau Account Payable Pencatatan akun hutang yang dilakukan oleh perusahaan. Akun ini akan masuk ke dalam bagian kredit pada buku kas besar dan neraca keuangan perusahaan.

3. Invoice

Dokumen tagihan/ invoice tidak dispesifikkan untuk dijelaskan pada menu Order Manager yang mencakup *sales* dan *purchase*. Namun dikelola pada menu *Accounting* untuk pembuatan dan pencarian tiap dokumen tagihan yang berhubungan dengan order yang telah dibuat. Dokumen ini dapat dibuat manual untuk *sales* order atau *purchase* order, serta dapat pula diteruskan dari *sales* order atau *purchase* order yang telah dibuat sebelumnya

4. Fixed Assets

Pengelolaan pada aset-aset yang dimiliki perusahaan. Pengelolaan ini terdiri atas pendaftaran macam aset, lokasi penempatan, kategorisasi, pembelian, penjualan, depresiasi, pelaporan, dan pemantauan aset.

5. Billing Account

Fitur ini digunakan untuk memantau tiap akun transaksi yang dimiliki perusahaan serta meninjau status akunnnya dengan melihat pada jumlah limit pada akun, tanggal mulai, dan tanggal akhir akun

6. Financial Accounts

Fitur ini berfungsi seperti Chart of Accounts, yakni menyimpan berbagai daftar rekening dalam bentuk daftar yang terstruktur. Daftar rekening ini terdiri dari rangkaian kode, nama, dan jenis akun. Daftar rekening ini berguna untuk memudahkan penyusunan dan pembacaan tiap akun dalam akuntansi

7. Reports

Fitur ini terdapat pada menu Order Manager dan *Accounting*, dimana cakupannya adalah berbagai laporan akuntansi dalam kurun waktu tertentu. Laporan yang dicakup antar lain, yakni:

- a) *Sales Journal*
- b) *Purchase Jurnal*
- c) General Ledger
- d) Trial Balance
- e) Profit and Loss
- f) Balance Sheet

5.2.10 xTuple Postbooks

Berikut gambar 5.48 menunjukkan tampilan halaman utama pada xTuple Postbooks

5.2.10.1 *Sales*

Pada modul ini terdapat beberapa fitur terkait pengelolaan penjualan yang dijalankan perusahaan. Gambar 5.49 ialah contoh tampilan halaman *sales* pada xTuple Postbooks.

1. Quote

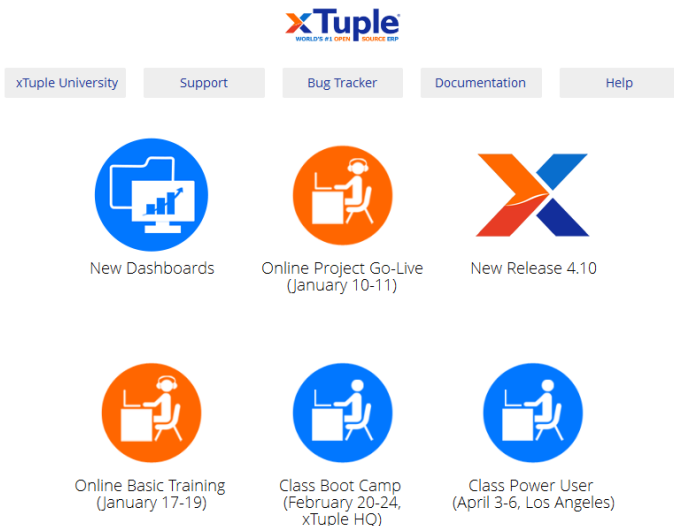
Pada bagian quote ini terdapat berbagai data yang perlu diisikan guna membuat quotation untuk diajukan kepada pelanggan

- General Information
Informasi umum dalam *quotation* yang terdiri atas nomor order, tanggal order, kode pelanggan, nama pelanggan, kode pengiriman, nama perusahaan.
- Header Information

Informasi detail dalam *quotation* yang akan ditinjau oleh pelanggan. Informasi ini dibagi ke dalam 3 bagian utama, yakni syarat dan aturan *quotation*, detail alamat pengiriman, dan detail alamat pengiriman.

- Line items

Informasi memuat berbagai detail *line* produk yang akan ditawarkan bagi pelanggan beserta total perhitungan harga dan diskon produk



Gambar 5.48 Tampilan Halaman Utama Postbooks

2. *Sales Order*

Pada *sales order* ini, Postbooks menyediakan 2 pilihan dalam membuat order. Perusahaan dapat membuat order bagi pelanggan melalui *quotation* ataupun langsung dari *sales order* jika pelanggan memesan tanpa *quotation*. Terdapat beberapa bagian dari *sales order*, yakni:

- New Order

Tahapan perusahaan dalam membuat *sales order* baru bagi pesanan pelanggan tanpa *generate* dari *quotation*

- List Order

Daftar *sales order* yang dimiliki perusahaan, baik dibuat melalui *quotation* yang direspon pelanggan, maupun dari *sales order* yang langsung dibuat

- Reschedule

Pilihan dalam pengelolaan perubahan jadwal dalam pemesanan, pengiriman, ataupun penagihan yang tertera sebelumnya pada *sales order* dan akan diubah atas kesepakatan dengan pelanggan.

The screenshot displays a software interface for managing sales orders. At the top, there are fields for 'Order #' (40014), 'Customer #' (with a dropdown arrow), 'Order Date' (1/1/17), 'Name', 'Ship To' (with a dropdown arrow), and 'Cust. PO #'. Below these are tabs for 'Header Information', 'Line Items', 'Relationships', 'Characteristics', 'Comments', 'Notes', and 'Documents'. The 'Header Information' tab is active, showing 'Terms', 'F.O.B.', 'Shipping Zone', 'Outside Rep.', and 'Ship Via'. A 'Quote Status' dropdown is set to 'Open'. The form is divided into two main sections: 'BILL-TO' and 'SHIP-TO'. Each section contains fields for 'Street Address', 'Country', 'City', 'State', and 'Postal'. Below these are fields for 'Name', 'Phone', 'Alternate', and 'Fax'. A 'Copy to Ship-to -->' button is located between the two sections.

Gambar 5.49 Contoh Tampilan Sales Postbooks

3. Billing

Fitur ini merupakan tahapan dalam penagihan pembayaran pada pelanggan setelah barang pesannya dikirimkan oleh perusahaan. Billing ini berkaitan erat dengan invoice. Invoice ini ialah tagihan yang dilayangkan oleh perusahaan kepada pelanggan mengenai detail barang yang dikirim beserta biaya yang harus dibayarkan.

4. Forms

Berbagai formulir yang dapat dicetak offline untuk kepentingan arsip perusahaan dalam penjualan yang ia lakukan serta pengiriman tagihan resmi bagi pelanggan

5. Returns Management

Pengelolaan pengembalian barang dari pelanggan pada perusahaan kembali. Pengembalian barang ini dapat

diakibatkan karena barang yang diterima pelanggan tidak sesuai dengan aturan dan kesepakatan yang telah dibuat pada *sales order*

6. Lookup
Pemantauan terkait penjualan produk yang dilakukan perusahaan, baik pada quotation, *sales order*, maupun status perkembangan *sales order*
7. Reports
Laporan dalam penjualan produk perusahaan yang mencakup backlog, pengiriman, pemesanan, inventori, serta perpajakan produk.
8. Analytics
Tools yang digunakan untuk analisa penjualan produk. Analisa ini terdiri atas bookings, *sales history*, rangkuman *sales history*, serta time phased pada bookings dan *sales history*.
9. Prospect
Manajemen dalam penempatan prospek bagi pelanggan terkait penjualan produk perusahaan di masa mendatang
10. Customer
Manajemen dalam pendaftaran dan tracking terhadap pelanggan beserta statusnya bagi penjualan perusahaan
11. Item Pricing
Pengelolaan terkait harga bagi tiap produk perusahaan, mulai dari daftar harga, pembaruan harga, penjadwalan, penjualan, laporan, serta penentuan harga bagi tiap jenis pelanggan
12. Master Information
Kumpulan berbagai informasi yang berhubungan dengan *sales*, baik karakteristik, reputasi, detail lokasi pengiriman, perpajakan, kategori dan aturan penjualan, serta pencatatan akuntansi *sales*.
13. Utilities
Berbagai keperluan tambahan mengenai *sales*, seperti penentuan jenis pelanggan, pembaruan status kredit, pembersihan (purge) invoice dan memo, alokasi reservasi order, dan pengarsipan *sales history*.

5.2.10.2 Purchase

Modul ini memuat dan mengelola berbagai informasi dalam pembelian dan pengadaan material bagi perusahaan yang diperoleh dari vendor. Gambar 5.50 ialah contoh tampilan halaman *purchase* pada xTuple Postbooks.

