



TESIS - IF185401

**PENGUKURAN KESERUPAAN SEMANTIK
MENGUNAKAN TACSIM DAN STRUKTURAL
MENGUNAKAN *GRAPH EDIT DISTANCE-GREEDY*
PADA DIAGRAM KASUS PENGGUNAAN**

**FATIMATUS ZULFA
NRP. 05111950010030**

**DOSEN PEMBIMBING
Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.**

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021**



TESIS - IF185401

**PENGUKURAN KESERUPAAN SEMANTIK
MENGUNAKAN TACSIM DAN STRUKTURAL
MENGUNAKAN *GRAPH EDIT DISTANCE-GREEDY*
PADA DIAGRAM KASUS PENGGUNAAN**

**FATIMATUS ZULFA
NRP. 05111950010030**

**DOSEN PEMBIMBING
Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.**

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

FATIMATUS ZULFA

NRP. 05111950010030

Dengan judul:

PENGUKURAN KESERUPAAN SEMANTIK MENGGUNAKAN TACSIM
DAN STRUKTURAL MENGGUNAKAN *GRAPH EDIT DISTANCE-GREEDY*
PADA DIAGRAM KASUS PENGGUNAAN

Tanggal Ujian : 04 Februari 2021

Periode Wisuda : 2021 Gasal

Disetujui oleh:

Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.
NIP. 19741123 200604 1 001

(Pembimbing 1)

Dr. Ahmad Saikhu, S.Si., MT
NIP. 19710718 200604 1 001

(Penguji 1)

Ir. Siti Rochimah, MT., Ph.D.
NIP. 19681002 199403 2 001

(Penguji 2)

Ary M. Shiddiqi, S.Kom., M.Comp.Sc., Ph.D.
NIP. 19810620 200501 2 001

(Penguji 3)

Kepala Departemen Teknik Informatika



Dr. Eng. Chastine Fatichah, S. Kom., M. Kom.
NIP. 19751220 200112 2 002

[Halaman ini sengaja dikosongkan]

**PENGUKURAN KESERUPAAN SEMANTIK
MENGUNAKAN TACSIM DAN STRUKTURAL
MENGUNAKAN *GRAPH EDIT DISTANCE-GREEDY* PADA
DIAGRAM KASUS PENGGUNAAN**

Nama : Fatimatus Zulfa
NRP : 05111950010030
Pembimbing : Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.

ABSTRAK

Unified Modeling Language (UML) merupakan sekumpulan diagram sebagai perancangan perangkat lunak. Terdapat beberapa jenis diagram UML, salah satunya diagram kasus penggunaan. Perancangan perangkat lunak sangat dibutuhkan dalam membangun sebuah sistem. Dalam dunia pendidikan khususnya mata pelajaran desain, seperti Rekayasa Perangkat Lunak, UML merupakan salah satu topik penting. Saat ini, dalam pembelajaran banyak dilakukan dengan sistem digital seperti e-learning. E-learning menyediakan banyak fitur pembelajaran, salah satunya fitur penilaian otomatis dalam bentuk soal dengan jawaban pilihan ganda, daftar pilihan, isian, dan uraian. Pada mata pelajaran kuliah Rekayasa Perangkat Lunak terdapat jawaban salah satunya berupa diagram kasus penggunaan, namun saat ini penilaian otomatis terhadap jawaban berbentuk diagram tersebut belum tersedia. Untuk itu dibutuhkan metode untuk penilaian otomatis terhadap bentuk jawaban tersebut.

Penelitian ini mengajukan metode untuk menghitung keserupaan antar dua buah diagram kasus penggunaan yang memodelkan sistem yang sama. Pengukuran dilakukan dengan membandingkan antara jawaban dan kunci jawaban yang memperhatikan dua aspek pengukuran, yaitu semantik dan struktural. Pengukuran semantik dilakukan dengan membandingkan informasi leksikal pada dua buah diagram kasus penggunaan dalam bentuk graf dengan menggunakan metode *Topology-Attributed Coupling Similarity* (TACSim), sedangkan pengukuran struktural dilakukan dengan memodelkan diagram kasus penggunaan menjadi graf dan dilakukan perhitungan keserupaan dengan menggunakan penggabungan *Graph Edit Distance* dan algoritma *Greedy*.

Hasil pengujian menunjukkan bahwa metode yang diusulkan untuk menghitung keserupaan antara dua buah kasus penggunaan dapat diandalkan seperti pengampu mata pelajaran Rekayasa Perangkat Lunak dalam melakukan penilaian jawaban siswa terhadap diagram kasus penggunaan. Nilai kesepakatan antara pengampu mata pelajaran dan metode yang diusulkan mencapai 0,86. Hal tersebut menunjukkan kesepakatan yang hampir sempurna (*almost perfect agreement*).

Kata kunci: UML, Diagram Kasus Penggunaan, *Topology-Attributed Coupling Similarity* (TACSim), *Graph Edit Distance* (GED), Algoritma *Greedy*.

[Halaman ini sengaja dikosongkan]

MEASUREMENT OF SEMANTIC SIMILARITY USING TACSIM AND STRUCTURAL USING GRAPH EDIT DISTANCE-GREEDY IN USE CASE DIAGRAM

Student's Name : Fatimatus Zulfa

Student ID : 05111950010030

Supervisor : Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng.

ABSTRACT

Unified Modeling Language (UML) is a collection of diagrams as software design. There are several types of UML diagrams, one of which is a use case diagram. Software design is needed in building a system. In the world of education especially design subjects, such as Software Engineering, UML is one of the important topics. Currently, learning is mostly done with digital systems such as e-learning. E-learning provides many learning features, one of which is an automatic assessment feature in the form of questions with multiple choice answers, list of choices, fields, and descriptions. In the subject of Software Engineering, there are answers to one of them in the form of use case diagrams, but at this time automatic assessment of answers in the form of diagrams is not yet available. This requires a method for automatic assessment of the form of the answer.

This study proposes a method for calculating the similarity between two use case diagrams that model the same system. Measurements are made by comparing answers and key answers that pay attention to two aspects of measurement, namely semantics and structural. Semantic measurements are performed by comparing lexical information on two graphs of use case diagrams using the Topology-Attributed Coupling Similarity (TACSim) method, while structural measurements are carried out by modeling a use case diagram into a graph and simulating similarity using a Graph Edit Distance and Greedy algorithm.

The results show that the proposed method for calculating the similarity between two use cases is as reliable as the software engineering subject in assessing students' answers to use case diagrams. The agreement value between subject teachers and the proposed method reaches 0.86. This shows an almost perfect agreement.

Keywords: *UML, Use Case Diagram, Topology-Attributed Coupling Similarity (TACSim), Graph Edit Distance (GED), Greedy Algorithm.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, atas limpahan rizki, berkah, nikmat, dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan penelitian dengan judul “Pengukuran Keserupaan Semantik Menggunakan TACSim dan Struktural Menggunakan *Graph Edit Distance-Greedy* pada Diagram Kasus Penggunaan”.

Penulis menyadari bahwa tanpa bantuan dari berbagai pihak, penelitian ini tidak akan terselesaikan seperti sekarang ini. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT yang memberikan nikmat iman, islam, kesehatan, waktu, petunjuk, serta berbagai kemudahan dari arah yang tidak disangka-sangka, sehingga penulis dapat menyelesaikan penelitian ini dengan baik.
2. Bapak Daniel Oranova Siahaan, S.Kom., M.Sc., PD.Eng. selaku pembimbing yang telah meluangkan waktu untuk mendidik dan membimbing penulis dalam menyelesaikan penelitian ini.
3. Bapak Reza Fauzan dan Ibu Evi Triandini yang telah meluangkan waktu untuk membimbing penulis dalam menyelesaikan penelitian ini.
4. Bapak Amenan, Almh. Ibu Sri Rahayu Ningsih, Bapak Bandriyo, Ibu Satiyemi, Heru Sofyantoro, Andik Ali Mustofa, Anik Yulianti, Rifky Hamzah Khoirudin dan Abid Maulana selaku keluarga yang selalu memberikan dukungan berupa materi, doa, dan semangat terhadap penulis.
5. Vivi, Septian, Fia, dan rekan-rekan mahasiswa S2 Teknik Informatika, ITS Surabaya.

Semoga Allah SWT senantiasa menyayangi, menguatkan, dan menunjukkan jalan yang terbaik atas semua kebaikan yang telah diberikan. Tidak dapat dipungkiri bahwa laporan penelitian ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan saran dan kritik sebagai bahan perbaikan di masa yang akan datang. Semoga laporan penelitian ini dapat bermanfaat bagi penulis dan pembaca.

Surabaya, Februari 2021



Fatimatus Zulfa

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

SAMPUL	i
COVER.....	iii
LEMBAR PENGESAHAN	v
ABSTRAK.....	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR	xix
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	4
1.3. Tujuan	4
1.4. Manfaat	4
1.5. Kontribusi Penelitian	4
1.6. Batasan Masalah	5
BAB 2 KAJIAN PUSTAKA	7
2.1. Unified Modelling Language.....	7
2.2. Diagram Kasus Penggunaan	8
2.3. StarUML	9
2.4. Kekerupaan Semantik.....	10
2.5. Pemrosesan Bahasa Alami.....	10
2.6. TACSim	12
2.7. Kekerupaan Struktural.....	13
2.8. Graf	14

2.9.	Hungarian.....	14
2.10.	<i>Graph Edit Distance</i> (GED)	15
2.11.	Gwet's AC1	18
BAB 3 METODOLOGI PENELITIAN		21
3.1.	Studi Literatur	21
3.2.	Pengumpulan Diagram Kasus Penggunaan	22
3.3.	Analisis Data Para Pakar.....	24
3.4.	Metode Pengukuran Keserupaan Diagram Kasus Penggunaan	25
3.4.1.	Keserupaan Semantik Diagram Kasus Penggunaan	26
3.4.2.	Keserupaan Struktural Diagram Kasus Penggunaan	39
3.5.	Penilaian antara Jawaban Diagram dan Kunci Jawaban Diagram.....	49
3.6.	Pengujian Metode	49
3.7.	Laporan Penelitian	50
3.8.	Jadwal Penelitian	50
BAB 4 HASIL DAN PEMBAHASAN		51
4.1.	Implementasi Sistem.....	51
4.1.1.	Akumulasi Data Para Pakar	51
4.1.2.	Deskripsi Diagram	52
4.1.3.	Keserupaan Semantik.....	52
4.1.4.	Keserupaan Struktural.....	53
4.2.	Hasil Pengujian dan Analisis	55
4.2.1.	Hasil Pengujian	55
4.2.2.	Hasil Kesepakatan Pakar.....	56
4.2.3.	Analisis Hasil	58
BAB 5 KESIMPULAN DAN SARAN		61
5.1.	Kesimpulan	61

5.2. Saran	62
DAFTAR PUSTAKA	65
LAMPIRAN.....	69
BIOGRAFI PENULIS	77

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Komponen diagram kasus penggunaan	8
Tabel 2.2 Pelabelan POS Tagging Stanford NLP	10
Tabel 2.3 Tabel Penulisan Hasil Pengamatan.....	18
Tabel 2.4 Interpretasi Nilai Gwet's AC1 (Landis and Koch 2012).....	19
Tabel 2.5 Data Pengamatan dari Contoh Kasus.....	20
Tabel 3.1 daftar informasi leksikal pada simpul graf1	31
Tabel 3.2 daftar informasi leksikal pada simpul graf2	31
Tabel 3.3 Informasi leksikal pada tepian graf1	32
Tabel 3.4 Informasi leksikal pada tepian graf2.....	32
Tabel 3.5 Perhitungan tepian	37
Tabel 3.6 Matriks Keserupaan Simpul.....	38
Tabel 3.7 Matriks Keserupaan Tepian	38
Tabel 3.8 Nilai Keserupaan Terpilih pada Matriks Keserupaan Simpul	38
Tabel 3.9 Nilai Keserupaan Terpilih pada Matriks Keserupaan Tepian.....	39
Tabel 3.10 Notasi Pemodelan Komponen Diagram Menjadi Graf.....	40
Tabel 3.11 Rencana Jadwal Kegiatan Penelitian	50

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Contoh Diagram Kasus Penggunaan dari sistem ISL	9
Gambar 2.2 Contoh Graf berarah.....	14
Gambar 2.3 Graf1	17
Gambar 2.4 Graf2	17
Gambar 2.5 Graf1 ¹	17
Gambar 3.1 Diagram Alur Metodologi Penelitian.....	21
Gambar 3.2 Contoh diagram kasus penggunaan.....	23
Gambar 3.3 Contoh hasil eksport XMI.....	24
Gambar 3.4 Metadata Diagram Kasus Penggunaan	26
Gambar 3.5 Contoh diagram kasus penggunaan pertama.....	27
Gambar 3.6 Spesifikasi kasus penggunaan utama <i>Issue Book</i>	28
Gambar 3.7 Spesifikasi kasus penggunaan utama <i>Return Book</i>	28
Gambar 3.8 Contoh diagram kasus penggunaan kedua	29
Gambar 3.9 Spesifikasi kasus penggunaan utama <i>See List Available Books</i>	29
Gambar 3.10 Spesifikasi kasus penggunaan utama <i>See All Books Data</i>	30
Gambar 3.11 Spesifikasi kasus penggunaan utama <i>seeListBorroweBooks</i>	30
Gambar 3.12 Perhitungan Keserupaan antara Dua Komponen Diagram	33
Gambar 3.13 Representasi dari Gambar 3.4 menjadi graf.....	41
Gambar 3.14 Representasi dari Gambar 3.7 menjadi graf.....	41
Gambar 3.15 Representasi dari Gambar 3.5 menjadi graf.....	46
Gambar 3.16 Representasi dari Gambar 3.6 menjadi graf	46
Gambar 3.17 Representasi Gambar 3.8 menjadi graf	46
Gambar 3.18 Representasi Gambar 3.9 menjadi graf	46
Gambar 3.19 Representasi Gambar 3.10 menjadi graf.....	46
Gambar 3.20 Pengujian Metode Keserupaan Diagram Kasus Penggunaan	49
Gambar 4.1 Kesepakatan pakar menggunakan Gwet's AC1	57

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Bab ini membahas tentang poin utama dalam pembuatan proposal penelitian yang meliputi latar belakang, perumusan masalah, tujuan, manfaat, kontribusi penelitian, dan batasan masalah.

1.1. Latar Belakang

Unified Modeling Language (UML) merupakan bahasa standart yang digunakan untuk membuat rancangan sebuah perangkat lunak dengan membangun dan menggambarkan dokumen artifak sebuah perangkat lunak (Booch et al. 2008). UML memungkinkan perancang untuk melakukan pemodelan terhadap interaksi antara sistem dan pengguna, interaksi antar objek, perilaku objek, dan struktur sistem (Chechik, Nejati, and Sabetzadeh 2012).

UML dibagi menjadi dua kategori yaitu diagram struktur dan diagram perilaku (Fowler 2004). Diagram UML di klasifikasikan dalam dua kategori. Diagram struktur terdiri dari enam diagram yaitu diagram kelas (*class diagram*), diagram struktur komposit (*composite structure diagram*), diagram objek (*object diagram*), diagram komponen (*componen diagram*), diagram peluncuran (*deployment diagram*), dan diagram paket (*package diagram*). Sedangkan diagram perilaku terdiri dari diagram aktivitas (*activity diagram*), diagram kasus penggunaan (*use case diagram*), diagram keberadaan (*statechart diagram*), dan diagram interaksi (*interaction diagram*). Diagram interaksi terdiri dari empat jenis yaitu diagram urutan (*sequence diagram*), diagram komunikasi (*commuication diagram*), diagram gambaran umum interaksi (*interaction overview diagram*), dan diagram waktu (*timing diagram*).

Diagram UML sangat dibutuhkan dalam perancangan perangkat lunak, maka dari itu UML menjadi topik pembelajaran yang penting dalam dunia pendidikan khususnya mata pelajaran desain, seperti Rekayasa Perangkat Lunak. Saat ini, sistem pembelajaran banyak dilakukan dengan sistem digital seperti e-learning. E-learning menyediakan banyak fitur pembelajaran, salah satunya fitur penilaian otomatis dalam bentuk soal dengan jawaban pilihan ganda, daftar pilihan, isian, dan

uraian. Metode yang dapat digunakan sebagai penilaian otomatis terhadap jawaban esai yang diterapkan pada e-learning yaitu menggunakan metode *Latent Semantic Analysis* atau LSA (Pramukantoro and Fauzi 2016). Pada penelitian lain menggunakan metode N-gram sebagai ekstraksi fitur dan *cosine similarity* sebagai metode dalam melakukan koreksi otomatis terhadap jawaban esai pada e-learning (Fauzi et al. 2017). Pada mata pelajaran Rekayasa Perangkat Lunak, selain terdapat bentuk jawaban seperti tersebut, terdapat bentuk jawaban dalam bentuk diagram UML. Untuk itu dibutuhkan juga metode penilaian otomatis terhadap bentuk jawaban diagram UML. Dalam perancangan perangkat lunak, diagram UML yang penting untuk dibuat salah satunya yaitu diagram kasus penggunaan. Diagram kasus penggunaan digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah perangkat lunak dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Sebagai contoh ketika akan membuat diagram kelas, kita dapat memanfaatkan kasus penggunaan yang telah dibuat sebelumnya menjadi sebuah kelas. Karena pentingnya diagram tersebut, penelitian ini berfokus pada diagram kasus penggunaan untuk dihitung keserupaannya.

Penelitian sebelumnya telah terdapat beberapa metode untuk menghitung keserupaan diagram diagram kelas akan tetapi fokus utama dari metode-metode tersebut adalah agar dapat menggunakan kembali sebuah sistem yang sudah ada untuk diterapkan pada sistem yang baru dengan menggunakan aspek struktural pada diagram (Adamu and Zainon 2017; Yuan, Yan, and Ma 2019). Pada penelitian lain untuk membandingkan keserupaan antar dua buah diagram kasus penggunaan menggunakan aspek pengukuran secara semantik dan struktural yang mengabaikan tipe relasi antar kasus penggunaan, antar aktor, dan antara kasus penggunaan dan aktor (Fauzan et al. 2019). Penelitian tersebut dilakukan dengan tujuan agar dapat menggunakan kembali perancangan perangkat lunak. Selain melakukan pengukuran keserupaan terhadap kasus penggunaan, juga dilakukan pengukuran keserupaan diagram kelas (Fauzan et al. 2018) dan keserupaan diagram urutan (Triandini et al. 2019) untuk penggunaan kembali diagram perancangan perangkat lunak. Sebelumnya telah dilakukan penelitian untuk menghitung keserupaan diagram kasus penggunaan, namun hanya berfokus pada keserupaan struktural diagram (Zulfa et al. 2020). Pengukuran keserupaan diagram UML dengan tujuan

penggunaan kembali perancangan perangkat lunak dilakukan dengan melakukan perbandingan antara satu diagram UML dengan diagram UML lainnya yang terdapat pada repositori dengan berbagai domain baik yang sama, maupun berbeda. Sedangkan pengukuran keserupaan diagram UML dengan tujuan penilaian dilakukan dengan membandingkan jawaban diagram UML dengan kunci jawaban yang berarti perbandingan dilakukan dalam domain yang sama, selain itu pada pengukuran keserupaan untuk tujuan penilaian terdapat toleransi terhadap hasil pengukuran. Sehingga aspek-aspek antara semantik dan struktural pada pengukuran diagram UML berbeda, tergantung dengan tujuan pengukuran tersebut.

Penelitian ini mengajukan metode untuk menghitung keserupaan antar dua buah diagram kasus penggunaan yang memodelkan sistem yang sama. Sehingga dapat mempermudah pihak pengampu mata pelajaran dalam Memberikan nilai berdasarkan tingkat keserupaan diagram kasus penggunaan oleh mahasiswa dengan kunci jawaban. Penelitian ini merupakan bagian dari penelitian doctoral oleh Reza Fauzan mengenai metode penilaian keserupaan semantik dan struktural rancangan perangkat lunak berdasarkan diagram *Universal Modeling Language* (UML). Pengukuran dilakukan dengan memperhatikan dua aspek pada diagram, yaitu semantik dan struktural. Pengukuran semantik dilakukan dengan membandingkan informasi leksikal pada diagram kasus penggunaan dengan menggunakan metode penggabungan Pemrosesan Bahasa Alami dan *Topology-Attributed Coupling Similarity* (TACSim), sedangkan pengukuran struktural dilakukan dengan memodelkan diagram kasus penggunaan menjadi graf dan dilakukan perhitungan dengan penggabungan metode *Graph Edit Distance* (GED) dan algoritma *Greedy*.

Metode yang diusulkan diharapkan dapat menghasilkan penilaian yang dapat diandalkan seperti pengampu mata pelajaran Rekayasa Perangkat lunak dalam melakukan penilaian jawaban siswa terhadap diagram kasus penggunaan. Hasil pengukuran keserupaan diagram kasus penggunaan, diharapkan dapat membantu dan menjadi bahan pertimbangan pihak pengampu mata pelajaran dalam melakukan penilaian terhadap jawaban diagram kasus penggunaan pada e-learning.

1.2. Perumusan Masalah

Rumusan masalah yang dibahas dalam penelitian ini dapat dipaparkan sebagai berikut:

1. Bagaimana memodelkan diagram kasus penggunaan ke dalam bentuk graf untuk menghitung keserupaan semantik dan struktural?
2. Berapa bobot parameter yang digunakan untuk menentukan keserupaan semantik dan struktural pada dua buah diagram kasus penggunaan?
3. Bagaimana mengukur keserupaan semantik dengan menggunakan pemrosesan bahasa alami dan TACSim?
4. Bagaimana mengukur keserupaan struktural dengan menggabungkan metode *Graph Edit Distance* (GED) dan algoritma *Greedy*?

1.3. Tujuan

Tujuan penelitian yang ingin dicapai adalah membangun metode untuk menghitung keserupaan antara dua buah diagram kasus penggunaan yang dapat diandalkan seperti pengampu mata pelajaran Rekayasa Perangkat Lunak dalam melakukan penilaian jawaban siswa terhadap diagram kasus penggunaan.

1.4. Manfaat

Penelitian ini diharapkan dapat mempermudah pihak pengampu mata pelajaran dalam memberikan nilai jawaban dalam bentuk diagram berdasarkan tingkat keserupaan antara diagram kasus penggunaan oleh siswa dan kunci jawaban.

1.5. Kontribusi Penelitian

Penelitian ini memiliki dua kontribusi. Kontribusi pertama yaitu pemodelan diagram kasus penggunaan dalam bentuk graf. Kontribusi kedua yaitu pengukuran keserupaan diagram kasus penggunaan dengan dua aspek yaitu keserupaan semantik menggunakan metode penggabungan Pemrosesan Bahasa Alami dan *Topology-Attributed Coupling Similarity* (TACSim) serta aspek keserupaan struktural menggunakan kombinasi *Graph Edit Distance* (GED) dan algoritma *Greedy*.

1.6. Batasan Masalah

Untuk memfokuskan permasalahan penelitian ini, batasan masalah yang ditentukan adalah sebagai berikut.

- Penelitian ini menggunakan UML versi 2.4.
- Diagram kasus penggunaan menggunakan Bahasa Inggris.

[Halaman ini sengaja dikosongkan]

BAB 2

KAJIAN PUSTAKA

Bab ini merupakan pembahasan dari referensi terkait yang telah dilakukan dalam menyelesaikan permasalahan sesuai dengan uraian pada latar belakang. Bab ini diawali dengan menjabarkan setiap kajian pustaka yang berkaitan dengan topik penelitian.

2.1. Unified Modelling Language

Unified Modeling Language (UML) merupakan Bahasa standart yang digunakan untuk membuat rancangan sebuah software dengan membangun dan menggambarkan dokumen artifak sebuah software (Booch et al. 2008). UML memiliki tujuan sebagai berikut (Chonoles 2018).

- a. Untuk melakukan pemodelan terhadap sistem menggunakan konsep berorientasi objek.
- b. Untuk membentuk artefak konseptual yang dapat dieksekusi.
- c. Untuk mengatasi masalah sistem yang kompleks.
- d. Untuk membuat Bahasa pemodelan yang dapat digunakan oleh manusia dan mesin.




