



TESIS - KI142502

# Pendekatan Kesamaan Semantik dan Struktur dalam Kasus Penggunaan untuk Mendapatkan Kembali Spesifikasi Kebutuhan Perangkat Lunak

Ferdika Bagus Pristiawan Permana  
NRP 5114 201 030

DOSEN PEMBIMBING  
Daniel Oranova Siahaan, S.Kom, M.Sc., P.D.Eng  
NIP 197411232006041001

PROGRAM MAGISTER  
BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2016



THESIS - KI142502

# Software Requirement Retrieval Using Use Case Structure and Semantic Similarity Computation

Ferdika Bagus Pristiawan Permana  
NRP 5114 201 030

SUPERVISOR

Daniel Oranova Siahaan, S.Kom, M.Sc., P.D.Eng  
NIP 197411232006041001

MAGISTER PROGRAM  
SOFTWARE ENGINEERING  
INFORMATICS ENGINEERING DEPARTMENT  
INFORMATION TECHNOLOGY FACULTY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2016

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

**FERDIKA BAGUS PRISTIAWAN PERMANA**

Nrp. 5114201030

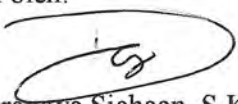
Dengan judul :

**Pendekatan Kesamaan Semantik dan Struktur dalam Kasus Penggunaan Untuk Mendapatkan Kembali Spesifikasi Kebutuhan Perangkat Lunak**

Tanggal Ujian : 20-6-2016

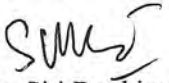
Periode Wisuda : 2015 Genap

Disetujui oleh:

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

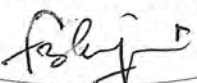
NIP. 197411232006041001

(Pembimbing 1)

  
Dr. Ir. Siti Rochimah, M.T

NIP. 196810021994032001

(Penguji 1)

  
Fajar Baskoro, S.Kom, M.T

NIP. 197404031999031002

(Penguji 2)

  
Rizky Januar Akbar, S.Kom, M.Eng

NIP. 198701032014041001

(Penguji 3)

Direktur Program Pasca Sarjana,



  
Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.

NIP. 196012021987011001

# **Pendekatan Kesamaan Semantik dan Struktur dalam Kasus Penggunaan untuk Mendapatkan Kembali Spesifikasi Kebutuhan Perangkat Lunak**

Nama mahasiswa : Ferdika Bagus Pristiawan Permana  
NRP : 5114 201 030  
Pembimbing : Daniel Oranova Siahaan, S.Kom, M.Sc., P.D.Eng

## **ABSTRAK**

Di dalam pengembangan perangkat lunak berskala besar, terdapat jumlah dokumen kebutuhan perangkat lunak yang sangat banyak dalam sekali proses elisitasi yang mungkin dihasilkan untuk domain yang berbeda dari setiap tim pengembang. Dokumen-dokumen ini mungkin digunakan kembali untuk mengurangi biaya dan waktu guna pengembangan perangkat lunak berikutnya. Oleh karena itu dibutuhkan suatu mekanisme untuk mendapatkan dokumen tersebut kembali yang sesuai dengan kebutuhan pengguna secara efektif dan efisien.

Tesis ini mengusulkan suatu pendekatan kesamaan semantik dan struktur dalam deskripsi kasus penggunaan guna mendapatkan kembali dokumen spesifikasi kebutuhan perangkat lunak. Dari kasus penggunaan yang didapatkan kembali tersebut kemudian dibuat suatu urutan atau ranking kesamaan kasus penggunaan berdasar suatu *threshold* yang telah didapatkan berdasar nilai kesepakatan tertinggi dengan pakar pada proses pengujian.

Pendekatan menggunakan kesamaan semantik pada kata dan kalimat ini merupakan pendekatan yang diajukan sebagai pengganti dari pendekatan sebelumnya yaitu menggunakan *term frequency* dan *keyword* atau kata kunci. Metode ini diujicobakan pada percobaan dengan menggunakan 20 kueri deskripsi kasus penggunaan. Tiga skenario yang berbeda disusun untuk menginvestigasi nilai *threshold* yang ideal, mengetahui perbedaan hasil atau *result set* dengan pendekatan sebelumnya dan untuk mengetahui efek kombinasi struktur deskripsi kasus penggunaan pada masukan kueri terhadap hasil kueri yang didapatkan.

**Kata kunci:** *Use Case Retrieval, Semantic Similarity, Information Retrieval*

# Software Requirement Retrieval Using Use Case Structure and Semantic Similarity Computation

By : Ferdika Bagus Pristiawan Permana  
Student ID/NRP : 5114 201 030  
Supervisor : Daniel Oranova Siahaan, S.Kom, M.Sc., P.D.Eng

## ABSTRACT

In software industry, there are tremendous number of software requirements documents in effect of large scale of software development. This collections of software requirements documents can be reused in order to cut off the development time and reduce the cost. There is a need to retrieve one or more of those documents which is suitable with the user's specifications for the new software development.

So, if we can retrieve many similar software requirements to a new project, development process will be less costly and less error because the retrieved software requirements can be tailored to the new case with fewer modifications.

This thesis presents a methodology to retrieve software requirements documents using structured base of use case description similarity computation. A use case description is used as a query to retrieve similar or exactly the same use cases in the collections repository in order to get the requirement documents to be used in the new software project development.

We introduce a semantic similarity computation approach as a substitute of term frequency and keyword approach. We validate the usefulness of our method through the experiment using 20 cases. Three experiment scenario are presented to investigate the ideal threshold value, the retrieval result differences with previous approach and to know the effect of various combinations of structural query to the retrieval result.

**Keywords:** *Use Case Retrieval, Semantic Similarity, Information Retrieval*

## KATA PENGANTAR

Segala puji dan syukur kehadirat Allah Subhanallahu Wataalla atas berkat Rahmat, Taufik dan Hidayah-Nya sehingga dapat diselesaikannya Tesis yang berjudul “Pendekatan Kesamaan Semantik dan Struktur dalam Kasus Penggunaan untuk Mendapatkan Kembali Spesifikasi Kebutuhan Perangkat Lunak”. Tesis ini diajukan sebagai bagian dari tugas akhir dalam rangka menyelesaikan studi di Program Magister Teknik Informatika di Institut Teknologi Sepuluh Nopember Surabaya bidang keahlian Rekayasa Perangkat Lunak.

Dalam penyelesaian Tesis ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Untuk itu penulis menyampaikan ucapan terima kasih setulusnya kepada:

1. Daniel Oranova Siahaan, S.Kom,M.Sc.,P.D.Eng. selaku dosen pembimbing, yang telah banyak membantu penulis dalam memberikan ide, saran dan kritiknya.
2. Istriku Ratih Ardia Sari dan anakku Ferdiana Aulia Azhar yang telah memberikan semangat dan motivasi kepada penulis.
3. Raden Arief Setyawan, S.T.,M.T. dan Achmad Basuki, S.T.,M.MG., Ph.D. di UPT TIK UB yang telah memberikan waktu dan kesempatan bagi penulis untuk melanjutkan studi di Magister T.Informatika – ITS.
4. Semua rekan-rekan di UPT TIK UB dan Magister T.Informatika – ITS yang telah banyak membantu penulis.

***“Tiada gading yang tak retak”***. Penulis menyadari masih banyak kekurangan dan kelemahan di dalam Tesis ini. Untuk itu kritik dan saran yang konstruktif untuk penyempurnaan, akan penulis terima dengan senang hati. Akhirnya, harapan penulis semoga Tesis ini dapat memberikan kontribusi di dunia pendidikan dan bermanfaat bagi kita semua.

Surabaya, Juni 2016

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	iv
ABSTRAK .....	vi
ABSTRACT .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Penelitian Terdahulu.....	2
1.3 Perumusan Masalah.....	3
1.3.1. Masukan dan Luaran .....	3
1.3.2. Permasalahan.....	3
1.3.3. Batasan Masalah.....	4
1.3.4. Sumber Data.....	4
1.4 Tujuan dan Manfaat Penelitian.....	5
1.5 Kontribusi Penelitian .....	5
BAB 2 KAJIAN PUSTAKA.....	7
2.1 Pemodelan Kasus Penggunaan .....	7
2.1.1. Komponen-komponen dalam Kasus Penggunaan.....	7
2.1.2. Relasi Dalam Diagram Kasus Penggunaan.....	8
2.2 Deskripsi Kasus Penggunaan .....	9
2.3 Sistem Temu Kembali Informasi.....	10
2.3.1. Cara Kerja Sistem Temu Kembali Informasi.....	10

2.4	Pemrosesan Bahasa Alamiah ( <i>Natural Language Processing</i> ) .....	11
2.4.1	Ekstraksi Fakta .....	12
2.4.2	<i>Part of Speech Tagging</i> .....	12
2.4.3	Penguraian Kalimat ( <i>Parsing</i> ) .....	13
2.4.4	Tokenisasi .....	13
2.4.5	Penghapusan <i>Stopword</i> .....	14
2.4.6	<i>Stemming</i> .....	15
2.5	Teknik Pengukuran Kesamaan Kata dan Kalimat .....	16
2.5.1	String-Based Similarity .....	17
2.5.2	Corpus-Based Similarity .....	18
2.5.3	Knowledge-Based Similarity .....	19
2.5.4	Hybrid Similarity Measures .....	19
2.6	Wordnet .....	19
2.7	Koefisien Kappa .....	20
2.8	Milestone Penelitian Terkait Penggunaan Kembali Spesifikasi Kebutuhan Perangkat Lunak .....	21
2.9	Sistem Temu Kembali Kebutuhan Perangkat Lunak Menggunakan Perhitungan Kesamaan Struktur Dan Istilah Dalam Kasus Penggunaan .....	24
2.10	Arsitektur dan Teknologi Kakas Bantu Pendukung .....	27
BAB 3 METODE PENELITIAN .....		29
3.1	Studi Literatur .....	29
3.2	Analisa dan Perancangan Metode .....	30
3.2.1	Penentuan Kueri Kasus Penggunaan .....	30
3.2.2	Perhitungan Kesamaan Semantik .....	31
3.2.3	Proses Perankingan dan <i>Threshold</i> .....	35
3.2.4	Skenario Pengujian .....	36



BAB 4 HASIL PENELITIAN DAN PEMBAHASAN .....	37
4.1    Lingkungan Pengujian.....	37
4.1.1.    Koleksi Deskripsi Kasus Penggunaan.....	37
4.1.2.    Pakar atau ahli .....	41
4.1.3.    Proses Pencarian.....	42
4.1.4.    Kueri Kasus Penggunaan .....	42
4.2    Hasil Pengujian.....	43
4.2.1.    Pengujian Pertama.....	43
4.2.2.    Pengujian Kedua .....	44
4.2.3.    Pengujian Ketiga .....	47
4.3    Analisa dan Evaluasi Hasil Pengujian.....	50
4.3.1.    Analisa Hasil Pengujian Pertama.....	50
4.3.2.    Analisa Hasil Pengujian Kedua.....	52
4.3.3.    Analisa Hasil Pengujian Ketiga .....	53
BAB 5 KESIMPULAN DAN SARAN .....	55
5.1    Kesimpulan.....	55
5.2    Saran.....	56
DAFTAR PUSTAKA .....	57
BIOGRAFI PENULIS .....	59