Order #: 20110 Save Cancel

Header Information | Line Items | Characteristics | Notes | Comments | Quick Entry | Documents

Order #: 20110 Terms: [] Purchasing Agent: admin

Order Date: 1/1/17 Tax Zone: None Print on Save

Release Date: [] Status: Unreleased Sales Order #: []

Vendor: [] Purchase Type: []

VENDOR ADDRESS

Address: []

Street Address: []

Country: []

City: [] State: [] Postal: []

Receiving Site: WH1

Name: Master Prodiem Toys Retail Customer

Street Address: Prodiem Toys Inc.
1 Playland Way

Country: United States

City: Norfolk State: [] Postal: 23180

Name: []

Phone: [] Title: []

Alternate: [] Email: []

Fax: [] Web: []

Name: Mr. Jed Hastings

Phone: 800-342-5657 Title: []

Alternate: [] Email: jh@prodiem.com

Fax: 800-342-1100 Web: []

FOB: [] Ship Via: []

Gambar 5.50 Contoh Tampilan *Purchase Postbooks*

1. *Purchase Request*

Purchase request (PR) merupakan formulir permintaan yang diajukan perusahaan bagi vendor. Requisition yang dibuat memuat item atau barang-barang yang dibutuhkan beserta daftar harganya. Terdapat 2 cara dalam pembuatan *purchase order*, yakni:

- **By Planner Code**
Pembuatan PR berdasarkan kode perencanaan dalam pemesanan produk yang telah dibuat di awal oleh perusahaan
- **By Item**
Pembuatan PR berdasarkan *item* atau produk yang diinginkan secara langsung oleh perusahaan. Produk yang

dipesan ini disesuaikan dengan daftar produk dan harganya yang disediakan oleh vendor

2. *Purchase Order*

Pada bagian *purchase order* (PO) ini terdapat berbagai data yang perlu diisikan guna membuat order untuk diajukan kepada vendor

- General Information

Informasi umum dalam order yang terdiri atas nomor order, tanggal order, tanggal rilis, kode vendor, aturan order, pajak, status, dan jenis pembelian

- Header Information

Informasi detail dalam order yang akan ditinjau oleh perusahaan vendor. Informasi ini dibagi ke dalam 2 bagian utama, yakni detail alamat vendor, dan detail alamat pengiriman

- Line items

Informasi memuat berbagai detail *line* produk yang akan dipesan perusahaan beserta total perhitungan harga dan diskon produk untuk ditujukan bagi vendor

3. Voucher

Manajemen penentuan voucher untuk produk-produk yang dipesan perusahaan, baik berupa pembuatan, macam/ jenis, serta daftar voucher

4. Forms

Berbagai formulir yang dapat dicetak offline untuk kepentingan arsip perusahaan dalam pembelian dan pengadaan barang yang ia lakukan serta kepentingan kebutuhannya bagi vendor

5. Reports

Laporan dalam pembelian material perusahaan yang mencakup situs dan sumber daya produk, PR, PO, penerimaan dan pengembalian produk, serta material yang bermasalah

6. Vendor
Manajemen dalam pendaftaran dan tracking terhadap vendor beserta statusnya bagi pembelian material perusahaan
7. Item Source
Pengelolaan terkait sumber daya bagi tiap material perusahaan, baik pembuatan dan pendaftaran materialnya
8. Master Information
Kumpulan berbagai informasi yang berhubungan dengan *purchase*, baik planner code, reject code, aturan pembelian, kategori biaya pengeluaran, serta pencatatan akuntansi *purchase*.
9. Utilities
Berbagai keperluan tambahan mengenai *purchase*, seperti material yang tidak memerlukan sumber daya serta penentuan material bagi planner code

5.2.10.3 Accounting

Akuntansi yang disediakan mencakup seluruh pencatatan akuntansi yang berhubungan dengan keuangan perusahaan. Akuntansi yang disediakan adalah dalam bentuk macam-macam dokumen terkait pencatatan keuangan. Gambar 5.51 ialah contoh tampilan halaman *accounting* pada xTuple Postbooks.

1. Accounts Payable
Pencatatan akun hutang yang dilakukan oleh perusahaan. Akun ini akan masuk ke dalam bagian kredit pada buku kas besar dan neraca keuangan perusahaan. Hutang yang tercatat dalam dokumen ini mencakup PO, voucher, memo pembelian, pembayaran, serta formulir dan pelaporan terkait hutang
2. Accounts Receivable
Pencatatan akun piutang yang diterima oleh perusahaan. Akun ini akan masuk ke dalam bagian debit pada buku kas besar dan neraca keuangan perusahaan. Piutang yang tercatat dalam dokumen ini mencakup invoice, memo penjualan, penerimaan kas, serta formulir dan pelaporan terkait piutang

Standard Journal ? X

Name: Cancel

Description: Save

Transactions:

Account	Notes	Debit	Credit

Totals: Debits Credits

Notes:

New
Edit
View
Delete

Gambar 5.51 Contoh Tampilan *Accounting Postbooks*

3. General Ledger

Buku kas yang menjadi repositori utama dalam pencatatan akuntansi. Dalam buku kas ini dibagi menjadi 3 bagian. Pertama ialah simple journal guna ringkasan yang lebih sederhana dari standard journal. Melalui dokumen ini, perusahaan dapat langsung memantau pada pencatatan akuntansi yang belum terselesaikan atau membutuhkan tindak lanjut. Kedua standard journal yang menjadi buku kas besar untuk pencatatan seluruh aliran kas masuk dan keluar beserta pembayaran yang dilakukan. Seluruh pencatatan dari modul-modul lain yang berhubungan dengan akun hutang dan piutang serta kondisi-kondisi yang menyebabkan adanya perubahan keuangan perusahaan akan tercatat secara menyeluruh pada dokumen ini.

4. Bank Reconciliation

Pengaturan terkait akun rekonsiliasi bank pada perusahaan. Tiap pembayaran dan pengeluaran yang dilakukan perusahaan akan mempengaruhi neraca keuangan. Pada pengaturan ini juga dilakukan penyamarataan neraca keuangan antara debit dan kredit

5. Fixed Assets
Manajemen terkait aset-aset yang dimiliki perusahaan. Pengelolaan ini mencakup pendaftaran, lokasi penempatan, kategorisasi, pembelian, penjualan, depresiasi, pelaporan, dan pemantauan aset.
6. Financial Statements
Berbagai jenis pelaporan terkait finansial yang dibutuhkan perusahaan. Pelaporan utama dibuat dan dihasilkan melalui fitur ini. Laporan yang dicakup oleh fitur ini ialah ad hoc balance, neraca keuangan basic, laporan laba rugi basic, laporan aliran kas, neraca keuangan official, dan laporan laba rugi official
7. Fiscal Calendar
Laporan yang bertujuan sebagai rangkuman pencatatan akuntansi akhir tahun perusahaan. Pencatatan laporan ini akan menyebabkan seluruh akuntansi dalam satu tahun ditutup/ diselesaikan. Tiap catatan yang telah ditutup berarti telah selesai dan tidak akan dibahas lagi pada buku kas besar di awal tahun.
8. Ledger Accounts
Pengelolaan mengenai akun buku besar. Laporan yang dicatat adalah terkait informasi yang tidak tercakup pada buku kas besar, yakni chart of accounts (daftar rekening), akun perusahaan, nilai keuntungan, nilai subakun, dan kategori subakun
9. Budget
Dokumen *budget* digunakan untuk membuat dan mengelola *budget* perusahaan dalam rangka tujuan khusus/ tertentu. Informasi terkait *budget* ini terdiri atas tanggal mulai, tanggal berakhir, kode *budget*, serta deskripsi penjelasannya
10. Tax
Manajemen dalam perpajakan yang perlu dikelola oleh perusahaan. Pajak ini berpengaruh pada aliran keuangan kas perusahaan sehingga diperlukan pengelolaan tepat di dalamnya. Fitur pajak ini terdiri atas otorisasi, zona, kelas/ tingkat, kode, *assignment*, dan registrasi pajak

11. Utilities

Berbagai keperluan tambahan mengenai *accounting*, seperti duplikasi nomor akun, pembersihan *invoice*, dan pembaruan status kredit pelanggan yang terlambat.

5.3 Aspek Komunitas

Implementasi pengerjaan perbandingan FOS ERP untuk aspek komunitas adalah berdasarkan keaktifan komunitas. Pengukuran keaktifan komunitas ini dibedakan atas pengguna aktif dan pengguna pasif. Indikator pengukuran keaktifan komunitas pada pengguna aktif dan developer adalah seperti yang telah dijelaskan pada bab 2, yakni terdiri atas:

1. Jumlah *user* aktif

Jumlah pengguna yang memakasi FOS ERP yang sama dan bersedia untuk bergabung untuk saling bertukar informasi terkait FOS ERP

2. Jumlah *knowledge shared*

Jumlah informasi-informasi yang berguna dan penting mengenai FOS ERP dan menjadi pengetahuan di mata developer dan pengguna serta telah dipublikasikan dan diperbincangkan pada portal komunitas

3. Tanggal pos terakhir

Tanggal suatu pos/ *thread* yang akan menjadi *knowledge shared* terakhir kali diposkan dalam forum atau portal komunitas

4. Tanggal rilis terakhir

Tanggal sistem aplikasi dari suatu FOS ERP untuk versi terbarunya dirilis/ dikeluarkan

5. Rata-rata waktu rilis

Rata-rata jarak waktu antar rilis/ keluarnya tiap versi (antara versi lama dengan versi yang lebih baru) dari aplikasi FOS ERP.

Berikut daftar portal komunitas tiap FOS ERP yang digunakan sebagai bahan implementasi penelitian:

No	Nama FOS ERP	Link Portal Komunitas
1	Odoo	https://www.odoo.com/forum/help-1
2	Openbravo	http://forums.openbravo.com/
3	ADempiere	https://sourceforge.net/p/adempiere/discussion/
		https://groups.google.com/forum/#!forum/idempiere
4	Dolibarr	https://www.dolibarr.org/forum
5	ERPNext	https://discuss.erpnext.com/
6	WebERP	http://www.weberp.org/forum/
7	ERP5	https://www.erp5.com/forum
8	Opentaps	http://www.opentaps.org/forum
9	Apache OFBiz	http://mailarchives.apache.org/mod_mbox/ofbiz-user/
10	xTuple Postbooks	https://www.xtuple.org/

Tiap FOS ERP memiliki 1 situs forum/ portal komunitas. Hanya ADempiere yang memiliki 2 situs. Hal ini karena forum utamanya pada Sourceforge tidak terlalu aktif lagi dan fungsinya lebih sebagai repositori *knowledge shared* yang telah diperbincangkan sebelumnya. Developer dan komunitas ADempiere telah banyak bermigrasi pada iDempiere, proyek baru ADempiere dengan menyertakan OSGi (platform *service* dan *module system* untuk Java). iDempiere memiliki portal komunitas baru yang lebih aktif di Google Mailing List. Oleh sebab itu, ADempiere disini ditinjau dari 2 portal komunitas. Sedangkan pengukuran keaktifan komunitas pada pengguna pasif dan awam adalah melalui analisis jumlah pencarian dan perbincangannya dari berbagai portal, situs, sosial media, maupun lingkungan *internet* lainnya. Pengukuran ini dibatasi hanya melalui Google Search Engine dan Google Trends. Pengukuran dengan 2 cara tersebut sebenarnya sudah dilakukan sebelumnya pada bab 4 untuk melakukan pemilihan 10 FOS ERP. Hasil dari pengukuran tersebut juga digunakan untuk

analisis pengukuran keaktifan komunitas pada pengguna pasif dan awam di tiap FOS ERP.