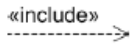


Salah satu tujuan utama dari pembangunan UML adalah membantu analisis desain perangkat lunak. Terdapat tiga jenis UML dalam mendukung untuk mencapai tujuan tersebut, yaitu analisis domain, analisis kasus penggunaan, dan analisis kebutuhan (Chonoles 2018). Tujuan tersebut dimaksudkan untuk menggambarkan pemodelan mengenai yang diinginkan pada sistem. Pemodelan analisis domain dilakukan dengan meninjau model untuk menentukan hubungan antara pemangku kepentingan dan sistem. Diagram yang terdapat pada pemodelan tersebut, yaitu diagram kelas, diagram keberadaan, dan diagram urutan. Sedangkan pemodelan analisis kasus penggunaan yaitu rencana pengguna terhadap penggunaan sistem di masa depan. Diagram yang digunakan pada pemodelan ini yaitu diagram kasus penggunaan. Pada analisis ini merupakan jembatan antara pengguna atau aktor dengan kegunaan sistem. Dan yang terakhir yaitu pemodelan analisis kebutuhan, dilakukan dengan memeriksa kebutuhan oleh pengguna yang

telah terdokumentasikan untuk mengidentifikasi kata benda dan kata kerja serta memetakan objek

2.2. Diagram Kasus Penggunaan

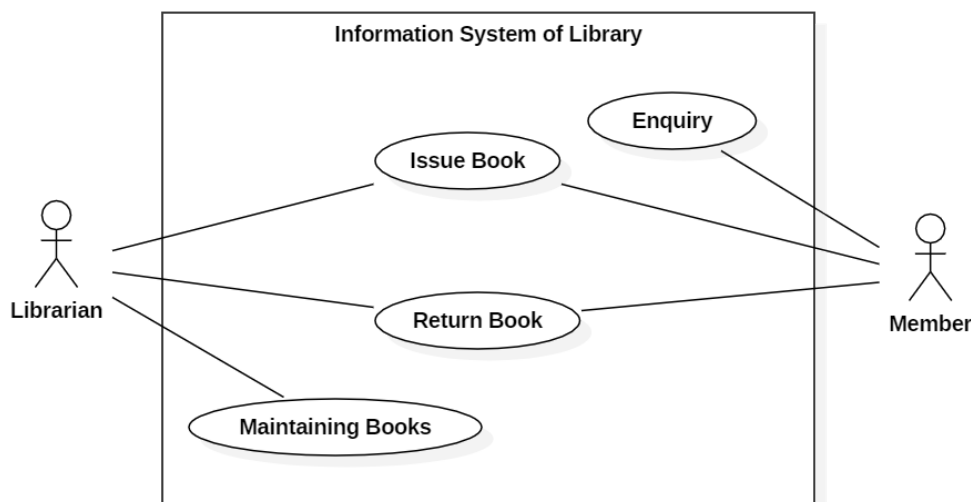
Kasus penggunaan merupakan serangkaian tindakan yang dilakukan oleh sistem, User yang diwakili oleh aktor, atau sistem lain yang berinteraksi dengan sistem yang dimodelkan (Spurrier and Topi 2018). Tabel 2.1 berikut merupakan penjabaran komponen yang terdapat pada diagram kasus penggunaan.

Tabel 2.1 Komponen diagram kasus penggunaan

Komponen	Notasi	Deskripsi
Aktor		Aktor adalah peran yang dimainkan oleh objek di luar sistem yang berinteraksi langsung dengan sistem. Sebuah objek tunggal dapat memainkan beberapa peran. Baik secara aktif maupun pasif, aktor berpartisipasi dalam satu atau lebih kasus penggunaan. Aktor dapat dituliskan dengan menggunakan kata benda.
Kasus penggunaan		Kasus penggunaan merupakan serangkaian tindakan yang dapat dilakukan oleh sistem dan dapat diamati yang biasanya berlaku bagi satu atau lebih aktor atau pemangku kepentingan lainnya dari sistem. Kasus penggunaan dapat dituliskan dengan kata kerja atau kata benda aktif.
Memperpanjang		Sebuah kasus penggunaan dapat digunakan untuk memperpanjang kasus penggunaan lain.
Termasuk		Sebuah kasus pengguna mungkin termasuk dalam serangkaian kasus penggunaan utama lainnya.
Generalisasi		Sebuah kasus pengguna mungkin menjadi generalisasi atau bagian dari kasus pengguna utama lainnya.
Asosiasi		Hubungan antara aktor dengan kasus penggunaan yang menggambarkan apa saja yang dapat dilakukan oleh aktor.

Gambar 2.1 merupakan contoh dari diagram kasus penggunaan. Terdapat dua buah aktor yaitu Librarian dan Member. Aktor *Librarian* dapat melakukan

Issue Book, *Return Book*, dan *Maintaining Book*. Sedangkan aktor *Member* dapat melakukan *Enquiry*, *Issue Book*, dan *Return Book*.



Gambar 2.1 Contoh Diagram Kasus Penggunaan dari sistem ISL

2.3. StarUML

StarUML merupakan alat yang digunakan untuk melakukan perancangan terhadap diagram UML. StarUML kompatibel dengan artifak UML dan mendukung sebelajs jenis diagram UML, yaitu diagram kelas, diagram objek, diagram kasus penggunaan, diagram komponen, diagram peluncuran, diagram struktur komposit, diagram urutan, diagram komunikasi, diagram keberadaan, dan diagram aktivitas (MKLab Co. 2018).

StarUML memiliki beberapa fitur sebagai berikut (MKLab Co. 2018).

- a. StarUML mendukung banyak cara cepat di Ubah Cepat untuk menciptakan elemen dan hubungan sekaligus seperti subkelas, antarmuka pendukung, dll.
- b. Semua diagram, teks, dan ikon sangat tajam dan dapat diekspor ke gambar DPI Tinggi (PNG dan JPEG).
- c. StarUML menyimpan model dalam format JSON (*Javascript Object Notation*) yang sangat sederhana.
- d. StarUML mendukung bahasa pemrograman utama termasuk Java, C #, dan C ++.
- e. Siapa saja dapat menulis ekstensi menggunakan HTML5, CSS3, JavaScript, dan Node.js.

- f. API yang memungkinkan untuk menambahkan Menu, UI, Dialog, bahkan bahasa pemodelan sendiri termasuk metamodel, notasi, dan aturan validasi.
- g. Banyak aturan validasi model UML telah ditentukan untuk StarUML dan aturannya diperiksa secara asinkron setiap kali Anda menyimpan atau membuka file model Anda.
- h. Mudah untuk mempublikasikan Dokumen HTML. Dokumen HTML yang dihasilkan dapat dilihat dengan sebagian besar browser web.

2.4. Keserupaan Semantik

Keserupaan semantik didapatkan dari informasi leksikal yang terdapat dalam diagram kasus penggunaan. Pada dua rangkaian kata dapat dilakukan keserupaan dari keduanya (Majumder et al. 2016). Penghitungan keserupaan kosinus dapat dilihat pada Rumus 2.2.

$$CosineSim(A, B) = \frac{\sum_{n=1}^j (nA \times nB)}{\sqrt{\sum_{n=1}^j (nA)^2} \times \sqrt{\sum_{n=1}^j (nB)^2}} \quad (2.1)$$

Sebagai contoh, A dan B adalah kumpulan kata yang akan dibandingkan keserupaannya. A adalah *take cash*. B adalah *save money*. Kata-kata tersebut sebelum dimasukkan ke dalam Rumus 2.1 harus dicari keserupaan antar kata menggunakan WuPalmer dan WordNet sesuai pada Rumus 2.2 (Gao, Zhang, & Chen, 2015).

$$WuP(a, b) = \frac{2 \times DepthLCS}{Depth1 + Depth2} \quad (2.2)$$

Dimana *DepthLCS* adalah nilai kedekatan antar a dan b dalam wordnet. *Depth1* adalah kedalaman kata a dalam wordnet, sedangkan *Depth2* adalah kedalaman kata b dalam wordnet.

2.5. Pemrosesan Bahasa Alami

Pemrosesan bahasa alami adalah bidang penelitian dan aplikasi yang membahas bagaimana komputer dapat digunakan untuk memahami dan memanipulasi teks atau pidato bahasa alami untuk melakukan hal-hal yang bermanfaat (Sevenster et al. 2015).

Tabel 2.2 Pelabelan POS Tagging Stanford NLP

No	Label	Definisi
1	CC	Coordinating conjunction

2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential there
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JR	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP\$	Possessive pronoun (prolog version PRP-S)
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particel
24	SYM	Symbol
25	TO	To
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or resent participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP\$	Possessive wh-pronoun (prolog version WP-S)
36	WRP	Wh-adverb

Dalam penghitungan keserupaan semantik di sub bab sebelumnya, proses awal dilakukan menggunakan pemrosesan bahasa alami dari tokenisasi, lemmatisasi, dan POS *tagging*. Keseluruhan proses tersebut dilakukan dengan bantuan kaskas bantu StanfordNLP. Tokenisasi adalah memecah kalimat menjadi kata-kata. Lemmatisasi adalah mengubah kata menjadi kata bentuk pertama atau kata dasar. Tagging adalah penandaan kata dalam kalimat untuk menentukan labelnya dengan menggunakan *Part of Speech* (POS) dari model *Pen Tree Bank*. Tabel 2.5 menunjukkan simbol dan definisi dari pelabelan menggunakan StanfordNLP dengan model *Pen Tree bank* pada tingkatan kata yang akan digunakan pada penelitian ini.

2.6. TACSim

Topology Attributed Coupling Similarity (TACSim) merupakan metode pengukuran keserupaan graf dengan menghubungkan atribut yang dimiliki dengan simpul dan tepian. Metode TACSim dapat diterapkan pada graf berarah (Chen et al. 2017). Tujuan utama dari metode ini adalah untuk menemukan struktur temporal mobilitas manusia. Tahap-tahap yang dilakukan pada metode ini meliputi

1. Perhitungan Keserupaan Simpul dan Tepian

Perhitungan dilakukan pada setiap pasang simpul dan tepian dari graf1 ke graf2. Perhitungan dilakukan berdasarkan informasi simpul tetangga. L adalah cakupan informasi tetangga yang digunakan. Sebagai contoh pada Gambar 2.2 apabila nilai L adalah 1 maka untuk mengukur keserupaan simpul B akan digunakan informasi dari simpul A dan simpul C, sedangkan untuk mengukur keserupaan simpul C hanya akan digunakan informasi dari simpul B.

2. Pembentukan Matriks Keserupaan Simpul dan Tepian

Setelah nilai keserupaan dari masing-masing pasangan simpul dan tepian sudah didapatkan, selanjutnya dimasukkan kedalam matriks keserupaan simpul(S) dan matriks keserupaan tepian(T). Matriks keserupaan simpul memiliki ukuran $X \times Y$ dimana X adalah jumlah simpul pada graf1 dan Y adalah jumlah simpul pada graf2. Matriks keserupaan tepian memiliki ukuran $M \times N$ dimana M adalah jumlah tepian pada graf1 dan N adalah jumlah tepian pada graf2. Persamaan (2.3) adalah bentuk matriks S dan persamaan (2.4) adalah bentuk matriks T.

$$S = \begin{bmatrix} s_1s_1' & \cdots & s_1s_Y' \\ \vdots & \ddots & \vdots \\ s_j s_1' & \cdots & s_X s_Y' \end{bmatrix} \quad (2.3)$$

Keterangan:

s_i = simpul pada graf1

s_j = simpul pada graf2

$$T = \begin{bmatrix} t_1t_1' & \cdots & t_1t_N' \\ \vdots & \ddots & \vdots \\ t_j t_1' & \cdots & t_M t_N' \end{bmatrix} \quad (2.4)$$

Keterangan:

t_i = tepian pada graf1

t_j = tepian pada graf2

3. Penentuan Nilai Keserupaan Simpul dan Tepian dengan algoritma Hungarian

Matriks keserupaan simpul dan tepian yang sudah diperoleh selanjutnya dicari nilai keserupaan maksimum dari pasangan-pasangan simpul dan tepian. Pencarian nilai maksimum dilakukan dengan menerapkan algoritma *Hungarian*. Pertama dicari nilai maksimum dari keseluruhan nilai pada matriks. Apabila satu nilai sudah diambil maka nilai yang memiliki kolom atau baris yang sama sudah tidak dapat digunakan lagi. Selanjutnya dicari nilai keserupaan maksimum pada matriks dengan nilai yang tersisa, begitu seterusnya hingga sudah tidak ada lagi nilai yang bisa diambil. Selanjutnya setiap nilai keserupaan yang dipilih dirata-rata untuk mendapatkan nilai keserupaan simpul dan tepian.

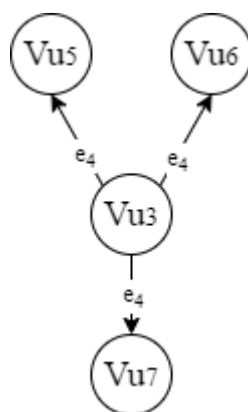
2.7. Keserupaan Struktural

Keserupaan struktural adalah keserupaan pada struktur diagram. Sebagai contoh, relasi antar kelas dan komponen kelas pada diagram kelas. Keserupaan struktural yang telah dilakukan pada penelitian sebelumnya menggunakan dua pendekatan, yaitu penggunaan informasi leksikal dari tetangga (Sevenster et al. 2015) dan penggunaan graf (Yuan, Yan, and Ma 2019). Pada penelitian tersebut melakukan pengukuran keserupaan untuk dapat menggunakan Kembali perancangan perangkat lunak. Pengukuran dilakukan dengan membandingkan UML Class Graph (UCG) dengan hanya memperhatikan aspek struktural dan tidak memperhatikan aspek semantik pada diagram. Untuk melakukan pengukuran

keserupaan pada graf berarah dapat menggunakan metode Graph Edit Distance (GED) yang ditambahkan dengan algoritma Greedy dengan tujuan agar komputasi perhitungan semakin cepat (Riesen et al. 2015).

2.8. Graf

Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$. Dalam hal ini, V merupakan himpunan tidak kosong dari simpul-simpul, dan E adalah himpunan tepian yang digambarkan dalam garis-garis yang menghubungkan sepasang simpul (Deo 2017). Berdasarkan orientasinya, jenis graf terbagi menjadi dua macam, yaitu graf berarah dan graf tidak berarah. Penelitian ini menggunakan jenis graf berarah.



Gambar 2.2 Contoh Graf berarah

Gambar 2.2 menjelaskan terdapat tiga buah simpul dengan nama A, B, dan C. kemudian memiliki dua buah tepian yang menghubungkan dari B ke A dan B ke C.

2.9. Hungarian

Hungarian merupakan algoritma yang digunakan untuk mendapatkan alokasi optimal. Ada dua jenis Hungarian yaitu Hungarian minimasi dan maksimasi. Hungarian minimasi yaitu algoritma yang digunakan untuk mendapatkan cost terkecil dari sebuah kasus, sedangkan Hungarian maksimasi yaitu algoritma yang digunakan untuk mendapatkan hasil atau keuntungan maksimal. Dalam kasus penelitian ini, algoritma yang digunakan adalah Hungarian maksimasi. Hal tersebut bertujuan untuk mendapatkan nilai keserupaan yang terbesar pada diagram kasus penggunaan. Perhitungan dengan algoritma Hungarian dilakukan setelah suatu kasus dirubah kedalam matriks yang berukuran sama, yaitu baris dan kolom berjumlah sama. Apabila ukuran matriks tidak sama, maka ditambahkan data

dummy pada kolom atau baris untuk menyamakan ukuran matriks. Proses perhitungan algoritma Hungarian maksimasi setelah terbentuk matriks yaitu sebagai berikut (Jungnickel 2013).

1. Mencari nilai terbesar dari setiap baris matriks, setelah itu kurangkan nilai tersebut dengan semua nilai pada setiap baris.
2. Pastikan bahwa setiap kolom memiliki nilai 0. Apabila terdapat kolom yang tidak memiliki 0, lakukan pencarian nilai terkecil setiap kolom dan kurangkan masing-masing kolom yang tidak memiliki 0.
3. Beri tanda dengan garis untuk baris atau kolom yang memiliki nilai 0. Pastikan semua kolom memiliki nilai 0. Apabila belum terdapat nilai 0 pada masing-masing kolom (nilai optimal), maka lakukan langkah 2 kembali hingga di dapatkan nilai optimal.
4. Setelah didapatkan matriks yang optimal, maka pilihlah nilai 0 di masing-masing kolom yang berbeda dan gantikan dengan nilai awal dari setiap matriks.

2.10. Graph Edit Distance (GED)

Graph Edit Distance (GED) merupakan metode untuk melakukan pengukuran keserupaan dua buah graf, baik berarah maupun tidak berarah. Dengan menggunakan GED g_1 akan diubah menjadi g_2 menggunakan beberapa operasi. Operasi yang ada pada GED adalah penambahan, penghapusan, dan penukaran pada simpul dan tepian. g_1 terdiri dari simpul-simpul u , dan g_2 terdiri dari simpul-simpul v . Notasi penukaran simpul u menjadi v adalah $u \rightarrow v$, notasi penghapusan simpul u adalah $u \rightarrow \varepsilon$, notasi penambahan simpul v adalah $\varepsilon \rightarrow v$. *Cost* dari masing-masing operasi adalah 1.

Langkah-langkah untuk melakukan perhitungan dengan GED yaitu sebagai berikut.

1. Pembentukan Cost Matriks C untuk setiap Pasangan Simpul pada Graf

Cost matriks C merupakan matriks yang berisi kumpulan nilai cost untuk mengubah graf pertama menjadi graf kedua. Cost (c) merupakan besarnya langkah yang dibutuhkan untuk mengubah simpul pertama menjadi simpul kedua atau untuk mengubah tepian pertama menjadi tepian kedua. Cost matriks memiliki ukuran $m + n$, dimana m merupakan jumlah simpul pada graf pertama dan n merupakan jumlah simpul pada graf kedua.

Sebagai contoh, pada graf pertama inter memiliki jumlah simpul sebesar 6 atau disebut m yang terdiri dari simpul $V_1 = \{Va_1, Vu_1, Vu_2, Vu_3, Vu_4, Va_2\}$ dan graf kedua memiliki simpul 7 yang disebut n yang terdiri dari simpul $V_2 = \{Va_3, Vu_9, Vu_{10}, Vu_{11}, Vu_{12}, Vu_{13}, Va_4\}$. Maka dari kedua graf tersebut di dapatkan $m + n$ sebesar 13 dengan simpul gabungan $V_{1+} = \{Va_1, Vu_1, Vu_2, Vu_3, Vu_4, Va_2, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$ dan $V_{2+} = \{Va_3, Vu_9, Vu_{10}, Vu_{11}, Vu_{12}, Vu_{13}, Va_4, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$. Setelah mendapatkan V_{1+} dan V_{2+} , selanjutnya dibentuk sebuah matriks cost C dari kedua graf yang terdiri dari 4 kuadran, yaitu c_{11} dan seterusnya merupakan kuadran I, $c_{1\varepsilon}$ dan seterusnya merupakan kuadran II, $c_{\varepsilon 1}$ dan seterusnya merupakan kuadran III, 0 dan seterusnya merupakan kuadran IV seperti pada Persamaan (2.5) berikut.

$$C = \begin{matrix} & c_{11} & c_{12} & \dots & c_{1m} & c_{1\varepsilon} & \infty & \dots & \infty \\ & c_{21} & c_{22} & \dots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n1} & c_{n2} & \dots & c_{nm} & \infty & \dots & \infty & c_{n\varepsilon} \\ c_{\varepsilon 1} & \infty & \dots & \dots & \infty & 0 & 0 & \dots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & \ddots & 0 \\ \infty & \dots & \infty & c_{\varepsilon m} & 0 & \dots & 0 & 0 & 0 \end{matrix} \quad (2.5)$$

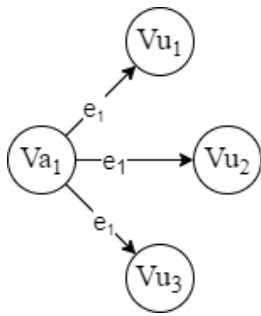
Keterangan:

- Kuadran I = cost yang di dapatkan untuk substitusi dari V_1 ke V_2
- Kuadran II = cost yang di dapatkan untuk menghapus V_1
- Kuadran III = cost yang di dapatkakan untuk menambahkan V_2
- Kuadran IV = cost yang di dapatkan untuk substitusi dari ε menjadi ε
- V_1 = simpul-simpul pada graf 1
- V_2 = simpul-simpul pada graf2

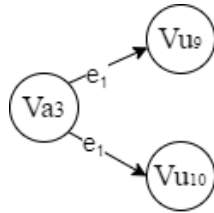
2. Perhitungan Kekeragaman

Setelah melakukan pembentukan matriks cost, dilakukan pencarian cost minimum pada matriks cost tersebut. Untuk mencari cost minimum pada GED diperlukan perhitungan dari masing-masing permutasi yang ada sehingga total permutasi pada GED yaitu $(m+n)!$.

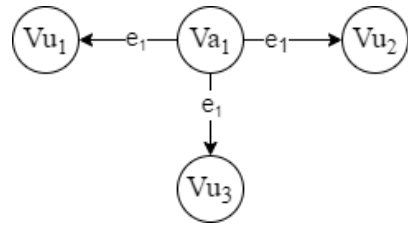
Sebagai contoh, apabila terdapat graf representasi dari diagram sesuai dengan aturan notasi pada Tabel 3.10 seperti pada Gambar 2.3 sebagai Graf1 dengan perhitungan *cost* pertama dilakukan dari simpul Va_1 , Gambar 2.4 sebagai Graf2



Gambar 2.3 Graf1



Gambar 2.4 Graf2



Gambar 2.5 Graf1¹

dengan perhitungan *cost* pertama dilakukan dari simpul Va_3 dan Gambar 2.5 yang merupakan Graf1 yang diputar sehingga dilakukan perhitungan *cost* pertama dilakukan dari simpul Vu_1 yang disebut Graf1¹. Untuk selanjutnya akan dilakukan perhitungan keserupaan antara Graf1 dan Graf2, serta Graf1¹ dan Graf 2.

Untuk perhitungan keserupaan antara Graf1 dan Graf2 dilakukan perhitungan *cost* pertama yaitu dengan menghitung *cost* antara Va_1 dan Va_3 , sehingga mendapatkan nilai *cost* sebesar 1. Selanjutnya dilakukan perhitungan *cost* antara Va_1 dan Vu_9 , sehingga mendapatkan nilai *cost* sebesar 4. Perhitungan *cost* dilakukan hingga seluruh simpul dari kedua graf telah dihitung nilai *cost*-nya dan nilai *cost* tersebut dituliskan pada matriks berukuran $(m+n) \times (m+n)$ yaitu 7×7 sebagai berikut.

$$C = \left[\begin{array}{ccc|cccc} 1 & 4 & 4 & 4 & \infty & \infty & \infty \\ 3 & 0 & 0 & \infty & 1 & \infty & \infty \\ 3 & 0 & 0 & \infty & \infty & 1 & \infty \\ 3 & 0 & 0 & \infty & \infty & \infty & 1 \\ \hline 3 & \infty & \infty & 0 & 0 & 0 & 0 \\ \infty & 1 & \infty & 0 & 0 & 0 & 0 \\ \infty & \infty & 1 & 0 & 0 & 0 & 0 \end{array} \right]$$

Setelah didapatkan *cost* matriks C seperti diatas, dilakukan pencarian nilai *cost* minimum dengan memperhatikan seluruh nilai pada matriks berukuran 7×7 tersebut. Hal tersebut akan menghasilkan kompleksitas sebesar $7!$ hingga mendapatkan nilai *cost* minimum pada setiap kolom dan baris matriks. Dari contoh tersebut didapatkan kombinasi nilai *cost* minimum sebagai berikut.

$$C = \left[\begin{array}{ccc|cccc} \textcircled{1} & 4 & 4 & 4 & \infty & \infty & \infty \\ 3 & \textcircled{0} & 0 & \infty & 1 & \infty & \infty \\ 3 & 0 & \textcircled{0} & \infty & \infty & 1 & \infty \\ 3 & 0 & 0 & \infty & \infty & \infty & \textcircled{1} \\ \hline 3 & \infty & \infty & \textcircled{0} & 0 & 0 & 0 \\ \infty & 1 & \infty & 0 & \textcircled{0} & 0 & 0 \\ \infty & \infty & 1 & 0 & 0 & \textcircled{0} & 0 \end{array} \right]$$

Dari pemilihan *cost* minimum didapatkan kombinasi dengan nilai *cost* 1,0,0,1,0,0,0. Sehingga jumlah *cost* yang didapatkan dari perhitungan keserupaan antara Graf1 dan Graf2 yaitu 2. Untuk selanjutnya dilakukan perhitungan antara Graf1¹ dan Graf2 dengan langkah yang sama seperti perhitungan Graf1 dan Graf2

sehingga didapatkan *cost* matriks C dan kombinasi nilai minimum *cost* sebagai berikut.

$$C = \left[\begin{array}{ccc|cccc} 3 & 0 & 0 & 1 & \infty & \infty & \infty \\ 1 & 4 & 4 & \infty & 4 & \infty & \infty \\ 3 & 0 & 0 & \infty & \infty & 1 & \infty \\ 3 & 0 & 0 & \infty & \infty & \infty & 1 \\ \hline 3 & \infty & \infty & 0 & 0 & 0 & 0 \\ \infty & 1 & \infty & 0 & 0 & 0 & 0 \\ \infty & \infty & 1 & 0 & 0 & 0 & 0 \end{array} \right]$$

Jumlah *cost* perhitungan keserupaan antara Graf1¹ dan Graf2 yaitu sebesar 2. Dari perhitungan antara Graf1 dan Graf2 serta Graf1¹ dan Graf2 didapatkan hasil GED yang sama yaitu nilai *cost* sebesar 2. Graf1¹ merupakan hasil pemutaran dari Graf1. Hal tersebut membuktikan bahwa pada hasil perhitungan GED tidak dipengaruhi oleh posisi simpul, sehingga tidak diperlukan simpul awal untuk melakukan perhitungan keserupaan antara dua buah graf menggunakan GED.

2.11. Gwet's AC1

AC1-statistic merupakan statistika untuk melakukan pengukuran terhadap indeks kesepakatan antara dua pengamat. Pengukuran nilai AC1 diusulkan pada tahun 2002 oleh Gwet. Dibandingkan dengan Cohen Kappa, Gwet's AC1 menunjukkan pendekatan yang lebih dapat diandalkan (Gwet 2002). Hasil observasi pengamat ditulis dalam bentuk table 2x2 untuk melakukan perhitungan AC1 seperti pada Tabel 2.3 berikut.