## DAFTAR TABEL

Tabel 1.3.4.1 Sumber Data .....	4
Tabel 2.4.2.1 Penandaan dari POS <i>Tagging</i> .....	12
Tabel 2.4.3.1 Label pada proses penguraian .....	13
Tabel 2.4.5.1 Contoh <i>stopword</i> dalam bahasa Inggris .....	14
Tabel 2.4.5.2 Contoh penghapusan <i>stopword</i> .....	14
Tabel 2.4.6.1 Aturan untuk <i>noun</i> pada <i>WordNet Stemmer</i> .....	15
Tabel 2.4.6.2 Contoh <i>stemming</i> untuk kata benda .....	15
Tabel 2.4.6.3 Aturan untuk <i>verb</i> pada <i>WordNet Stemmer</i> .....	16
Tabel 2.4.6.4 Contoh <i>stemming</i> untuk kata kerja .....	16
Tabel 2.4.6.1 Beberapa pendekatan atau metode pengukuran kesamaan teks. ....	17
Tabel 2.5.4.1 Interpretasi nilai koefisien Kappa.....	20
Tabel 2.5.4.2 Hasil eksperimen E1 .....	20
Tabel 2.5.4.1 Milestone penelitian terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak .....	21
Tabel 2.5.4.1 Form Masukan Kueri Kasus Penggunaan .....	25
Tabel 3.2.1.1 Form kueri berdasar struktur deskripsi kasus penggunaan.....	30
Tabel 3.2.2.1 Contoh kalimat-kalimat pada use-case.....	33
Tabel 3.2.2.2 <i>Sentence preprocessing (Bag-of-word)</i> .....	33
Tabel 3.2.2.3 Nilai Masukan dari $V_1$ dan $V_2$ .....	34
Tabel 3.2.2.4 Nilai Vektor dari $V_1$ dan $V_2$ .....	35
Tabel 3.2.2.5 Nilai Vektor Akhir dari $V_1$ dan $V_2$ .....	35
Tabel 4.1.1.1 Daftar dataset berdasar nama sistem dan sumber referensi.....	38
Tabel 4.1.1.2 Domain kategori Kasus Penggunaan.....	40
Tabel 4.1.1.3 Sebaran kueri deskripsi kasus penggunaan berdasar kondisi setiap fitur/atribut deskripsi kasus penggunaan. ....	40
Tabel 4.2.1.1 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.1 sampai dengan no.6 .....	43
Tabel 4.2.1.2 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.7 sampai dengan no.13 .....	43

Tabel 4.2.1.3 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.14 sampai dengan no.20 .....	44
Tabel 4.2.2.1 Deskripsi Kasus Penggunaan “Withdraw Cash” .....	44
Tabel 4.2.2.2 Hasil Pengujian pada <i>use case</i> “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda. ....	45
Tabel 4.2.2.3 Deskripsi Kasus Penggunaan “Program Switch” .....	46
Tabel 4.2.2.4 Hasil Pengujian pada <i>use case</i> “Program Switch” dengan kombinasi masukan komponen yang berbeda. ....	47
Tabel 4.2.3.1 Beberapa contoh kata-kata yang mempunyai kesamaan semantik. ....	47
Tabel 4.2.3.2 Hasil kueri semantik dengan menggunakan kata frase “Pick gun” pada komponen nama <i>use case</i> . ....	47
Tabel 4.2.3.3 Hasil kueri non-semantik( <i>String-based similarity-dice coefficient</i> ) dengan menggunakan kata frase “Pick gun” pada komponen nama <i>use case</i> . ....	48
Tabel 4.2.3.4 Hasil kueri semantik dengan menggunakan kata frase “Get coin” pada komponen nama <i>use case</i> . ....	48
Tabel 4.2.3.5 Hasil kueri non-semantik( <i>String-based similarity-dice coefficient</i> ) dengan menggunakan kata frase “Get coin” pada komponen nama <i>use case</i> . ....	49
Tabel 4.2.3.6 Hasil kueri semantik dengan menggunakan kata frase “Add Formula” pada komponen nama <i>use case</i> .....	49
Tabel 4.2.3.7 Hasil kueri non-semantik dengan menggunakan kata “Add Formula” pada komponen nama <i>use case</i> .....	49
Tabel 4.3.1.1 Nilai <i>threshold</i> dan rata-rata nilai indeks Kappa dari keseluruhan data uji kueri deskripsi kasus penggunaan. ....	50
Tabel 4.3.1.2 R ata-rata Kappa untuk setiap nilai <i>threshold</i> antara 0,90 sampai dengan 0,98. ....	51
Tabel 4.3.2.1 Hasil Pengujian pada <i>use case</i> “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda. ....	53

## DAFTAR GAMBAR

Gambar 2.3.1 Cara kerja sistem temu kembali informasi .....	11
Gambar 3.2.1 Alur metodologi penelitian .....	29
Gambar 4.3.1 Grafik nilai Kappa untuk setiap nilai <i>threshold</i> antara 0,1 sampai dengan 0,9.....	51
Gambar 4.3.2 Grafik nilai Kappa untuk setiap nilai <i>threshold</i> antara 0,1 sampai dengan 0,9.....	52

[Halaman ini sengaja dikosongkan]



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di dalam pengembangan perangkat lunak, banyak permasalahan yang berakar pada keterbatasan pemahaman pengembang di dalam memahami kebutuhan pengguna terhadap perangkat lunak yang dibangun. Hal ini bisa disebabkan oleh berbagai faktor diantaranya yaitu keterbatasan data dan informasi yang didapatkan pada waktu proses pengumpulan, penganalisaan, penspesifikasian, verifikasi dan validitas kebutuhan dari perangkat lunak yang hendak dibangun (Siahaan, 2012). Oleh karena itu proses tersebut memerlukan usaha, waktu dan biaya tidak sedikit.

Pada saat ini, terdapat persaingan yang sangat ketat diantara para pengembang di industri perangkat lunak. Para pengembang mencari cara untuk mengurangi biaya proses pengembangan perangkat lunak sebanyak mungkin guna menekan biaya produksi sehingga memperoleh keuntungan dalam pasar.

Oleh karena itu penggunaan kembali komponen perangkat lunak seperti pada penelitian (Mcclure, 1997) dan (Krueger, 1992) dinilai lebih efektif memberikan dampak positif. Untuk mengurangi biaya dan alokasi waktu pada tahap ini berbagai pemodelan kebutuhan digunakan oleh para analis. Diantaranya yaitu melalui pemodelan kasus penggunaan seperti pada penelitian (Gomaa, 2000). Kasus penggunaan dipilih karena merupakan diagram yang seringkali digunakan dan dianggap efektif membantu di dalam proses analisa kebutuhan perangkat lunak. Dengan alasan keterbatasan waktu, sering kali para analis memanfaatkan kasus penggunaan yang pernah mereka buat untuk merancang kasus penggunaan atau spesifikasi kebutuhan yang baru. Dalam hal ini mereka menggunakan konsep *reuse* atau *information retrieval* di dalam analisa kebutuhan. Akan tetapi di dalam penerapannya timbul permasalahan diantaranya yaitu bagaimana mengukur tingkat kesamaan diantara kasus penggunaan.

## 1.2 Penelitian Terdahulu

Berbagai penelitian telah dilakukan terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak ini yaitu diantaranya yang dilakukan oleh (M. C. Blok, and J. L. Cybulski, 1998), (Saeki, Reusing use case descriptions for requirements specification: towards use case patterns, 1999), (Saeki, Patterns and aspects for use cases reuse techniques for use case descriptions, 2000), (H. G. Woo and W. N. Robinson, 2002), (Marir & Haouam, 2004), (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) dan (Srisura & Daengdej, 2010). Namun dari hasil penelitian-penelitian tersebut masih terdapat beberapa kekurangan seperti; proses yang kompleks, respon yang lambat dari *retrieval system* dan adanya proses yang masih dilakukan secara manual. Di dalam penelitiannya, (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) mengusulkan pendekatan *software requirement specification retrieval* dalam bentuk deskripsi kasus penggunaan menggunakan struktur *use case* dan komputasi kesamaan antara *term of use case query* dan *use case* di dalam repository.

Pada penelitian yang lain, (Srisura & Daengdej, 2010) mengusulkan menggunakan metode *case-based reasoning* di dalam menyelesaikan permasalahan pemanggilan kembali pengalaman-pengalaman sebelumnya untuk mendukung penggunaan *use case diagram*. Untuk membuat perangkat lunak yang baru, spesifikasi kebutuhan yang sudah ada/lama tersimpan di dalam suatu repository diambil kembali berdasarkan kesamaan diagram kasus penggunaan. Diagram kasus penggunaan dan spesifikasi kebutuhan yang memiliki tingkat kesamaan yang paling tinggi akan dijadikan sebagai pedoman atau acuan guna menentukan spesifikasi perangkat lunak yang baru. Akan tetapi di dalam penelitian tersebut masih terdapat kekurangan terkait mekanisme pengukuran kesamaan actor dan use case. Pencocokan kata masih menggunakan terbatas pada "*exactly matching word*" atau pencocokan kata yang sama persis.

(H. G. Woo and W. N. Robinson, 2002) mengusulkan teknik otomatisasi yang berfungsi untuk membantu proses penggunaan kembali diagram kasus penggunaan. Mereka mengembangkan kaskas bantu yang disebut "ScenAsst" untuk mengumpulkan dan mendapatkan kembali use case secara praktis. ScenAsst mengubah isi dari *use case* ke dalam bentuk graph, meng-clusternya dan



menyimpannya ke dalam sebuah koleksi *use case*. Di dalam proses *retrieval*, query dari user juga diubah ke dalam bentuk graph. Derajat kesamaan antara *query user* dan *graph* yang dihasilkan di dalam koleksi penyimpanan dibandingkan dengan menggunakan algoritma “SUBDUE”. Teknik ini menghasilkan respon proses yang lama karena faktor kompleksitas. (M. C. Blok, and J. L. Cybulski, 1998).

### **1.3 Perumusan Masalah**

Pada bagian ini akan dipaparkan mengenai permasalahan yang akan menjadi perhatian dalam penelitian ini. Penjabaran akan dibagi menjadi 3 bagian yakni pembahasan input dan output, garis permasalahan yang hendak diselesaikan dalam penelitian ini dan disertai dengan penentuan terhadap batasan masalah, sehingga penelitian lebih terarah.

#### **1.3.1. Masukan dan Luaran**

Masukan dari implementasi metode yang diusulkan yaitu berupa *query use case* dalam format *UC description*. Sedangkan luaran yang dihasilkan berupa urutan ranking UC pada *UC Repository* berdasar parameter dan threshold *query UC* yang dicari.

#### **1.3.2. Permasalahan**

Penelitian sebelumnya (Udomchaiorn, Prompoon, & Kanongchaiyos, 2006) menerangkan bahwa masih terdapat permasalahan terkait metode *Use Case Retrieval* yaitu pada kurangnya tingkat akurasi pada mekanisme pengukuran kesamaan kata dan kalimat dan metode pembobotan fitur yang masih dilakukan *manual* oleh pengguna. Oleh karena itu, penelitian ini mengusulkan solusi terhadap permasalahan tersebut yaitu berupa optimasi pada pengukuran kesamaan diagram *use case* dengan memasukkan aspek kesamaan semantik atau arti pada kata, frase dan kalimat menggunakan pendekatan *knowledge-based similarity* dan pembobotan fitur secara otomatis sesuai tingkat atau nilai akumulasi kesamaan fitur secara keseluruhan.

Beberapa permasalahan dalam penelitian ini dirumuskan sebagai berikut:

1. Bagaimana menambahkan aspek perhitungan kesamaan kata atau kalimat secara semantik atau arti kata di dalam temu kembali kasus penggunaan?
2. Bagaimana mengevaluasi atau mengukur tingkat akurasi dan efektifitas dari metode pengukuran kesamaan kasus penggunaan?

### 1.3.3. Batasan Masalah

Dalam penelitian ini, batasan masalah yang dibahas diuraikan sebagai berikut:

1. Bahasa yang digunakan dalam pemodelan kasus penggunaan adalah Bahasa Inggris.
2. Objek yang diteliti yaitu deskripsi kasus penggunaan dalam bentuk teks.

### 1.3.4. Sumber Data

Kasus penggunaan yang digunakan dalam penelitian ini sebagian besar berasal dari proyek-proyek akademik dan contoh-contoh pemodelan kasus penggunaan.