BAB VI

HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil analisa dari implementasi penelitian yang telah dilaksanakan. Hasil analisa yang diberikan adalah terkait perbandingan keseluruhan FOS ERP beserta pembahasan dalam menentukan kesesuaian FOS ERP bagi penggunaannya.

6.1 Perbandingan Aspek Arsitektur

Subbab ini merupakan rangkuman hasil implementasi perbandingan FOS ERP pada aspek arsitektur. Hasil implementasi ini ditampilkan dalam bentuk tabel data hasil serta dijabarkan lebih lanjut pada pembahasannya

6.1.1 Data Hasil Perbandingan Arsitektur

Implementasi terkait gambaran dan detail arsitektur untuk masing-masing FOS ERP digabungkan untuk menjadi data hasil perbandingan. Pada tabel 6.1 menjelaskan mengenai hasil perbandingan antar FOS ERP pada aspek arsitektur

Tabel 6.1 Data Hasil Perbandingan Arsitektur

Kategori	Odoo	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
Jenis arsitektur	4 tier	3 tier	3 tier	3 tier	3 tier	3 tier	3 tier	3 tier	3 tier	3 tier
User Client	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Presentat ion	✓	✓	✓					✓	✓	✓
Service	✓			✓	✓					
Applicati on logic	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Data Access	✓	✓	✓	✓		✓		✓		✓
Data Store	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Bahasa	Java script, Python	Java, Java script,	Java	PHP	Java script, Python	ASP. Net, PHP	Python	Java	Free Marker, Java	Java Script
Database	Postgre SQL	Postgre SQL	Postgre SQL	My SQL	My SQL	My SQL	My SQL	My SQL	Postgre SQL	Postgre SQL

Kategori	Odo	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
Dukungan database lain	-	Apache Derby, MySQL, Oracle, HSQL DB	Oracle, MySQL	Postgre SQL, Maria DB	Maria DB	Maria DB	Postgre SQL, DB2	Postgre SQL	MySQL, Oracle, SQL Server, Sybase, DB2	Postgre SQL
Platform	Web	Web	Web	Web	Web	Web	Web	Web	Web	Desktop
Dukungan platform lain	Mobile	-	Desktop, Mobile	Mobile	Mobile	-	-	-	-	-
Sistem Operasi	Windows	Windows	Windows	Windows	Linux-Ubuntu dan CentOS	Windows	Windows	Windows	Windows	Windows
Dukungan Sistem Operasi lain	Linux, Mac OS X, Unix, Android, iOS	Linux, Mac OS X, Unix	Linux, Mac OS X, Unix, Android	Linux, Mac OS X, Unix, Android	Android	Windows	Linux, Mac OS X, Unix	Linux, Mac OS X, Unix	Linux, Mac OS X, Unix	Linux, Mac OS X, Unix

6.1.2 Pembahasan Perbandingan Arsitektur

1. Jenis arsitektur

Jenis arsitektur yang digunakan oleh FOS ERP dapat merepresentasikan 3 macam hal [1], yakni:

a) Flexibility

Fleksibilitas adalah tingkat sejauh mana sistem ERP dapat beradaptasi dengan perusahaan. Kriteria ini dapat dicapai melalui tingkat arsitektur yang dimiliki FOS ERP. Perusahaan menginginkan agar FOS ERP mampu menyesuaikan dengan keinginan dan kebutuhan mereka. Perubahan pada FOS ERP akan melihat pada arsitekturnya, yakni sejauh mana tingkat kompleksitas perubahan FOS ERP di tiap tingkat arsitektur agar sesuai dengan keinginan perusahaan

Dari 10 FOS ERP yang diteliti, terdapat 9 FOS ERP yang memiliki arsitektur 3 *tier*, dan 1 FOS yang berarsitektur 4 *tier*. Sesuai dengan penjelasan dari Motiwalla dan Thompson [1], maka keuntungan dari arsitektur 3 *tier* pada kriteria fleksibilitas ialah pada kemudahan partisipasi antar sistem. Sistem ERP dapat dibagi menjadi 3 macam tingkatan, yakni *database*, *application*, dan *presentation* (UI). Pembagian ini dapat memudahkan perusahaan mengubah bagian dari FOS ERP, tanpa harus mempengaruhi bagian lain. Jadi, apabila hanya diperlukan perubahan pada *database* yang digunakan, maka perusahaan tidak perlu repot mengkonfigurasi lagi bagian *application tier* dan *presentation layer* pada FOS ERP. FOS ERP yang memiliki arsitektur 3 *tier* adalah sebagai berikut:

- Openbravo
- ADempiere
- Dolibarr
- ERPNext
- WebERP
- ERP5
- Opentaps
- Apache OFBiz

- xTuple Postbooks

Sedangkan pada arsitektur 4 *tier* kemudahan di sisi fleksibilitasnya adalah pada integrasi antar sistemnya. Hal ini dikarenakan *presentation layer* dipartisi menjadi *Web service* dengan *Web browser*. *Web service* berfokus pada alur dalam akses *request* pada *application tier*. Sedangkan *Web browser* hanya fokus pada tampilan UI saja. Perusahaan dapat mengubah tampilan UI FOS ERP tanpa perlu mengubah konfigurasi jalur akses hubungan antara UI dengan *application tier*. Jadi melalui partisi sistem yang lebih kompleks dari arsitektur 3 *tier*, perusahaan pengguna dapat mengkonfigurasi FOS ERP lebih mudah dan mengintegrasikan komunikasi antar sistem secara lebih efektif. FOS ERP yang memiliki arsitektur 4 *tier* adalah Odoo.

b) Scalability

Skalabilitas merupakan tingkat sistem ERP dapat turut tumbuh sesuai dengan perkembangan perusahaan. Seluruh FOS ERP yang diujikan memiliki jenis arsitektur berlapis, dimana terdiri atas 9 FOS ERP berarsitektur 3 *tier* dan 1 FOS ERP berarsitektur 4 *tier*. Semakin banyak *tier* yang dimiliki, maka penambahan *hardware (device)* atau *software (aplikasi)* dapat semakin kompleks dan beragam. Jadi dengan begitu penambahan *user* dapat dimungkinkan dengan jumlah yang semakin banyak.

Perusahaan yang hendak memilih FOS ERP dapat mempertimbangkannya berdasarkan arsitektur FOS ERP dengan kebutuhan *hardware, software, dan user* yang mereka memiliki. Semakin beragam kebutuhan *hardware* dan *software*, maka semakin kompleks tingkat arsitektur FOS ERP yang dibutuhkan. Begitu pula dengan jumlah *user*, semakin banyak jumlah *user* yang harus mengakses sistem ERP tersebut, maka semakin banyak tingkatan arsitektur FOS ERP yang harus dimiliki.

Berdasarkan jumlah *tier* yang dicakup dalam arsitektur FOS ERP, maka FOS ERP dengan arsitektur 3 *tier* lebih

cocok digunakan oleh perusahaan skala kecil dan menengah/ disebut juga dengan istilah Usaha, Mikro, Kecil, dan Menengah (UMKM) di Indonesia. Hal ini dikarenakan jenis perusahaan tersebut memiliki kebutuhan *hardware* dan *software* yang tidak terlalu kompleks serta jumlah *user* yang minimal. Perusahaan skala besar tentu memiliki kebutuhan *hardware* dan *software* yang lebih beragam serta jumlah *user* pengakses sistem ERP yang lebih banyak. Oleh sebab itu, perusahaan skala besar lebih dianjurkan untuk menggunakan FOS ERP dengan arsitektur 4 *tier*

c) Security

Tingkat keamanan termasuk dalam kriteria penilaian FOS ERP di bidang arsitektur. FOS ERP yang memiliki tingkat keamanan yang baik akan dapat menjamin integritas dan kerahasiaan data penggunanya. Dalam FOS ERP yang diakses melalui *web browser*, maka diperlukan koneksi keamanan melalui *Secure Socket Layer* (SSL). *Secure Socket Layer* adalah protokol keamanan guna mengamankan semua bentuk transaksi yang ada di *website*. Identifikasi keamanan *website* dari FOS ERP yang paling mudah dapat ditelusuri pada protokol HTTP (*Hypertext Transfer Protocol*) yang dimiliki. *Website* FOS ERP yang aman adalah situs dengan protokol HTTPS (*Hypertext Transfer Protocol Secure*). Berdasarkan tabel 4.2 mengenai situs developer FOS ERP, maka FOS ERP yang memiliki tingkat keamanan situs yang baik adalah sebagai berikut:

- Odoo
- Dolibarr
- ERP5
- Apache OFBiz
- xTuple Postbooks

Selain ketiga penilaian di atas, FOS ERP juga dapat ditinjau dari jumlah komponen arsitektur yang dicakup. Terdapat 6 macam komponen arsitektur *software* yang menjadi acuan arsitektur FOS ERP [10]. Dari 6 komponen tersebut (*client*,

presentation, service, application logic, data access, dan data store), maka dapat disimpulkan bahwa Odoo memiliki arsitektur yang paling kompleks dan lengkap, karena mampu mencakup keseluruhan komponen arsitektur. FOS ERP lainnya yang paling mendekati kelengkapan komponen arsitektur adalah Openbravo, ADempiere, Opentaps dan xTuple Postbooks. Masing-masing FOS ERP tersebut hanya mencakup 5 komponen, dengan tidak menyediakan komponen *service*.

2. Bahasa pemrograman

Bahasa pemrograman yang digunakan untuk membangun FOS ERP juga menjadi pertimbangan dalam pemilihan FOS ERP bagi perusahaan. Perusahaan akan menyesuaikan FOS ERP yang dipilih dengan bahasa pemrograman yang dapat digunakan oleh teknisinya. Tujuannya agar teknisi perusahaan dapat dengan mudah mengembangkan dan melakukan kustomisasi FOS ERP tersebut agar sesuai dengan kebutuhan perusahaan. Berdasarkan artikel yang ditulis oleh Stephen Cass [34], maka daftar bahasa pemrograman terbaik dan paling banyak digunakan adalah sebagai berikut:

- 1) C
- 2) Java
- 3) Python
- 4) C++
- 5) R
- 6) C#
- 7) PHP
- 8) JavaScript
- 9) Ruby
- 10) Go

Daftar ini yang menjadi acuan dalam menentukan bahasa pemrograman FOS ERP yang sesuai bagi perusahaan. Berdasarkan daftar tersebut, maka tabel 6.2 menjelaskan peringkat/ urutan FOS ERP yang memiliki bahasa pemrograman yang baik dan sesuai bagi perusahaan.