Tabel 2.3 Tabel Penulisan Hasil Pengamatan

Pengamat 1	Pengamat 2		
	Ya	Tidak	Total
Ya	A	B	B1=A+B
Tidak	C	D	B2=C+D
Total	A1=A+C	A2=B+D	N

Pada Tabel 2.3 terdapat dua pengamat melakukan klasifikasi subjek berjumlah N kedalam dua kategori. Dua kategori tersebut diberi label “Ya” dan “Tidak”. Pengamat 1 dan 2 mengklasifikasikan A sebagai “Ya”. Pengamat 1 mengklasifikasikan B sebagai “Ya” dan pengamat 2 mengklasifikasikan B sebagai “Tidak”. Pengamat 1 mengklasifikasikan C sebagai “Tidak” dan pengamat 2 mengklasifikasikan C sebagai “Ya”. Pengamat 1 dan 2 mengklasifikasikan D sebagai “Tidak”. A1 dan A2 menunjukkan jumlah subjek yang diklasifikasikan

oleh pengamat 1. B1 dan B2 menunjukkan jumlah subjek yang diklasifikasikan oleh pengamat 2.

Perhitungan AC-statistic:

Kesepakatan yang terobservasi seperti pada Persamaan (2.6) berikut.

$$P = \frac{A+D}{N} \quad (2.6)$$

Dimana A adalah banyaknya data yang dikelompokkan ke dalam kategori “Ya” oleh kedua pengamat. D adalah banyaknya data yang dikelompokkan ke dalam kategori “Tidak” oleh kedua pengamat.

Probabilitas *change-agreement* seperti pada Persamaan (2.7) berikut.

$$e(\gamma) = 2P_1(1 - P_1) \quad (2.7)$$

Dimana $P_1 = \frac{(A1+B1)/2}{N}$ merepresentasikan perkiraan kemungkinan seorang pengamat (1 atau 2) mengelompokkan data ke dalam kategori “Ya”. A1 dan B1 masing-masing adalah jumlah pengamat 1 atau 2 mengelompokkan data ke dalam kategori “Ya”, sedangkan N adalah jumlah data.

AC1-statistic dituliskan seperti Persamaan (2.8) berikut.

$$AC1 = \frac{P-e(\gamma)}{1-e(\gamma)} \quad (2.8)$$

Nilai Gwet’s AC1 dapat menentukan tingkat kesepakatan antar dua pakar.

Tabel 2.4 Memberikan interpretasi nilai Gwet’s AC1.

Tabel 2.4 Interpretasi Nilai Gwet’s AC1 (Landis and Koch 2012)

Nilai Indeks AC1	Proporsi Kesepakatan
< 0	Rendah (<i>less than chance agreement</i>)
0.01 – 0.20	Sedikit (<i>slight agreement</i>)
0.21 – 0.40	Cukup (<i>fair agreement</i>)
0.41 – 0.60	Sedang (<i>moderate agreement</i>)
0.61 – 0.80	Substansial (<i>substansial agreement</i>)
0.81 – 1	Hampir Sempurna (<i>almost perfect agreement</i>)

Minimum Indeks nilai AC1 yaitu $-\infty$ dan nilai maksimumnya yaitu 1. Apabila kedua pakar tidak memiliki kesepakatan sama sekali, maka nilai AC1 yang diperoleh yaitu nilai minimum dengan nilai $AC1 = 0$. Sebaliknya apabila didapatkan kesepakatan dari kedua pakar, maka nilai AC1 yang diperoleh yaitu nilai maksimum dengan nilai $AC1 = 1$. Apabila nilai AC1 yang didapatkan lebih mendekati 1, maka tingkat kesepakatan antar pakar akan semakin tinggi. Namun

apabila nilai kappa yang di dapatkan kurang dari 0, maka tingkat kesepakatan antar pakar semakin rendah.

Berikut adalah contoh perhitungan AC1-static dalam suatu contoh kasus tertentu sebagai berikut.

Terdapat dua orang pengamat yang dimintai pendapatnya terhadap 100 buah perkuliahan, apakah bermanfaat atau tidak. Hasil pengamatan terlihat seperti Tabel 2.5 berikut.

Tabel 2.5 Data Pengamatan dari Contoh Kasus

Pengamat 1	Pengamat 2		
	Ya	Tidak	Total
Ya	40	9	49
Tidak	6	45	51
Total	46	54	100

$$P = \frac{40 + 45}{100} = 0,85$$

$$e(\gamma) = 2 \left(\frac{(49 + 46)/2}{100} \right) \left(1 - \frac{(49 + 46)/2}{100} \right)$$

$$= 2 \left(\frac{49 + 46}{2 \times 100} \right) \left(1 - \frac{49 + 46}{2 \times 100} \right)$$

$$= 2 \times 0,475 \times 0,525$$

$$= 0,49$$

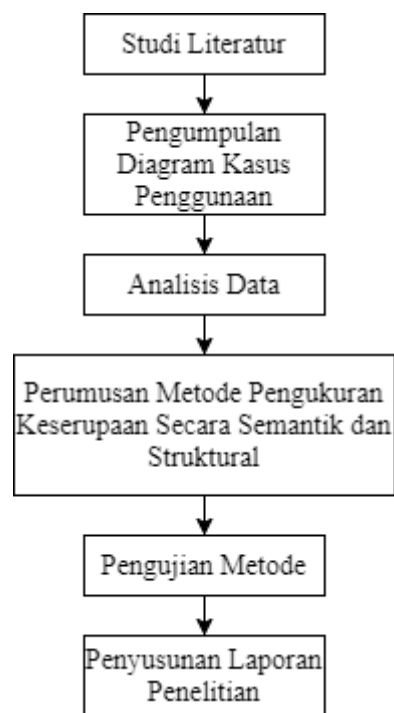
$$AC1 = \frac{0,85 - 0,49}{1 - 0,49} = 0,70$$

Sehingga nilai indeks kesepakatannya termasuk pada level substansial.

BAB 3

METODOLOGI PENELITIAN

Seperti yang diilustrasikan pada Gambar 3.1, penelitian ini meliputi beberapa tahap penelitian yang terdiri dari 1) studi literatur, 2) pengumpulan diagram kasus penggunaan, 3) analisis data, 4) perumusan metode pengukuran keserupaan secara semantik dan struktural, 5) pengujian metode, serta 6) penyusunan laporan penelitian. Setiap tahapan pada metodologi penelitian ini akan dijelaskan lebih rinci pada setiap subbab.



Gambar 3.1 Diagram Alur Metodologi Penelitian

3.1. Studi Literatur

Subbab ini menjelaskan studi literatur yang penulis kaji berdasarkan topik yang berkaitan dengan diagram kasus penggunaan, pemodelan diagram kasus penggunaan dalam bentuk graf dan metode pengukuran keserupaan antar diagram baik secara semantik maupun struktural. Studi literatur tentang penelitian ini yaitu materi-materi yang berhubungan dengan keserupaan antar diagram kasus penggunaan yang diperoleh dari buku, jurnal, buku elektronik, dan lain-lain. Pada tahap ini dilakukan pengkajian literatur terhadap sejumlah hasil penelitian yang

membahas mengenai keserupaan dokumen yang dilakukan oleh penelitian sebelumnya.

3.2. Pengumpulan Diagram Kasus Penggunaan

Data yang dikumpulkan merupakan data diagram kasus penggunaan yang dibuat dari perangkat lunak yang diperoleh dari Github. Perangkat lunak yang digunakan dapat berupa perangkat lunak dengan berbagai jenis, seperti desktop maupun website. Hal tersebut dikarenakan pembuatan diagram kasus penggunaan tidak bergantung pada jenis perangkat lunak apapun.

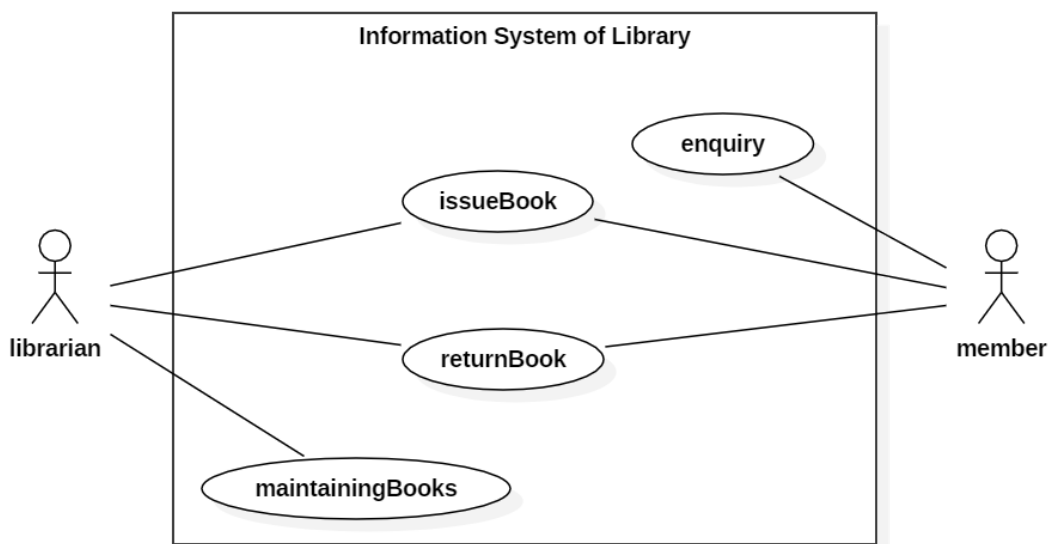
Table 3.1 Responden Diagram Jawaban Kasus Penggunaan

No	Sistem	Kode Diagram	Deskripsi	Responden	Rata-rata Jumlah Aktor	Rata-rata jumlah kasus penggunaan	Rata-rata jumlah relasi	Rata-rata Kata Unik
1	<i>Outlay</i>	A	Aplikasi untuk melakukan pencatatan keuangan	11	1	7	7	13
2	<i>Quick Bill</i>	B	Aplikasi untuk melakukan manajemen penjualan di kasir	9	2	15	15	26
3	<i>Restaurant Management System</i>	C	Aplikasi untuk melakukan manajemen terhadap penjualan dan karyawan di restoran	16	1	13	13	17

Diagram kasus penggunaan dibuat oleh beberapa mahasiswa yang telah mendapatkan pendidikan mengenai diagram UML, khususnya diagram kasus penggunaan. Perangkat lunak yang diberikan berjumlah 3 jenis perangkat lunak

yang berbeda, diberikan kepada 36 mahasiswa. Responden jawaban diagram kasus penggunaan secara lengkap dapat dilihat pada tabel 3.1. Dari pengumpulan data tersebut didapatkan 36 jawaban diagram kasus penggunaan yang berbeda. Setelah itu dilakukan pembuatan kunci jawaban diagram dari masing-masing perangkat lunak. Data tersebut digunakan untuk melakukan pengukuran keserupaan diagram kasus penggunaan.

Gambar 3.2 merupakan salah satu contoh kasus penggunaan dari sistem *Information System of Library*. Setiap diagram kasus penggunaan yang didapatkan akan diolah menggunakan Star UML menjadi format berbentuk XMI (XML Metadata Interchange). Gambar 3.3 merupakan contoh potongan hasil ekspor diagram kasus penggunaan menjadi format berbentuk XMI.



Gambar 3.2 Contoh diagram kasus penggunaan

```

...
<packagedElement xmi:id="AAAAAAFweEgr3iBExtY=" name="Librarian" ...
xmi:type="uml:Actor">
  <ownedMember xmi:id="AAAAAAFweFNcoiMyrpw=" ... xmi:type="uml:Association" ... >
  ...
  <ownedEnd xmi:id="AAAAAAFweFNcoiM0zAc=" ... xmi:type="uml:Property" ...
type="AAAAAAFweE+0IyD1U60=">
  ...
</ownedMember>
<ownedMember xmi:id="AAAAAAFweFN1liOim3g=" ... xmi:type="uml:Association" ... >

```

```

...
<ownedEnd xmi:id="AAAAAAFweFN1lyOk7ZM=" ... xmi:type="uml:Property" ...
type="AAAAAAFweE/OXCF8DF4=">
...
</ownedMember>
<ownedMember xmi:id="AAAAAAFweFOKzCQhf0=" ... xmi:type="uml:Association" ... >
...
<ownedEnd xmi:id="AAAAAAFweFOKzCQjVOI=" ... xmi:type="uml:Property" ...
type="AAAAAAFweE/aTCHWiBo=">
...
</ownedMember>
</packagedElement>
<packagedElement xmi:id="AAAAAAFweE+0IyDIU60=" name="Issue Book" ...
xmi:type="uml:UseCase">
...
<packagedElement xmi:id="AAAAAAFweE/OXCF8DF4=" name="Return Book" ...
xmi:type="uml:UseCase">
..
<packagedElement xmi:id="AAAAAAFweE/aTCHWiBo=" name="Maintaining Books" ...
xmi:type="uml:UseCase"/>
...

```

Gambar 3.3 Contoh hasil ekspor XMI

Contoh diatas dapat diartikan terdapat sebuah elemen dengan nama *Librarian* bertipe aktor memiliki hubungan asosiasi dengan properti *AAAAAAFweE+0IyDIU60=* memiliki nama *Issue Book*. Kemudian *Librarian* juga memiliki asosiasi *AAAAAAFweE/OXCF8DF4=* dengan nama *Return Book*. Selain itu, *Librarian* juga memiliki asosiasi *AAAAAAFweE/aTCHWiBo=* dengan nama *Maintaining Books*. Jadi dapat disimpulkan sebuah aktor dengan nama *Librarian* memiliki relasi asosiasi kepada kasus penggunaan *Issue Book*, *Return Book*, dan *Maintaining Books*.

3.3. Analisis Data Para Pakar

Data yang telah terkumpul, dilakukan analisa menggunakan uji validitas dan uji kehandalan agar dapat menjadi acuan pengujian metode pengukuran keserupaan diagram kasus penggunaan. Sebelum dilakukan analisa terhadap data, pengolahan data dilakukan dengan Memberikan kuisisioner kepada para pakar mengenai nilai keserupaan antara jawaban diagram kasus penggunaan dan kunci jawabannya.

Pakar yang dimaksud yaitu orang yang telah mendapatkan materi mengenai perancangan perangkat lunak khususnya diagram kasus penggunaan minimal selama dua tahun. Kuisisioner yang diberikan berisi jawaban diagram kasus penggunaan dan kunci jawabannya, serta rentang penilaian untuk Memberikan nilai keserupaan antar dua buah diagram kasus penggunaan tersebut. Nilai keserupaan diisi dengan menggunakan skala dari nilai 1 sampai 5 dengan kriteria dari tidak mirip hingga mirip. Setelah itu, jawaban para pakar akan diuji validitasnya menggunakan Pearson dan akan diuji kehandalan pakar menggunakan Alpha Cronbach. Hasil dari penilaian para pakar yang telah teruji dan handal, akan digunakan sebagai acuan nilai standart keserupaan antar dua buah diagram kasus penggunaan agar dapat menghitung kehandalan metode yang diusulkan.

3.4. Metode Pengukuran Keserupaan Diagram Kasus Penggunaan

Pengukuran keserupaan antar diagram kasus penggunaan terbagi menjadi dua, yaitu berdasarkan semantik dan struktural. Untuk pengukuran keserupaan diagram kasus penggunaan secara semantik diperoleh dari melakukan pemetaan informasi pada diagram kasus penggunaan, yang meliputi nama aktor, relasi, dan kasus penggunaan. Sedangkan untuk pengukuran keserupaan secara struktural dibedakan menjadi dua komponen, yaitu inter-struktural dan intra-struktural. Pengukuran keserupaan secara struktural diperoleh dari pemodelan diagram kasus penggunaan dalam bentuk graf dan selanjutnya dihitung menggunakan penggabungan metode *Graph Edit Distance* (GED) dengan algoritma *Greedy*. Perhitungan keserupaan antar dua buah diagram kasus penggunaan ($ucdSim(d_1, d_2)$) dapat dilihat pada Rumus 3.1.

$$ucdSim(d_1, d_2) = w_{sem} \times semSim(d_1, d_2) + w_{struc} \times strucSim(d_1, d_2) \quad (3.1)$$

keterangan:

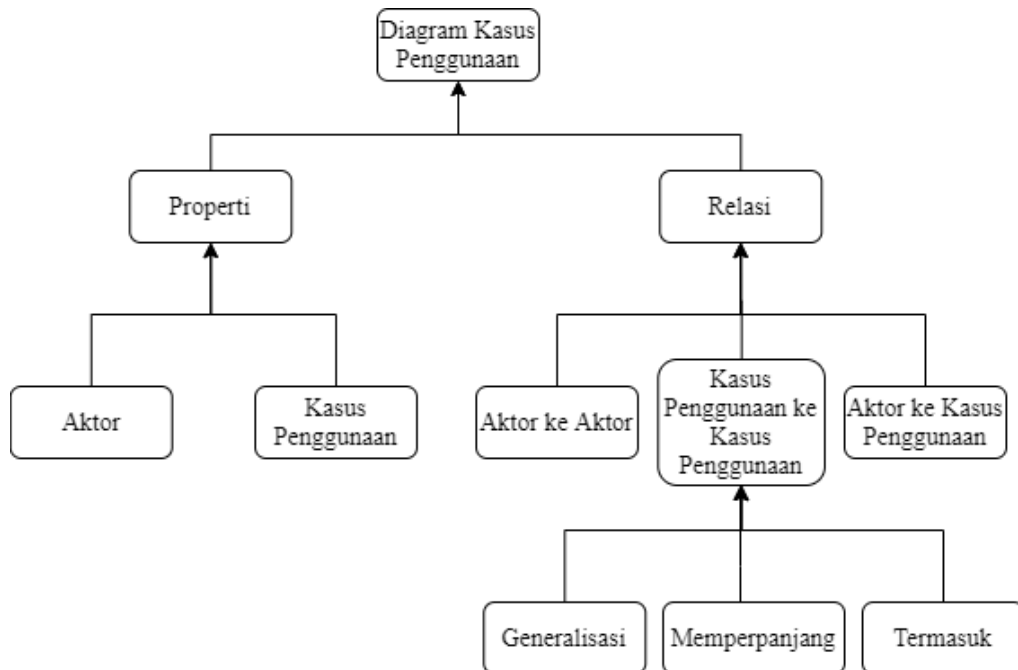
- d_1 = diagram kasus penggunaan pertama,
- d_2 = diagram kasus penggunaan kedua,
- w_{sem} = bobot keserupaan semantik,
- w_{struc} = bobot keserupaan struktural,
- $semSim$ = keserupaan semantik diagram kasus penggunaan,

strucSim = keserupaan struktural diagram kasus penggunaan.

Hasil perhitungan keserupaan semantik dan struktural yang didapatkan selanjutnya digunakan untuk perhitungan keserupaan antara dua buah diagram kasus penggunaan. Sebagai contoh, perhitungan bobot keserupaan semantik dan bobot keserupaan struktural masing-masing 0,5. Apabila nilai keserupaan semantik adalah 0,46 dan nilai keserupaan struktural adalah 0,99 maka didapatkan nilai keserupaan antara dua buah diagram kasus penggunaan sebesar 0,73.

3.4.1. Keserupaan Semantik Diagram Kasus Penggunaan

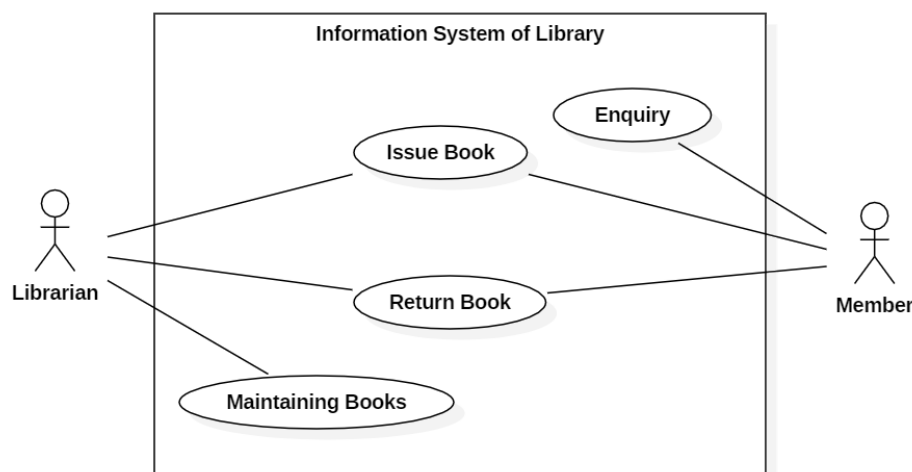
Keserupaan semantik diagram kasus penggunaan diukur dengan menggunakan *Topology-Attributed Coupling Similarity* (TACSim) yang dimodifikasi menggunakan pemrosesan bahasa alami. Metode ini digunakan untuk mengukur keserupaan informasi leksikal yang terdapat pada diagram kasus penggunaan. Informasi yang terkandung dalam metadata diagram kasus penggunaan dibagi dalam beberapa komponen yang dapat dilihat pada Gambar 3.4.



Gambar 3.4 Metadata Diagram Kasus Penggunaan

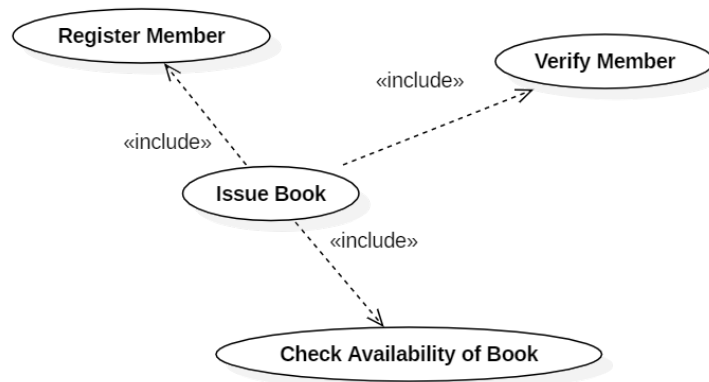
Gambar 3.4 menunjukkan dua bagian utama yang terdapat pada diagram kasus penggunaan yaitu properti dan relasi. Properti memiliki dua jenis yaitu aktor dan kasus penggunaan. Sedangkan relasi memiliki tiga jenis yaitu aktor ke aktor, kasus penggunaan ke kasus penggunaan, dan aktor ke kasus penggunaan. Relasi

antar kasus penggunaan memiliki tiga jenis yaitu generalisasi, memperpanjang, dan termasuk. Pada diagram kasus penggunaan terdapat dua jenis kasus penggunaan, yaitu kasus penggunaan utama (*main use case*) dan spesifikasi kasus penggunaan (*use case specification*).



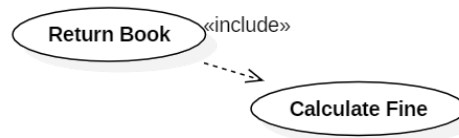
Gambar 3.5 Contoh diagram kasus penggunaan pertama

Gambar 3.5 menunjukkan contoh kunci jawaban diagram kasus penggunaan sistem *Information System of Library* (ISL) sebagai contoh diagram kasus penggunaan pertama (UCD1). Pada UCD1 terdapat beberapa komponen yang meliputi aktor pertama (va_1) yaitu *Librarian* dan memiliki tiga kasus penggunaan utama. Kasus penggunaan tersebut meliputi kasus penggunaan pertama (vu_1) yaitu *Maintaining Books*, kasus penggunaan kedua (vu_2) yaitu *Return Book*, dan kasus penggunaan ketiga (vu_3) yaitu *Issue Book*. Pada aktor kedua (va_2) yaitu *Member*, memiliki kasus penggunaan keempat (vu_4) yaitu *Enquiry* dan memiliki kasus penggunaan ketiga (vu_3) yaitu *Issue Book*, kasus penggunaan kedua (vu_2) yaitu *Return Book*.



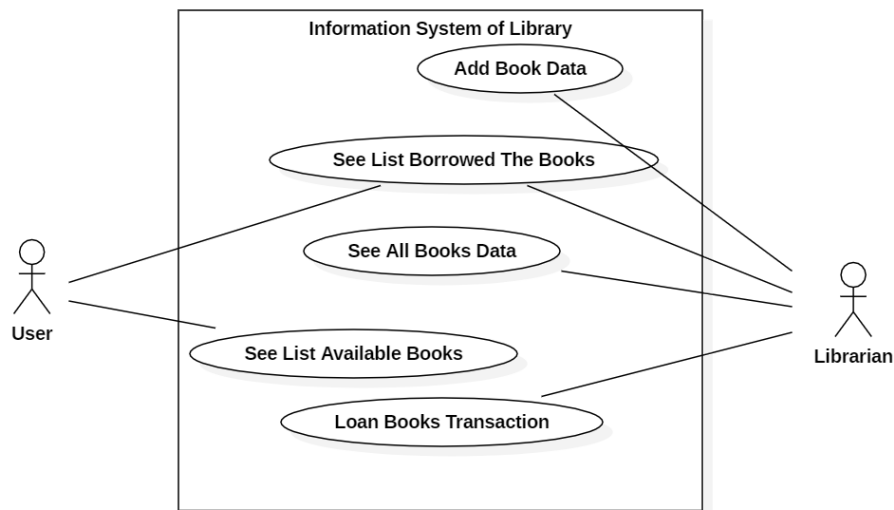
Gambar 3.6 Spesifikasi kasus penggunaan utama *Issue Book*

Gambar 3.6 menunjukkan komponen spesifikasi dari kasus penggunaan utama (vu₃) yaitu *Issue Book*. Vu₃ memiliki kasus penggunaan kelima yang termasuk (vu₅) yaitu *Register Member*, memiliki kasus penggunaan keenam yang termasuk (vu₆) yaitu *Verify Member*, dan memiliki kasus penggunaan ketujuh yang termasuk (vu₇) yaitu *checkAvailableOfBook*.