Tabel 1.3.4.1 Sumber Data

Jenis Sumber	Referensi
<i>Text book</i>	<ol style="list-style-type: none"> <li>1. (Siahaan, 2012)</li> <li>2. UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, Martin Fowler , 2003</li> <li>3. Managing Software Requirements: A Use Case Approach, Second Edition, Dean Leffingwell, Don Widrig , 2003</li> <li>4. UML Weekend Crash Course Thomas A. Pender, 2002</li> <li>5. Use Case Modeling, Kurt Bittner, Ian Spence , 2002</li> <li>6. Fast Track UML 2.0, Kendall Scott, Apress, 2004</li> <li>7. Enterprise Java with UML, CT Arlington, 2001</li> </ol>

	8. Use Case Driven Object Modeling with UML Theory and Practice, Doug Rosenberg, Matt Stephens, apress, 2007
--	--

Sumber data disimpan di dalam repositori dalam bentuk dokumen terstruktur (*structured document*).

Langkah-langkah ekstraksi dataset:

- Dataset dari *text book*, proses ekstraksi dilakukan secara manual dengan cara memasukkan deskripsi kasus penggunaan yang bersesuaian dengan menggunakan UML Modelling tool.
- Setelah semua data terekam dalam bentuk diagram kasus penggunaan beserta deskripsinya, data tersebut kemudian diekspor ke dalam bentuk file XMI.
- Diagram kasus penggunaan dalam format XMI tersebut kemudian di-*import* ke dalam repository basis data guna pengolahan atau proses berikutnya.

#### **1.4 Tujuan dan Manfaat Penelitian**

Tujuan dari penelitian ini adalah untuk mengoptimasi atau meningkatkan akurasi temu kembali dokumen spesifikasi kebutuhan perangkat lunak melalui deskripsi kasus penggunaan pada penelitian sebelumnya oleh (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006).

Manfaat dari penelitian ini, yaitu dihasilkannya solusi yang lebih optimal dalam proses temu kembali spesifikasi kebutuhan perangkat lunak melalui konsep penggunaan kembali (*reuse*) deskripsi kasus penggunaan sehingga didapatkan efisiensi dan efektifitas waktu proses rekayasa kebutuhan.

#### **1.5 Kontribusi Penelitian**

Adapun kontribusi dalam penelitian ini yaitu:

1. Penambahan perhitungan kesamaan kata dan kalimat secara semantik menggunakan *text similarity based on semantic nets* pada *use case description* untuk meningkatkan tingkat akurasi.

2. Pengembangan kakas bantu dan repositori kasus penggunaan untuk mempercepat dan mempermudah pengujian metode yang diusulkan.

## **BAB 2**

### **KAJIAN PUSTAKA**

Dalam subbab ini diuraikan kajian pustaka penelitian terkait perhitungan kesamaan semantik dan struktur pada kasus penggunaan di dalam analisa spesifikasi kebutuhan perangkat lunak.

#### **2.1 Pemodelan Kasus Penggunaan**

Melalui pemodelan Kasus Penggunaan (*Use-Case Modeling*), kebutuhan dasar aktor diidentifikasi dan bagaimana kebutuhan tersebut akan dipenuhi oleh sistem dijelaskan. Aktor merepresentasikan kelas pengguna dari sistem. Setiap kelas pengguna memiliki alasan yang spesifik mengapa ia menggunakan sistem. Alasan ini juga mewakili suatu *goal* dari setiap seorang pengguna ketika ia menggunakan sistem. Hubungan interaksi antara aktor dan sistem inilah yang kita sebut sebagai kasus penggunaan atau *Use Cases*. Kasus penggunaan menjelaskan fungsi-fungsi atau layanan-layanan yang harus dilakukan sistem dari perspektif aktor-aktor tertentu. Kasus-kasus penggunaan tersebut menangkap kebutuhan-kebutuhan dan membantu melacak realisasi dari kebutuhan tersebut selama proses pengembangan perangkat lunak (Siahaan, 2012).

##### **2.1.1. Komponen-komponen dalam Kasus Penggunaan**

Dari mengetahui semua komponen yang ada di dalam kasus penggunaan, pengembang dapat mengetahui siapa saja aktor-aktor yang berhubungan dengan sistem dan apa saja yang dilakukan sistem. Berikut ini adalah komponen-komponen yang terdapat pada pemodelan kasus penggunaan:

###### **a. Aktor**

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah seseorang atau sesuatu yang dapat berinteraksi dengan sistem, yaitu siapa dan apa yang menggunakan sistem. Aktor mempresentasikan peran atau profesi, bukan pemakai individu dari sistem. Aktor memiliki nama. Nama yang dipilih seharusnya menyatakan peran atau profesi aktor.

## b. Kasus Penggunaan

Kasus penggunaan (*Use-Case*) merupakan deskripsi dari interaksi antara aktor dan sistem yang ditujukan untuk memenuhi kebutuhan pengguna. Suatu kasus penggunaan merupakan deskripsi langkah-langkah interaksi antara aktor dan sistem untuk mencapai hasil tertentu. Kasus penggunaan seringkali merepresentasikan fungsional yang dapat dilakukan oleh aktor dari sistem. Kasus penggunaan juga merefleksikan tujuan atau goal seorang aktor ketika menggunakan sistem (Siahaan, 2012).

## c. Relasi dalam Kasus Penggunaan

Relasi digunakan sebagai penghubung antar aktor dengan kasus penggunaan. Jenis relasi dijelaskan pada sub bab berikutnya.

### 2.1.2. Relasi Dalam Diagram Kasus Penggunaan

#### a. Asosiasi

Asosiasi, yaitu hubungan statis antar elemen. Umumnya menggambarkan element yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.

#### b. Generalisasi dan spesialisasi

Spesialisasi, yaitu hubungan hirarkis antar elemen. Elemen dapat diturunkan dari elemen lain dan mewarisi semua atribut dan metode elemen asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari elemen yang diwarisinya. Kebalikan dari spesialisasi adalah generalisasi.

#### c. Ketergantungan atau *Dependency*

Ketergantungan adalah suatu jenis hubungan yang menandakan bahwa satu elemen, atau kelompok elemen, bertindak sebagai klien tergantung pada unsur lain atau kelompok elemen yang berlaku sebagai penyalur. Ini merupakan suatu hubungan lemah yang menandakan bahwa jika penyalur klien diubah maka klien secara otomatis akan terpengaruh oleh perubahan tersebut. Ini merupakan suatu hubungan searah. Ketergantungan terbagi menjadi dua, yaitu *include* dan *extend*. *Include* adalah relasi dimana kasus penggunaan yang di-*include*-kan adalah bagian dari proses kasus penggunaan utama. Sedangkan *extend* adalah relasi yang

memungkinkan suatu kasus penggunaan memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh kasus penggunaan lainnya.

## 2.2 Deskripsi Kasus Penggunaan

Deskripsi Kasus Penggunaan merupakan suatu dokumen yang mendeskripsikan detail-detail dari kasus penggunaan di dalam diagram kasus penggunaan dan ditulis dalam bentuk bahasa alamiah. Komponen-komponen dasar dari deskripsi kasus penggunaan yang mencakup isi-isi penting dan biasa digunakan terdiri dari sembilan komponen yaitu:

1. *Use Case Name: Use case name* merupakan nama dari kasus penggunaan. Setiap kasus penggunaan harus mempunyai nama guna membedakan antara kasus penggunaan satu dengan yang lainnya.
2. *Objective*: Merupakan suatu elemen yang mendeskripsikan tujuan dari kasus penggunaan.
3. *Actor*: Merupakan seseorang atau sesuatu/peran diluar sistem yang berinteraksi dengan sistem.
4. *Relationship*: Merupakan koneksi semantik antar elemen-elemen model. Di dalam deskripsi kasus penggunaan, hubungan terdiri dari *association*, *include*, *extend* dan *generalization*. Komponen-komponen hubungan mengandung nama kasus penggunaan dari kasus penggunaan yang terkait.
5. *Pre-Condition: Precondition* adalah sebuah konstrain yang harus bernilai *true* ketika kasus penggunaan dipanggil.
6. *Post-Condition: Postcondition* merupakan konstrain yang harus bernilai *true*/benar ketika kasus penggunaan berakhir.
7. *Normal flow of events*: Merupakan elemen dari kasus penggunaan yang menjelaskan implementasi secara normal tanpa adanya *error*/kesalahan.
8. *Subflow*: Dibeberapa kasus, aliran normal dapat didekomposisi menjadi beberapa subflow.
9. *Alternative or exceptional flow of events*: Merupakan elemen dari kasus penggunaan yang menjelaskan implementasi alternatif.

## 2.3 Sistem Temu Kembali Informasi

*Information Retrieval System* (IRS) merupakan sistem yang digunakan untuk menemukan kembali (*retrieve*) dokumen yang relevan terhadap kebutuhan *user* dari suatu kumpulan informasi berdasarkan kata kunci atau *keywords* atau *query* dari *user*. Selain menemukan dokumen yang relevan terhadap *query*, IRS juga melakukan perankingan terhadap hasil pencarian tersebut. Suatu dokumen yang memiliki ranking yang lebih tinggi daripada dokumen lain akan dianggap lebih relevan terhadap *query* (M. J. McGill and G. Salton, 1983).

### 2.3.1. Cara Kerja Sistem Temu Kembali Informasi

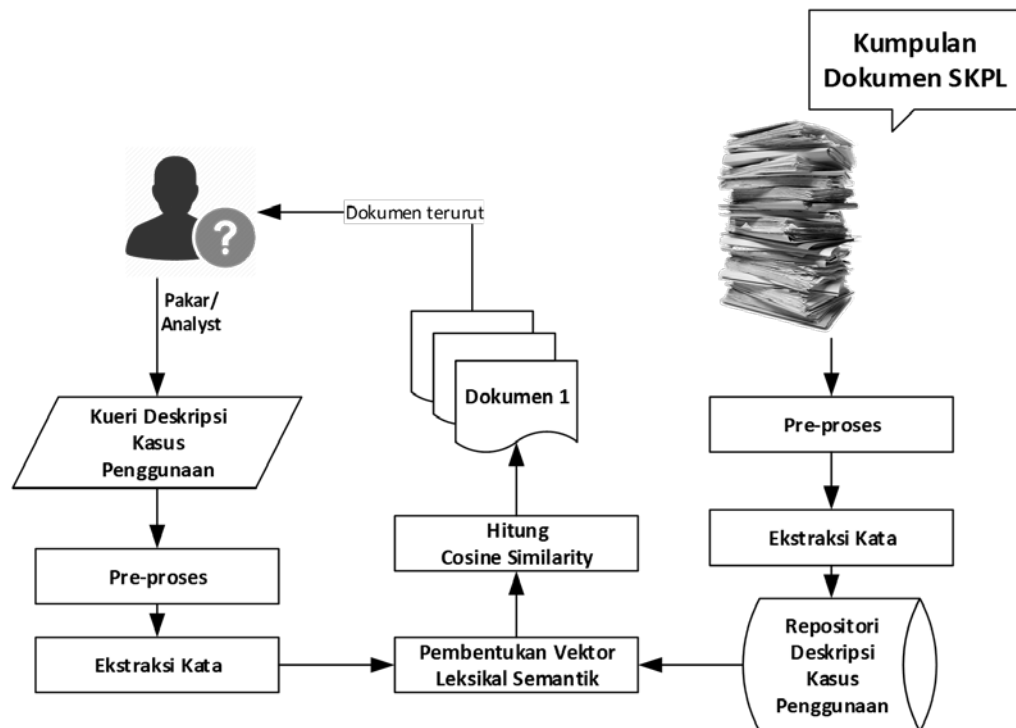
Model sistem temu kembali informasi menentukan detail sistem temu kembali. Informasi yaitu meliputi representasi dokumen maupun kueri, fungsi pencarian (*retrieval function*) dan notasi kesesuaian (*relevance notation*) dokumen terhadap kueri. Salah satu model sistem temu kembali informasi yang paling awal digunakan adalah model *boolean*. Model boolean mempresentasikan dokumen sebagai suatu himpunan kata-kunci (*set of keywords*). Sedangkan kueri direpresentasikan sebagai ekspresi *boolean*. Kueri dalam ekspresi *boolean* merupakan kumpulan kata kunci yang saling dihubungkan melalui operator *boolean* seperti *AND*, *OR* dan *NOT* serta menggunakan tanda kurung untuk menentukan *scope* operator. Hasil pencarian dokumen dari model *boolean* adalah himpunan dokumen yang relevan.