Tabel 6.2 Peringkat berdasarkan Bahasa Pemrograman

Peringkat	Nama FOS ERP	Bahasa Pemrograman
1	Openbravo	Java, JavaScript
2	Odoo	JavaScript, Python
3	ERPNext	JavaScript, Python
4	ADempiere	Java
5	Opentaps	Java
6	ERP5	Python
7	Apache OFBiz	Free Marker, Java
8	Dolibarr	PHP
9	WebERP	ASP. Net, PHP
10	xTuple Postbooks	JavaScript

Urutan peringkat ini berdasarkan jumlah bahasa pemrograman yang sesuai dengan hasil penelitian pada artikel Stephen Cass [34].

3. Jenis *database*

Database merupakan salah satu komponen penting pada FOS ERP. Seluruh FOS ERP yang diteliti membutuhkan *database* pada *tier* pertama arsitekturnya. Hal ini karena kebutuhan *database* sangat esensial sebagai repositori seluruh data-data yang akan diolah dan dikelola dalam FOS ERP. Pilihan *database* yang tepat juga penting bagi perusahaan. Perusahaan skala kecil dan menengah mungkin cukup dengan jenis *database* yang mudah instalasinya dan ringan untuk dijalankan di sisi server/ komputer. Berbeda dengan perusahaan besar yang membutuhkan *database* dengan repositori besar namun tetap memiliki respon tinggi dalam pemrosesannya.

Jumlah jenis *database* yang dapat mendukung FOS ERP juga penting dalam pemilihan FOS ERP bagi perusahaan. FOS ERP yang memiliki banyak jenis *database* pendukung memang belum tentu menjamin performanya bagus ketika dijalankan. Namun perusahaan juga perlu menyesuaikan

kemampuan teknisnya terkait jenis *database* yang dapat ia gunakan. Perusahaan juga mempertimbangkan pada jenis *database* yang baik digunakan dalam jangka panjang dan mampu mengikuti perkembangan perusahaan. Oleh sebab itu, pilihan jenis *database* yang banyak mendukung FOS ERP tersebut dapat pula menjadi pertimbangan agar perusahaan fleksibel dalam memilih *database* yang mereka inginkan untuk konfigurasi FOS ERP.

Dari kesepuluh FOS ERP yang diujikan, terdapat 5 FOS ERP yang menggunakan PostgreSQL dan sisanya menggunakan MySQL sebagai jenis *database* yang utama. Berdasarkan data statistik dari DB-Engines [35], diperoleh peringkat jenis *database* terbaik dan paling banyak digunakan oleh *user*, yaitu antara lain:

- 1) Oracle
- 2) MySQL
- 3) Microsoft SQL Server
- 4) PostgreSQL
- 5) MongoDB
- 6) IBM DB2
- 7) Cassandra
- 8) Microsoft Access
- 9) Redis
- 10) SQLite

Daftar ini yang menjadi acuan dalam menentukan jenis *database* FOS ERP yang sesuai bagi perusahaan. Apabila hanya melihat dari jenis *database* utamanya, maka FOS ERP dengan *database* MySQL lebih disarankan bagi perusahaan karena tingkat popularitasnya lebih tinggi dari PostgreSQL. Namun jika melihat pada cakupan jenis *database* yang dapat didukung, maka pada tabel 6.3 dijelaskan peringkat/ urutan FOS ERP yang memiliki cakupan jenis *database* yang baik dan sesuai bagi perusahaan

Tabel 6.3 Peringkat berdasarkan Jenis Database

Peringkat	Nama FOS ERP	Jenis Database
1	Apache OFBiz	PostgreSQL, MySQL, Oracle, SQL Server, Sybase, DB2
2	Openbravo	Apache Derby, MySQL, PostgreSQL, Oracle, HSQLDB
3	ADempiere	PostgreSQL, Oracle, MySQL
4	ERP5	PostgreSQL, MySQL, DB2
5	Dolibarr	MySQL, PostgreSQL, MariaDB
6	Opentaps	MySQL, PostgreSQL
7	WebERP	MySQL, MariaDB
8	Odoo	PostgreSQL
9	xTuple Postbooks	PostgreSQL
10	ERPNext	MariaDB

Urutan peringkat ini berdasarkan jumlah jenis *database* yang sesuai dengan hasil data statistik *database* dari DB-Engines [35].

4. Platform aplikasi

Perusahaan dalam memilih FOS ERP juga dapat melihat pada platform aplikasi yang mendukung sistem ERP tersebut. Semakin beragam jenis platform yang dapat dijalankan, maka FOS ERP tersebut dapat memudahkan perusahaan dalam menggunakannya. Platform aplikasi ini dibedakan menjadi 3 jenis, yakni *desktop application*, *web application*, dan *mobile application*. Sistem ERP yang membutuhkan integrasi data antar departemen/ divisi tentu harus dapat dijalankan dengan mudah oleh tiap departemennya. *Web application* merupakan platform aplikasi yang paling mudah untuk dapat mengintegrasikan antar *user* pada satu server utama. Hal ini dikarenakan akses aplikasinya dapat dijalankan melalui *web browser*. Berbeda dengan *desktop application* yang membutuhkan instalasi aplikasi terlebih dahulu untuk disematkan ke dalam tiap

komputer *user*. Sedangkan *mobile application* merupakan alternatif platform yang dapat digunakan di samping penggunaan *web* atau *desktop application*. Karena kemudahannya untuk dapat diakses secara *mobile*, maka sistem ERP jadi memiliki kemampuan mobilitas yang tinggi. Namun dengan pertimbangan keterbatasan yang dimiliki *mobile application*, maka fungsionalitas yang dicakup tidak akan selengkap *web* atau *desktop application*. Oleh sebab itu penggunaannya lebih difokuskan sebagai alternatif platform untuk kegiatan-kegiatan yang sederhana terhadap sistem ERP.

Berdasarkan data hasil perbandingan yang diperoleh, maka dapat disimpulkan bahwa seluruh FOS ERP dapat dijalankan melalui *web application*. Karena tidak ada perbedaan pada platform utamanya, jadi pertimbangan selanjutnya adalah pada cakupan platform aplikasi lain yang dapat ditunjang. Berikut tabel 6.4 menjelaskan urutan peringkat FOS ERP berdasarkan cakupan platform aplikasinya

Tabel 6.4 Peringkat berdasarkan Platform Aplikasi

Peringkat	Nama FOS ERP	Platform Aplikasi
1	ADempiere	<i>Web, Mobile, Desktop</i>
2	Odoo	<i>Web, Mobile</i>
3	ERPNext	<i>Web, Mobile</i>
4	Dolibarr	<i>Web, Mobile</i>
5	Openbravo	<i>Web</i>
6	WebERP	<i>Web</i>
7	ERP5	<i>Web</i>
8	Opentaps	<i>Web</i>
9	Apache OFBiz	<i>Web</i>
10	xTuple Postbooks	<i>Web</i>

5. Sistem operasi pendukung

Kriteria terakhir dalam aspek arsitektur FOS ERP adalah mengenai sistem operasi yang mendukung. Tiap perusahaan tentu menggunakan sistem operasi yang berbeda-beda. Bahkan dalam satu perusahaan pun dapat pula memakai

sistem operasi yang tidak sama. Oleh sebab itu, FOS ERP yang didukung sistem operasi yang beragam akan dapat menyesuaikan tiap perusahaan yang menggunakannya. Pada kesepuluh FOS ERP yang diuji, 9 FOS ERP dapat dijalankan di sistem operasi yang paling umum digunakan, yakni Windows. Hanya ERPNext yang tidak didukung oleh Windows sehingga sistem operasi utamanya adalah pada Linux. Selain dari sistem operasi utamanya, jumlah sistem operasi yang mendukung juga dapat menjadi penilaian. Tabel 6.5 menjelaskan urutan peringkat FOS ERP berdasarkan jumlah sistem operasi yang dapat didukung.

Tabel 6.5 Peringkat berdasarkan Sistem Operasi

Peringkat	Nama FOS ERP	Sistem Operasi
1	Odoo	Linux, Mac OS X, Unix, Windows, Android, iOS
2	ADempiere	Linux, Mac OS X, Unix, Windows, Android
3	Dolibarr	Linux, Mac OS X, Unix, Windows, Android
4	Openbravo	Linux, Mac OS X, Unix, Windows
5	ERP5	Linux, Mac OS X, Unix, Windows
6	Opentaps	Linux, Mac OS X, Unix, Windows
7	Apache OFBiz	Linux, Mac OS X, Unix, Windows
8	xTuple Postbooks	Linux, Mac OS X, Unix, Windows
9	WebERP	Linux, Windows
10	ERPNext	Linux (Ubuntu dan CentOS)

Melalui berbagai hasil perbandingan di tiap kategori dalam aspek arsitektur ini, maka disimpulkan 3 ERP yang paling baik dan sesuai untuk diterapkan di perusahaan. ERP tersebut antara lain adalah:

1. Odoos
 - Memiliki arsitektur 4 *tier*
 - Mencakup 6 komponen arsitektur *software*
 - Tingkat keamanan web yang baik
 - Bahasa pemrograman JavaScript dan Python
 - Dapat berjalan pada platform *web* dan *mobile application*
 - Dukungan sistem operasi yang beragam (Linux, Mac OS X, Unix, Windows, Android, iOS)
2. Openbravo
 - Memiliki arsitektur 3 *tier*
 - Mencakup 5 komponen arsitektur *software*
 - Bahasa pemrograman JavaScript dan Java
 - Dukungan jenis *database* yang beragam (Apache Derby, MySQL, PostgreSQL, Oracle, HSQLDB)
 - Dukungan sistem operasi yang beragam (Linux, Mac OS X, Unix, Windows)
3. ADempiere
 - Memiliki arsitektur 3 *tier*
 - Mencakup 6 komponen arsitektur *software*
 - Bahasa pemrograman Java
 - Dapat berjalan pada platform *web*, *desktop* dan *mobile application*
 - Dukungan jenis *database* yang beragam (MySQL, PostgreSQL, Oracle)
 - Dukungan sistem operasi yang beragam (Linux, Mac OS X, Unix, Windows, Android)