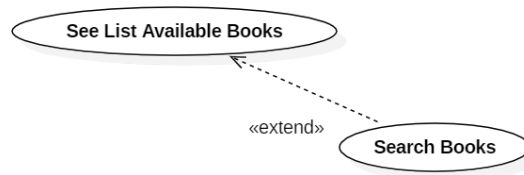
Gambar 3.7 Spesifikasi kasus penggunaan utama *Return Book*

Pada Gambar 3.7 menunjukkan komponen spesifikasi dari kasus penggunaan utama (vu₂) yaitu *Return Book*. Vu₂ memiliki kasus penggunaan kedelapan yang termasuk (vu₈) yaitu *calcuteFine*.



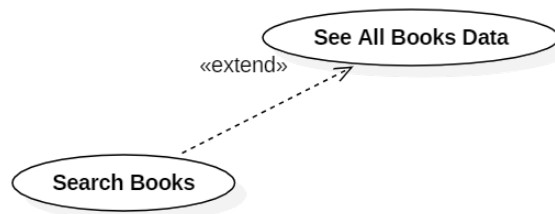
Gambar 3.8 Contoh diagram kasus penggunaan kedua

Gambar 3.8 menunjukkan contoh salah satu jawaban diagram kasus penggunaan sistem ISL yang merupakan contoh diagram kasus penggunaan kedua (UCD2). Komponen pada diagram kasus penggunaan tersebut meliputi dua aktor, untuk aktor pertama (va_3) *User*, kasus penggunaan kesembilan (vu_9) yaitu *See List Borrowed the Books*, kasus penggunaan ke sepuluh (vu_{10}) yaitu *See List Available Books*. Dan aktor kedua (va_4) yaitu *Librarian* memiliki kasus penggunaan ke sebelas (vu_{11}) yaitu *Add Book Data*, kasus penggunaan ke dua belas (vu_{12}) yaitu *See All Books Data*, dan kasus penggunaan ketiga belas (vu_{13}) yaitu *Loan Books Transaction*.



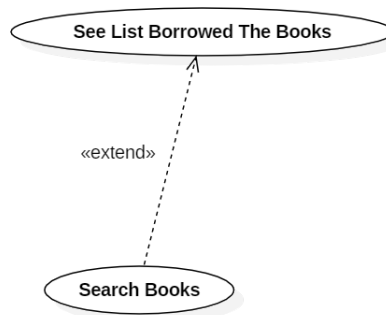
Gambar 3.9 Spesifikasi kasus penggunaan utama *See List Available Books*

Gambar 3.9 menunjukkan kasus penggunaan utama kesembilan (vu_9) yaitu *See List Available Books*, memiliki kasus penggunaan ke empat belas (v_{14}) yaitu *searchBook*.



Gambar 3.10 Spesifikasi kasus penggunaan utama *See All Books Data*

Gambar 3.10 menunjukkan kasus penggunaan utama kedua belas (vu_{12}) yaitu *See All Books Data*, memiliki kasus penggunaan keempat belas (vu_{14}) yaitu *searchBook*.



Gambar 3.11 Spesifikasi kasus penggunaan utama *seeListBorroweBooks*

Gambar 3.11 menunjukkan kasus penggunaan utama kesepuluh (vu_{10}) yaitu *See List Borrowed Book*, memiliki kasus penggunaan keempat belas (vu_{14}) yaitu *Search Book*.

Untuk melakukan perhitungan keserupaan semantik diagram kasus penggunaan. Perhitungan dilakukan dengan mengikuti Rumus 3.2.

$$\text{semSim}(d_1, d_2) = w_{\text{simpul}} \times \text{simpulSim}(d_1, d_2) + w_{\text{tepiian}} \times \text{tepiianSim}(d_1, d_2) \quad (3.2)$$

Keterangan:

d_1 = diagram kasus penggunaan pertama,

d_2 = diagram kasus penggunaan kedua,

w_{simpul} = bobot keserupaan simpul,

w_{tepiian} = bobot keserupaan tepiian,

simpulSim = keserupaan simpul diagram kasus penggunaan,

tepiianSim = keserupaan tepiian diagram kasus penggunaan.

semSim = keserupaan semantik diagram kasus penggunaan

Nilai keserupaan simpul dan tepian yang telah didapatkan pada proses sebelumnya digunakan sebagai input pada Rumus 3.1. Asumsi bobot keserupaan simpul dan tepian adalah sama yaitu masing-masing 0,5. Dan nilai keserupaan simpul adalah 0,25 dan nilai keserupaan tepian adalah 0.62. Maka nilai keserupaan semantik yang didapatkan adalah 0,44.

Tahap-tahap yang dilakukan untuk mengukur keserupaan semantik diagram kasus penggunaan adalah sebagai berikut.

3.4.1.1 *Pemodelan Diagram menjadi Graf*

Langkah pertama yang dilakukan adalah memodelkan diagram kasus penggunaan dalam bentuk graf. Pemodelan ini mengikuti aturan yang digunakan pada pengukuran keserupaan inter-struktural. Bentuk graf yang dihasilkan akan berbeda dengan graf yang digunakan sebagai input pada metode TACSim. Apabila pada graf input TACSim atribut disimpan dalam bentuk simpul baru, pada penelitian ini atribut hanya akan disimpan pada simpul terkait. Notasi simpul dan tepian dapat dilihat pada Tabel 3.1 dan Tabel 3.2. Perbedaan antara graf untuk pengukuran keserupaan semantik dengan graf untuk pengukuran keserupaan inter-struktur adalah pada graf ini setiap simpul menyimpan informasi leksikal dari diagram kasus penggunaan yang direpresentasikan. Sebagai contoh pengukuran keserupaan semantik antara diagram kasus pengguna menggunakan UCD1 (Gambar 3.5) dan UCD2 (Gambar 3.8). Model graf1 dapat dilihat pada Gambar 3.13 untuk diagram 1 dan model graf2 pada Gambar 3.14 untuk diagram 2.

Tabel 3.1 daftar informasi leksikal pada simpul graf1

Label Simpul	Informasi yang Disimpan
V _{a1}	<i>Librarian</i>
V _{a2}	<i>Member</i>
V _{u1}	<i>Maintaining Books</i>
V _{u2}	<i>Return Book</i>
V _{u3}	<i>Issue Book</i>
V _{u4}	<i>Enquiry</i>

Tabel 3.2 daftar informasi leksikal pada simpul graf2

Label Simpul	Informasi yang Disimpan
V _{a3}	<i>User</i>

V _{a4}	<i>Librarian</i>
V _{u9}	<i>See List Available Books</i>
V _{u10}	<i>See List Borrowed Books</i>
V _{u11}	<i>Add Book Data</i>
V _{u12}	<i>See All Books Data</i>
V _{u13}	<i>Loan Book Transaction</i>

Contoh informasi leksikal yang disimpan pada simpul Va adalah nama aktor, sedangkan informasi yang disimpan pada simpul Vu adalah nama kasus penggunaan. Untuk informasi leksikal pada setiap relasi pada diagram contoh dapat dilihat pada Tabel 3.3 untuk graf1 dan Tabel 3.4 untuk graf2. Pada masing-masing tepian terdapat 3 jenis informasi leksikal, yaitu tipe tepian (t), simpul sumber dari tepian (srcV), dan simpul target dari tepian (tgtV).

Tabel 3.3 Informasi leksikal pada tepian graf1

Kode Tepian	tipe (t ₁)	simpul sumber (srcV ₁)	simpul target (tgtV ₁)
V _{a1} , V _{u1}	<i>asosiasi</i>	<i>Librarian</i>	<i>Maintaining Books</i>
V _{a1} , V _{u2}	<i>asosiasi</i>	<i>Librarian</i>	<i>Return Book</i>
V _{a1} , V _{u3}	<i>asosiasi</i>	<i>Librarian</i>	<i>Issue Book</i>
V _{a1} , V _{u4}	<i>asosiasi</i>	<i>Librarian</i>	<i>Enquiry</i>
V _{a2} , V _{u2}	<i>asosiasi</i>	<i>Member</i>	<i>Return Book</i>
V _{a2} , V _{u3}	<i>asosiasi</i>	<i>Member</i>	<i>Issue Book</i>
V _{a2} , V _{u4}	<i>asosiasi</i>	<i>Member</i>	<i>Enquiry</i>

Tabel 3.4 Informasi leksikal pada tepian graf2

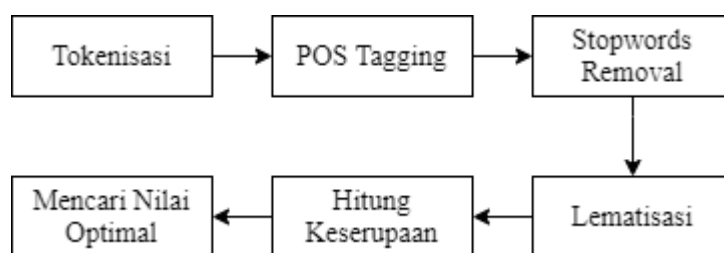
Kode Tepian	tipe (t ₂)	simpul sumber (srcV ₂)	simpul target (tgtV ₂)
V _{a3} , V _{u9}	<i>asosiasi</i>	<i>User</i>	<i>See List Borrowed the Books</i>
V _{a3} , V _{u10}	<i>asosiasi</i>	<i>User</i>	<i>See List Available Books</i>
V _{a4} , V _{u10}	<i>asosiasi</i>	<i>User</i>	<i>See List Available Books</i>
V _{a4} , V _{u11}	<i>asosiasi</i>	<i>User</i>	<i>Add Book Data</i>
V _{a4} , V _{u12}	<i>asosiasi</i>	<i>Librarian</i>	<i>See All Books Data</i>
V _{a4} , V _{u13}	<i>asosiasi</i>	<i>Librarian</i>	<i>Loan Books Transaction</i>

Contoh informasi leksikal yang disimpan pada V_{a1}, V_{u1} pada graf1 yaitu tipe tepian *asosiasi*, simpul sumber *Librarian*, dan simpul target *Maintaining Books*. Sedangkan untuk contoh informasi leksikal yang disimpan pada V_{a3}, V_{u9}

pada graf2 yaitu tipe tepian *asosiasi*, simpul sumber *User*, dan simpul target *See List Borrowed the Books*.

3.4.1.2 *Perhitungan Kekerupaan Simpul dan Tepian*

Perhitungan kekerupaan semantik dibagi mejadi dua tahap, yaitu kekerupaan simpul dan kekerupaan tepian. Pengukuran kekerupaan antar simpul dan tepian dilakukan dengan pemrosesan bahasa alami. Proses pertama dalam melakukan perhitungan kekerupaan antara dua komponen dilakukan dengan Pemrosesan Bahasa Alami. Proses perhitungan kekerupaan antar dua komponen dapat dilihat pada Gambar 3.12. Proses tersebut diawali dengan melakukan tokenisasi untuk mendapatkan token dari setiap masukan, selanjutnya dilakukan POS tagging untuk mendapatkan jenis kata setiap token. Langkah selanjutnya dilakukan stopwords Removal yaitu menghilangkan kata hubung yang tidak berpengaruh, selanjutnya Lematisasi untuk mendapatkan kata dasar dari setiap token. Selanjutnya menghitung kekerupaan antar dua komponen dan yang terakhir yaitu mencari nilai optimal dari nilai yang di dapat dari perhitungan.



Gambar 3.12 Perhitungan Kekerupaan antara Dua Komponen Diagram

1. **Tokenisasi**

Proses tokenisasi dilakukan dengan membagi kalimat/kata-kata masukan menjadi token-token. Pesan pada masukan yang terdapat pada data diagram kasus penggunaan tidak dapat diproses secara langsung karena memiliki bentuk khusus dibanding dengan kalimat yang ada pada aturan EYD. Jadi, tokenisasi ini membutuhkan syarat khusus sebagai berikut.

- a. Apabila pada suatu kalimat ditemukan spasi di tengah kata, maka akan dilakukan pemisahan pada kata tersebut menjadi sebuah token.

Berdasarkan syarat yang berlaku, beberapa luaran dari proses tokenisasi untuk contoh diagram pada UCD1, kasus penggunaan yang dimiliki oleh aktor 1 *Librarian* (V_{a1}) yaitu *Issue*, *Book* (V_{u3}), kasus penggunaan yang dimiliki oleh aktor

2 *Member* (Va₂) yaitu *return*, *Book* (Vu₂) dan *Enquiry* (Vu₄). Untuk contoh beberapa hasil tokenisasi pada UCD2, kasus penggunaan yang dimiliki oleh aktor 3 *User* (Va₃) yaitu *See*, *List*, *Borrowed*, *The*, *Books* (Vu₉), kasus penggunaan yang dimiliki oleh aktor 4 *Librarian* (Va₄) yaitu *Add*, *Book*, *Data* (Vu₁₁).

2. Pos Tagging

Sebelum proses POS tagging, dilakukan perubahan semua token ke dalam huruf kecil. Proses pos tagging dilakukan untuk mengetahui jenis token. Jenis token dapat berupa kata kerja, kata sifat, kata benda, dan lain sebagainya. Proses ini dilakukan dengan menggunakan *Stanford Postagger* dengan model Bahasa Inggris oleh *Stanford NLP*.

Berdasarkan syarat yang berlaku, beberapa luaran dari proses POS Tagging untuk contoh diagram pada UCD1, kasus penggunaan yang dimiliki oleh aktor 1 *librarian* (Va₁) yaitu *issue/NN*, *book/VB* (Vu₃), kasus penggunaan yang dimiliki oleh aktor 2 *member* (Va₂) yaitu *return/NN*, *book/NN* (Vu₂) dan *enquiry/NN* (Vu₄). Untuk contoh beberapa hasil tokenisasi pada UCD2, kasus penggunaan yang dimiliki oleh aktor 3 *user* (Va₃) yaitu *see/VB*, *list/NN*, *borrowed/VBN*, *the/DT*, *books/NNS* (Vu₉), kasus penggunaan yang dimiliki oleh aktor 4 *librarian* (Va₄) yaitu *add/VB*, *book/VB*, *data/NNS* (Vu₁₁).

3. Stopwords Removal

Penghilangan stopwords bertujuan agar nilai keserupaan antara dua rangkaian kata menjadi meningkat berdasarkan makna sebenarnya. Luaran dari proses ini berdasarkan masukkan proses sebelumnya adalah sebagai berikut.

Berdasarkan syarat yang berlaku, beberapa luaran dari proses POS Tagging untuk contoh diagram pada UCD1, kasus penggunaan yang dimiliki oleh aktor 1 *librarian* (Va₁) yaitu *issue/NN*, *book/VB* (Vu₃), kasus penggunaan yang dimiliki oleh aktor 2 *member* (Va₂) yaitu *return/NN*, *book/NN* (Vu₂) dan *enquiry/NN* (Vu₄). Untuk contoh beberapa hasil tokenisasi pada UCD2, kasus penggunaan yang dimiliki oleh aktor 3 *user* (Va₃) yaitu *see/VB*, *list/NN*, *borrowed/VBN*, ~~*the/DT*~~, *books/NNS* (Vu₉), kasus penggunaan yang dimiliki oleh aktor 4 *librarian* (Va₄) yaitu *add/VB*, *book/VB*, *data/NNS* (Vu₁₁).

4. Lematisasi

Proses lematisasi adalah proses mengubah token menjadi kata dasar. Hal ini bertujuan agar menambah akurasi pada proses berikutnya. Lematisasi dibantu oleh Stanford NLP. Berdasarkan syarat yang berlaku, beberapa luaran dari proses Lematisasi untuk contoh diagram pada UCD1, kasus penggunaan yang dimiliki oleh aktor 1 *librarian* (V_{a1}) yaitu *issue/NN*, *book/VB* (V_{u3}), kasus penggunaan yang dimiliki oleh aktor 2 *member* (V_{a2}) yaitu *return/NN*, *book/NN* (V_{u2}) dan *enquiry/NN* (V_{u4}). Untuk contoh beberapa hasil tokenisasi pada UCD2, kasus penggunaan yang dimiliki oleh aktor 3 *user* (V_{a3}) yaitu *see/VB*, *list/NN*, *borrowed/VBN*, *books/NNS* (V_{u9}), kasus penggunaan yang dimiliki oleh aktor 4 *librarian* (V_{a4}) yaitu *add/VB*, *book/VB*, *datum/NNS* (V_{u11}).

5. Hitung Keserupaan

Pada proses ini perhitungan dilakukan dengan tiga langkah, yaitu menghitung keserupaan antar kasus penggunaan pada UCD1, antar kasus penggunaan pada UCD2, dan menghitung keserupaan antara kasus penggunaan UCD1 dengan kasus penggunaan UCD2. Untuk melakukan perhitungan keserupaan, digunakan keserupaan kosinus. Proses keserupaan kosinus dilakukan dengan 3 tahapan, yaitu pembuatan vektor, perhitungan keserupaan kata dan perhitungan nilai kosinus. Berikut akan dijelaskan proses untuk menghitung keserupaan antara V_{u3} pada UCD1 ($UCD1_V_{u3}$) dan V_{u11} pada UCD2 ($UCD2_V_{u11}$).

a. Pembuatan Vektor

Vektor dibangun berdasarkan kata-kata yang berbeda yang muncul dari dua rangkaian kata yang ingin dihitung keserupaannya. Setiap kata akan dihitung keserupaannya terhadap kata yang lain agar mendapatkan kata dengan makna berbeda. Jadi vektor yang dihasilkan dari $UCD1_V_{u3}$ dan $UCD2_V_{u11}$ adalah “*issue/NN*, *book/NN*, *add/VB*, *datum/NNS*”.

b. Perhitungan Keserupaan Kata

Keserupaan kata dihitung menggunakan WuPalmer dan Wordnet. Sebelum dihitung menggunakan WuPalmer, jenis dari kata asal (hasil POS tagging) tetap diperhatikan. Jadi, jika jenis kata berbeda, maka nilai keserupaan dianggap nol. Jika penghitungan tidak dapat diselesaikan menggunakan WuPalmer karena berbeda jenis kata, maka penghitungan akan dilakukan menggunakan *Levensthein Distance*.

Sedangkan kata akan dianggap serupa jika memiliki jenis kata yang sama dan memiliki makna yang serupa.

b. Perhitungan Nilai Kosinus

Hasil perhitungan dari masukkan diatas yaitu sebagai berikut.

Vektor : *issue/NN* *book/NN* *add/VB* *datum/NNS*

M1 : *issue/NN* *book/NN*

M2 : *add/VB* *book/VB* *datum/NNS*

V-M1 : 0 1

V-M2 : 0 1 0

Hasil : 0.25

Jadi nilai keserupaan UCD1_Vu₃ dengan UCD2_Vu₁₁ yaitu 0.25.

Sedangkan untuk nilai keserupaan UCD1_Vu₃ dengan UCD2_Vu₉ yaitu sebesar 0.2. Untuk masing-masing penjelasan perhitungan pada keserupaan simpul dan keserupaan tepian adalah sebagai berikut.

3.4.1.2.1 Keserupaan simpul

Keserupaan simpul didapatkan dengan membandingkan informasi leksikal antar simpul pada graf1 dengan simpul pada graf2. Apabila simpul memiliki label yang berbeda maka nilai keserupaan yang didapatkan adalah 0. Simpul dengan label yang sama selanjutnya diukur keserupaannya menggunakan pemrosesan bahasa alami.

3.4.1.2.2 Keserupaan tepian

Keserupaan tepian didapatkan dengan membandingkan tipe tepian, simpul sumber tepian, dan simpul target antara graf1 dan graf2. Tipe tepian, simpul sumber dan simpul target akan diukur keserupaannya menggunakan pemrosesan bahasa alami. Untuk mendapatkan nilai keserupaan pada tepian digunakan Rumus 3.3 sebagai berikut.

$$\text{tepianSim}(g_1, g_2) = \frac{\text{CosineSim}(t_1, t_2) + \text{CosineSim}(srcV_1, srcV_2) + \text{CosineSim}(tgtV_1, tgtV_2)}{3} \quad (3.3)$$

Keterangan:

g₁ = graf pertama,

g₂ = graf kedua,

t₁ = tipe tepian pada graf pertama,

- t_2 = tipe tepian pada graf kedua,
- $srcV_1$ = simpul sumber dari tepian graf pertama,
- $srcV_2$ = simpul sumber dari tepian graf kedua,
- $tgtV_1$ = simpul target dari tepian graf pertama,
- $tgtV_2$ = simpul target dari tepian graf kedua.

Tabel 3.5 Perhitungan tepian

tepiian g_1	tepiian g_2	<i>CosineSim</i> (t_1, t_2)	<i>CosineSim</i> ($srcV_1, srcV_2$)	<i>CosineSim</i> ($tgtV_1, tgtV_2$)	tepiianSim (g_1, g_2)
V_{a1}, V_{u1}	V_{a3}, V_{u9}	1,00	0,00	0,20	0,40
V_{a1}, V_{u1}	V_{a3}, V_{u10}	1,00	0,00	0,20	0,40
V_{a1}, V_{u1}	V_{a4}, V_{u10}	1,00	1,00	0,20	0,73

Sebagai contoh perhitungan tepian seperti pada Tabel 3.5. Pada tabel tersebut ditampilkan contoh perhitungan tepian yang memiliki informasi leksikal tipe tepian, sumber simpul, dan target simpul. Masing-masing informasi leksikal yang terdapat pada tepian dilakukan perhitungan keserupaan dengan pemrosesan bahasa alami dan dihitung menggunakan Rumus 3.3.

3.4.1.3 *Pembentukan Matrik Keserupaan Simpul dan Tepian*

Setelah diperoleh nilai keserupaan simpul dari setiap pasangan simpul dan nilai keserupaan tepian dari setiap pasangan tepian selanjutnya nilai-nilai tersebut dimasukkan kedalam matriks keserupaan simpul dan matriks keserupaan tepian. matriks keserupaan simpul memiliki ukuran $X \times Y$ dimana X adalah jumlah simpul pada graf1 dan Y adalah jumlah simpul pada graf2. Tabel 3.6 adalah hasil matriks keserupaan simpul. Sebagai contoh nilai $V_{a1}V_{a3}$ yang merupakan perbandingan dari simpul (*Librarian*) V_{a1} dan (*User*) V_{a3} adalah sebesar 0,00. Tabel 3.7 adalah hasil matriks keserupaan tepian.

Contoh selanjutnya yaitu pembentukan matrik tepian yang dapat dilihat pada Tabel 3.7. pada tabel dituliskan hasil dari perhitungan keserupaan pasangan tepian menggunakan pemrosesan bahasa alami dengan tiga jenis informasi leksikal yang ada pada setiap tepian, yaitu tipe tepian, simpul sumber, dan simpul target. Pada contoh, nilai keserupaan dari tepian V_{a1}, V_{u1} dan V_{a3}, V_{u9} adalah 0,40 dengan perhitungan yang telah dijelaskan pada point keserupaan tepian yang selanjutnya di

masukkan pada matrik yang memiliki ukuran $P \times Q$, dimana P merupakan jumlah tepian pada graf1 dan Q merupakan jumlah tepian pada graf2.

Tabel 3.6 Matriks Keserupaan Simpul

graf1/graf2	Va ₃	Va ₄	Vu ₉	Vu ₁₀	Vu ₁₁	Vu ₁₂	Vu ₁₃
Va ₁	0,00	1,00	0,00	0,00	0,00	0,00	0,00
Va ₂	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Vu ₁	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₂	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₃	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₄	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Tabel 3.7 Matriks Keserupaan Tepian

graf1/graf2	Va ₃ , Vu ₉	Va ₃ , Vu ₁₀	Va ₄ , Vu ₁₀	Va ₄ , Vu ₁₁	Va ₄ , Vu ₁₂	Va ₄ , Vu ₁₃
Va ₁ , Vu ₁	0,40	0,40	0,73	0,75	0,73	0,75
Va ₁ , Vu ₂	0,40	0,40	0,73	0,75	0,73	0,75
Va ₁ , Vu ₃	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₂	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₃	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₄	0,33	0,33	0,67	0,67	0,67	0,67

3.4.1.4 *Penentuan Nilai Keserupaan Simpul dan Tepian dengan algoritma Hungarian*

Matriks keserupaan simpul dan tepian yang sudah diperoleh selanjutnya dicari nilai keserupaan maksimum dari pasangan-pasangan simpul dan tepian. Pencarian nilai maksimum dilakukan dengan menerapkan algoritma *Hungarian*.

Tabel 3.8 Nilai Keserupaan Terpilih pada Matriks Keserupaan Simpul

graf1/graf2	Va ₃	Va ₄	Vu ₉	Vu ₁₀	Vu ₁₁	Vu ₁₂	Vu ₁₃
Va ₁	0,00	1,00	0,00	0,00	0,00	0,00	0,00
Va ₂	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Vu ₁	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₂	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₃	0,00	0,00	0,20	0,20	0,25	0,20	0,25
Vu ₄	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Vd ₁	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Tabel 3.9 Nilai Keserupaan Terpilih pada Matriks Keserupaan Tepian

graf1/graf2	Va ₃ , Vu ₉	Va ₃ , Vu ₁₀	Va ₄ , Vu ₁₀	Va ₄ , Vu ₁₁	Va ₄ , Vu ₁₂	Va ₄ , Vu ₁₃
Va ₁ , Vu ₁	0,40	0,40	0,73	0,75	0,73	0,75
Va ₁ , Vu ₂	0,40	0,40	0,73	0,75	0,73	0,75
Va ₁ , Vu ₃	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₂	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₃	0,40	0,40	0,73	0,75	0,73	0,75
Va ₂ , Vu ₄	0,33	0,33	0,67	0,67	0,67	0,67

Pertama dicari nilai maksimum dari keseluruhan nilai pada matriks, sebagai contoh pada matriks keserupaan simpul diperoleh nilai keserupaan maksimal adalah 1 pada pasangan simpul Va₁ dan Va₄. Sehingga semua pasangan simpul yang terdapat Va₁ atau Va₄ nilai keserupaannya sudah tidak dapat dipakai lagi. Selanjutnya dicari nilai keserupaan maksimum pada matriks dengan nilai yang tersisa, begitu seterusnya hingga sudah tidak ada lagi nilai yang bisa diambil. Pada Tabel 3.8 dan Tabel 3.9 kolom yang diberi warna hijau adalah kolom yang nilai keserupaannya dipilih. Karena jumlah dari kolom dan baris tidak sama, pada algoritma Hungarian menambahkan satu baris/kolom Vd₁ (kolom/baris *dummy*) untuk menyamakan keduanya.