Gambar 2.3.1 menunjukkan proses temu kembali secara umum.

1. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam kueri maupun dokumen dalam pentransformasian dokumen atau kueri menjadi *terms index* (indeks dari kata-kata).
2. *Query Formulation* (formulasi terhadap kueri) yang memberi bobot pada indeks kata-kata kueri.
3. *Ranking*, mencari dokumen-dokumen yang relevan terhadap kueri dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan kueri.
4. *Indexing*, membangun data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen, sistem temu balik informasi menerima kueri dari pengguna, kemudian melakukan perankingan terhadap pada koleksi



berdasarkan kesesuaiannya dengan kueri. Hasil perangkaian yang diberikan kepada pengguna merupakan dokumen yang sistem, relevan dengan kueri, namun relevansi dokumen terhadap suatu kueri merupakan penilaian pengguna yang subyektif dan dipengaruhi banyak faktor.



Gambar 2.3.1 Cara kerja sistem temu kembali informasi

#### 2.4 Pemrosesan Bahasa Alamiah (*Natural Language Processing*)

Pemrosesan bahasa alamiah adalah metode yang digunakan untuk melakukan proses komputasi dari bahasa sehingga terciptanya interaksi antara manusia dengan komputer menggunakan bahasa alamiah. Salah satu alat yang digunakan dalam pemrosesan Bahasa alamiah adalah StanfordNLP. StanfordNLP adalah sistem yang dikembangkan oleh The Stanford NLP Group dengan fokus penelitian pada *sentence understanding*, *probabilistic parsing and tagging*, *biomedical information extraction*, *grammar introduction*, *word sense disambiguation*, dan *automatic question answering* (nlp.stanford.edu, 2013).

### 2.4.1 Ekstraksi Fakta

Ekstraksi fakta bertujuan untuk mengambil fakta-fakta yang ada pada setiap kalimat yang diinputkan dengan cara memotong kalimat menjadi klausa-klausa. Kalimat adalah satuan bahasa berupa kata atau rangkaian kata yang dapat berdiri sendiri dan menyatakan makna yang lengkap. Kalimat mengungkapkan pikiran yang utuh, baik dengan cara lisan maupun tulisan. Proses untuk ekstraksi fakta dilakukan dengan dua tahap yaitu POS tagging dan Parsing.

### 2.4.2 Part of Speech Tagging

*Part of Speech Tagging* (POS Tagging) adalah proses untuk mendeteksi *part of speech* (*verb*, *adverb*, *personal pronoun*) dari tiap token dengan menggunakan kamus *tag* dan model latin. Keluaran dari proses POS tagging menggunakan model *Penn Treebank*. POS tagging memberikan label kata secara otomatis pada suatu kalimat. POS tagging dapat diklasifikasikan menjadi delapan jenis yaitu *noun* (kata benda), *pronoun* (kata ganti), *adjective* (kata sifat), *verb* (kata kerja), *adverb* (kata keterangan), *preposition* (kata depan), *conjunction* (kata penghubung) dan *interjection* (kata seru).

Tabel 2.4.2.1 Penandaan dari POS Tagging

Tag	Deskripsi	Tag	Deskripsi
CC	Coordinating conjunction	UH	Interjection
CD	Cardinal number	VBD	Verb, past tense
DT	Determiner	VBZ	Verb, 3rd ps.sing. present
EX	Existential there	MD	Modal
FW	Foreign word	WDT	Wh-determiner
VB	Verb, base form	NN	Noun, singular or mass
IN	Preposition/subordinate conjunction	WP	Wh-pronoun
JJ	Adjective	NNP	Proper noun, singular
VBG	Verb, gerund/present participle	WPS	Possesive wh-pronoun
JJR	Adjective, comparative	NNPS	Proper noun, plural
VBN	Verb, past participle	NNS	Noun, plural
JJS	Adjective, superlative	PDT	Predeterminer

VBP	Verb, non-3rd ps. sing. Present	POS	Possesive ending
LS	List item marker	PRP	Personal pronoun
RP	Particle	PRP\$	Possesive pronoun
SYM	Symbol	RB	Adverb
TO	To	RBR	Adverb, comparative
RBS	Adverb, superlative	WRB	Wh-adverb

### 2.4.3 Penguraian Kalimat (*Parsing*)

Penguraian Kalimat (*Parsing*) adalah suatu proses menguraikan kalimat berdasarkan struktur kalimat dan tata bahasa yang menggunakan *Penn TreeBank* sebagai dasarnya. Gramatika yang dipakai juga sangat berkaitan dengan proses penguraian kalimat apa yang digunakan. Penguraian kalimat dilakukan dengan cara memecah-mecah suatu rangkaian masukan yang akan menghasilkan suatu pohon urai yang akan digunakan pada tahap analisis semantik.

Tabel 2.4.3.1 Label pada proses penguraian

Label	Deskripsi
ADJP	Adjective Phrase
ADVP	Adverb Phrase
CONJP	Conjunction Phase
INTJ	Interjection
LST	List Maker
PP	Prepositional Phrase
PRT	Particle
SBAR	Clause Introduced by a Subordinating conjunction
UCP	Unlike Coordinate Phrase
VP	Verb Phrase
NP	Noun Phrase

### 2.4.4 Tokenisasi

Tokenisasi adalah tahap pemotongan kata berdasarkan tiap kata yang menyusunnya. Sebelum melakukan tokenisasi, setiap huruf dalam dokumen harus diubah ke dalam huruf kecil. Hanya huruf „a” sampai dengan „z” yang diterima.

Karakter-karakter tertentu seperti digit, angka, tanda hubung, dan tanda baca akan dihilangkan. Berikut contoh hasil proses tokenisasi:

K1 : “The”, “system”, “must”, “be”, “able”, “to”, “receives”, “many”, “request”, “in”, “a”, “time”.

### 2.4.5 Penghapusan *Stopword*

*Stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Pendekatan *bag-of-words* merupakan penyederhanaan asumsi yang digunakan pada pemrosesan bahasa alami dan temu kembali informasi. Dalam pendekatan ini, sebuah teks (misalnya sebuah kalimat atau sebuah dokumen) dipresentasikan sebagai sebuah koleksi kalimat yang tidak berurutan, serta tidak mengindahkan tata bahasa dan urutan kata.

*Stopwords* perlu dihapus karena dua alasan, yaitu menghemat tempat dan waktu pencarian, serta untuk menghindari kesalahan pemaknaan oleh *parser*. Sebagai contoh, “karena” dan ”sebab” adalah dua *stopword* yang memiliki makna yang sama.

Tabel 2.4.5.1 Contoh *stopword* dalam bahasa Inggris

<b><i>Stopword</i> dalam bahasa Inggris</b>
<i>a, about, above, after, again, all, am, before, being, cannot, could, during, each, few, has, have, her, here, how, I, into, is, let’s, me, more, dan lain-lain.</i>

Tahap penghapusan *stopwords* atau penyaringan adalah tahap mengambil kata-kata penting dari hasil tokenisasi. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) arau *wordlist* (menyimpan kata penting). Contoh hasil penghapusan *stopword* dapat dilihat di Tabel 2.4.4.

Tabel 2.4.5.2 Contoh penghapusan *stopword*

<b>Teks Masukan</b>	<b>Teks Luaran</b>
“An”, “administrator”, “has”, “the”, “task,”, ”of”, “configuring”, “and “, “maintaining”, “the”, “system”	Administrator task configuring maintaining system

### 2.4.6 Stemming

Stemming adalah tahap mencari akar kata dari tiap kata hasil penyaringan. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Algoritma *stemming* yang digunakan adalah *wordnet stemmer*. Algoritma ini merupakan algoritma berbasis kamus yang menggunakan kamus *wordnet* sebagai acuannya. *Stemming* dilakukan berdasarkan aturan yang telah ditentukan. Bila kata hasil *stemming* tersebut ada di dalam kamus *wordnet*, maka kata yang ditemukan pada kamus *wordnet* yang menjadi keluaran dari *WordNet stemmer*. Namun, jika kata tersebut tidak ditemukan dalam kamus *wordnet*, maka kata yang dimasukkan akan dianggap sebagai luaran dari *Wordnet Stemmer*. Aturan-aturan *stemming* untuk tiap *part of speech* dalam *wordnet stemmer* dapat dilihat pada Tabel 2.4.5.

Tabel 2.4.6.1 Aturan untuk *noun* pada *WordNet Stemmer*

Part of Speech	Suffix	Replacement
Noun	S	NULL
	ses	s
	xes	x
	ches	ch
	shes	sh
	Ies	y
	Ves	f
	Zes	z
	Men	man

Untuk setiap *part of speech noun*, bila kata masukan memiliki akhiran seperti pada kolom *suffix*, maka akhiran awal akan diganti dengan akhiran seperti pada kolom *replacement*. Contoh *stemming* untuk kata benda dapat dilihat pada Tabel 2.4.6.

Tabel 2.4.6.2 Contoh stemming untuk kata benda

Kata Masukan	Kata Luaran
Application	application
Taxes	Tax

Ashes	Ash
Lives	Life
Women	Woman

Tabel 2.4.6.3 Aturan untuk verb pada WordNet *Stemmer*

Part of Speech	Suffix	Replacement
Verb	s	Null
	ies	Y
	es	e
	es	Null
	ed	e
	ing	e
	ing	Null

Untuk setiap *part of speech* kata kerja, bila kata masukan memiliki akhiran seperti pada kolom *suffix*, maka akan di akhiran awal akan diganti dengan akhiran seperti pada kolom *replacement*. Contoh stemming untuk *verb* dapat dilihat di tabel 2.4.8

Tabel 2.4.6.4 Contoh *stemming* untuk kata kerja

Kata Masukan	Kata Luaran
Performs	Perform
Clustering	Cluster
Worked	Work
Living	Live
Presenting	Present

## 2.5 Teknik Pengukuran Kesamaan Kata dan Kalimat

Pengukuran derajat atau tingkat kesamaan antar kata, kalimat, paragraf dan dokumen merupakan komponen yang penting di dalam berbagai disiplin ilmu seperti: *information retrieval*, *document clustering*, *word-sense disambiguation*, *automatic essay scoring*, *short answer grading*, *machine translation* dan *text*

*summarization*. Berikut beberapa teknik atau metode perhitungan tingkat kesamaan teks yang dihasilkan dari beberapa penelitian terakhir:

Tabel 2.4.6.1 Beberapa pendekatan atau metode pengukuran kesamaan teks.

No	Kategori	Sub-Kategori	Nama Metode
1	<i>String-Based Similarity</i> : Pengukuran kesamaan berdasarkan pada rangkaian atau urutan <i>string</i> dan komposisi karakter.	<i>Character-Based Similarity</i>	LCS
			Dameraue-Levenshtein
			Jaro
			Jaro-Winkler
			Needleman-Wunsch
			Smith-Waterman
			N-gram
		<i>Term-Based</i>	Blok Distance
			Cosine Similarity
			Dice's Coefficient
			Euclidean Distance
			Jaccard Similarity
			Matching Coefficient
			Overlap Coefficient
2	<i>Corpus-Based Similarity</i> : Pengukuran kesamaan semantic berdasar informasi yang didapatkan dari basis data <i>corpora</i> . (misal: <i>Brown Corpus</i> )	HAL	
		LSA	GLSA
		ESA	CL-ESA
		PMI-IR	SOC-PMI
		NGD	
		DISCO	DISCO1
			DISCO2
3	<i>Knowledge-Based Similarity</i> : Pengukuran kesamaan semantik berdasar tingkat kesamaan antar kata-kata menggunakan informasi yang didapatkan dari <i>semantic network</i> seperti; <i>WordNet</i> .	<i>Similarity</i>	<i>Information Content</i>
			• res
			• lin
			• jcn
			<i>Path Length</i>
		• lch	
		• wup	
		• path	
		<i>Relatedness</i>	hso
			lesk
vector			
4	<i>Hybrid Similarity</i>	<i>Semantic Text Similarity (STS)</i>	

### 2.5.1 String-Based Similarity

*String-Based Similarity* adalah pengukuran kesamaan string yang beroperasi pada urutan string dan komposisi karakter. Sebuah metrik *string* adalah sebuah metrik yang mengukur tingkat kesamaan maupun ketidaksamaan antara dua buah teks *string* untuk pencocokan atau perbandingan secara akurat. *String-Based*

*Similarity* terbagi menjadi dua kategori atau tipe yaitu *Character-Based* dan *Term-Based Similarity*.