6.2 Perbandingan Aspek Fungsionalitas

Subbab ini merupakan rangkuman hasil implementasi perbandingan FOS ERP pada aspek fungsionalitas. Hasil implementasi ini ditampilkan dalam bentuk tabel data hasil serta dijabarkan lebih lanjut pada pembahasannya

6.2.1 Data Hasil Perbandingan Fungsionalitas

Implementasi terkait instalasi dan uji coba pada fungsionalitas untuk masing-masing FOS ERP (modul *sales*, *purchase*, dan *accounting*) digabungkan untuk menjadi data hasil perbandingan. Berikut hasil hasil perbandingan antar FOS ERP pada aspek fungsionalitas:

1. Tabel 6.2 menjelaskan mengenai perbandingan fitur/aktivitas utama pada modul *sales*
2. Tabel 6.3 menjelaskan mengenai fitur tambahan pada modul *sales*
3. Tabel 6.4 menjelaskan mengenai perbandingan fitur/aktivitas utama pada modul *purchase*
4. Tabel 6.5 menjelaskan mengenai fitur tambahan pada modul *purchase*
5. Tabel 6.6 menjelaskan mengenai perbandingan fitur/aktivitas utama pada modul *accounting*
6. Tabel 6.7 menjelaskan mengenai fitur tambahan pada modul *accounting*

Tabel 6.6 Perbandingan Fitur Utama pada Modul Sales

Fitur/ Aktivitas Utama	Odoo	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
<i>Quotation</i>	✓	✓	✓		✓	✓			✓	✓
<i>Sales order</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Good shipment</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Return from customer</i>	✓	✓							✓	✓
<i>Return material receipt</i>		✓								
<i>Sales invoice</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Sales payment</i>	✓			✓	✓		✓		✓	

Tabel 6.7 Perbandingan Fitur Tambahan pada Modul Sales

Odoo	Openbravo	ADempiere	Dolibarr	ERP Next	Web ERP	ERP5	Opentaps	OFBiz	Postbooks
<i>Sales Report</i>	<i>Generate Invoice</i>	<i>Order Setting</i>		<i>Sales Stock Ledger</i>	<i>Counter Sales</i>	<i>Journalise Transaction</i>	<i>Order Setting</i>	<i>Find Order</i>	<i>Sales Report</i>
	<i>Commission Payment</i>	<i>Subscription</i>		<i>Sales Accounting</i>	<i>Counter Returns</i>		<i>Order Organize</i>	<i>Detail Order</i>	<i>Sales Prospect</i>
	<i>Sales and Marketing Setup</i>	<i>Sales and Marketing Setup</i>			<i>Outstand Sales Order</i>		<i>Order Summary</i>		<i>Sales Master Information</i>
	<i>Sales Analysis Tools</i>				<i>Recurring Sales Order</i>				<i>Sales Analysis Tools</i>

Tabel 6.8 Perbandingan Fitur Utama pada Modul *Purchase*

Fitur/ Aktivitas Utama	Odoo	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
<i>RFQ</i>	✓		✓							
<i>Requisition</i>		✓	✓		✓	✓			✓	✓
<i>Purchase order</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Good receipt</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Return to Vendor</i>	✓	✓							✓	
<i>Return to Vendor Shipment</i>		✓								
<i>Purchase Invoice</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Purchase payment</i>	✓			✓	✓		✓		✓	

Tabel 6.9 Perbandingan Fitur Tambahan pada Modul Purchase

Odoo	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
<i>Purchase Report</i>	<i>Manage requisiton</i>	<i>Purchase Matching</i>		<i>Purchase Stock Ledger</i>	<i>Authorize Purchase Order</i>	<i>Journalise Transacti on</i>	<i>Order Setting</i>	<i>Find Order</i>	<i>Purchase Report</i>
	<i>Matched Invoice</i>			<i>Purchase Accounti ng</i>	<i>Amending Purchase Order</i>		<i>Order Organize</i>	<i>Detail Order</i>	<i>Purchase Voucher</i>
	<i>Purchase Analysis Tools</i>						<i>Order Summary</i>		<i>Purchase Master Informati on</i>
	<i>Landed Cost</i>								<i>Purchase Contract</i>

Tabel 6.10 Perbandingan Fitur Utama pada Modul Accounting

Fitur/ Aktivitas Utama	Odoos	Open bravo	ADem piere	Doli barr	ERP Next	Web ERP	ERP5	Opent aps	OFBiz	Post books
<i>Sales/ Purchase Journal</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>General Ledger</i>	✓	✓	✓		✓	✓	✓	✓	✓	✓
<i>Trial Balance</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Balance Sheet</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Profit and Loss</i>	✓	✓	✓		✓	✓	✓	✓	✓	
<i>Budget</i>		✓	✓			✓		✓		✓
<i>Cash Flow Statement</i>			✓			✓		✓		✓
<i>Chart of Account</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

6.2.2 Pembahasan Perbandingan Fungsionalitas

Analisis pembahasan dari data hasil perbandingan FOS ERP pada aspek fungsionalitas dibagi ke dalam 3 bagian sesuai dengan modul yang diteliti, yakni pembahasan pada modul *sales*, *purchase*, dan *accounting*

6.2.2.1 Pembahasan Perbandingan Modul *Sales*

Dalam modul *sales* terdapat 7 aktivitas utama, dimana tiap 1 aktivitasnya dikatakan sebagai 1 fitur. Fitur/ aktivitas utama yang dicakup dalam modul *sales* adalah sebagai berikut:

1. Quotation
Formulir penawaran barang bagi pelanggan
2. *Sales Order*
Pemesanan barang dari pelanggan
3. Good Shipment
Pengiriman barang kepada pelanggan
4. Return from Customer
Pengembalian barang dari pelanggan (biasanya diakibatkan permasalahan barang yang diterima pelanggan)
5. Return Material Receipt
Penerimaan pengembalian barang dari pelanggan kepada perusahaan
6. *Sales Invoices*
Tagihan barang yang dibeli pelanggan
7. *Sales Payment*
Pembayaran barang dari pelanggan pada perusahaan.

Berdasarkan data hasil perbandingan FOS ERP pada fitur/ aktivitas utama modul *sales*, maka dapat disimpulkan bahwa tidak ada FOS ERP yang dapat memenuhi keseluruhan fitur tersebut. Namun terdapat 3 FOS ERP yang paling banyak memenuhi fitur utama modul *sales*, yakni Odoo, Openbravo, dan Apache OFBiz. Odoo memenuhi 6 aktivitas modul *sales*, kecuali *return material receipt*. Openbravo memenuhi 6 aktivitas modul *sales*, kecuali *sales payment*. Sedangkan Apache OFBiz sama halnya dengan Odoo, memenuhi 6 aktivitas modul *sales* kecuali *return material receipt*.

FOS ERP lainnya tidak dapat memenuhi 2 hingga 4 fitur utama pada modul *sales*. FOS ERP yang paling sedikit memenuhi fitur

utama modul *sales* adalah Opentaps. FOS ERP ini hanya memenuhi 3 aktivitas modul *sales*, yakni *sales order*, *good shipment*, dan *sales invoice*. Jadi, dapat pula disimpulkan bahwa fitur/ aktivitas utama yang dimiliki oleh seluruh FOS ERP pada modul *sales* adalah *sales order*, *good shipment*, dan *sales invoice*.

Selain fitur utama, terdapat pula fitur tambahan yang ditawarkan oleh tiap FOS ERP dalam modul *sales*. Fitur tambahan ini memiliki nama dan fungsi yang berbeda-beda. Penjelasan untuk masing-masing fitur tambahan telah dijabarkan pada bab 5.2 bagian implementasi perbandingan tiap FOS ERP untuk modul *sales*. FOS ERP yang memiliki fitur tambahan paling banyak adalah Openbravo, WebERP, dan xTuple Postbooks, yakni sebanyak 4 fitur. Masing-masing fitur tambahan yang ditawarkan antara lain:

1. Openbravo
Generate invoice, commission payment, sales and marketing setup, dan sales analysis tools
2. WebERP
Counter sales, counter returns, outstand sales order, recurring sales order
3. xTuple Postbooks
Sales report, sales prospect, sales master information, dan sales analysis tools

Berdasarkan analisis pada kelengkapan fitur utama dan keberagaman fitur tambahan yang dimiliki, maka FOS ERP yang sesuai untuk diterapkan perusahaan pada modul *sales* adalah Openbravo. Hal ini dikarenakan Openbravo termasuk FOS ERP yang paling banyak memenuhi fitur utama dan memiliki jumlah fitur tambahan yang paling beragam pada modul *sales*.

6.2.2.2 Pembahasan Perbandingan Modul *Purchase*

Dalam modul *purchase* terdapat 8 aktivitas utama, dimana tiap 1 aktivitasnya dikatakan sebagai 1 fitur. Fitur/ aktivitas utama yang dicakup dalam modul *purchase* adalah sebagai berikut:

1. RFQ
Pengajuan *quotation* dari perusahaan bagi vendor
2. Requisition
Permintaan barang yang ingin dibeli kepada vendor
3. *Purchase Order*
Pembelian barang kepada vendor
4. Good Receipt
Penerimaan barang yang dikirim dari vendor
5. Return to Vendor
Pengembalian barang kepada vendor (biasanya diakibatkan permasalahan barang yang diterima perusahaan)
6. Return Vendor Shipment
Pengiriman pengembalian barang kepada vendor
7. *Purchase Invoices*
Tagihan barang yang dibeli perusahaan
8. *Purchase Payment*
Pembayaran barang dari perusahaan pada vendor

Berdasarkan data hasil perbandingan FOS ERP pada fitur/ aktivitas utama modul *purchase*, maka dapat disimpulkan bahwa tidak ada FOS ERP yang dapat memenuhi keseluruhan fitur tersebut. Namun terdapat 3 FOS ERP yang paling banyak memenuhi fitur utama modul *purchase*, yakni Odoo, Openbravo, dan Apache OFBiz. Odoo memenuhi 6 aktivitas modul *purchase*, kecuali *requisition* dan *return to vendor shipment*. Openbravo memenuhi 6 aktivitas modul *purchase*, kecuali RFQ dan *purchase payment*. Sedangkan Apache OFBiz memenuhi 6 aktivitas modul *purchase*, kecuali RFQ dan *return to vendor shipment*.