Selanjutnya setiap nilai keserupaan yang dipilih dirata-rata untuk mendapatkan nilai keserupaan simpul dan tepian. Sehingga diperoleh keserupaan simpul dengan nilai 0,25 dan keserupaan tepian dengan nilai 0,62.

3.4.2. Keserupaan Struktural Diagram Kasus Penggunaan

Keserupaan struktural diagram kasus penggunaan dihitung dengan mempresentasikan diagram kasus penggunaan menjadi graf. Simpul yang dimiliki graf dalam merepresentasikan diagram kasus penggunaan meliputi aktor dan kasus penggunaan. Tepian yang dimiliki adalah aktor, asosiasi, generalisasi, memperpanjang, dan termasuk masing-masing diberi penandaan. Sebagai contoh, untuk tepian aktor diberi penandaan e_a. Berdasarkan penjelasan tersebut, secara lengkap dituliskan pada Tabel 3.10.

Keserupaan struktural dibagi menjadi dua komponen yaitu inter-struktural dan intra-struktural.

Tabel 3.10 Notasi Pemodelan Komponen Diagram Menjadi Graf

No	Tipe Elemen	Nama	Penandaan
1	Simpul	Simpul Aktor	va
2	Simpul	Simpul Use Case	vu
3	Tepian	Tepian Aktor	e _a
4	Tepian	Tepian Asosiasi	e ₁
5	Tepian	Tepian Generalisasi	e ₂
6	Tepian	Tepian Memperpanjang	e ₃
7	Tepian	Tepian Termasuk	e ₄

Untuk mendapatkan nilai keserupaan intra-struktural ($strucSim(d_1, d_2)$) dilakukan dengan mengikuti rumus 3.4.

$$strucSim(d_1, d_2) = w_{inter} \times interSim(d_1, d_2) + w_{intra} \times intraSim(d_1, d_2) \quad (3.4)$$

keterangan:

d_1 = diagram kasus penggunaan pertama,

d_2 = diagram kasus penggunaan kedua,

w_{inter} = bobot keserupaan inter-struktural,

w_{intra} = bobot keserupaan intra-struktural,

$interSim$ = keserupaan inter-struktural,

$intraSim$ = keserupaan intra-struktural.

Sebagai contoh nilai keserupaan inter-struktural sebesar 0,62 dan nilai keserupaan intra-struktural sebesar 0,67. Kemudian untuk bobot dari kedua perhitungan keserupaan inter-struktural dan intra-struktural masing-masing dimisalkan 0,5. Maka akan di dapatkan keserupaan pada pengukuran struktural sebesar 0,65. Untuk perhitungan keserupaan struktural diagram kasus penggunaan dijelaskan dengan detail sebagai berikut.

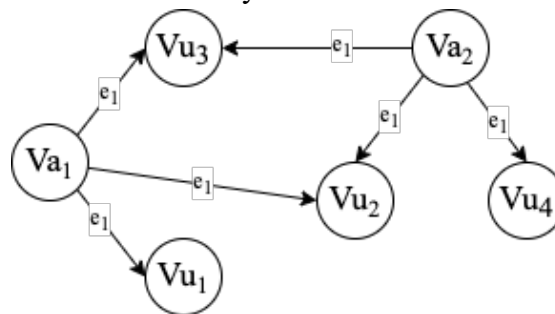
3.4.2.1. Pengukuran Keserupaan Inter-struktural

Pengukuran keserupaan inter-struktural dilakukan dengan 3 tahapan, yaitu pemodelan diagram kasus penggunaan menjadi graf, pembentukan cost matriks C untuk setiap pasangan simpul pada graf, dan yang terakhir perhitungan keserupaan inter-struktural dengan metode *Graph Edit Distance* (GED) yang ditambahkan

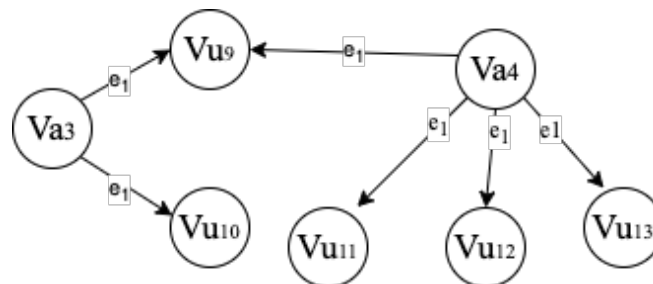
dengan algoritma *Greedy*. Untuk detail dari setiap tahapan, dijelaskan sebagai berikut.

3.4.2.1.1. Pemodelan Diagram Kasus Penggunaan Menjadi Graf

Sebelum melakukan perhitungan keserupaan inter-struktural antara dua buah diagram kasus penggunaan, dilakukan pemodelan diagram kasus penggunaan menjadi graf. Untuk penandaan simpul maupun tepian dalam pemodelan, secara lengkap dapat dilihat pada Tabel 3.10. Sebagai contoh representasi dari diagram kasus penggunaan menjadi graf dapat dilihat pada Gambar 3.13 yang merupakan representasi dari contoh diagram kasus penggunaan pertama (UCD1) dan Gambar 3.14. merupakan representasi dari contoh diagram kasus penggunaan kedua (UCD2) pada gambar tersebut masing-masing simpul dan tepian dinotasikan sesuai dengan Tabel 3.10. Sebagai contoh, aktor 1 pada UCD1 yaitu *Librarian*, dinotasikan menjadi Va_1 dan seterusnya.



Gambar 3.13 Representasi dari Gambar 3.4 menjadi graf



Gambar 3.14 Representasi dari Gambar 3.7 menjadi graf

3.4.2.1.2. Pembentukan *Cost* Matrik C untuk setiap Pasangan Simpul pada Graf

Cost matriks C merupakan matrik yang berisi kumpulan nilai *cost* untuk mengubah graf pertama menjadi graf kedua. *Cost* (*c*) merupakan besarnya langkah yang dibutuhkan untuk mengubah simpul pertama menjadi simpul kedua atau untuk mengubah tepian pertama menjadi tepian kedua. *Cost* matriks memiliki ukuran $m + n$, dimana m merupakan jumlah simpul pada graf pertama dan n merupakan jumlah simpul pada graf kedua.

Sebagai contoh, pada graf pertama inter-struktural memiliki jumlah simpul sebesar 6 atau disebut m yang terdiri dari simpul $V_1 = \{Va_1, Vu_1, Vu_2, Vu_3, Vu_4, Va_2\}$ dan graf kedua memiliki simpul 7 yang disebut n yang terdiri dari simpul $V_2 = \{Va_3, Vu_9, Vu_{10}, Vu_{11}, Vu_{12}, Vu_{13}, Va_4\}$. Maka dari kedua graf tersebut di dapatkan m + n sebesar 13 dengan simpul gabungan $V_{1+} = \{Va_1, Vu_1, Vu_2, Vu_3, Vu_{14}, Va_2, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$ dan $V_{2+} = \{Va_3, Vu_9, Vu_{10}, Vu_{11}, Vu_{12}, Vu_{13}, Va_{14}, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$. Setelah mendapatkan V_{1+} dan V_{2+} , selanjutnya dibentuk sebuah matriks cost C dari kedua graf yang terdiri dari 4 kuadran, yaitu c_{11} dan seterusnya merupakan kuadran I, $c_{1\varepsilon}$ dan seterusnya merupakan kuadran II, $c_{\varepsilon 1}$ dan seterusnya merupakan kuadran III, 0 dan seterusnya merupakan kuadran IV seperti Persamaan (3.5).

$$C = \begin{matrix} & c_{11} & c_{12} & \dots & c_{1m} & c_{1\varepsilon} & \infty & \dots & \infty \\ & c_{21} & c_{22} & \dots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \dots & c_{nm} & \infty & \dots & \infty & c_{n\varepsilon} \\ c_{\varepsilon 1} & \infty & \dots & \infty & 0 & 0 & \dots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \dots & \infty & c_{\varepsilon m} & 0 & \dots & 0 & 0 \end{matrix} \quad (3.5)$$

Keterangan:

- Kuadran I = cost yang di dapatkan untuk substitusi dari V_1 ke V_2
- Kuadran II = cost yang di dapatkan untuk menghapus V_1
- Kuadran III = cost yang di dapatkan untuk menambahkan V_2
- Kuadran IV = cost yang di dapatkan untuk substitusi dari ε menjadi ε
- V_1 = simpul-simpul pada graf 1
- V_2 = simpul-simpul pada graf2

Sebagai contoh matriks cost C yang di dapatkan dari graf 1 yaitu representasi UCD1 dan graf 2 yaitu representasi dari UCD2 pada komponen inter-struktural sebagai berikut.

$$C = \left[\begin{array}{cccccc|cccccc} 1 & 4 & 4 & 1 & 4 & 4 & 4 & 4 & \infty & \infty & \infty & \infty & \infty \\ 3 & 1 & 0 & 5 & 0 & 0 & 0 & 0 & \infty & 1 & \infty & \infty & \infty & \infty \\ 3 & 0 & 1 & 5 & 1 & 1 & 1 & 1 & \infty & \infty & 1 & \infty & \infty & \infty \\ 1 & 4 & 4 & 1 & 4 & 4 & 4 & 4 & \infty & \infty & \infty & 4 & \infty & \infty \\ 3 & 0 & 1 & 5 & 1 & 1 & 1 & 1 & \infty & \infty & \infty & \infty & 1 & \infty \\ 3 & 1 & 0 & 5 & 0 & 0 & 0 & 0 & \infty & \infty & \infty & \infty & \infty & 1 \\ \hline 3 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & 1 & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & 1 & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & 5 & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & 1 & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & 1 & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

Aturan-aturan yang berlaku untuk menghitung cost yaitu:

- Tepian merupakan milik dari simpul asal. Sebagai contoh pada Gambar 3.13, simpul Va_1 memiliki dua bagian, yaitu 1 simpul dan 3 tepian. Dan untuk Va_2 juga memiliki dua bagian, yaitu 1 simpul dan 3 tepian.
- Simpul target hanya terdiri dari simpul itu sendiri.

Pada contoh matriks *cost* diatas, *cost* pertama pada kuadran I, di dapatkan dari besarnya *cost* dari V_{a1} menjadi V_{a3} , yaitu di dapatkan *cost* sebesar 1. Contoh lainnya, *cost* kedua pada kuadran I, di dapatkan dari besarnya *cost* dari V_{a1} menjadi V_{u4} , yaitu di dapatkan *cost* sebesar 4.

3.4.2.1.3. Perhitungan Keserupaan Inter-struktural

Perhitungan keserupaan inter-struktural antara dua buah diagram kasus penggunaan dilakukan dengan metode *Graph Edit Distance* (GED) dengan menambahkan algoritma *Greedy*. Tujuan dari penambahan algoritma *Greedy* pada pemilihan permutasi yaitu untuk mencari *cost* minimum yang diperlukan untuk mengubah graf 1 menjadi graf 2. Pada 2.9 telah dijelaskan, untuk mencari *cost* minimum pada GED diperlukan perhitungan dari masing-masing permutasi yang ada sehingga total permutasi pada GED yaitu $(m+n)!$. Maka dari itu ditambahkan algoritma *Greedy* pada tahap ini. Jumlah perhitungan yang diperlukan untuk mencari *cost* minimum pada GED yang menggunakan algoritma *Greedy* yaitu $O(m^2)$ dimana m lebih dari sama dengan n .

Algoritma *Greedy* dilakukan dua kali pada tahap pemilihan permutasi ini, Adapun aturan pertama dari pemilihan minimum *cost* adalah sebagai berikut.

- Apabila $m > n$, maka lakukan pencarian *cost* minimum pertama pada Matriks *cost* Kuadran II.
- Apabila $m < n$, maka lakukan pencarian *cost* minimum pertama pada Matriks *cost* Kuadran III.
- Apabila $m = n$ dengan jumlah simpul sama dan tepian tepian dari keduanya, maka lakukan pencarian *cost* minimum pertama pada Matriks *cost* Kuadran I.
- Apabila $m = n$ dengan jumlah simpul dan tepian dari keduanya sama, maka lakukan pencarian *cost* minimum pertama pada Matriks *cost* Kuadran II.

Untuk aturan dalam pemilihan minimum *cost* kedua, pencarian dilakukan pada Kuadran 4. Hal tersebut karena pada kuadrat 4 terdiri dari *cost* 0, dimana *cost* 0 yang merupakan *cost* minimum. Selanjutnya, untuk pemilihan *cost* minimum selanjutnya dilakukan dengan memilih *cost* minimum dari *cost* yang tersisa pada matriks *cost*. Berikut merupakan contoh pemilihan *cost* minimum dari graf 1 dan graf 2.

$$C = \begin{bmatrix} 1 & 4 & 4 & 1 & 4 & 4 & 4 & 4 & \infty & \infty & \infty & \infty & \infty & \infty \\ 3 & 1 & 0 & 5 & 0 & 0 & 0 & 0 & \infty & 1 & \infty & \infty & \infty & \infty \\ 3 & 0 & 1 & 5 & 1 & 1 & 1 & 1 & \infty & \infty & 1 & \infty & \infty & \infty \\ 1 & 4 & 4 & 1 & 4 & 4 & 4 & 4 & \infty & \infty & \infty & 4 & \infty & \infty \\ 3 & 0 & 1 & 5 & 1 & 1 & 1 & 1 & \infty & \infty & \infty & \infty & 1 & \infty \\ 3 & 1 & 0 & 5 & 0 & 0 & 0 & 0 & \infty & \infty & \infty & \infty & \infty & 1 \\ 3 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & 1 & \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & 1 & \infty & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & 5 & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & 1 & \infty & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & 1 & \infty & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty & 0 & 0 & 0 & 0 & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dari matriks *cost* diatas, untuk melakukan pemilihan *cost* pertama dilakukan pencarian nilai *cost* minimum pada kuadran III karena $m < n$. Sehingga di dapatkan 1 sebagai *cost* pertama. Untuk pencarian *cost* kedua dilakukan pada Kuadran IV dan didapatkan nilai *cost* 0. Dan pencarian dilakukan hingga di dapatkan masing-masing nilai *cost* minimum dari baris dan kolom yang berbeda. Sehingga di dapatkan *cost* minimum 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0. Untuk mendapatkan total *cost* pada perhitungan keserupaan inter-struktural dilakukan penjumlahan dari semua *cost* minimum yang di dapatkan, yaitu sebesar 5. Berarti besarnya nilai *cost* keserupaan inter-struktural atau *intraSim* (d_1, d_2) adalah sebesar 5.

Setelah didapatkan *cost* minimum, dilakukan perhitungan keserupaan inter-struktural dengan cara mengurangi 1 dengan *cost* minimum dibagi dengan nilai paling signifikan dari jumlah simpul dan tepian pada graf 1 dan graf 2. Di kurangkan dengan 1 karena kedua diagram dikatakan serupa apabila nilai keserupaannya sebesar 1. Cara menghitungnya dapat dilihat pada persamaan 3.5 berikut.

$$interSim = 1 - \frac{d_{\lambda min}(g_1, g_2)}{\max(|g_1|, |g_2|)} \quad (3.5)$$

Keterangan:

interSim = keserupaan inter-struktural

$d_{\lambda min}$ = *cost* minimum

g_1 = graf 1

g_2 = graf 2

$\max(|g_1|, |g_2|)$ = nilai paling signifikan dari jumlah simpul dan tepian pada graf 1 dan graf 2.

3.4.2.2. Pengukuran Keserupaan Intra-struktural

Pengukuran keserupaan intra-struktural dilakukan dengan melakukan perhitungan terhadap struktur diagram secara spesifik dan keseluruhan. Untuk

menghitung keserupaan intra-struktural antara dua buah diagram kasus penggunaan dilakukan berdasarkan Rumus 3.5 berikut

$$\text{intraSim}(d_1, d_2) = w_a \times a\text{Sim}(d_1, d_2) + w_u \times u\text{Sim}(d_1, d_2) \quad (3.5)$$

Keterangan:

d_1 = diagram kasus penggunaan pertama,

d_2 = diagram kasus penggunaan kedua,

w_a = bobot keserupaan intra-struktural pada aktor,

w_u = bobot keserupaan intra-struktural pada kasus penggunaan,

$a\text{Sim}$ = cost keserupaan intra-struktural pada aktor,

$u\text{Sim}$ = cost keserupaan intra-struktural pada kasus penggunaan.

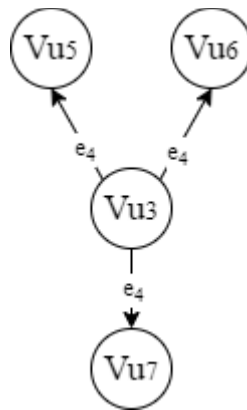
IntraSim = keserupaan intra-struktural diagram kasus penggunaan

Sebagai contoh perhitungan terhadap keserupaan intra-struktural diagram, apabila nilai *cost* keserupaan intra-struktural pada aktor sebesar 0 dan nilai *cost* keserupaan intra-struktural sebesar 5. sedangkan untuk bobot keserupaan intra-struktural pada aktor dan intra-struktural pada kasus keserupaan masing-masing dicontohkan sebesar 0.5. Maka didapatkan nilai *cost* keserupaan intra-struktural sebesar 2.5.

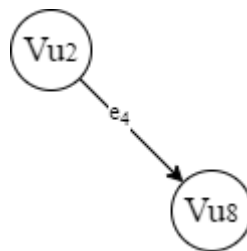
Untuk melakukan pengukuran keserupaan intra-struktural dilakukan dengan 3 tahapan, yaitu pemodelan diagram kasus penggunaan menjadi graf, pembentukan cost matriks C untuk setiap pasangan simpul pada graf, dan yang terakhir perhitungan keserupaan inter-struktural dengan metode *Graph Edit Distance* (GED) yang ditambahkan dengan algoritma Greedy. Untuk detail dari setiap tahapan, dijelaskan sebagai berikut.

3.4.2.2.1. Pemodelan Diagram Kasus Penggunaan Menjadi Graf

Sebelum melakukan perhitungan keserupaan intra-struktural antara dua buah diagram kasus penggunaan, dilakukan pemodelan spesifikasi kasus penggunaan menjadi graf. Untuk penandaan simpul maupun tepian dalam pemodelan, secara lengkap dapat dilihat pada Tabel 3.10. Sebagai contoh representasi dari diagram kasus penggunaan menjadi graf dapat dilihat pada Gambar 3.15 yang merupakan representasi dari contoh spesifikasi kasus penggunaan ketiga (Vu_3) dari diagram kasus penggunaan pertama (UCD1) dan Gambar 3.15. merupakan representasi dari contoh spesifikasi kasus penggunaan kedua (Vu_2) dari diagram kasus penggunaan pertama (UCD1).

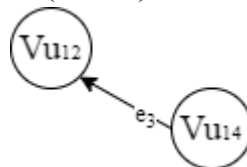


Gambar 3.15 Representasi dari Gambar 3.5 menjadi graf

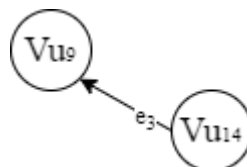


Gambar 3.16 Representasi dari Gambar 3.6 menjadi graf

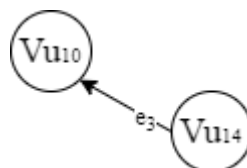
Gambar 3.17 merupakan representasi dari contoh spesifikasi kasus penggunaan kedua belas (Vu_{12}) dari diagram kasus penggunaan kedua (UCD2), Gambar 3.18. merupakan representasi dari contoh spesifikasi kasus penggunaan kesembilan (Vu_9) dari diagram kasus penggunaan kedua (UCD2) dan Gambar 3.19. merupakan representasi dari contoh spesifikasi kasus penggunaan kesepuluh (Vu_{10}) dari diagram kasus penggunaan kedua (UCD2).



Gambar 3.17 Representasi Gambar 3.8 menjadi graf



Gambar 3.18 Representasi Gambar 3.9 menjadi graf



Gambar 3.19 Representasi Gambar 3.10 menjadi graf

3.4.2.2.2. Pembentukan Cost Matriks C untuk setiap Pasangan Simpul pada Graf

Cost matriks C merupakan matriks yang berisi kumpulan nilai *cost* untuk mengubah graf pertama menjadi graf kedua. *Cost* (c) merupakan besarnya langkah yang dibutuhkan untuk mengubah simpul pertama menjadi simpul kedua atau untuk mengubah tepian pertama menjadi tepian kedua. *Cost* matriks memiliki ukuran $m + n$, dimana m merupakan jumlah simpul pada graf pertama dan n merupakan jumlah simpul pada graf kedua.

Sebagai contoh, pada graf pertama inter memiliki jumlah simpul sebesar 2 atau disebut m yang terdiri dari simpul $V_1 = \{Vu_2, Vu_8\}$ dan graf kedua memiliki simpul 2 yang disebut n yang terdiri dari simpul $V_2 = \{Vu_9, Va_{14}\}$. Maka dari kedua graf tersebut di dapatkan $m + n$ sebesar 4 dengan simpul gabungan $V_{1+} = \{Vu_2, Vu_8, \varepsilon, \varepsilon\}$ dan $V_{2+} = \{Vu_9, Va_{14}, \varepsilon, \varepsilon\}$. Setelah mendapatkan V_{1+} dan V_{2+} , selanjutnya dibentuk sebuah matriks cost C dari kedua graf yang terdiri dari 4 kuadran, yaitu c_{11} dan seterusnya merupakan kuadran I, $c_{1\varepsilon}$ dan seterusnya merupakan kuadran II, $c_{\varepsilon 1}$ dan seterusnya merupakan kuadran III, 0 dan seterusnya merupakan kuadran IV seperti Persamaan (3.5).

Sebagai contoh matriks cost C yang di dapatkan dari graf 1 yaitu representasi dari spesifikasi kasus penggunaan *Return Book* (Vu_2) dan graf 2 yaitu representasi dari spesifikasi kasus penggunaan *See List Available Books* (Vu_9) pada komponen intra-struktural sebagai berikut.

$$C = \left[\begin{array}{cc|cc} 1 & 0 & 2 & \infty \\ 0 & 1 & \infty & 1 \\ \hline 1 & \infty & 0 & 0 \\ 0 & 2 & 0 & 0 \end{array} \right]$$

Aturan-aturan yang berlaku untuk menghitung cost yaitu:

- Tepian merupakan milik dari simpul asal. Sebagai contoh pada Gambar 3.13, simpul Vu_2 memiliki dua 1 bagian, yaitu 1 simpul. Dan untuk Vu_9 juga memiliki 1 bagian, yaitu 1 simpul.
- Simpul target hanya terdiri dari simpul itu sendiri.

Pada contoh matriks *cost* diatas, *cost* pertama pada kuadran I, di dapatkan dari besarnya *cost* dari Vu_2 menjadi Vu_9 , yaitu di dapatkan *cost* sebesar 1. Contoh lainnya, *cost* kedua pada kuadran I, di dapatkan besarnya *cost* dari Vu_2 menjadi Vu_{14} , yaitu di dapatkan *cost* sebesar 0.

3.4.2.2.3. Perhitungan Keserupaan Intra-struktural

Perhitungan keserupaan intra-struktural antara dua buah diagram kasus penggunaan dilakukan dengan menghitung seluruh spesifikasi pada diagram kasus penggunaan, baik antar aktor maupun antar kasus penggunaan. Perhitungan dilakukan dengan metode *Graph Edit Distance* (GED) dengan menambahkan algoritma *Greedy*. Tujuan dari penambahakan algoritma *Greedy* pemilihan

permutasi yaitu untuk mencari cost minimum yang diperlukan untuk mengubah graf 1 menjadi graf 2. Pada 2.9 telah dijelaskan, untuk mencari cost minimum pada GED diperlukan perhitungan dari masing-masing permutasi yang ada sehingga total permutasi pada GED yaitu $(m+n)!$. Maka dari itu ditambahkan algoritma *Greedy* pada tahap ini.

Algoritma *Greedy* dilakukan dua kali pada tahap pemilihan permutasi ini, Adapun aturan pertama dari pemilihan minimum cost adalah sebagai berikut.

- Apabila $m > n$, maka lakukan pencarian cost minimum pertama pada Matriks cost Kuadran II.
- Apabila $m < n$, maka lakukan pencarian cost minimum pertama pada Matriks cost Kuadran III.
- Apabila $m = n$ dengan jumlah simpul sama dan tepian berbeda dari keduanya, maka lakukan pencarian cost minimum pertama pada Matriks cost Kuadran I.
- Apabila $m = n$ dengan jumlah simpul dan tepian dari keduanya sama, maka lakukan pencarian cost minimum pertama pada Matriks cost Kuadran II.