*Character-Based Similarity Measures*:

- Algoritma *Longest Common SubString* (LCS) memperhitungkan kesamaan antara dua buah string berdasar pada panjang *contiguous chain* dari karakter-karakter yang muncul pada kedua string.
- *Damerau-Levenshtein* mendefinisikan jarak antara dua buah string dengan menghitung jumlah minimum operasi yang dibutuhkan untuk mengubah satu string ke string yang lain. Dimana sebuah operasi dapat berupa penyisipan, penghapusan atau pergantian dari sebuah karakter atau *transposition* dari dua *adjacent* karakter.

*Term-Based Similarity Measures*:

- *Cosine similarity* adalah metode pengukuran kesamaan antara dua buah vektor dari sebuah *inner product space* yang mengukur *cosinus* sudut dari kedua vektor tersebut.
- *Jaccard similarity* dihitung sebagai jumlah *terms* yang di-*share* diatas atau berbanding jumlah semua terms unik yang terdapat pada kedua buah string.

## 2.5.2 Corpus-Based Similarity

*Corpus-Based Similarity* merupakan metode pengukuran kesamaan teks secara semantik yang menentukan kesamaan antar kata berdasar informasi yang didapatkan dari *corpora* yang besar. Sebuah *Corpus* adalah sekumpulan koleksi yang besar dari teks tertulis maupun lisan yang digunakan untuk penelitian bahasa.

- *Latent Semantic Analysis* (LSA) merupakan teknik *Corpus-Based similarity* yang paling populer. LSA mengasumsikan bahwa kata-kata yang memiliki kesamaan atau kedekatan arti akan muncul pada potongan-potongan teks yang sama.
- *Extracting DIStributionally similar word using Cooccurrence* (DISCO) merupakan kesamaan terdistribusi antar kata yang mengasumsikan kata yang



mempunyai kesamaan arti muncul pada konteks yang sama. Koleksi teks yang besar dianalisa secara statistik untuk memperoleh kesamaan yang terdistribusi.

### 2.5.3 Knowledge-Based Similarity

*Knowledge-Based Similarity* merupakan salah satu metode pengukuran kesamaan secara semantik yang berbasis pada pengidentifikasian derajat kesamaan antar kata menggunakan informasi yang didapatkan dari *semantic networks*. *WordNet* merupakan salah satu bentuk *semantic network* yang populer digunakan pada area *knowledge-based similarity*.

### 2.5.4 Hybrid Similarity Measures

*Metode hybrid similarity* menggunakan atau mengkombinasikan beberapa metode pengukuran kesamaan secara bersamaan.

- *Semantic Text Similarity (STS)*

Metode ini menentukan tingkat kesamaan dua buah teks berdasar kombinasi dari informasi semantik dan informasi sintaks/tata bahasa. Di dalam metode ini terdapat 2 metode *mandatory* (*string similarity* dan *semantic word similarity*) dan satu fungsi *optional* (*common word order similarity*).

## 2.6 Wordnet

*Wordnet* adalah basis data leksikal untuk bahasa Inggris yang merupakan hasil penelitian dari Princeton University dalam bidang komputasi linguistik. Basis data leksikal adalah kumpulan data yang menyimpan informasi relasi semantik antar sinonim set (*synset*). Dua jenis relasi antar *synset* terdiri dari leksikal dan semantik. Relasi leksikal merupakan relasi berdasarkan pada susunan kata sedangkan semantik adalah relasi berdasarkan pada arti kata. *Wordnet* memiliki empat tipe *part of speech* (POS) yaitu *noun* (N), *verb* (V), *adjective* (Adj) dan *adverb* (Av). *Wordnet* juga memiliki beberapa relasi semantik, yaitu sinonim, antonym, hiponim, meronym, troponim dan entailment (Milner, 1995).

Hal yang paling penting dalam *wordnet* adalah kemiripan makna karena penentuan hubungan antar bentuk kata merupakan prasyarat untuk merepresentasikan makna dalam matriks leksikal. Dua buah kata dianggap

bersinonim jika penggantian kata tersebut oleh kata yang lain tidak akan merubah makna kalimat. Jarang sekali ditemukan kata yang bersinonim dan dapat digunakan dalam konteks yang sama. Kata “mobil” dan “truk” memiliki makna serupa tetapi keduanya memiliki perbedaan konteks penggunaan.

## 2.7 Koefisien Kappa

Koefisien Kappa sering digunakan untuk mengukur kesepakatan dari dua pengamat terhadap karakteristik yang menjadi perhatian penelitian. Koefisien Kappa hanya diterapkan pada hasil pengukuran data kualitatif.

Tabel 2.5.4.1 Interpretasi nilai koefisien Kappa

Nilai K	Keeratan Kesepakatan ( <i>Strength of agreement</i> )
< 0,20	Rendah (Poor)
0,21 – 0,40	Lumayan (Fair)
0,41 – 0,60	Cukup (Moderate)
0,61 – 0,80	Kuat (Good)
0,81 – 1.00	Sangat Kuat (Very good)

Berikut ini adalah contoh kasus perhitungan menggunakan koefisien Kappa menurut (Gwet, 2002).

Tabel 2.5.4.2 Hasil eksperimen E1

Pengamat B	Pengamat A		
	“1”	“2”	Total
“1”	40	9	49
“2”	6	45	51
Total	46	54	100

Dengan  $e(\gamma)$  merupakan derajat probabilitas *change-agreement* dan  $P_1$  merepresentasikan aproksimasi kemungkinan bahwa pengamat (A atau B) mengklasifikasikan sebuah objek ke dalam kategori “1”.

$$e(\gamma) = 2 P_1(1 - P_1) \dots\dots\dots (1)$$

$$P_1 = \frac{(A1+B1)/2}{N} \dots\dots\dots (2)$$

$$Kappa(AC1) = \frac{p-e(\gamma)}{1-e(\gamma)} \dots\dots\dots (3)$$

$$p = \frac{A+D}{N} \dots\dots\dots (4)$$

Dari rumus atau persamaan (1), (2), (3) dan (4) didapat hasil perhitungan Kappa sebagai berikut:

$$e(\gamma) = 2 \left( \frac{46 + 49}{2 \times 100} \right) \left( 1 - \frac{46 + 49}{2 \times 100} \right) = 0,49875$$

Sehingga didapatkan koefisien Kappa (AC1) sebesar:

$$AC1 = \frac{0,85 - 0,49875}{1 - 0,49875} = 0,7008$$

## 2.8 Milestone Penelitian Terkait Penggunaan Kembali Spesifikasi Kebutuhan Perangkat Lunak

Tabel 2.5.4.1 Milestone penelitian terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak.

No	Tahun	Judul Penelitian	Penjelasan
1	1999	M. C. Blok, and J. L. Bybulski, "Reusing UML Specifications in a Constrained Application Domain", Proceeding of the 5th Asia Pacific Software Engineering Conference (APSEC),1998.	Merupakan metode pertama dari <i>use case retrieval</i> yang mengukur kesamaan diagram <i>sequence</i> dari akumulasi alur aksi diagram (diagram <i>event flow accumulation</i> ).
2	2002	H. G. Woo, and W. N. Robinson, "Reuse of Scenario Specifications Using an Automated Relational Learner: A Lightweight Approach", Requirement Engineering Conference (SAIS), 2002.	Mengusulkan teknik otomatisasi dalam perhitungan tingkat kesamaan diagram <i>sequence</i> . Diagram <i>sequence</i> ini kemudian ditransformasi ke dalam bentuk graph kemudian dilakukan pencocokan <i>graph</i> guna mengukur tingkat kesamaan diagram.
3	2006	A. Udomchaiporn, N. Prompoon, and P.Kanongchaiyos, "Software	Mengusulkan metode retrieval menggunakan <i>Vector Space Model</i>

		Requirements Retrieval Using Use Case Terms and Structure Similarity Computation”, The 13th Conference in Software Engineering (APSEC), Dec 2006, p.113-120	(VSM) untuk mengukur tingkat kesamaan <i>use case description</i> . Keterbasan: <ul style="list-style-type: none"> <li>• Perhitungan kesamaan kata tidak memperhitungkan kesamaan semantik.</li> <li>• Pemberian bobot untuk masing-masing elemen struktur kasus penggunaan masih dilakukan secara manual oleh pengguna. Sehingga ketidaktepatan pemberian bobot sangat mempengaruhi akurasi pada hasil retrieval.</li> </ul>
4	2006	S. Suksaard, and N. Prompoon, “Use Case Retrieval Using Use Case Clustering and User Relevance Feedback”, The 3rd Joint Conference on Computer Science of Software Engineering, 2006	Mengusulkan ide tambahan untuk meningkatkan metode <i>use case retrieval</i> yang diusulkan oleh Udomchiporn’s dengan menggunakan teknik relevansi <i>feedback</i> .
5	2007	Use Case Extractor: XML Parser for Automated Extraction and Storage of Use Case Diagram	Mengusulkan kakas bantu untuk <i>Use Case Extractor</i> dengan tujuan: <ul style="list-style-type: none"> <li>• Membaca dan menyimpan data <i>use-case</i> ke dalam basis data.</li> <li>• Otomatisasi evaluasi diagram <i>use-case</i></li> </ul>
6	2009	Ilyas, M., & Kung, J. (2009). A Similarity Measurement Framework for Requirements Engineering. Fourth International Multi-Conference on Computing in the Global Information	Mengusulkan suatu kerangka kerja (SimReq) guna penggunaan kembali <i>design</i> dan <i>code</i> . Metodologi: <ul style="list-style-type: none"> <li>• Persiapan Data</li> </ul>

		Technology (pp. 31-35). IEEE Computer Society.	<p>Analisa kelengkapan spesifikasi kebutuhan dan analisis <i>lexical</i>.</p> <ul style="list-style-type: none"> <li>• Pengukuran Kesamaan Spesifikasi Kebutuhan</li> <li>• Penggunaan Kembali Rancangan dan Kode Sumber.</li> </ul>
7	2010	Srisura, B., & Daengdej, J. (2010). Retrieving Use Case Diagram With Case-based Reasoning Approach. Journal of Theoretical and Applied Information Technology, 19(2), 68-78.	<p>Mendapatkan kembali diagram-diagram yg mirip dengan membandingkan dua dimensi: dimensi use case dan <i>actor-text-based</i> format, dan dimensi hubungan – <i>structure-based</i> format, dan kemudian merankingnya.</p> <p>Skor yang sesuai dan bobot yg masuk akal dari setiap dimensi dihitung secara teliti berdasarkan <i>the nearest neighbor matching</i> dan <i>ranking</i> untuk menentukan <i>use case diagram</i> yang paling mirip.</p>
8	2012	Morales, B. B., Sergio, C., & Clunie, C. (2012, January). Reuse of Use Cases Diagrams: An Approach based on Ontologies and Semantic Web Technologies. IJCSI International Journal of Computer Science Issues, 9(1), 24-29.	<p>Mengusulkan sebuah kaskas bantu untuk menggunakan kembali diagram kasus penggunaan dengan menyimpan informasi ke dalam Ontologi (OWL) dan penggunaan teknologi situs semantik.</p> <p>Metodologi:</p> <ul style="list-style-type: none"> <li>• Pembuatan diagram kasus penggunaan dan meng-ekspornya ke dalam format XMI.</li> <li>• Menyimpan informasi-informasi pada diagram kasus</li> </ul>

			<p>penggunaan ke dalam Ontologi OWL.</p> <ul style="list-style-type: none"> <li>• Menyimpan informasi Ontologi OWL ke dalam DBMS MySQL</li> <li>• Mencari dan mendapatkan kembali diagram kasus penggunaan dalam format XML.</li> </ul>
9	2013	Reza Fauzan “Rekomendasi Kasus Penggunaan Berdasarkan Skenario Naratif Menggunakan Teknologi Semantik”	<p>Ekstraksi metadata dari scenario naratif yg memiliki potensi menjadi use case.</p> <p>Metodologi yang dilakukan:</p> <ol style="list-style-type: none"> <li>1. Analis</li> <li>2. Skenario</li> <li>3. Ekstraksi Metadata(StanfordNLP)</li> <li>4. Pencarian dlm repository use case</li> </ol>

## 2.9 Sistem Temu Kembali Kebutuhan Perangkat Lunak Menggunakan Perhitungan Kesamaan Struktur Dan Istilah Dalam Kasus Penggunaan

Pada penelitian (Udomchaiyorn, Prompoon, & Kanongchaiyos, 2006) diusulkan suatu metode penggunaan kembali spesifikasi kebutuhan perangkat lunak melalui perhitungan kesamaan struktur kasus penggunaan. Pendekatan ini diharapkan dapat menjadi solusi untuk mengurangi biaya dan waktu pada setiap pengembangan perangkat lunak.