FOS ERP lainnya tidak dapat memenuhi 3 hingga 5 fitur utama pada modul *purchase*. FOS ERP yang paling sedikit memenuhi fitur utama modul *purchase* adalah Opentaps. FOS ERP ini hanya memenuhi 3 aktivitas modul *purchase*, yakni *purchase order*, *good receipt*, dan *purchase invoice*. Jadi, dapat pula disimpulkan bahwa fitur/ aktivitas utama yang dimiliki oleh seluruh FOS ERP pada modul *purchase* adalah *purchase order*, *good receipt*, dan *purchase invoice*.

Selain fitur utama, terdapat pula fitur tambahan yang ditawarkan oleh tiap FOS ERP dalam modul *purchase*. Fitur tambahan ini memiliki nama dan fungsi yang berbeda-beda. Penjelasan untuk masing-masing fitur tambahan telah dijabarkan pada bab 5.2 bagian implementasi perbandingan tiap FOS ERP untuk modul *purchase*. FOS ERP yang memiliki fitur tambahan paling banyak adalah Openbravo, dan xTuple Postbooks, yakni sebanyak 4 fitur. Masing-masing fitur tambahan yang ditawarkan antara lain:

1. Openbravo
Manage requisition, matched invoice, purchase analysis tools, dan landed cost
2. xTuple Postbooks
Purchase report, purchase voucher, purchase master information, dan purchase contract

Berdasarkan analisis pada kelengkapan fitur utama dan keberagaman fitur tambahan yang dimiliki, maka FOS ERP yang sesuai untuk diterapkan perusahaan pada modul *purchase* adalah Openbravo. Hal ini dikarenakan Openbravo termasuk FOS ERP yang paling banyak memenuhi fitur utama dan memiliki jumlah fitur tambahan yang paling beragam pada modul *purchase*.

6.2.2.3 Pembahasan Perbandingan Modul *Accounting*

Dalam modul *purchase* terdapat 7 jenis laporan dan 1 akun utama, dimana masing-masingnya dikatakan sebagai 1 fitur (total terdapat 8 fitur). Fitur utama yang dicakup dalam modul *accounting* adalah sebagai berikut:

1. *Sales/ Purchase Journal*
Jurnal pencatatan transaksi yang dilakukan perusahaan. Terdiri atas jurnal pembelian (*sales journal*) dan jurnal penjualan (*purchase journal*)
2. General Ledger
Buku kas besar untuk pencatatan seluruh aliran kas masuk/ piutang (debit) dan keluar/ hutang (kredit) beserta pembayaran yang dilakukan
3. Trial Balance

Neraca saldo yang memuat seluruh akun dengan saldo totalnya yang bersumber dari *general ledger*

4. Balance Sheet
Neraca keuangan yang menampilkan pengeluaran (akun hutang) dan pendapatan (akun piutang) beserta total keuntungan/ kerugian yang didapat
5. Profit and Loss Statement
Pencatatan seluruh keuntungan yang diperoleh dan kerugian yang dikeluarkan
6. Budget
Pembuatan dan pengelolaan anggaran sebagai pedoman aktivitas transaksi di masa mendatang
7. Cash Flow Statement
Laporan keuangan berupa aliran kas masuk dan keluar pada periode tertentu
8. Chart of Account
Daftar akun rekening yang terstruktur untuk memudahkan penyusunan dan pembacaan tiap akun dalam akuntansi

Berdasarkan data hasil perbandingan FOS ERP pada fitur/ aktivitas utama modul *accounting*, maka dapat disimpulkan bahwa terdapat 4 FOS ERP yang dapat memenuhi keseluruhan fitur tersebut, yakni ADempiere, WebERP, Opentaps, dan xTuple Postbooks. Keempat FOS ERP tersebut mampu menghasilkan ketujuh laporan utama dan memiliki pengelolaan akun rekening.

FOS ERP lainnya tidak dapat memenuhi 2 hingga 4 fitur utama pada modul *accounting*. FOS ERP yang paling sedikit memenuhi fitur utama modul *accounting* adalah Dolibarr. FOS ERP ini hanya memenuhi 4 fitur modul *accounting*, yakni *sales/purchase journal*, *trial balance*, *balance sheet*, dan *chart of account*. Fitur utama yang dimiliki oleh seluruh FOS ERP pada modul *accounting* adalah *sales/purchase journal*, *trial balance*, dan *balance sheet*, dan *chart of account*.

Selain fitur utama, terdapat pula fitur tambahan yang ditawarkan oleh tiap FOS ERP dalam modul *accounting*. Fitur tambahan ini memiliki nama dan fungsi yang berbeda-beda. Penjelasan untuk masing-masing fitur tambahan telah dijabarkan pada bab 5.2

bagian implementasi perbandingan tiap FOS ERP untuk modul *accounting*. FOS ERP yang memiliki fitur tambahan paling banyak adalah Odoo, yakni sebanyak 5 fitur. Sedangkan pada FOS ERP yang memiliki fitur utama paling lengkap, terdapat WebERP dengan 3 fitur tambahan dan Opentaps dengan 4 fitur tambahan

Berdasarkan analisis pada kelengkapan fitur utama yang dimiliki, maka FOS ERP yang sesuai untuk diterapkan perusahaan pada modul *accounting* adalah Opentaps. Hal ini dikarenakan Opentaps merupakan FOS ERP yang mampu memenuhi seluruh fitur utama pada modul *accounting* dan mempunyai fitur tambahan yang banyak (4 fitur).

6.3 Perbandingan Aspek Komunitas

Subbab ini merupakan rangkuman hasil implementasi perbandingan FOS ERP pada aspek komunitas. Hasil implementasi ini ditampilkan dalam bentuk data hasil serta dijabarkan lebih lanjut pada pembahasannya

6.3.1 Data Hasil Perbandingan Komunitas

Implementasi terkait keaktifan komunitas dijalankan dibedakan atas pengguna aktif dan pasif. Pengguna aktif dan *developer* adalah berdasarkan 5 indikator yang dihimpun dari tiap situs *developer* dan forum/ portal komunitas FOS ERP. Sedangkan pengguna pasif dan awam diukur melalui analisis Google Search Engine dan Google Trends. Berikut pada tabel 6.8 menjelaskan mengenai hasil perbandingan antar FOS ERP pada aspek komunitas untuk pengguna aktif. Pencarian tiap indikator pada aspek komunitas untuk pengguna aktif dilakukan pada tanggal yang sama, yakni 12 Januari 2017.

Tabel 6.12 Data Hasil Perbandingan Komunitas Pengguna Aktif

Jenis Sistem ERP	Jumlah <i>user</i> aktif	Jumlah <i>knowledge shared</i>	Tanggal pos terakhir	Tanggal rilis terakhir	Rata-rata waktu rilis	Keterangan
Odoo	8549 <i>user</i>	36426 pos	12 Januari 2017	Oktober 2016	1 tahun	Tersedia halaman forum komunitas di situs resmi vendor
ADempiere	1153 <i>user</i>	15079 pos	12 Januari 2017	31 Oktober 2015	1 tahun	Forum melalui Google mailing list
Openbravo	18183 <i>user</i>	73783 pos (20620 topik)	8 Januari 2017	16 Desember 2016	1 bulan	Tersedia halaman forum komunitas di situs resmi vendor
ERPNext	4437 <i>user</i>	14002 pos	12 Januari 2017	10 Januari 2017	1 minggu	Tersedia halaman forum komunitas di situs resmi vendor
Opentaps	-	12589 pos (3735 topik)	22 September 2015	11 Januari 2013	2 tahun	Tersedia halaman forum komunitas di situs resmi vendor
Dolibarr	8755 <i>user</i>	12663 pos (3951 topik)	11 Januari 2017	Juni 2016	6 bulan	Tersedia halaman forum komunitas di situs resmi vendor

Jenis Sistem ERP	Jumlah <i>user</i> aktif	Jumlah <i>knowledge shared</i>	Tanggal pos terakhir	Tanggal rilis terakhir	Rata-rata waktu rilis	Keterangan
WebERP	23133 <i>user</i>	5612 pos (1361 topik)	12 Januari 2017	27 November 2016	5 bulan	Mailing list mati. Forum komunitas baru dirintis
Apache OFBiz	1270 <i>user</i>	50514 pos	12 Januari 2017	27 November 2016	9 bulan	Tersedia halaman forum komunitas di situs resmi vendor
ERP5	3 <i>user</i>	3 pos	4 Januari 2017	3 November 2014	1,5 tahun	Tersedia halaman forum komunitas di situs resmi vendor
xTuple Post books	-	16382 pos (3961 topik)	21 Desember 2016	Oktober 2016	4 bulan	Tersedia halaman forum komunitas di situs resmi vendor

Selanjutnya, hasil perbandingan antar FOS ERP pada aspek komunitas untuk pengguna pasif dan awam adalah berdasarkan Google Search Engine dan Google Trends. Analisis pada kedua cara ini dilakukan pada tanggal yang sama, yakni 10 November 2016. Berikut jumlah hasil pencarian tiap FOS ERP pada Google Search Engine:

- 1) Odoo : ± 11.400.000
- 2) Openbravo : ± 3.920.000
- 3) ADempiere : ± 2.570.000
- 4) Dolibarr : ± 579.000
- 5) ERPNext : ± 136.000
- 6) WebERP : ± 273.000
- 7) ERP5 : ± 706.000
- 8) Opentaps : ± 79.600
- 9) Apache OFBiz : ± 121.000
- 10) xTuple Postbooks : ± 508.000

Analisis tiap FOS ERP pada Google Trends membutuhkan detail untuk pengukuran yang spesifik. Berikut detail pengukuran spesifik yang digunakan:

- a) Area pencarian : Seluruh dunia
- b) Rentang waktu : 5 tahun terakhir (2011-2016)
- c) Kategori pencarian : Semua kategori
- d) Metode penelusuran : Penelusuran *web*

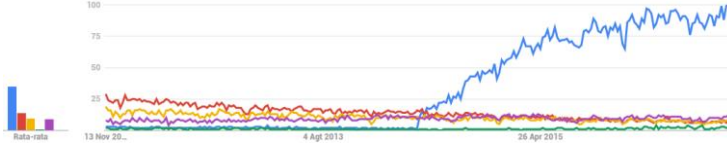
Hasil pencarian melalui Google Trends secara umum adalah berupa diagram garis untuk tiap FOS ERP sesuai rentang waktunya. Diagram garis tersebut dirangkum dalam bentuk perhitungan rata-rata yang disajikan melalui diagram batang. Karena perbandingan dalam Google Trends hanya dibatasi untuk 5 macam *keyword* saja, maka 10 FOS ERP ini dibandingkan dari hasil peringkat pada cara pertama (jumlah pencarian melalui Google Search Engine) dengan dibedakan atas beberapa kategori FOS ERP.