Untuk aturan dalam pemilihan minimum cost kedua, pencarian dilakukan pada Kuadran 4. Hal tersebut karena pada kuadrat 4 terdiri dari cost 0, dimana cost 0 merupakan cost minimum. Selanjutnya, untuk pemilihan cost minimum dilakukan dengan memilih cost minimum dari cost yang tersisa pada matriks cost. Berikut merupakan contoh pemilihan cost minimum dari graf 1 dan graf 2.

$$C = \begin{bmatrix} 1 & 0 & 2 & \infty \\ 0 & 1 & \infty & 1 \\ 1 & \infty & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

Dari matriks cost diatas, untuk melakukan pemilihan cost pertama dilakukan pencarian nilai cost minimum pada kuadran II karena $m = n$, jumlah simpul dan tepian dari keduanya sama. Sehingga di dapatkan 1 sebagai cost pertama. Untuk pencarian cost kedua dilakukan pada Kuadran IV dan didapatkan nilai cost 0. Dan pencarian dilakukan hingga di dapatkan masing-masing nilai cost minimum dari baris dan kolom yang berbeda. Sehingga di dapatkan cost minimum 0, 1, 0, 0. Untuk mendapatkan total cost pada perhitungan keserupaan inter-struktural dilakukan penjumlahan dari semua cost minimum yang di dapatkan, yaitu sebesar 1.

Setelah didapatkan cost minimum, dilakukan perhitungan keserupaan inter-struktural dengan cara mengurangi 1 dengan cost minimum dibagi dengan nilai paling signifikan dari jumlah simpul dan tepian pada graf 1 dan graf 2. Untuk

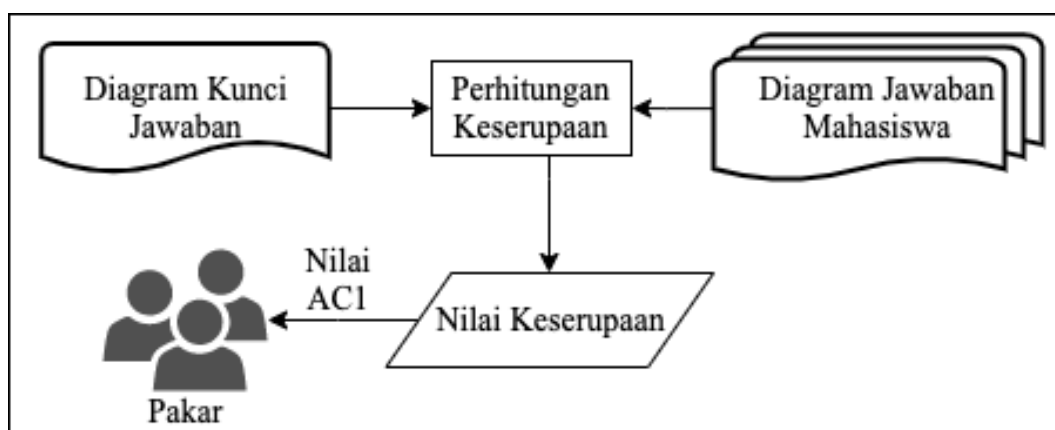
cara menghitungnya dapat dilihat pada persamaan 3.5. Perhitungan untuk mendapatkan keserupaan intra-struktural antara dua buah graf sama dengan perhitungan keserupaan pada inter-struktural, namun karena jumlah pasangan graf yang dihitung pada keserupaan intra-struktural lebih dari 1 maka harus dicari keserupaan optimal dari kumpulan keserupaan intra-struktural.

3.5. Penilaian antara Jawaban Diagram dan Kunci Jawaban Diagram

Nilai keserupaan dari dua buah diagram kasus penggunaan digunakan sebagai bahan pertimbangan dalam Memberikan penilaian suatu jawaban pada E-Learning. Untuk selanjutnya metode keserupaan ini akan menghasilkan penilaian otomatis pada dua buah kasus penggunaan.

3.6. Pengujian Metode

Pengujian untuk mengetahui kehandalan metode, dilakukan dengan pembangunan sistem yang mengimplementasikan metode tersebut. Ilustrasi pengujian dapat dilihat pada Gambar 3.20. Hasil yang didapatkan dari sistem tersebut akan dibandingkan dengan hasil kuesioner oleh para pakar berdasarkan analisis data sebelumnya. Untuk perbandingan antara nilai keserupaan dengan pakar digunakan nilai AC1. Cara mendapatkan nilai AC1 yaitu dengan metode Gwet's AC1. Pengujian terhadap metode usulan dalam penelitian ini dapat dilihat pada Gambar 3.20 berikut.



Gambar 3.20 Pengujian Metode Keserupaan Diagram Kasus Penggunaan

Pada pengujian metode ini terdapat dua variabel, yaitu variabel bebas dan variabel terikat. Variabel bebas pada pengujian ini yaitu bobot perhitungan dan variabel terikatnya yaitu nilai kesepakatan para pakar. Pengujian dilakukan secara

berulang hingga mendapatkan komposisi kesepakatan pakar yang terbaik (*substantial agreement*) menurut Landis & Koch (Landis and Koch 2012).

3.7. Laporan Penelitian

Dokumentasi dan laporan, merupakan tahap akhir dari penelitian ini. Dimana seluruh proses penelitian dan pengujian didokumentasikan dalam bentuk laporan.

3.8. Jadwal Penelitian

Tabel 3.11 Rencana Jadwal Kegiatan Penelitian

No.	Kegiatan	Bulan																					
		I			II			III			IV												
1.	Studi Literatur	■	■	■	■																		
2.	Perancangan dan Implementasi Metode			■	■	■	■																
3.	Pengujian						■	■	■	■													
4.	Pengukuran Performa dan Analisis									■	■	■	■	■									
5.	Penyusunan Laporan Penelitian									■	■	■	■	■	■	■	■	■	■	■	■	■	■

BAB 4

HASIL DAN PEMBAHASAN

Pada pembahasan ini diberikan pemaparan mengenai implementasi sistem serta pengujian dari sistem berdasarkan skenario yang telah dirancang pada Bab Tiga. Proses implementasi dilakukan berdasarkan tahapan yang telah diberikan pada pembahasan sebelumnya. Selanjutnya pengujian sistem dilakukan dengan beberapa kondisi yang disesuaikan dengan skenario pengujian. Dari hasil pengujian yang telah didapatkan, selanjutnya diberikan pembahasan dan analisa dari setiap pengujian yang dilakukan dengan tujuan untuk mendapatkan hasil dari penelitian, sehingga mendapatkan kesimpulan yang diberikan pada pembahasan selanjutnya.

4.1. Implementasi Sistem

Dalam penelitian ini system untuk implementasi dan ujicoba akan dibangun dalam lingkungan pengembangan sebagai berikut:

Sistem operasi	: MacOS Catalina 10.15.7
RAM	: 4
Processor	: 2.5 GHz Dual-Core Intel Core i5
IDE	: Apache Netbeans 12.0
Library	: Stanford corenlp 3.9.2, ws4j 1.01 (<i>wordnet</i> dan <i>wuPalmer</i>).

Untuk langkah implementasi sistem secara detail dijelaskan pada sub bab berikut.

4.1.1. Akumulasi Data Para Pakar

Penilaian oleh para pakar didapatkan dengan melakukan penyebaran kuesioner. Kuesioner dibuat dengan menyajikan sepasang diagram jawaban mahasiswa dan kunci jawaban. Kuesioner tersebut disertai dengan deskripsi soal yang diberikan pada mahasiswa. Kriteria ahli yang diminta untuk melakukan penilaian yaitu dosen khusus yang mengajar mata kuliah di bidang Rekayasa Perangkat Lunak dan rata-rata telah mengajar minimal 1 tahun. Selain itu, para ahli telah menilai jawaban mahasiswa mengenai diagram kasus penggunaan di kelas yang telah diajarkan. Para ahli ini berasal dari beberapa Universitas. Responden dari kuesioner didapatkan sebanyak 22 ahli.

Hasil penilaian oleh para pakar yang telah terkumpul digunakan sebagai standar emas. Standar emas merupakan tolak ukur kehandalan dari metode yang diusulkan dalam penelitian ini. Intrepetasi kesepakatan sesuai dengan Tabel 2.3. Para ahli diminta melakukan penilaian terhadap pasangan diagram yang disajikan dengan rentang nilai 10 sampai 100. Nilai 10 adalah ketika diagram jawaban mahasiswa dan diagram kunci jawaban sangat berbeda, sedangkan nilai 100 ketika diagram jawaban mahasiswa dan diagram kunci jawaban sangat serupa. Untuk memudahkan perbandingan dengan metode penilaian yang diusulkan, maka penilaian oleh para pakar akan diubah ke dalam skala linker. Kisaran nilai skala linker meliputi 1 yaitu antara 10 sampai 19, 2 yaitu antara 20 sampai 39, 3 yaitu antara 40 sampai 59, 4 yaitu antara 60 sampai 79, dan 5 yaitu antara 80 sampai 100. Nilai 1 yaitu menyatakan bahwa pasangan diagram yang dibandingkan sangat berbeda dan nilai 5 apabila pasangan diagram yang dibandingkan serupa.

4.1.2. Deskripsi Diagram

Jumlah diagram yang digunakan pada penelitian ini yaitu 36 diagram kasus penggunaan. Secara lengkap, diagram yang didapatkan dapat dilihat pada Tabel 3.1. Diagram tersebut di dapatkan dari mahasiswa yang sedang menempuh mata kuliah Rekayasa Perangkat Lunak dan telah mendapatkan materi mengenai diagram UML, khususnya diagram kasus penggunaan. Diagram dibuat dengan menggunakan Star UML.

Sebelum dilakukan perhitungan keserupaan diagram, diagram yang telah terkumpul di translasi ke dalam format XMI. Contoh translasi diagram menjadi format XMI dapat dilihat pada Gambar 3.2. Dalam format XMI tersebut yang akan menjadi input program untuk melakukan perhitungan keserupaan diagram. Dari input tersebut akan didapatkan informasi mengenai semantik dan struktural dari diagram. Informasi tersebut meliputi aktor, kasus penggunaan, dan relasi dari diagram kasus penggunaan.

4.1.3. Keserupaan Semantik

4.1.3.1. Diagram Pre-processing Semantik

Dalam pre-processing semantik, diagram kasus penggunaan dalam format XMI selanjutnya dirubah ke dalam bentuk graf. Salah satu contoh translasi dari

format XMI dapat dilihat pada Gambar 3.13. Graf tersebut disertai dengan informasi leksikal terhadap diagram yang meliputi nama aktor, kasus penggunaan, dan relasi. Salah satu contoh informasi leksikal pada graf hasil translasi dari format XMI dapat dilihat pada Tabel 3.1. Informasi leksikal tersebut yang akan dihitung keserupaannya antar diagram kasus penggunaan.

4.1.3.2. Perhitungan Keserupaan Semantik

Perhitungan keserupaan semantik dilakukan dengan menggunakan metode TACSim. Langkah yang dilakukan setelah dilakukan pre-processing terhadap diagram dengan mengubah diagram berformat XMI ke dalam bentuk graf yang memiliki informasi leksikal yaitu dilakukan perhitungan keserupaan setiap simpul dan tepian. Keserupaan dihitung menggunakan library Wordnet dan Wupalmer. Perhitungan keserupaan tersebut telah dijelaskan pada Bab 3. Setiap nilai keserupaan kemudian ditulis dalam bentuk matriks. Matriks yang dibentuk ada dua jenis, yaitu matriks untuk nilai keserupaan antar simpul dan matriks untuk nilai keserupaan antar tepian. Setelah matriks terbentuk, masing-masing matriks dihitung keserupaan yang optimal dari kumpulan nilai keserupaan tersebut.

Untuk mendapatkan keserupaan yang optimal digunakan algoritma Hungarian maksimasi. Langkah-langkah perhitungan keserupaan yang optimal tersebut telah dijelaskan pada bab 3. Setelah di dapatkan keserupaan yang optimal dari matriks simpul dan matriks tepian, nilai keserupaan optimal keduanya dijumlahkan dengan bobot masing-masing untuk mendapatkan keserupaan semantik secara keseluruhan antara kedua diagram.

4.1.4. Keserupaan Struktural

4.1.4.1. Digaram Pre-processing Struktural

Dalam pre-processing struktural, diagram kasus penggunaan dalam format XMI selanjutnya juga akan dirubah ke dalam bentuk graf. Salah satu contoh translasi dari format XMI dapat dilihat pada Gamabr 3.13. Namun graf tersebut hanya disertai simbol yang membedakan antara aktor, use case, dan relasi diagram. Pada perhitungan keserupaan struktural akan mengabaikan informasi leksikal setiap simpul dan tepiannya dan hanya berfokus pada strukturalnya. Struktur graf pada diagram kasus penggunaan dibedakan menjadi dua yaitu inter-struktural dan intra-

struktural. Inter-struktural yaitu *main* struktural dari diagram kasus penggunaan dan intra-struktural yaitu spesifikasi dari diagram kasus penggunaan.

4.1.4.2. Perhitungan Kekerupaan Inter-struktural

Pada inter-struktural terdiri dari hanya satu graf. Jadi pada perhitungan kekerupaan inter-struktural hanya membandingkan dua buah graf dari main kasus penggunaan masing-masing diagram. Setelah dilakukan pengolahan input diagram dari format XMI yang menjadi graf, pada inter-struktural dilakukan pembentukan matrix C sesuai dengan Persamaan (3.5) yang telah dijelaskan pada Bab 3. Pembentukan Matrix dilakukan dengan pembentukan *cost* yang ditulis pada setiap kuadrannya, yaitu kuadran I, II, III, dan IV. Setelah itu melakukan pencarian *cost* terendah dengan menggunakan metode GEG Greedy. Ketika *cost* terendah didapatkan selanjutnya dilakukan perhitungan kekerupaan antara kedua diagram kasus penggunaan terhadap *cost* minimum yang didapatkan.

4.1.4.3. Perhitungan Kekerupaan Intra-struktural

Pada intra-struktural terdiri dari beberapa graf sesuai dengan jumlah kasus penggunaan suatu diagram, yaitu bisa jadi jumlahnya lebih dari satu. Hal tersebut karena intra-struktural merupakan spesifikasi dari setiap kasus penggunaan pada diagram. Langkah-langkah perhitungan kekerupaan pada intra-struktural hampir sama dengan perhitungan kekerupaan inter-struktural, namun pada intra-struktural terdapat langkah untuk mencari nilai optimal kekerupaan dari masing-masing pasangan graf dari diagram pertama dan kedua yang telah dihitung kekerupaannya. Langkah pertama sebelum dilakukan perhitungan kekerupaan antara dua buah graf kasus penggunaan, dilakukan pembentukan matrix C sesuai dengan Rumus () yang telah dijelaskan pada Bab 3. Cost matriks tersebut terdiri dari Kuadran I, II, III, dan IV. Setelah itu melakukan pencarian *cost* terendah dengan menggunakan metode GED Greedy. Setelah *cost* terendah didapatkan selanjutnya dilakukan pencarian nilai *cost* minimum yang optimal dari setiap pasangan graf yang dibandingkan. Setelah didapatkan nilai *cost* minimum yang optimal, dilakukan perhitungan kekerupaan intra-struktural dari *cost* minimum yang didapatkan.

4.1.4.4. Penggabungan Keserupaan Inter-struktural dan Intra-struktural

Setelah didapatkan nilai keserupaan inter-struktural dan intra-struktural, kemudian dilakukan penggabungan kedua nilai tersebut. Kedua nilai keserupaan tersebut ditambahkan dengan bobot masing-masing. Bobot tersebut didapatkan dari kesepakatan penilaian para pakar yang telah dikumpulkan dan dihitung nilainya menggunakan Gwet's AC. Hal tersebut bertujuan agar penilaian dengan menggunakan metode yang diusulkan pada penelitian ini menjadi reliable dengan kesepakatan para pakar.

4.2. Hasil Pengujian dan Analisis

Untuk mengevaluasi hasil dari metode usulan dalam penelitian ini, dilakukan pengujian metode usulan dengan membandingkan antara hasil penilaian dari usulan metode dan penilaian oleh para pakar. Penilaian para pakar sangat berbeda-beda, maka dari itu dilakukan perhitungan nilai kesepakatan dengan menggunakan Gwet's AC1. Untuk hasil pengujian dan analisis terhadap metode usulan dijelaskan dengan detail pada sub bab berikut.

4.2.1. Hasil Pengujian

Hasil dari implementasi program yang melakukan perhitungan keserupaan menggunakan metode Graph Edit Distance dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Hasil keserupaan diagram dari implementasi program

	Kombinasi bobot inter-struktural dan intra-struktural										
	1	2	3	4	5	6	7	8	9	10	11
Rata-rata	61,93	61,44	60,95	60,46	59,97	59,49	59,00	58,51	58,02	57,53	57,05
Min	17,14	20,67	24,21	27,75	31,29	34,82	37,73	37,07	35,97	34,88	33,78
Median	68,10	66,24	64,78	63,70	63,18	62,51	61,58	60,64	58,27	57,67	58,23
Max	95,10	91,76	88,85	85,95	83,04	81,34	80,44	79,55	78,66	77,77	76,88
Stdev	23,38	21,58	19,83	18,15	16,56	15,08	13,77	12,65	11,80	11,27	11,11

Pada Tabel 4.1 dapat dilihat hasil keserupaan terhadap diagram kunci jawaban dan jawaban mahasiswa dalam beberapa kombinasi bobot antara keserupaan semantik dan keserupaan struktural. Ditampilkan kombinasi dari 1 hingga 11. kombinasi 1 yaitu bobot keserupaan semantik sebesar 0 dan bobot keserupaan struktural sebesar 1. Untuk kombinasi 2 yaitu bobot keserupaan

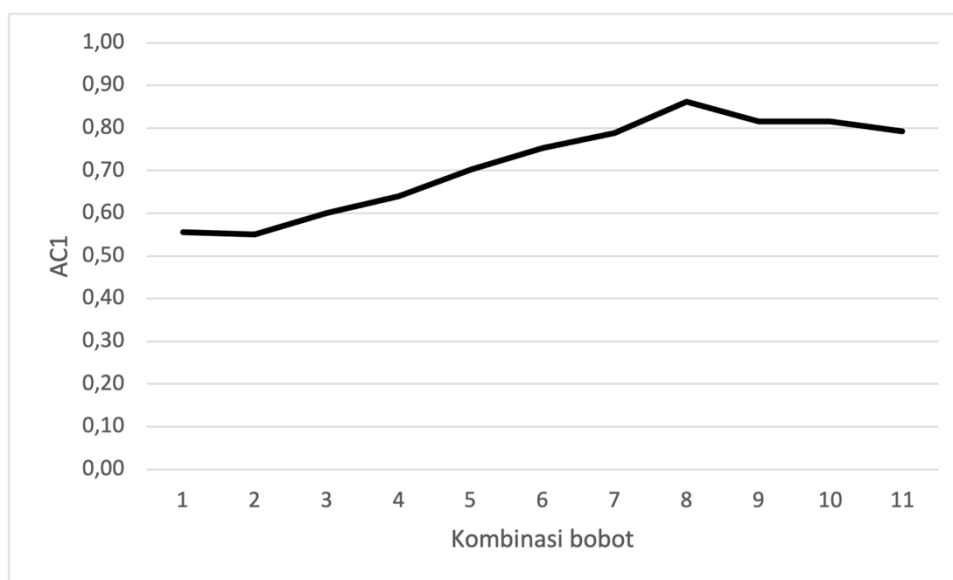
semantik sebesar 0,1 dan bobot keserupaan struktural sebesar 0,9. Kombinasi 2 yaitu bobot keserupaan semantik sebesar 0,2 dan bobot keserupaan struktural sebesar 0,8. Kombinasi 3 yaitu bobot keserupaan semantik sebesar 0,3 dan bobot keserupaan struktural sebesar 0,7. Dan kombinasi selanjutnya hingga kombinasi 11 yaitu bobot keserupaan semantik sebesar 1 dan bobot keserupaan struktural sebesar 0.

Hasil penilaian dalam beberapa kombinasi sangat bervariasi. Rata-rata penilaian terendah didapatkan dari kombinasi 11 dengan nilai 57,05 dan rata-rata penilaian tertinggi didapatkan dari kombinasi 1. Minimal penilaian terendah didapatkan dari kombinasi 1 dengan nilai 17,14 dan minimal penilaian tertinggi didapatkan dari kombinasi 7 dengan nilai 37,73. Median penilaian terendah didapatkan dari kombinasi 10 dengan nilai 57,67 dan median penilaian tertinggi didapatkan dari kombinasi 1 dengan nilai 68,10. Maksimal penilaian terendah didapatkan dari kombinasi 11 dengan nilai 76,88 dan maksimal penilaian tertinggi didapatkan dari kombinasi 1 dengan nilai 95,10. Standart deviasi penilaian terendah didapatkan dari kombinasi 11 dengan nilai 11,11 dan standart deviasi penilaian tertinggi didapatkan dari kombinasi 1 dengan nilai 23,38.

4.2.2. Hasil Kesepakatan Pakar

Setelah penilaian keserupaan oleh para ahli dikumpulkan, kesepakatan para ahli dibuat untuk penilaian tersebut. Hal ini dilakukan karena beberapa penilaian oleh para ahli berbeda-beda. Nilai kesepakatan para ahli dihitung menggunakan Gwet's AC1. Hasil kesepakatan pakar digunakan dalam menentukan bobot dalam perhitungan keserupaan semantik, struktural dan keserupaan gabungan antara keduanya. Pada perhitungan keserupaan semantik digunakan untuk menentukan bobot antara kerupaan simpul dan tepian. Dengan menggunakan perhitungan Gwet's AC1, diperoleh nilai AC1 tertinggi yaitu 0,79 dengan bobot keserupaan simpul sebesar 0,6 dan bobot keserupaan tepian sebesar 0,4. Sedangkan pada perhitungan keserupaan struktural digunakan untuk menentukan bobot antara keserupaan inter-struktural dan intra-struktural. Dengan menggunakan perhitungan Gwet's AC1, diperoleh nilai AC1 tertinggi yaitu 0,56 dengan bobot keserupaan inter-struktural sebesar 0,4 dan bobot intra-struktural sebesar 0,6.

Selain digunakan untuk menentukan bobot pada perhitungan semantik dan struktural, nilai kesepakatan para pakar juga digunakan dalam menentukan bobot dalam menggabungkan kedua perhitungan keserupaan tersebut. Hasil pengukuran Gwet's AC1 terhadap bobot keserupaan semantik dan struktural dapat dilihat pada Gambar 4.1 berikut.

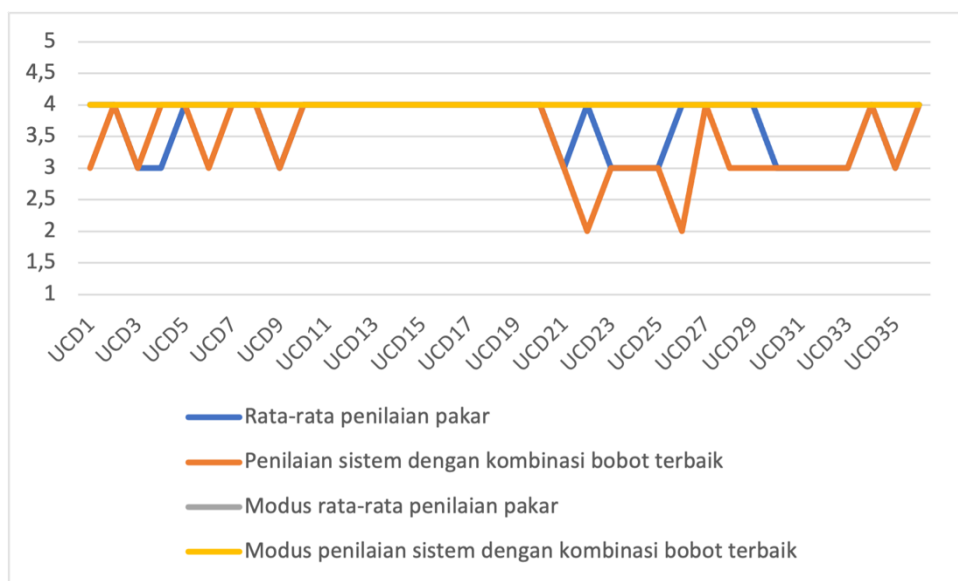


Gambar 4.1 Kesepakatan pakar menggunakan Gwet's AC1

Dari Gambar 4.1 dituliskan kombinasi 1 yaitu bobot keserupaan semantik sebesar 0 dan bobot keserupaan struktural sebesar 1. Untuk kombinasi 2 yaitu bobot keserupaan semantik sebesar 0,1 dan bobot keserupaan struktural sebesar 0,9. Kombinasi 2 yaitu bobot keserupaan semantik sebesar 0,2 dan bobot keserupaan struktural sebesar 0,8. Kombinasi 3 yaitu bobot keserupaan semantik sebesar 0,3 dan bobot keserupaan struktural sebesar 0,7. Dan kombinasi selanjutnya hingga kombinasi 11 yaitu bobot keserupaan semantik sebesar 1 dan bobot keserupaan struktural sebesar 0. Dari gambar tersebut, dapat kita lihat bahwa nilai AC1 terbesar didapatkan pada kombinasi 8 yang berarti bobot semantik sebesar 0,7 dan bobot keserupaan struktural sebesar 0,3 dengan nilai 0,86. Jadi kombinasi yang dapat diterapkan pada sistem sesuai dengan nilai kesepakatan tertinggi yaitu kombinasi 8.

4.2.3. Analisis Hasil

Setelah diperoleh kesepakatan mengenai kombinasi bobot terbaik dengan para pakar, maka dilakukan perhitungan keserupaan diagram kasus penggunaan sesuai dengan kombinasi bobot terbaik. Berdasarkan Gambar 4.1 diperoleh kombinasi 8 sebagai kombinasi terbaik dengan bobot keserupaan semantik sebesar 0,7 dan bobot keserupaan struktural sebesar 0,3. Hasil perhitungan keserupaan diagram kasus penggunaan dengan menggunakan kombinasi terbaik dapat dilihat pada Gambar 4.2.