Konsep Sistem Temu Kembali Informasi diimplementasikan dalam penelitian tersebut, yaitu meliputi empat langkah utama:

### 1) *Automatic Indexing*:

- a. Parsing isi dari *use case*; memisahkan setiap kata di dalam *use case*.
- b. Menghapus atau menghilangkan kata-kata yang termasuk dalam *stop list*.

c. *Stemming word* ke dalam bentuk *grammatical root*. Salah satu model yang sering digunakan dalam *words stemming* yaitu metode yang dikembangkan oleh (Porter, 1980).

d. Menghilangkan *high frequency words*.

2) *Term weighting system*

Merupakan suatu proses untuk menentukan bobot pada masing-masing *term*. Pada saat ini terdapat banyak sekali metode pembobotan. Akan tetapi pada penelitian tersebut menggunakan metode yang disebut *Inverse Document Frequency (IDF)*.

3) *Vector Similarity Computation*

Metode perhitungan kesamaan vektor yang digunakan dalam penelitian tersebut yaitu *Dice's coefficients*.

4) *Retrieval evaluation*

Tujuan dari evaluasi yaitu untuk mengukur tingkat efektifitas dan efisiensi dari sebuah sistem temu kembali dan penyimpanan informasi. Untuk proses evaluasi, terdapat tiga metrik yang populer; *recall*, *precision* dan *harmonic mean*. Berikut penjelasan singkat ketiga metrik tersebut:

- *Recall* mendefinisikan proporsi dari dokumen yang diterima kembali dan relevan terhadap semua dokumen relevan di dalam koleksi.
- *Precision* adalah proporsi dari dokumen yang relevan dan diterima kembali terhadap semua dokumen yang diterima kembali di dalam koleksi.
- *Harmonic mean* merupakan pengukuran tunggal yang mengkombinasikan antara *recall* dan *precision*.

Kueri dan Proses Temu Kembali

Pertama, pengguna membangkitkan atau menentukan sebuah kasus penggunaan sebagai kueri beserta bobot masing-masing elemen struktur kasus penggunaan. Nilai bobot merupakan *integer* dalam interval 1-5.

Tabel 2.5.4.1 Form Masukan Kueri Kasus Penggunaan

No	Elemen Deskripsi Kasus Penggunaan	Teks Kueri	Bobot (1-5)
1	Name		
2	Objective		

3	Actor			
4	Relationship	Association		
5		Include		
6		Extend		
7		Generalization		
8	Precondition			
9	Postcondition			
	Flow of Events			
10		Normal Flow		
11		SubFlow		
12	Alternative or Exceptional Flow of events			

Tingkat atau nilai kesamaan antara masing-masing elemen pada kueri kasus penggunaan dan elemen kasus penggunaan pada koleksi dihitung dengan persamaan berikut:

$$\text{Similarity}(E_{mi}, E_{mj}) = \frac{2[\sum_{k=l}^t (\text{Term}_{mik} W_{ik}) \text{Term}_{mjk}]}{\sum_{k=l}^t \text{Term}_{mik} + \sum_{k=l}^t \text{Term}_{mjk}} \dots\dots\dots (2.1)$$

$E_{mi}$  = elemen m dari *use case* i.

$E_{mj}$  = elemen m dari *use case* j.

$W_{ik}$  = nilai bobot IDF *term* k pada *use case* i.

$\text{Term}_{mik}$  = 1 jika *term* k muncul pada elemen m dari *use case* i, sebaliknya 0

$\text{Term}_{mjk}$  = 1 jika *term* k muncul pada elemen m dari *use case* j, sebaliknya 0

Sedangkan nilai kesamaan kueri kasus penggunaan dengan kasus penggunaan dalam koleksi dihitung berdasar persamaan (2.2).

$$\text{Similarity}(UC_i, Query_j) = \frac{\sum_{m=l}^e [\text{Similarity}(E_{mi}, E_{mj}) \cdot WE_m]}{\text{TotalWeight}} \dots\dots\dots (2.2)$$

$\text{Similarity}(UC_i, Query_j)$  = Skor tingkat kesamaan antara kueri *use case* dari pengguna dan masing-masing *use case* di dalam repositori.

$\text{Similarity}(E_{mi}, E_{mj})$  = Skor kesamaan masing-masing elemen

$WE_m$  = Bobot setiap elemen yang ditentukan secara otomatis berdasar nilai tingkat kesamaan total dari masing-masing elemen kasus penggunaan.

Total Weight = total akumulasi bobot pada semua komponen.



Hasil koleksi kasus penggunaan (*Topmost T similarity score*) kemudian ditampilkan kepada pengguna aplikasi.

Hipotesis dari eksperimen penelitian tersebut adalah temu kembali kasus penggunaan dengan struktur kasus penggunaan lebih efektif daripada temu kembali melalui kata kunci pada kasus penggunaan. Berdasarkan nilai *precision*, nilai *precision improvement* (P+) adalah terdapat pada rating tinggi untuk semua subjek sample. Nilai rata-rata P+ (3) dari (1) dan P+(3) dari (2) adalah 55,78% dan 44,21%.

Namun demikian, pendekatan yang dilakukan pada penelitian tersebut masih mempunyai keterbatasan. Pertama, proses temu kembali hanya memperhitungkan pada kesamaan kata, tetapi tidak memperhitungkan kesamaan semantik. Keterbatasan yang kedua yaitu pengguna masih harus menentukan bobot untuk masing-masing elemen secara manual. Jika pengguna memilih bobot yang tidak sesuai maka rata-rata *recall* menjadi lebih rendah.

## **2.10 Arsitektur dan Teknologi Kakas Bantu Pendukung**

Guna mendukung pengujian dan evaluasi dari metode yang diusulkan dalam tesis ini, dibuat suatu kakas bantu berbasis web. Kakas bantu ini terbagi menjadi tiga komponen utama yaitu;

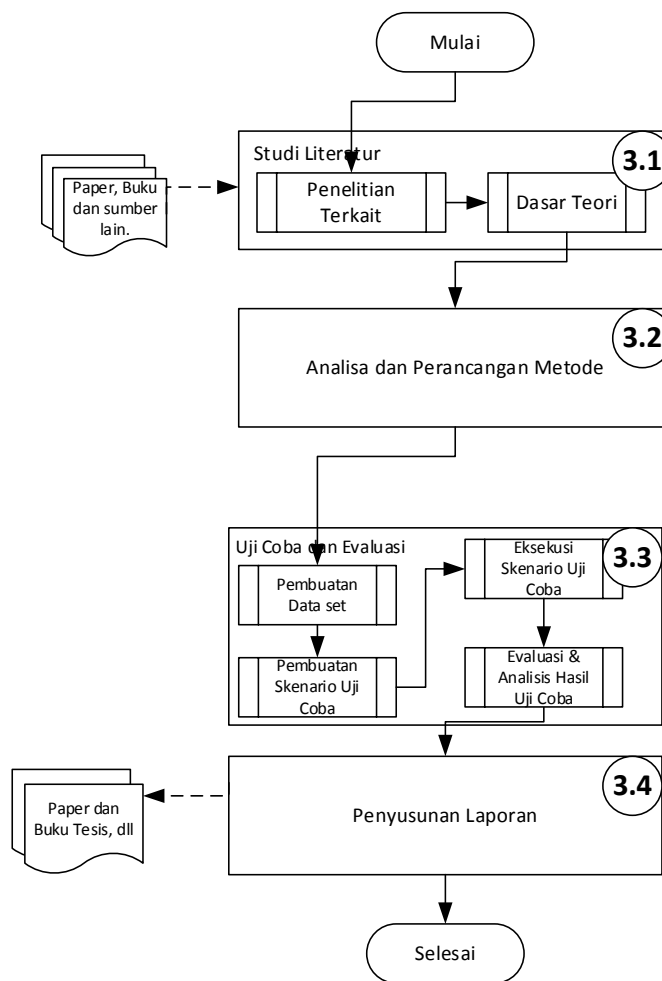
1. Repositori Kasus Penggunaan. Repositori ini dibuat disimpan dalam bentuk basis data. Adapun DBMS yang digunakan yaitu MySQL.
2. Teknologi *Java web service* digunakan untuk melakukan proses perhitungan kesamaan semantik antara dua buah kata atau kalimat. Pemrosesan *Natural Language Processing* (NLP) juga dilakukan pada bagian ini. Teknologi Java dipilih karena mempunyai keunggulan dalam hal kecepatan proses eksekusi dan kehandalan dalam pemrosesan bahasa alamiah.
3. Aplikasi *web-client* menggunakan teknologi PHP: semua masukan maupun luaran dari dan ke pengguna ditangani dalam bagian ini.
  - Masukan: berupa form isian kueri deskripsi kasus penggunaan
  - Luaran: berupa daftar deskripsi kasus penggunaan yang didapatkan kembali.

[Halaman ini sengaja dikosongkan]

## BAB 3

### METODE PENELITIAN

Secara umum, penelitian ini diawali dengan studi literatur yang meliputi penelitian terkait, dasar teori dan analisa. Kemudian dilanjutkan dengan perancangan metode yang meliputi perumusan masalah, usulan metode penyelesaiannya serta diakhiri dengan uji coba dan evaluasi metode. Sedangkan penulisan laporan tesis dimulai pada awal sampai akhir penelitian. Secara lebih detil, penelitian ini dirancang dengan urutan sebagai berikut:



Gambar 3.22.10.1 Alur metodologi penelitian

### 3.1 Studi Literatur

Studi literatur merupakan tahap awal dari penelitian. Pada tahap ini dilakukan studi literatur berkaitan dengan metode-metode dan dasar teori yang

digunakan dalam penelitian, meliputi konsep Kasus Penggunaan, Sistem Temu Kembali Informasi, Metode perhitungan kesamaan teks, dan penelitian-penelitian terkait permasalahan *Use Case Retrieval*. Pada Bab 2 dibahas lebih detil mengenai hasil dari studi literatur yang telah dilakukan.

### 3.2 Analisa dan Perancangan Metode

Pada tahap ini dilakukan analisa menyeluruh mengenai perancangan solusi metode yang dibangun untuk menyelesaikan permasalahan dalam penelitian meliputi: rancangan detil langkah-langkah yang akan ditempuh dan kriteria-kriteria yang akan digunakan. Metode yang diusulkan meliputi tiga tahapan. Gambar 3.2.2 menggambarkan alur proses metode *Use Case Retrieval* yang diusulkan.