Kategori FOS ERP yang pertama dibandingkan ialah pada 5 FOS ERP yang memiliki jumlah pencarian pada Google Search terbanyak, yakni:

1. Odoo : bertanda biru
2. Openbravo : bertanda merah

3. ADempiere : bertanda kuning
4. ERP5 : bertanda hijau
5. Dolibarr : bertanda ungu

Berikut gambar 6.1 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori pertama



Gambar 6.1 Diagram 1 Perbandingan FOS ERP menurut Google Trends

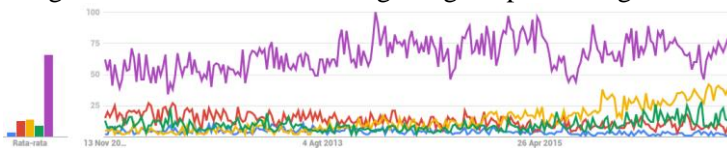
Dari hasil perbandingan tersebut, diperoleh hasil rata-rata pencarian yang disajikan dalam diagram batang dengan jumlah sebagai berikut:

1. Odoo : 35
2. Openbravo : 14
3. ADempiere : 10
4. ERP5 : 1
5. Dolibarr : 9

Selanjutnya 2 FOS ERP dengan rata-rata terendah dibandingkan dibandingkan dengan daftar FOS ERP berikutnya. Jadi, kategori FOS ERP kedua yang dibandingkan adalah:

1. ERP5 : bertanda hijau
2. Dolibarr : bertanda ungu
3. xTuple Postbooks : bertanda biru
4. WebERP : bertanda merah
5. ERPNext : bertanda kuning

Berikut gambar 6.2 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori kedua



Gambar 6.2 Diagram 2 Perbandingan FOS ERP menurut Google Trends

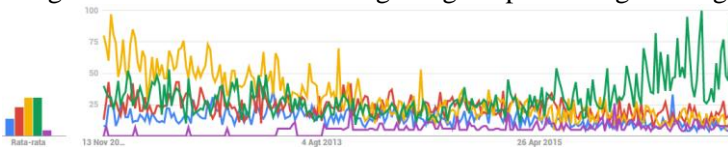
Dari hasil perbandingan tersebut, diperoleh hasil rata-rata pencarian yang disajikan dalam diagram batang dengan jumlah sebagai berikut:

1. ERP5 : 9
2. Dolibarr : 66
3. xTuple Postbooks : 4
4. WebERP : 13
5. ERPNext : 14

Lalu 2 FOS ERP dengan rata-rata terendah dibandingkan dibandingkan dengan daftar FOS ERP berikutnya. Jadi, kategori FOS ERP ketiga yang dibandingkan adalah:

1. ERP5 : bertanda hijau
2. xTuple Postbooks : bertanda biru
3. Apache OFBiz : bertanda merah
4. Opentaps : bertanda kuning

Berikut gambar 6.3 memaparkan hasil perbandingan pada Google Trends dalam bentuk diagram garis pada kategori ketiga



Gambar 6.3 Diagram 3 Perbandingan FOS ERP menurut Google Trends

Dari hasil perbandingan tersebut, diperoleh hasil rata-rata pencarian yang disajikan dalam diagram batang dengan jumlah sebagai berikut:

1. ERP5 : 31
2. xTuple Postbooks : 14
3. Apache OFBiz : 23
4. Opentaps : 31

6.3.2 Pembahasan Perbandingan Komunitas

1. Perbandingan Komunitas Pengguna Aktif

Berdasarkan data hasil perbandingan FOS ERP pada aspek komunitas, maka keaktifan komunitas dapat disimpulkan masing-masing berdasarkan pengguna aktif dan pasifnya.

Pada perbandingan keaktifan komunitas untuk pengguna aktif dan developer, terdapat beberapa temuan berbeda yang diperoleh. Terdapat 2 FOS ERP yang tidak dapat diidentifikasi jumlahnya, yakni Opentaps dan xTuple Postbooks. Hal ini dikarenakan kedua FOS ERP tersebut hanya menampilkan daftar pos dan topik (*knowledge shared*) tanpa adanya total *user/ member* yang ada di dalam forum tersebut. Terdapat pula FOS ERP yang jumlah *user* dan *knowledge shared* yang sangat sedikit, yakni ERP5. Hal ini dikarenakan portal komunitas lamanya pada *mailing list* telah mati dan tidak digunakan lagi. Kemudian ERP5 mengembangkan forum barunya sebagai portal komunitas pada tahun 2016. Oleh sebab itu, ERP5 hanya memiliki 3 *user* aktif dan 2 pos di forum tersebut.

Berikut pada tabel 6.13 menjelaskan urutan peringkat perbandingan komunitas FOS ERP pada pengguna aktif untuk masing-masing indikator pengukuran.

Tabel 6.13 Peringkat Perbandingan Komunitas Pengguna Aktif

Peringkat	Jumlah <i>user</i> aktif	Jumlah <i>knowledge shared</i>	Tanggal pos terakhir	Tanggal rilis terakhir	Rata-rata waktu rilis
1	WebERP	Openbravo	Odoo	ERPNext	ERPNext
2	Openbravo	Apache OFBiz	Adempiere	Openbravo	Openbravo
3	Dolibarr	Odoo	ERPNext	WebERP	xTuple Postbooks
4	Odoo	xTuple Postbooks	WebERP	Apache OFBiz	WebERP
5	ERPNext	Adempiere	Apache OFBiz	Adempiere	Dolibarr
6	Apache OFBiz	ERPNext	Dolibarr	Odoo	Apache OFBiz
7	Adempiere	Dolibarr	Openbravo	xTuple Postbooks	Odoo
8	ERP5	Opentaps	ERP5	Dolibarr	Adempiere

Peringkat	Jumlah <i>user</i> aktif	Jumlah <i>knowledge shared</i>	Tanggal pos terakhir	Tanggal rilis terakhir	Rata-rata waktu rilis
9	Opentaps	WebERP	xTuple Postbooks	ERP5	ERP5
10	xTuple Postbooks	ERP5	Opentaps	Opentaps	Opentaps

Pada tabel 6.13, indikator yang dapat membandingkan secara kontras adalah pada jumlah *user* aktif, jumlah *knowledge shared*, tanggal rilis terakhir, dan rata-rata waktu rilis. Pada tanggal pos terakhir, terdapat 5 FOS ERP dengan tanggal pos terakhir yang sama dengan tanggal pencarian pada saat penelitian dilakukan, yakni 12 Januari 2017. Sehingga 5 peringkat teratas pada indikator tersebut tidak membedakan antar FOS ERP secara kontras.

Berdasarkan analisis pada kelima indikator pengukuran tersebut, maka FOS ERP yang sesuai untuk diterapkan perusahaan berdasarkan keaktifan komunitas penggunaannya adalah Openbravo. Hal ini dikarenakan Openbravo merupakan FOS ERP yang paling banyak menempati peringkat teratas dari keseluruhan indikator yang diukur.

2. Perbandingan Komunitas Pengguna Pasif

Data hasil perbandingan FOS ERP terkait pengguna pasif yang telah diperoleh akan diperingkatkan juga. Hasil dari jumlah pencarian FOS ERP pada Google Search Engine ini diperingkatkan dari hasil pencarian terbanyak hingga terendah. Sedangkan hasil rata-rata pencarian FOS ERP pada Google Trends juga diperingkatkan, namun tidak hanya pada nilai rata-ratanya. Pemerinkatan dilakukan dengan turut mempertimbangkan tiap kategori/ kelompok FOS ERP yang dibandingkan (Tiap kategori perbandingan terdiri atas 5 FOS ERP). Berikut tabel 6.14 adalah hasil peringkat FOS ERP berdasarkan perbandingan komunitas untuk pengguna pasif.

Tabel 6.14 Peringkat Perbandingan Komunitas Pengguna Pasif

Peringkat	Google Search Engine	Google Trends
1	Odoo	Odoo
2	Openbravo	Openbravo
3	ADempiere	ADempiere
4	ERP5	Dolibarr
5	Dolibarr	ERPNext
6	xTuple Postbooks	WebERP
7	WebERP	ERP5
8	ERPNext	Opentaps
9	Apache OFBiz	Apache OFBiz
10	Opentaps	xTuple Postbooks

Berdasarkan analisis pada kedua cara pengukuran tersebut, maka FOS ERP yang sesuai untuk diterapkan perusahaan berdasarkan komunitas pada pengguna pasifnya adalah Odoo. Hal ini dikarenakan Odoo merupakan FOS ERP yang memiliki jumlah pencarian terbanyak pada Google Search Engine dan hasil rata-rata tertinggi pada Google Trends.

BAB VII

KESIMPULAN DAN SARAN

Pada bab terakhir ini dijelaskan hasil kesimpulan penelitian dan saran untuk penelitian selanjutnya. Kesimpulan penelitian ini adalah berupa rekomendasi FOS ERP yang sesuai untuk diterapkan perusahaan. Saran penelitian adalah mengenai saran pengembangan penelitian berikutnya seputar perbandingan FOS ERP.