Gambar 4.2 Hasil penilaian pasangan diagram

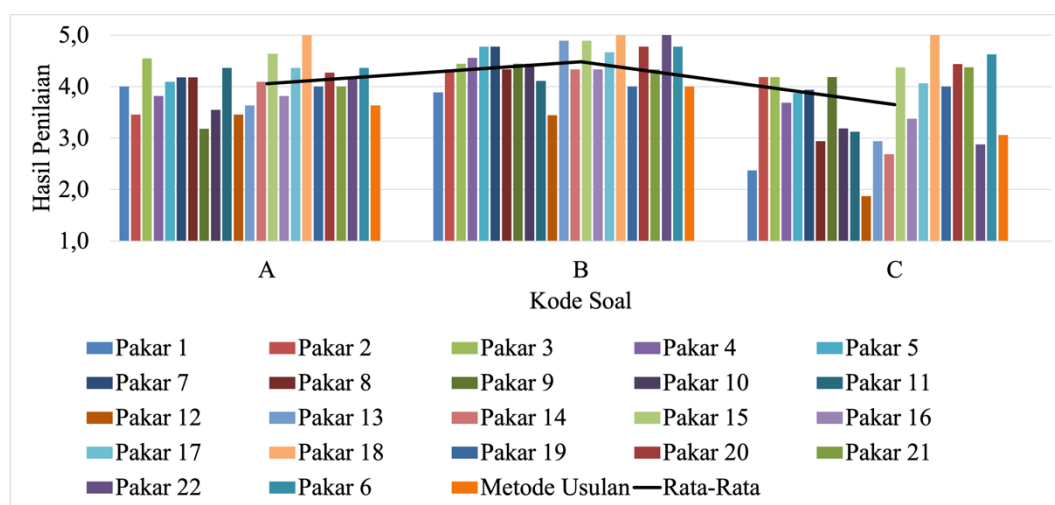
Pada Gambar 4.2 menunjukkan bahwa rata-rata nilai gap antara penilaian oleh para pakar dan sistem dengan menggunakan metode usulan sebesar 0,25. Kombinasi dengan bobot terbaik memberikan nilai gap antara keduanya menjadi kecil. Beberapa penilaian mendapatkan nilai gap yang tinggi. Dalam skala linker gap terbesarnya yaitu 2, pada UCD23 dan UCD27. Hal tersebut dikarenakan mungkin karena beberapa pakar mengabaikan arah relasi pada diagram. Namun, dalam metode yang diusulkan arah relasi dari diagram juga dipertimbangkan. Hal tersebut sangat berpengaruh dalam penilaian diagram kasus penggunaan.

Penelitian ini menguji sebanyak 36 jawaban diagram kasus penggunaan terhadap 3 kunci jawaban diagram kasus penggunaan. Metode yang diusulkan tidak dipengaruhi oleh banyaknya jumlah simpul pada diagram. Namun pada penelitian

ini, jumlah simpul dan relasi yang telah diuji masih terbatas. Banyaknya jumlah aktor, kasus penggunaan, relasi, dan kata unik pada diagram telah disebutkan pada Tabel 3.1 di Bab sebelumnya.

Untuk membuktikan kehandalan metode yang diusulkan, dilakukan perbandingan antara penilaian oleh sistem dengan penilaian oleh para pakar. Reliabilitas penilaian metode ini telah dibandingkan dengan bobot gabungan terbaik dengan penilaian para pakar yang telah dihitung menggunakan metode Gwet's AC1. Nilai kesepakatan antara penilaian dengan metode usulan dan penilaian oleh para pakar diperoleh sebesar 0,86. Menurut Landis & Koch (Landis and Koch 2012), nilai ini menunjukkan kesepakatan yang hampir sempurna (*almost perfect agreement*). Oleh karena itu, penilaian dengan metode yang diusulkan dapat menggantikan penilaian pakar dalam menilai jawaban diagram kasus penggunaan mahasiswa.

Selain itu, kita juga dapat melihat pada gambar tersebut bahwa modus dari hasil penilaian oleh para pakar dan sistem dengan metode usulan sama yaitu 4. Ini menunjukkan bahwa metode yang diusulkan untuk melakukan penilaian diagram kasus penggunaan sama dengan penilaian oleh para pakar. Salah satu tujuan dari metode yang diusulkan ini adalah untuk mengusulkan metode penilaian kasus penggunaan yang handal dan reliabel dengan para ahli. Hal tersebut telah dibuktikan.



Gambar 4.3 Perbandingan Penilaian Pakar dan Metode Usulan

Dari Gambar 4.3 menunjukkan bahwa pakar menilai diagram jawaban siswa dengan hasil yang bervariasi. Penilaian oleh para pakar telah diubah menjadi skala linker. Rata-rata penilaian tertinggi terhadap kode soal A yaitu oleh pakar 18 dengan skala linker sebesar 5, sedangkan yang terendah oleh pakar 9 yaitu sebesar 3,2. Rata-rata penilaian tertinggi terhadap kode soal B yaitu oleh pakar 18 dan 22 yaitu sebesar 5, sedangkan terendah oleh pakar 12 dengan nilai 3,4. Rata-rata penilaian tertinggi terhadap kode soal C yaitu oleh pakar 18 dengan nilai 5, sedangkan terendah oleh pakar 12 dengan nilai 1,9. Hasil penilaian dari metode yang diusulkan memiliki selisih yang paling rendah dengan nilai rata-rata dari para pakar. Dari kode soal A selisihnya yaitu 0,4, kode soal B selisihnya yaitu 0,5, dan dari kode soal C selisihnya yaitu 0,6. Hal tersebut menunjukkan bahwa metode yang diusulkan memiliki penilaian yang dapat diandalkan seperti para pakar, karena rata-rata pakar tersebut menjadi standar emas pada penelitian ini.

Pada kode soal A, rata-rata penilaian oleh pakar 9 dan pakar 18 memiliki selisih yang besar yaitu 1,5. Pada kode soal B, rata-rata penilaian oleh pakar 12 dan pakar 22 memiliki selisih yang besar juga yaitu 1,6. Pada kode soal C, rata-rata penilaian oleh pakar 12 dan pakar 18 memiliki selisih yang sangat besar yaitu 3,1. Hal tersebut menunjukkan bahwa para pakar dalam melakukan penilaian terhadap diagram kasus penggunaan sangat bervariasi. Penilaian para pakar terhadap diagram kasus penggunaan sangat bervariasi, mungkin karena kelelahan dalam melakukan penilaian diagram yang sangat banyak. Untuk mengantisipasi hal tersebut mungkin dalam melakukan penilaian terhadap diagram, soal yang diberikan kepada ahli dapat dikurangi. Hal lain yang dapat dilakukan yaitu mungkin dengan menambah jumlah penilai atau pakar.

BAB 5

KESIMPULAN DAN SARAN

Pada bab terakhir ini, ditarik beberapa kesimpulan yang didapat dari hasil penelitian dan saran-saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan atau riset selanjutnya.

5.1. Kesimpulan

Keserupaan antara dua buah kasus penggunaan dapat dihitung berdasarkan dua aspek diagram, yaitu semantik dan struktural. Keserupaan semantik dihitung menggunakan metode pemrosesan bahasa alami dan *Topology-Attributed Coupling Similarity* (TACSim), sedangkan keserupaan struktural dihitung menggunakan metode *Graph Edit Distance* (GED) dan algoritma *Greedy*. Dalam penelitian ini dapat diambil beberapa kesimpulan.

Kesimpulan pertama, metode pengukuran keserupaan antara dua buah kasus penggunaan dalam penelitian ini dapat diandalkan seperti penilaian pengampu mata pelajaran Rekayasa Perangkat Lunak. Penilaian yang dilakukan yaitu penilaian jawaban siswa terhadap diagram kasus penggunaan.

Kesimpulan kedua, pemodelan diagram kasus penggunaan menjadi graf pada aspek semantik dilakukan dengan mengambil informasi leksikal diagram. Informasi leksikal tersebut terdiri dari nama dan tipe dari setiap aktor, kasus penggunaan, dan relasi diagram kasus penggunaan. Sedangkan pada aspek struktural, pemodelan diagram kasus penggunaan menjadi graf dengan mengambil struktur dari aktor, kasus penggunaan, dan arah relasi dari diagram kasus penggunaan.

Kesimpulan ketiga, bobot untuk semantik sebesar 0,7 dan struktural sebesar 0,3 dengan nilai kesepakatan sebesar 0,86. Nilai tersebut menunjukkan kesepakatan yang hampir sempurna (*almost perfect agreement*).

Kesimpulan keempat, sebelum dilakukan pengukuran keserupaan semantik dilakukan dengan pemodelan diagram menjadi graf yang disertai informasi leksikal. Perhitungan keserupaan semantik dibedakan menjadi dua, yaitu keserupaan simpul dan keserupaan tepian. Keserupaan simpul yaitu menghitung keserupaan antar informasi simpul diagram dan keserupaan tepian yaitu

menghitung antar informasi tepian diagram. Setelah dilakukan perubahan ke dalam bentuk graf yang disertai informasi leksikal, dilakukan pemrosesan bahasa alami untuk mendapatkan informasi leksikal sehingga dapat dilakukan perbandingan informasi leksikal dari masing-masing pasangan graf untuk dihitung keserupaannya dengan menggunakan *Wordnet* dan *Wupalmer*. Kemudian nilai keserupaan ditulis pada matriks. Metode TACSim dilakukan dengan menggunakan algoritma *Hungarian* dalam menemukan nilai keserupaan yang maksimal dari perbandingan graf yang tertulis pada matriks. Setelah didapatkan nilai keserupaan maksimal, seluruh nilai dirata-ratakan sehingga mendapatkan keserupaan semantik pasangan diagram. Untuk mendapatkan keserupaan semantik diagram kasus penggunaan dilakukan penambahan hasil keserupaan simpul dengan bobot 0,6 dan keserupaan tepian dengan bobot 0,4.

Kesimpulan kelima, pengukuran keserupaan struktural sebelumnya dilakukan dengan memodelkan diagram menjadi graf, masing-masing pasangan graf dihitung keserupaannya. Graf pada keserupaan struktural ini dibedakan menjadi inter-struktural dan intra-struktural. Inter-struktural yaitu *main* diagram kasus penggunaan dan intra-struktural yaitu spesifikasi dari diagram kasus penggunaan. Masing-masing dihitung keserupaannya dengan mengabaikan informasi leksikal dari diagram. Metode GED dilakukan dengan cara membentuk matriks biaya untuk melakukan pengubahan, penambahan, dan penghapusan dari graf satu ke graf kedua. Setelah itu dilakukan pemilihan biaya terkecil dengan menambahkan algoritma *Greedy* agar permutasi yang dihasilkan dalam pencarian tidak terlalu besar. Setelah didapatkan biaya terkecil, dilakukan perhitungan keserupaan pasangan diagram. Untuk mendapatkan keserupaan struktural diagram kasus penggunaan dilakukan penambahan hasil keserupaan inter-struktural dengan bobot 0,4 dan keserupaan intra-struktural dengan bobot 0,6.

5.2. Saran

Penelitian ini menggunakan hasil penilaian oleh 22 pengampu mata pelajaran untuk mendapatkan kesepakatan nilai dengan metode yang diusulkan. Namun hasil analisis dari pengujian menunjukkan bahwa penilaian jawaban diagram kasus penggunaan oleh para pengampu mata pelajaran sangat bervariasi. Hal tersebut

dapat terjadi karena kelelahan pengampu mata pelajaran dalam melakukan penilaian terhadap diagram yang sangat banyak. Maka dari itu untuk saran yang dapat diberikan untuk pengembangan lebih lanjut terhadap penelitian ini yaitu dengan mengurangi jumlah soal diagram yang diberikan kepada masing-masing pengampu mata pelajaran dan menambah jumlah pengampu mata pelajaran dalam melakukan penilaian jawaban siswa terhadap diagram kasus penggunaan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Adamu, Alhassan, and Wan Mohd Nazmee Wan Zainon. 2017. "Multiview Similarity Assessment Technique of UML Diagrams." *Procedia Computer Science* 124: 311–18.
- Booch, Grady et al. 2008. "Object-Oriented Analysis and Design with Applications." *ACM SIGSOFT software engineering notes* 33(5): 29.
- Chechik, Marsha, Shiva Nejati, and Mehrdad Sabetzadeh. 2012. "A Relationship-Based Approach to Model Integration." *Innovations in Systems and Software Engineering* 8(1): 3–18.
- Chen, Xiaming et al. 2017. "Discovering and Modeling Meta-Structures in Human Behavior from City-Scale Cellular Data." *Pervasive and Mobile Computing* 40: 464–79.
- Chonoles, Michael Jesse. 2018. "What Is UML?" *OCUP Certification Guide*: 17–41.
<https://www.sciencedirect.com/science/article/pii/B9780128096406000039?via%3Dihub> (April 26, 2020).
- Deo, Narsingh. 2017. *Graph Theory with Applications to Engineering and Computer Science*. Courier Dover Publications.
- Fauzan, Reza, Daniel Siahaan, Siti Rochimah, and Evi Triandini. 2018. "Class Diagram Similarity Measurement: A Different Approach." In *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, , 215–19.
- . 2019. "Use Case Diagram Similarity Measurement: A New Approach." In *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, , 3–7.
- Fauzi, M Ali, Djoko Cahyo Utomo, Budi Darma Setiawan, and Eko Sakti Pramukantoro. 2017. "Automatic Essay Scoring System Using N-Gram and Cosine Similarity for Gamification Based E-Learning." In *Proceedings of the International Conference on Advances in Image Processing*, , 151–55.
- Fowler, Martin. 2004. *UML Distilled: A Brief Guide to the Standard Object*

Modeling Language. Addison-Wesley Professional.

- Gwet, Kilem. 2002. "Kappa Statistic Is Not Satisfactory for Assessing the Extent of Agreement between Raters." *Statistical methods for inter-rater reliability assessment* 1(6): 1–6.
- Jungnickel, Dieter. 2013. *Algorithms and Computation in Mathematics* 5. Fourth Edi. eds. Manuel Bronstein et al. Germany: Springer Science & Business Media.
- Landis, J Richard, and Gary G Koch. 2012. "The Measurement of Observer Agreement for Categorical Data." *biometrics*: 159–74.
- Majumder, Goutam, Partha Pakray, Alexander Gelbukh, and David Pinto. 2016. "Semantic Textual Similarity Methods, Tools, and Applications: A Survey." *Computación y Sistemas* 20(4): 647–65.
- MKLab Co. 2018. "What Is UML | Unified Modeling Language." <http://uml.org/what-is-uml.htm> (April 21, 2020).
- Pramukantoro, Eko Sakti, and M Ali Fauzi. 2016. "Comparative Analysis of String Similarity and Corpus-Based Similarity for Automatic Essay Scoring System on e-Learning Gamification." In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, , 149–55.
- Riesen, Kaspar, Miquel Ferrer, Rolf Dornberger, and Horst Bunke. 2015. "Greedy Graph Edit Distance." In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, , 3–16.
- Sevenster, Merlijn, Jeffrey Bozeman, Andrea Cowhy, and William Trost. 2015. "A Natural Language Processing Pipeline for Pairing Measurements Uniquely across Free-Text CT Reports." *Journal of biomedical informatics* 53: 36–48.
- Spurrier, Gary, and Heikki Topi. 2018. "Resolving the Pedagogical Disconnect between User Stories and Use Cases in Systems Analysis and Design Textbooks."
- Triandini, Evi, Reza Fauzan, Daniel O Siahaan, and Siti Rochimah. 2019. "Sequence Diagram Similarity Measurement: A Different Approach." In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, , 348–51.
- Yuan, Zhongchen, Li Yan, and Zongmin Ma. 2019. "Structural Similarity Measure

between UML Class Diagrams Based on UCG.” *Requirements Engineering*: 1–17.

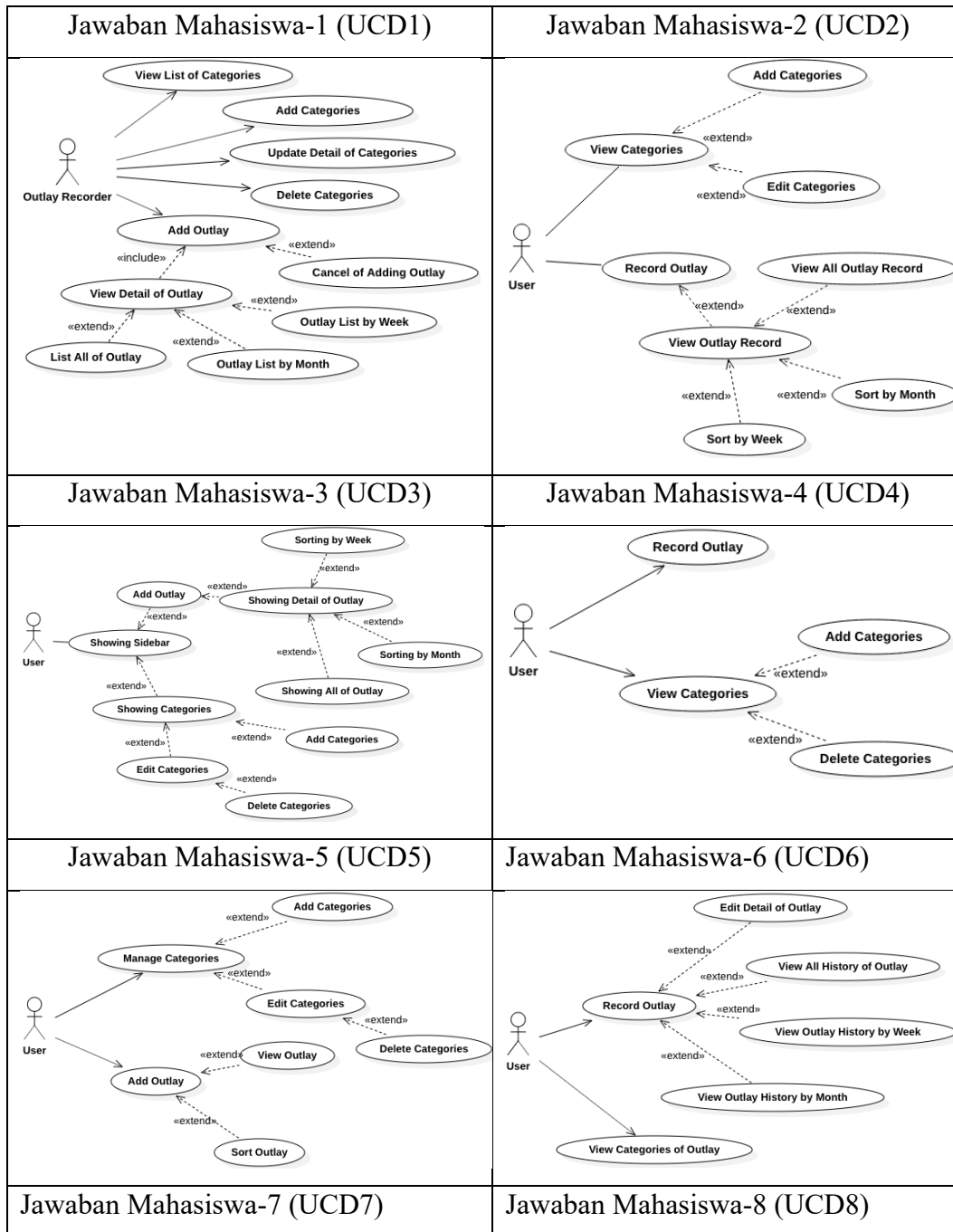
Zulfa, Fatimatus, Daniel Oranova Siahaan, Reza Fauzan, and Evi Triandini. 2020. “Inter-Structure and Intra-Structure Similarity of Use Case Diagram Using Greedy Graph Edit Distance.” In *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, , 1–6.

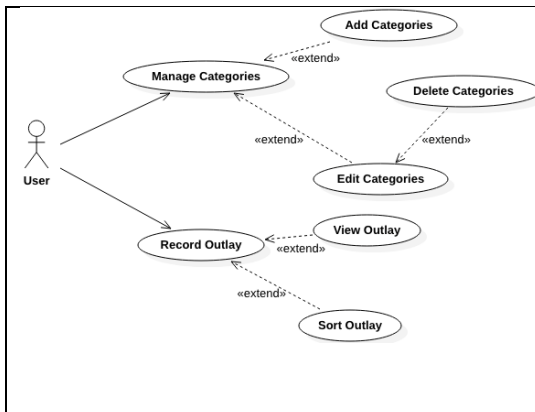
[Halaman ini sengaja dikosongkan]

LAMPIRAN

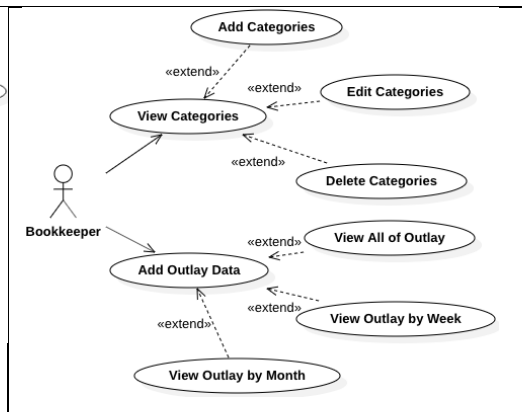
Lampiran 1. Diagram Kasus Penggunaan Jawaban Mahasiswa

Aplikasi Outlay

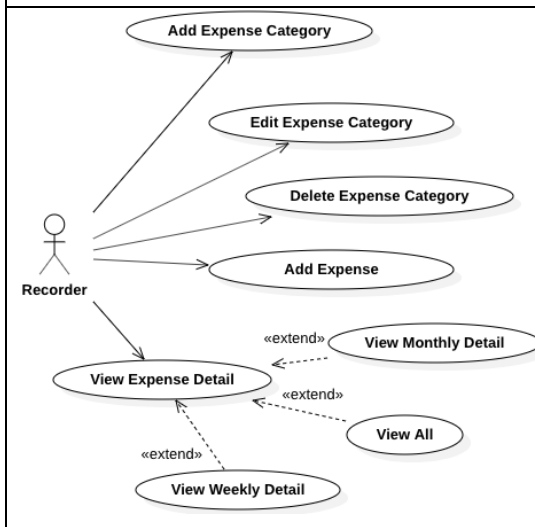




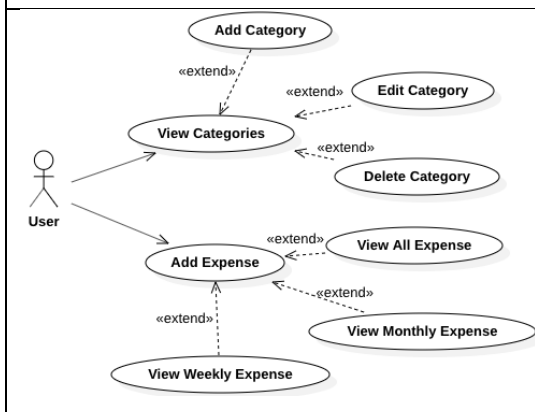
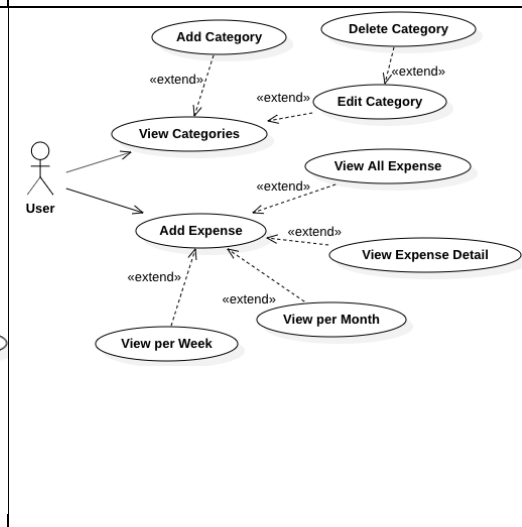
Jawaban Mahasiswa-9 (UCD9)



Jawaban Mahasiswa-10 (UCD10)



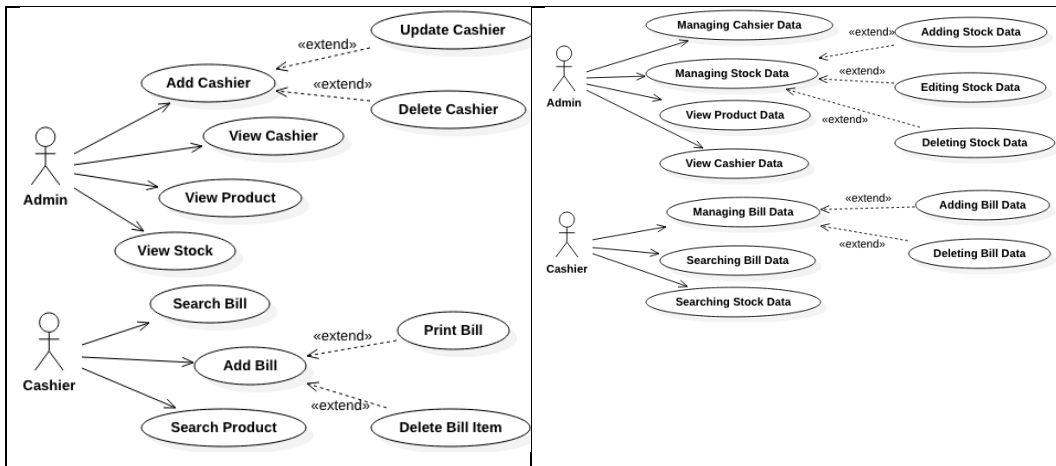
Jawaban Mahasiswa-11 (UCD11)