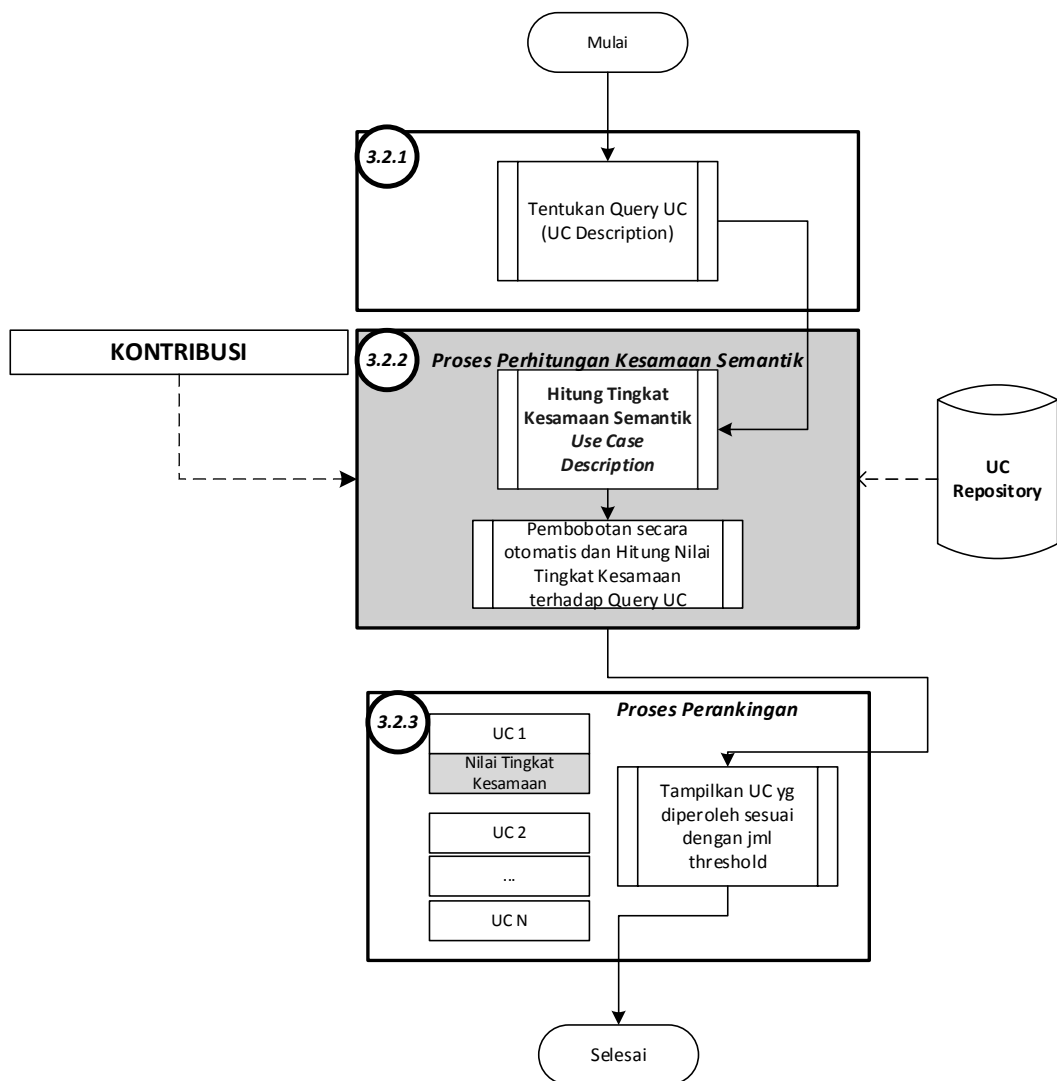
#### 3.2.1 Penentuan Kueri Kasus Penggunaan

Pertama-tama, pengguna menentukan sebuah kasus penggunaan sebagai kueri dengan mengisi form deskripsi kasus penggunaan seperti pada Tabel 3.2.1

Tabel 3.2.1.1 Form kueri berdasar struktur deskripsi kasus penggunaan

No	<i>Use Case Description Elements</i>		<i>Query</i>
1	Name		
2	Objective/Description		
3	Actor		
4	Trigger		
5	Precondition		
6	Postcondition		
7	Flow of Events		
		Normal/Basic Flow Events	
		Alternative or Exceptional Flow of events	
8	Special Requirement		
9	Assumptions		
10	Includes		

Pada form diatas (Tabel 3.2.1.1) terdiri dari sepuluh elemen deskripsi kasus penggunaan yang harus diisikan oleh pengguna.

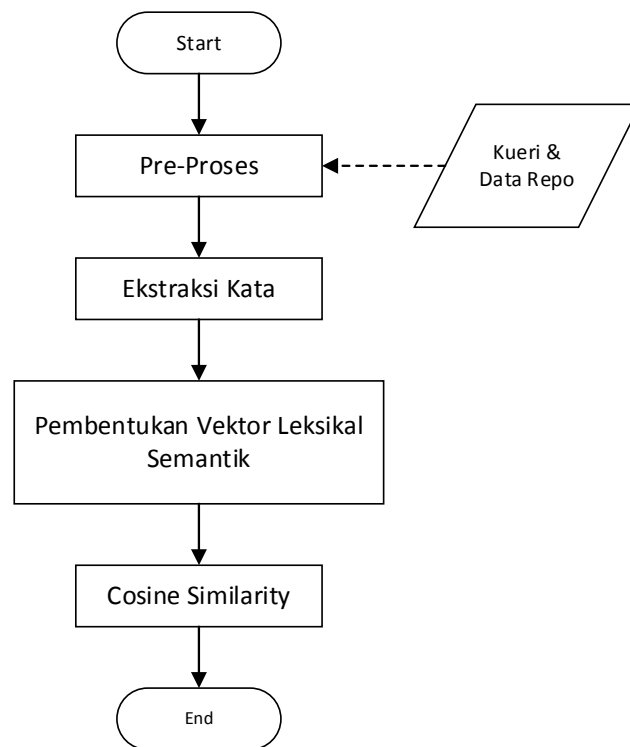


Gambar 3.2.2 Alur proses temu kembali Kasus Penggunaan

### 3.2.2 Perhitungan Kesamaan Semantik

Setelah kueri deskripsi kasus penggunaan ditentukan, proses selanjutnya yaitu melakukan perhitungan kesamaan kata secara semantik antara kueri deskripsi kasus penggunaan dan deskripsi kasus penggunaan di dalam repositori. Dalam proses ini, kami mengusulkan menggunakan *Wordnet* dalam melakukan proses perhitungan kesamaan semantik (Liu & Wang, 2013).

Gambar 3.2.3 menunjukkan proses perhitungan kesamaan semantik pada kalimat atau frasa.



Gambar 3.2.3 Alur metodologi perhitungan kesamaan semantik

Proses perhitungan kesamaan semantik terbagi menjadi empat bagian utama:

1. Pre-Proses  
Meliputi proses transformasi kalimat ke dalam *bag-of-word* atau tokenisasi.
2. Ekstraksi Kata  
Proses pelabelan kata sesuai jenis kata (*Part of Speech Tagging*).
3. Pembentukan Vektor Leksikal Semantik menggunakan wordnet
4. Menghitung nilai Cosine Similarity

Kesamaan semantik antara dua kata  $w_1$  dan  $w_2$  dinotasikan sebagai  $sim(w_1, w_2)$ . Nilai kesamaan ini didapat melalui analisa dari sebuah *lexical knowledge base* seperti *Wordnet* dimana kata-kata diorganisir ke dalam sebuah *synonym sets* (*synsets*).

- Jika kedua target kata adalah identik atau terdapat di dalam synset yang sama maka nilai kesamaannya adalah 1.
- Jika salah satu target kata tidak terdapat di dalam *Wordnet*, maka nilai kesamaannya adalah 0,

- Jika kedua target kata tidak terdapat di dalam synset yang sama tetapi masih di dalam hirarki *Wordnet*, maka nilai kesamaan dihitung berdasarkan relevansi *node* di dalam hirarki.

Berikut dijeaskan contoh proses penghitungan kesamaan kalimat menggunakan dua buah kalimat pada Tabel 3.2.2 diambil dari salah satu elemen deskripsi kasus penggunaan (misal: *use case objective/description* atau tujuan kasus penggunaan), satu *query use case* oleh pengguna dan satu lagi diambil dari *use case* pada *repository*.

Tabel 3.2.2.1 Contoh kalimat-kalimat pada use-case.

Kalimat S <sub>1</sub> ( <i>Query UC Objective</i> )	Kalimat S <sub>2</sub> ( <i>UC Objective Repo</i> )
Consumers would still have to get a descrambling security card from their cable operator to plug into the set.	To watch pay television, consumers would insert into the set a security card provided by their cable service.

Untuk setiap data set, *Wordnet* digunakan sebagai sumber untuk menghitung kesamaan kata. Untuk dapat menggunakan *Wordnet*, pertama kali kita harus mengubah kalimat S<sub>1</sub> dan S<sub>2</sub> ke dalam representasi *bag-of-word*.

$T_1 = \{\text{kata-kata di dalam } S_1\}$

$T_2 = \{\text{kata-kata di dalam } S_2\}$

Kita juga membentuk sebuah superset T, yang merupakan gabungan dari T<sub>1</sub> dan T<sub>2</sub>

$T = T_1 \cup T_2$ .

Tabel 3.2.2.2 *Sentence preprocessing (Bag-of-word)*

Notasi	Kumpulan Kata
T <sub>1</sub>	{"consumer","would","still","have","to","get","a","descramble","security","card","from","their","cable","operator","to","plug","into","the","set"}
T <sub>2</sub>	{"to","watch","pay","television","consumer","would","insert","into","the","set","a","security","card","provide","by","their","cable","service"}
T	{"consumer","would","still","have","get","a","descramble","security","card","from","their","cable","operator","to","plug","into","the","set","watch","pay","television","insert","provide","by","service"}

Langkah selanjutnya yaitu pembentukan vektor.

Tujuan dari pembentukan T, yang mengandung seluruh kata dari T<sub>1</sub> dan T<sub>2</sub> adalah untuk membentuk vector semantik bagi T<sub>1</sub> dan T<sub>2</sub>. Vektor yang diturunkan dari gabungan kumpulan kata disebut lexical semantik vektor yang dinotasikan dengan symbol V<sub>1</sub> dan V<sub>2</sub>. Dimensi dari lexical semantik vektor adalah sesuai dengan jumlah kata-kata pada gabungan kumpulan kata.