7.1 Kesimpulan

Kesimpulan dalam penelitian ini ialah berupa rekomendasi FOS ERP yang sesuai untuk diterapkan perusahaan, yakni antara lain:

1. Alternatif FOS ERP yang populer dan layak untuk digunakan ialah Odoo, Openbravo, ADempiere, Dolibarr, ERPNext, WebERP, ERP5, Opentaps, Apache OFBiz, dan xTuple Postbooks
2. Pada aspek arsitektur, pemilihan FOS ERP harus mempertimbangkan dari kebutuhan perusahaan. Perusahaan kecil dan menengah lebih disarankan menggunakan FOS ERP dengan arsitektur 3 *tier*. Rekomendasi FOS ERP yang paling sesuai adalah Openbravo dan ADempiere. Sedangkan, perusahaan besar yang memiliki kebutuhan lebih kompleks disarankan menggunakan FOS ERP dengan arsitektur 4 *tier*. Rekomendasi FOS ERP yang paling sesuai adalah Odoo.
3. Pada aspek fungsionalitas, terdapat 3 modul yang diteliti. Pada modul *sales* dan *purchase*, FOS ERP yang memiliki kelengkapan fitur utama dan tambahan adalah Openbravo. Pada modul *accounting*, FOS ERP yang mempunyai kelengkapan dan tambahan adalah Opentaps
4. Pada aspek komunitas, FOS ERP dengan keaktifan komunitas paling baik dari sisi pengguna aktif dan developer adalah Openbravo. Sedangkan FOS ERP yang paling populer bagi pengguna pasif dan awam adalah Odoo.

7.2 Saran

Saran yang dapat disampaikan untuk perbaikan atau penelitian selanjutnya di masa mendatang, antara lain adalah:

1. Perbandingan FOS ERP perlu meninjau pada studi kasus suatu perusahaan atau UMKM tertentu untuk meneliti kebutuhan terhadap FOS ERP dalam dunia nyata
2. Penelitian FOS ERP pada aspek penggunaannya, seperti teknis instalasi dan konfigurasi, tampilan UI, maupun keseluruhan modul yang dapat dicakup.

DAFTAR PUSTAKA

- [1] L. F. Motiwalla and J. Thompson, *Enterprise Systems for Management (Second Edition)*, New Jersey: Pearson, 2012.
- [2] S. Wibisono, "Enterprise Resource Planning (ERP) Solusi Sistem Informasi Terintegrasi," *Jurnal Teknologi Informasi Dinamik*, vol. 10(3), pp. 150-159, 2005.
- [3] M. Griffiths, A. Heinze and A. Ofoegbu, "The Real SAP® Business One Cost: A Case Study of ERP Adoption in an SME. International Journal of Management Practice," *International Journal of Management Practice*, vol. 6(2), pp. 199-215, 2013.
- [4] R. A. d. Carvalho, "Issues on Evaluating Free/Open Source ERP Systems," *Research and practical issues of enterprise information systems*, pp. 667-675, 2006.
- [5] T. Herzog, *A Comparison of Open Source ERP Systems*, Vienna: Institute of Information Systems and Operations, Department of Business Management and Information Systems, Vienna University of Economics and Business, 2006.
- [6] V. G. Fougatsaro, *A Study of Open Source ERP Systems*, Swedia: School of Management, Blekinge Institute of Technology, 2009.
- [7] F. G. d. S. Gripe and I. A. Rodello, "A Theoretical Analysis of Key Points when Choosing Open Source ERP Systems," *JISTEM-Journal of Information Systems and Technology Management*, vol. 8(2), pp. 441-458, 2011.
- [8] Y. Delsart and C. V. Nieuwenhuysen, *OpenERP Evaluation with SAP as Reference*, Feridis SA, 2011.
- [9] N. Jindal and K. S. Dhindsa, "Comparative Study of OpenERP and Its Technologies," *International Journal of Computer Applications*, vol. 73(20), pp. 42-47, 2013.
- [10] O. G. Standard, *TOGAF® Version 9.1*, vol. 1, United State of America: The Open Group, 2011, pp. 18-31.

- [11] P. W. Handayani, A. N. Hidayanto and I. Budi, "Business Process Requirements for Indonesian Small Medium Enterprises (SMEs) in Implementing Enterprise Resource Planning (ERP) and ERP Systems Comparison," *Journal of Computers*, vol. 8(9), pp. 2437-2441, 2013.
- [12] S. M. Al-Saleem, "A Comparative Analysis and Evaluation of Open Source ERP Systems," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 13(4), pp. 24-28, 2013.
- [13] J. Lee, K. Siau and S. Hong, "Enterprise Integration with ERP and EAI," *Communications of the ACM*, vol. 46(2), pp. 54-60, 2003.
- [14] J. R. Muscatello, M. H. Small and I. J. Chen, "Implementing Enterprise Resource Planning (ERP) Systems in Small and Midsize Manufacturing Firms," *International Journal of Operations & Production Management*, vol. 23(8), pp. 850-871, 2003.
- [15] F. Calisir and F. Calisir, "The Relation of Interface Usability Characteristics, Perceived Usefulness, and Perceived Ease of Use to End-User Satisfaction with Enterprise Resource Planning (ERP) Systems," *Computers in Human Behavior*, vol. 20, no. 4, pp. 505-515, 2004.
- [16] M. H. Fahmi, "Pelokalan dan Kustomisasi Aplikasi ERP Open Source Openbravo ERP Untuk Implementasi pada UKM Furniture," *DINAMIKA DOTCOM, Jurnal Pengembangan Manajemen Informatika & Komputer*, vol. 1(2), pp. 156-175, 2010.
- [17] I. A. Dewi, I. Darmawan and R. W. Witjaksono³, "Penerapan Sistem Manufacturing pada UMKM Konveksi Rajutan Berbasis OpenERP dengan Metode Sure Step Implementation of Manufacturing System at Small and Medium-Sized Enterprises Knitting Factory Based on OpenERP using Sure Step Method," *Tel-U Collection*, 2015.

- [18] Odoo, "MVC Architecture," 18 07 2014. [Online]. Available: https://doc.odoo.com/v6.1/developer/02_architecture.html.
- [19] D. Winer, "XML-RPC Specification," 15 06 1999. [Online]. Available: <http://xmlrpc.scripting.com/>.
- [20] OpenERP, "OpenERP as a multitenant three-tiers architecture," 12 11 2012. [Online]. Available: http://odoo-docs.readthedocs.io/en/latest/02_architecture.html.
- [21] Openbravo, "Openbravo 3 Architecture," 07 11 2016. [Online]. Available: http://wiki.openbravo.com/wiki/Openbravo_3_Architecture#Caching_and_refreshing_of_static_js_files.
- [22] ADempiere, "Adempiere Architecture 3 tier," 25 09 2008. [Online]. Available: http://www.adempiere.com/Adempiere_Architecture_3_tier.
- [23] Oracle, "Package java.sql," 08 02 2012. [Online]. Available: <https://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html>.
- [24] Dolibarr, "Prerequisite," 26 03 2015. [Online]. Available: <https://wiki.dolibarr.org/index.php/Prerequisite>.
- [25] Dolibarr, "Installation - Upgrade," 25 10 2014. [Online]. Available: https://wiki.dolibarr.org/index.php/Installation_-_Upgrade.
- [26] P. Vyas, "ERPNext High Availability," 01 09 2014. [Online]. Available: <https://frappe.io/blog/technology/erpnext-ha>.
- [27] VKInfotek, "Develop Web ERP Software Using VS 2015," 04 05 2010. [Online]. Available: <http://www.vkinfotek.com/develop-erp-asp-software.html>.
- [28] D. Camata, "ERP5, Wendelin and Jupyter," 13 06 2016. [Online]. Available: <https://www.erp5.com/latest/P-WIA-Presentation.PyData.2016>.

- [29] Opentaps, "Opentaps Technical Reference," 08 11 2011. [Online]. Available: http://www.opentaps.org/docs/index.php/Opentaps_Technical_Reference#opentaps_Architecture.
- [30] Opentaps, "POJO Service Engine," 21 08 2008. [Online]. Available: http://www.opentaps.org/docs/index.php/POJO_Service_Engine.
- [31] D. E. Jones, "Apache OFBiz Project Overview," 01 03 2003. [Online]. Available: <https://ofbiz.apache.org/apache-ofbiz-project-overview.html>.
- [32] FreeMarker, "What is Apache FreeMarker?," 05 07 2015. [Online]. Available: <http://freemarker.org/>.
- [33] J. Rogelstad, "JavaScript, JavaScript, JS all the way," 28 11 2012. [Online]. Available: <https://xtuple.com/blog/jrogelstad/JavaScript-JavaScript-js-all-way>.
- [34] S. Cass, "The 2016 Top Programming Languages," 26 07 2016. [Online]. Available: <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>.
- [35] DB-Engines, "DB-Engines Ranking," 12 2016. [Online]. Available: <http://db-engines.com/en/ranking>.

BIODATA PENULIS



Penulis lahir di Jember pada tanggal 23 Juni 1995. Penulis merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal di sekolah negeri mulai dari SDN Jember Lor 1 hingga lulus pada tahun 2007, SMPN 2 Jember hingga lulus pada tahun 2010, dan SMAN 1 Jember hingga lulus pada tahun 2013. Setelah lulus, penulis melanjutkan ke jenjang perguruan tinggi negeri di Surabaya, yakni Jurusan Sistem Informasi Institut

Teknologi Sepuluh Nopember Surabaya. Sebagai mahasiswa penulis aktif dalam urusan akademik, non akademik maupun organisasi. Tercatat penulis pernah menjadi staff pada Departemen Information Media di Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf) ITS Surabaya, serta staff Syiar (Media Informasi) dan Kepala Departemen Relasi Ukhuwah (Hubungan Internal dan Eksternal) pada Kajian Islam Sistem Informasi (KISI) ITS. Selain organisasi formal, penulis juga pernah mengikuti organisasi non-formal, yakni Unit Kegiatan Mahasiswa pada ITS Badminton Community (IBC). Selain organisasi, penulis juga aktif dalam kepanitiaan, baik panitia dalam organisasi yang diikutinya, maupun di luar organisasi. Penulis juga pernah menjalani kerja praktik di PT. Telekomunikasi Indonesia di Jakarta selama 1,5 bulan pada tahun 2016.

Untuk mendapatkan gelar Sarjana Komputer (S.Kom), penulis mengambil laboratorium bidang minat Sistem Enterprise (SE) dengan topik tugas akhir pada Enterprise Resource System (ERP). Untuk kepentingan penelitian penulis juga dapat dihubungi melalui e-mail: tesarakram@gmail.com.