Aplikasi QuickBill

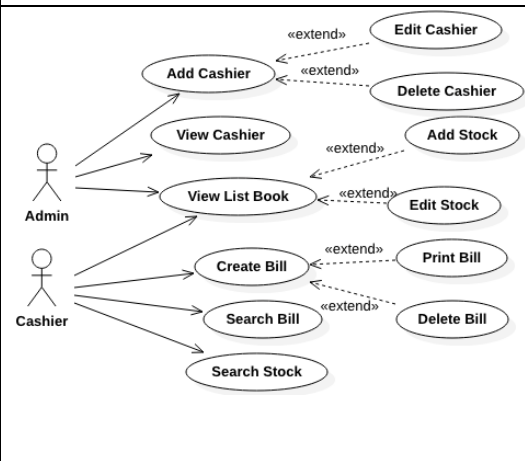
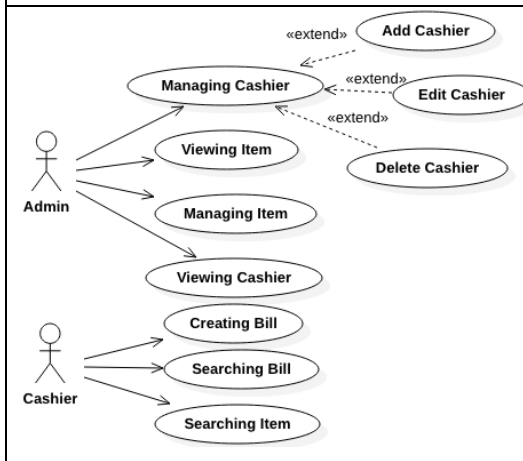
Jawaban Mahasiswa-12 (UCD12)

Jawaban Mahasiswa-13 (UCD13)



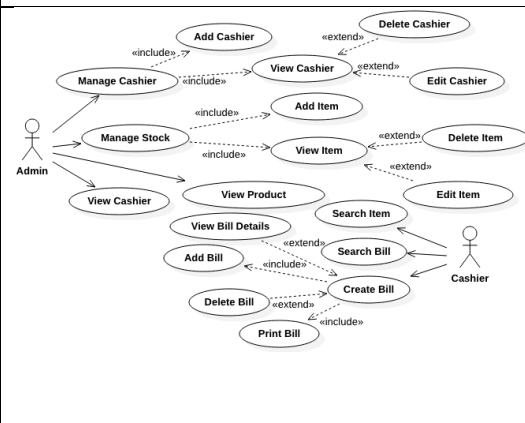
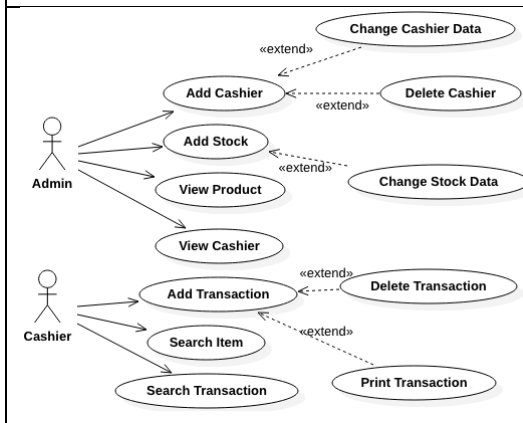
Jawaban Mahasiswa-14 (UCD14)

Jawaban Mahasiswa-15 (UCD15)



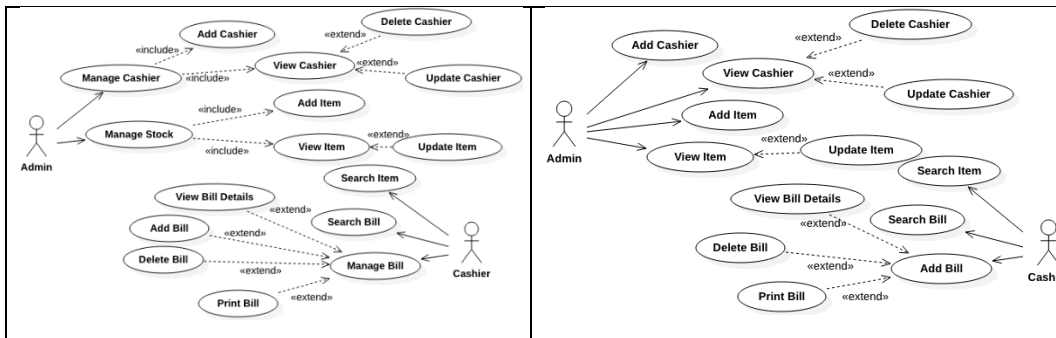
Jawaban Mahasiswa-16 (UCD16)

Jawaban Mahasiswa-17 (UCD17)

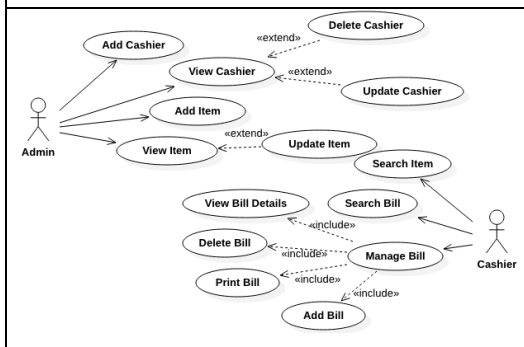


Jawaban Mahasiswa-18 (UCD18)

Jawaban Mahasiswa-19 (UCD19)

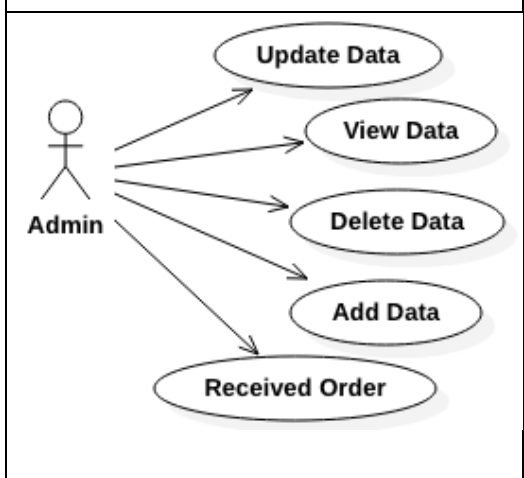


Jawaban Mahasiswa-20 (UCD20)

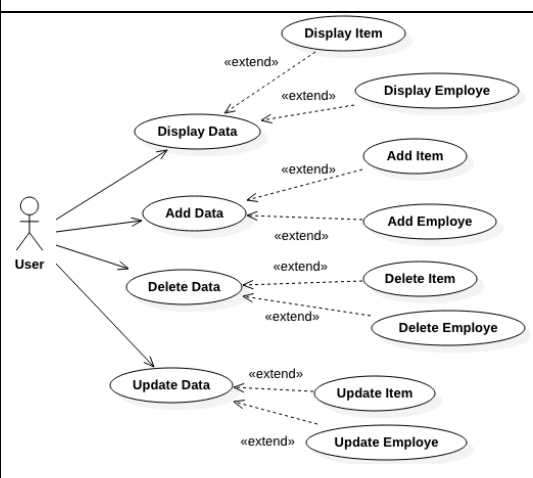


Aplikasi Restaurant Management System

Jawaban Mahasiswa-21 (UCD21)

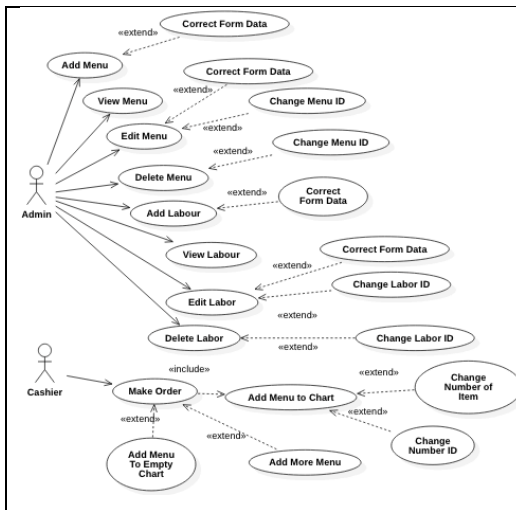


Jawaban Mahasiswa-22 (UCD22)

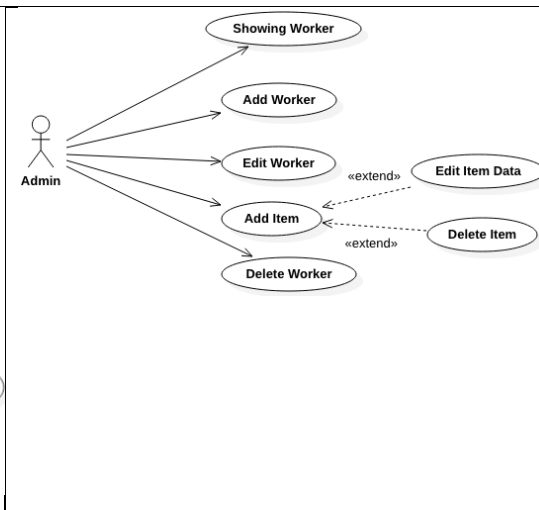


Jawaban Mahasiswa-23 (UCD23)

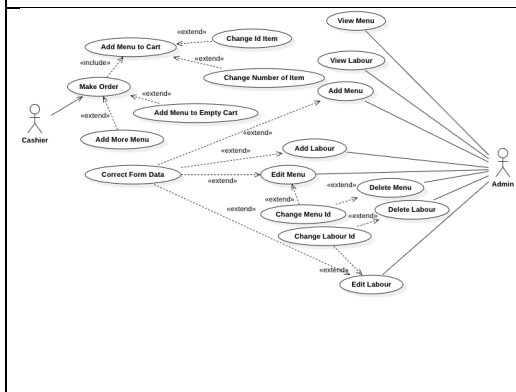
Jawaban Mahasiswa-24 (UCD24)



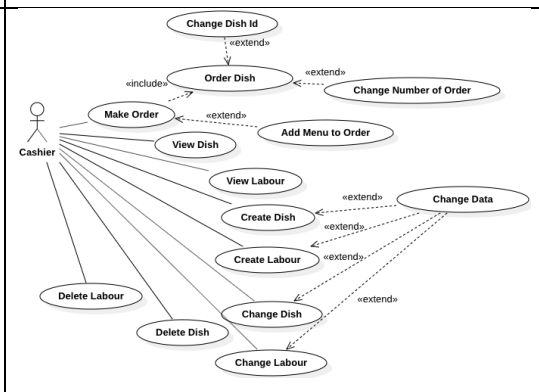
Jawaban Mahasiswa-25 (UCD25)



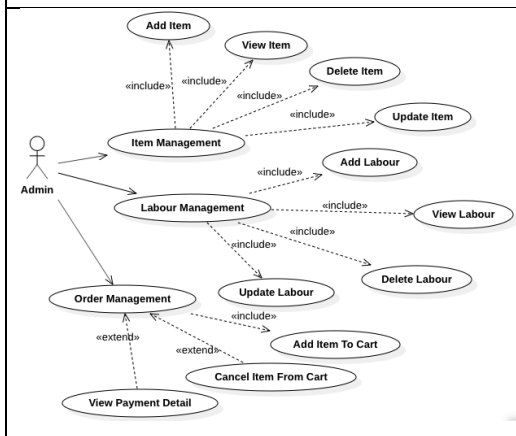
Jawaban Mahasiswa-26 (UCD26)



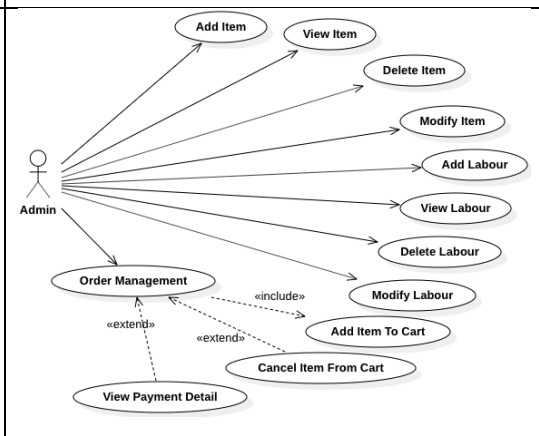
Jawaban Mahasiswa-27 (UCD27)



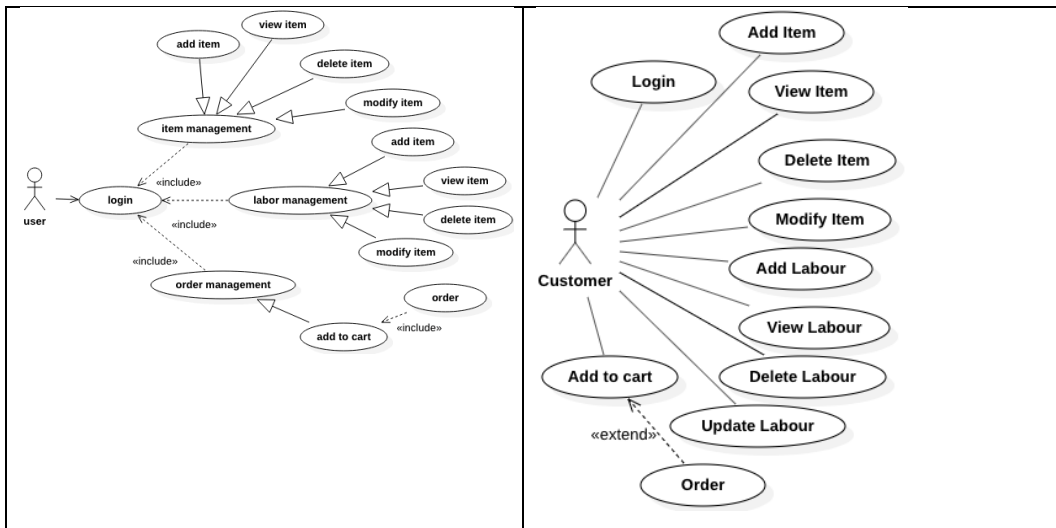
Jawaban Mahasiswa-28 (UCD28)



Jawaban Mahasiswa-29 (UCD29)

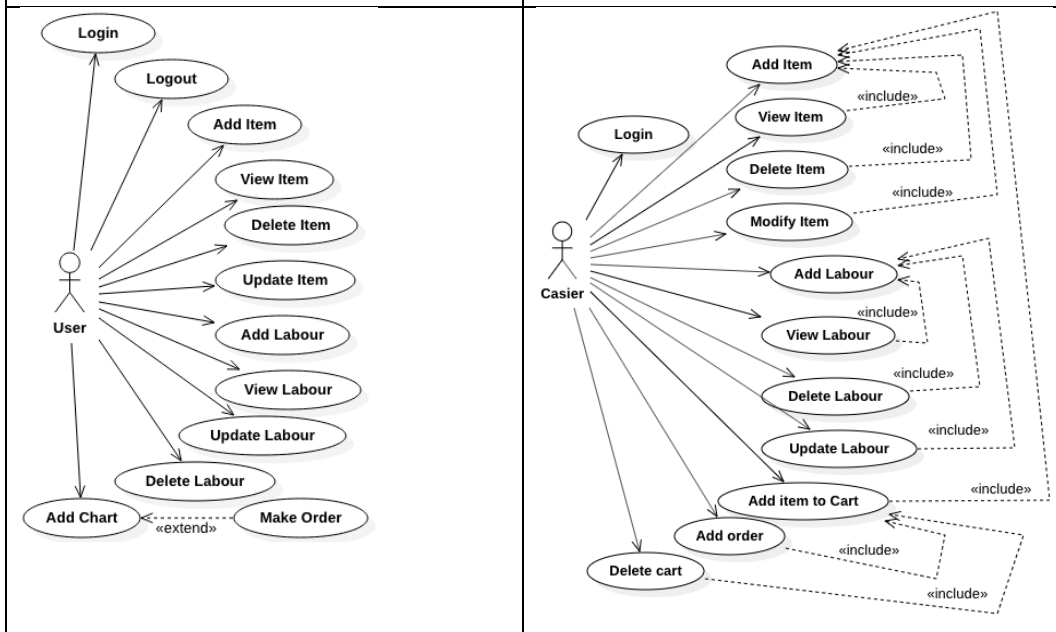


Jawaban Mahasiswa-30 (UCD30)



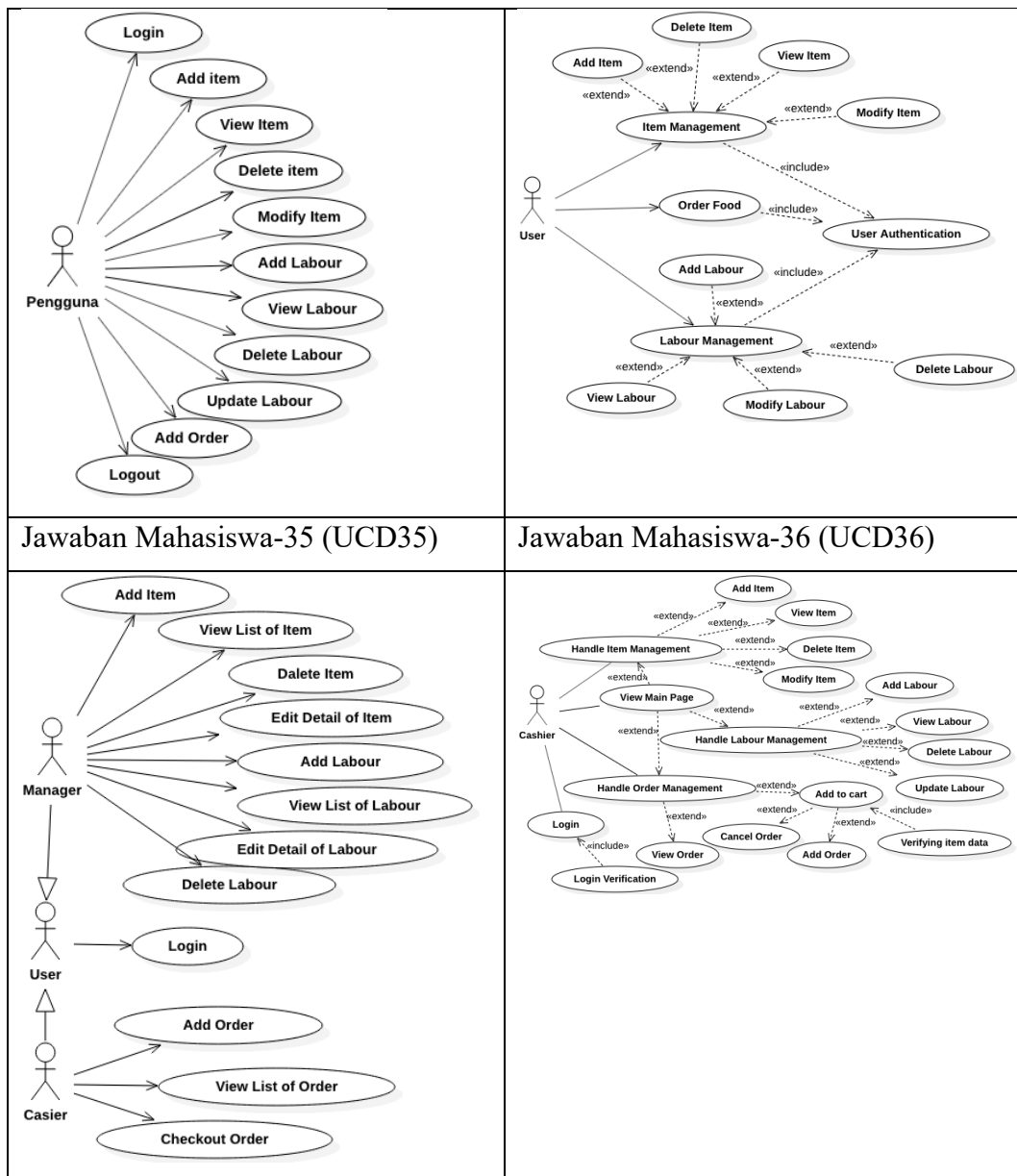
Jawaban Mahasiswa-31 (UCD31)

Jawaban Mahasiswa-32 (UCD32)



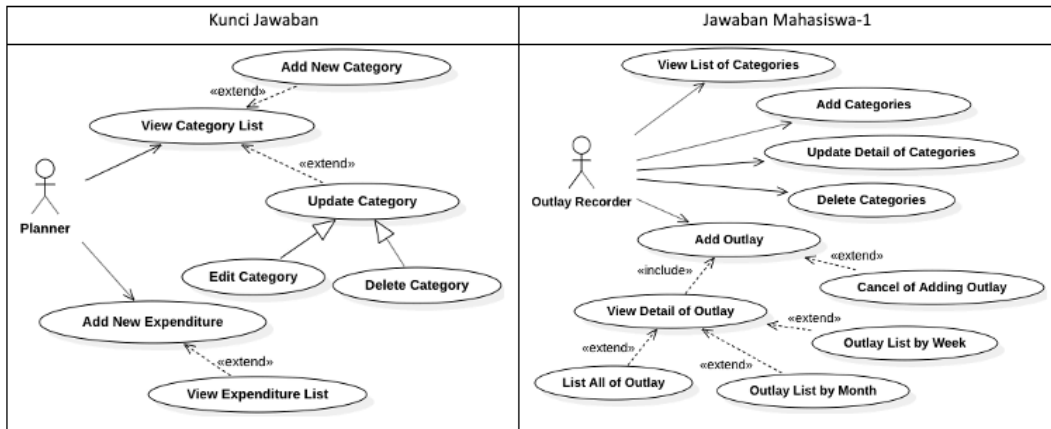
Jawaban Mahasiswa-33 (UCD33)

Jawaban Mahasiswa-34 (UCD34)

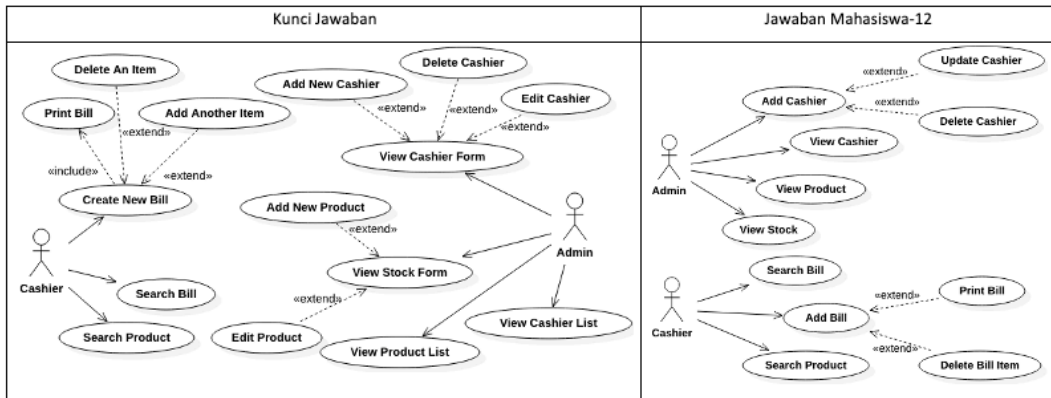


Lampiran 2. Contoh pasangan diagram yang dihitung keserupaannya.

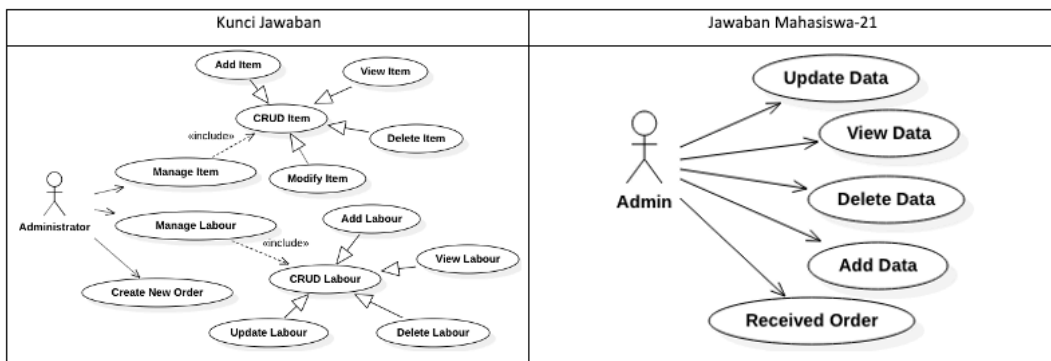
Kunci jawaban (UCDA) dan Jawaban Mahasiswa 1 (UCD1) terhadap aplikasi Outlay



Kunci jawaban (UCDB) dan Jawaban Mahasiswa 12 (UCD12) terhadap aplikasi QuickBill



Kunci Jawaban (UCDC) dan Jawaban Mahasiswa 21 (UCD21) terhadap aplikasi Restaurant Management System



BIOGRAFI PENULIS



Fatimatus Zulfa lahir di Madiun, 11 April 1997. Penulis biasa dipanggil Zulfa. Penulis menempuh Pendidikan formal di SDN Pilangkenceng 1 (2003-2009), SMPN 1 Pilangkenceng (2009-2012), dan SMAN 1 Mejayan (2012-2015). Pada Tahun 2015-2019 penulis melanjutkan studinya di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember. Pada Tahun 2019, penulis melanjutkan Pendidikan Magister S2 di Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember. Di Departemen Teknik Informatika, Rumpun Mata Kuliah (RMK) yang diambil oleh penulis adalah Rekayasa Perangkat Lunak (RPL). Penulis dapat dihubungi melalui alamat surel fatimatuszulfa@gmail.com.