Tabel 3.2.2.3 Nilai Masukan dari V<sub>1</sub> dan V<sub>2</sub>

Nilai dari V <sub>1</sub>	Nilai dari V <sub>2</sub>
("a","a"): 1.0,	("a","a"): 1.0,
("by","a"): 0,0,	("by","by"): 1.0,
("cable","cable"): 1.0,	("cable","cable"): 1.0,
("card","card"): 1.0,	("card","card"): 1.0,
("consumer","consumer"): 1.0,	("consumer","consumer"): 1.0,
("descrambling","descrambling"): 1.0,	("descrambling","a"): 0,0,
("from","from"): 1.0,	("from","a"): 0,0,
("get","get"): 1.0,	("get","pay"): 0,807
("have","get"): 1.0,	("have","pay"): 0,836
("insert","insert"): 0,935	("insert","insert"): 1.0,
("into","into"): 1.0,	("into","into"): 1.0,
("operator","operator"): 1.0,	("operator","card"): 0,0,
("pay","have"): 0,836	("pay","pay"): 1.0,
("plug","plug"): 1.0,	("plug","set"): 0,995
("provide","set"): 0,999	("provide","provide"): 1.0,
("security","security"): 1.0,	("security","security"): 1.0,
("service","set"): 0,996	("service","service"): 1.0,
("set","set"): 1.0,	("set","set"): 1.0,
("still","still"): 1.0,	("still","television"): 0,792
("television","cable"): 0,999	("television","television"): 1.0,
("the","the"): 1.0,	("the","the"): 1.0,
("their","their"): 1.0,	("their","their"): 1.0,
("to","to"): 1.0,	("to","to"): 1.0,
("watch","set"): 0,857	("watch","watch"): 1.0,
("would","would"): 1.0	("would","would"): 1.0



Tabel 3.2.2.4 Nilai Vektor dari V<sub>1</sub> dan V<sub>2</sub>

Vektor	Nilai Vektor
V <sub>1</sub>	(1.0, 0,0, 1.0, 1.0, 1.0, 1.0,1.0,1.0,1.0,0,935,1.0,1.0,0,836,1.0,0,999,1.0,0,996,1.0,1.0,0,999,1.0,1.0,1.0,0,857,1.0)
V <sub>2</sub>	(1.0,1.0,1.0,1.0,1.0,0,0,0,0,0,807,0,836,1.0,1.0,0,0,1.0,0,995,1.0,1.0,1.0,1.0,0,792,1.0,1.0,1.0,1.0,1.0,1.0)

Tabel 3.2.2.5 Nilai Vektor Akhir dari V<sub>1</sub> dan V<sub>2</sub>

Vektor	Nilai Vektor
V <sub>1</sub>	(0,2,0,0,1.0,1.0,1.0,1.0,0,2,1.0,1.0,0,935,0,2,1.0,0,836,1.0,0,999,1.0,0,996,1.0,0,2,0,999,0,2,0,2,0,2,0,857,0,2)
V <sub>2</sub>	0,2,0,2,1.0,1.0,1.0,0,0,0,0,0,807,0,836,1.0,0,2,0,0,1.0,0,995,1.0,1.0,1.0,1.0,0,792,1.0,0,2,0,2,0,2,1.0,0,2)

Sehingga nilai dari *cosine similarity* antara S<sub>1</sub> dan S<sub>2</sub> dapat dihitung dengan persamaan:

$$\text{Sim}(S_1, S_2) = \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|} \dots\dots\dots (3.1)$$

$$\text{Sim}(S_1, S_2) = 0,907$$

Khusus untuk elemen deskripsi kasus penggunaan *Normal Flows/Basic Flows*, perhitungan nilai kesamaan juga mempertimbangkan urutan dari aliran deskripsinya. Aliran normal no.1 dari kueri kasus penggunaan dibandingkan dengan aliran normal no.1 deskripsi kasus penggunaan di repositori. Aliran normal no.2 dari kueri deskripsi kasus penggunaan dibandingkan dengan aliran normal no.2 kasus penggunaan di repositori dan seterusnya.

### 3.2.3 Proses Perankingan dan *Threshold*

Langkah terakhir yaitu proses perankingan hasil kueri kasus penggunaan berdasarkan nilai kesamaan atau kemiripan secara semantik. Sebelum melakukan proses perankingan diperlukan pencarian nilai *threshold* yang ideal untuk

menentukan atau memvalidasi data hasil kueri. Nilai *threshold* ini didapat dari hasil pengujian melalui pengukuran indeks Kappa. *Threshold* dengan nilai rata-rata Kappa yang tertinggi dari hasil masing-masing kueri pada rentang 0,1 sampai dengan 0,9 akan diambil sebagai *threshold*.

#### **3.2.4 Skenario Pengujian**

Pengujian dilakukan dengan tiga skenario yang berbeda:

- Pengujian untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem terhadap penilaian pakar menggunakan indeks Kappa.
- Pengujian kueri kasus penggunaan dengan menggunakan kesamaan semantik di salah satu elemen atau item dari *use case description*.
- Pengujian *query use case* dengan kombinasi masukan elemen atau item *use case* yang berbeda-beda.

## **BAB 4**

### **HASIL PENELITIAN DAN PEMBAHASAN**

Di dalam bagian ini dibahas perihal hasil penelitian dan pembahasan yang meliputi beberapa bagian yaitu: pembahasan variabel pengujian, hasil pengujian dan analisa evaluasi hasil pengujian. Lingkungan pengujian terdiri dari empat aspek yaitu meliputi Koleksi Deskripsi Kasus Penggunaan, Pakar atau Ahli, Proses Pencarian dan Kueri Deskripsi Kasus Penggunaan. Hasil pengujian memperlihatkan data statistik yang didapat dari proses eksekusi kueri deskripsi kasus penggunaan menggunakan metode yang berbeda-beda yang dilakukan oleh para ahli.

Tahap yang terakhir yaitu evaluasi meliputi analisa pengukuran hasil kueri deskripsi kasus penggunaan oleh pakar terhadap hasil yang didapatkan oleh sistem. Tujuan utama dari eksperimen adalah untuk menguji asumsi dari penelitian yaitu apakah temu kembali kasus penggunaan berdasar struktur dengan memasukkan aspek semantik lebih efektif dan akurat dibandingkan dengan temu kembali berdasar struktur tanpa memasukkan aspek semantik. Pengukuran indeks Kappa digunakan untuk mengetahui tingkat keakuratan dari pendekatan yang dilakukan pada penelitian ini

#### **4.1 Lingkungan Pengujian**

Eksperimen ini dirancang untuk menghilangkan bias, sehingga mempunyai beberapa faktor yang harus dikontrol. Faktor-faktor ini terbagi menjadi empat variabel yaitu koleksi deskripsi kasus penggunaan, pakar atau ahli, proses pencarian dan kueri deskripsi kasus penggunaan. Berikut penjelasan selengkapnya mengenai empat variabel kontrol tersebut:

##### **4.1.1. Koleksi Deskripsi Kasus Penggunaan**

Sebagaimana dijelaskan di Bab 1, dataset penelitian ini diambil dari beberapa *textbook* seputar pemodelan kasus penggunaan. Dataset dari *textbook* ditulis dan digambarkan ulang ke dalam bentuk diagram kasus penggunaan beserta deskripsinya menggunakan suatu kakas bantu. Total kasus penggunaan yang dapat

diambil sebagai dataset sejumlah 137 kasus penggunaan dari 20 sistem atau projek aplikasi.

Tabel 4.1.1.1 Daftar dataset berdasar nama sistem dan sumber referensi.

No	SID	Sistem	Sumber Referensi
1	TS	Trading System	UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition By Martin Fowler Publisher : Addison Wesley Pub Date : September 15, 2003 ISBN : 0-321-19368-7 Pages : 208
2	TCS	TimeCard System	Enterprise Java with UML by CT Arlington, 2001 John Wiley & Sons, Inc.
3	TRS	Theater Management System	UML Bibble by Tom Pender ISBN:0764526049 John Wiley - Sons 2003 (940 pages)
4	PMS	The project management system	Learning UML By Sinan Si Alhir Publisher : OReilly Pub Date : July 2003 ISBN : 0-596-00344-7 Pages : 252 Slots : 1
5	STS	The simple telephone system	Use Case Modeling By Kurt Bittner, Ian Spence Publisher : Addison Wesley Pub Date : August 21, 2002 ISBN : 0-201-70913-9 Pages : 368
6	BS	The Bookstore	Fast Track UML 2.0 b y Kendall Scott ISBN:1590593200 Apress 2004
7	ACME	the ACME Super ATM	Use Case Modeling By Kurt Bittner, Ian Spence Publisher : Addison Wesley Pub Date : August 21, 2002 ISBN : 0-201-70913-9 Pages : 368
8	PS	Payroll System	<a href="http://faculty.kfupm.edu.sa/ICS/mwaslam/Rich%20Files/doc/Payroll%20System%20Use-Case%20Model.htm">http://faculty.kfupm.edu.sa/ICS/mwaslam/Rich%20Files/doc/Payroll%20System%20Use-Case%20Model.htm</a>
9	OSS	Order Shipment Management System	UML Weekend Crash Course by Thomas A. Pender Published by Wiley Publishing, Inc. Copyright 2002 by Wiley LOC: 2002103278 ISBN: 0-7645-4910-3

10	OPS	Order Processing System	IBM - UNITED <a href="http://www.ibm.com/developerworks/rational/library/content/legacy/parttwo/1000/0670/0670_Schneider_Ch07.pdf">http://www.ibm.com/developerworks/rational/library/content/legacy/parttwo/1000/0670/0670_Schneider_Ch07.pdf</a>
11	OBS	Online Books Store	Enterprise Java with UML by CT Arlington, 2001 John Wiley & Sons, Inc.
12	HOLIS	Home Lighting Automation System	Managing Software Requirements: A Use Case Approach, Second Edition By Dean Leffingwell, Don Widrig Publisher : Addison Wesley Pub Date : May 05, 2003 ISBN : 0-321-
13	CBS	Cinema Booking System	www.itu.dk Course: Modelling and Design (fall 2007) Submitted by: Toomas Kutt(toomas@itu.dk), Fraz Tabassam(tabassam@itu.dk), Jens kaae Christensen(jenschr@itu.dk)
14	ARIS	ARIS System	www.opuscollege.net
15	REMS	Real Estate Management System	Use Cases: Requirements in Context, Second Edition By Daryl Kulak, Eamonn Guiney Publisher : Addison Wesley Pub Date : July 25, 2003 ISBN : 0-321-15498-3 Pages : 272
16	IME	Instant Messaging Encryption	Use Cases: Requirements in Context, Second Edition By Daryl Kulak, Eamonn Guiney Publisher : Addison Wesley Pub Date : July 25, 2003 ISBN : 0-321-15498-3 Pages : 272
17	HR	Hotel Reservation	UML 2 for Dummies by Michael Jesse Chonoles and James A. Schardt ISBN:0764526146 Hungry Minds 2003 (412 pages)
18	HMS	Hotel Management System	Aspect-Oriented Software Development with Use Cases By Ivar Jacobson, Pan-Wei Ng Publisher : Addison Wesley Professional Pub Date : December 30, 2004 ISBN : 0-321-26888-1 Pages : 464
19	HD	HomeDirect	Developing Enterprise Java Applications with J2EE and UML by Khawar Zaman Ahmed, Cary E. Umrysh Paperback: 288 pages ; Dimensions (in inches): 0,66 x

			9.28 x 7.32 Publisher: Addison-Wesley Pub Co; ISBN: 0201738295; 1st edition (December 15, 2001)
20	ERS	e-retail system	Building Web Applications with UML Second Edition By Jim Conallen Publisher : Addison Wesley Pub Date : October 04, 2002 ISBN : 0-201-73038-3 Pages : 496

Tabel 4.1.1.2 Domain kategori Kasus Penggunaan

ID	Domain	Keterangan	Jumlah	
			Sistem	UC
A	Finantial Calculation	Domain sistem terkait permasalahan transaksi keuangan	3	20
B	e-Commerce/Trading System	Domain terkait system transaksi jual beli barang dan jasa.	5	37
C	Information Management System	Domain terkait sistem informasi secara umum atau general	8	54
D	Teaching-Studying System	Domain terkait sistem manajemen kegiatan belajar mengajar secara umum.	1	13
E	Other Unspecified Domain	Domain lainnya diluar domain No 1 hingga 4.	3	13
Jumlah			20	137

Pada Tabel 4.1.2 menunjukkan pembagian domain sistem serta sebaran jumlah sistem dan kasus penggunaan untuk masing-masing domain pada dataset. Terdapat 20 sistem dengan total kasus penggunaan 137 yang diambil dari 20 sumber *textbook* berbahasa inggris.

Tabel 4.1.1.3 Sebaran kueri deskripsi kasus penggunaan berdasar kondisi setiap fitur/atribut deskripsi kasus penggunaan.

QID	Name	Actor	Desc.	Pre-Cond	Post-Cond	Norm Flw	Alt Flw	Assm	Inc	Note	Trigg
1	√	√	√	√	√	√					
2	√	√	√	√	√	√					

3	√	√	√	√		√					
4	√	√	√	√	√	√					
5	√	√									
6	√		√								
7	√										
8	√	√				√					
9	√	√									
10	√		√								
11						√					
12	√	√	√	√	√	√					
13	√		√			√					
14	√					√					
15	√	√	√			√					
16	√	√		√		√					
17	√	√		√		√					
18	√	√		√		√					
19	√	√		√		√					
20	√	√		√		√					
	19	14	9	10	4	15	0	0	0	0	0

Terdapat sekitar 20 kueri deskripsi kasus penggunaan yang disusun oleh pakar yang turut berpartisipasi dalam penelitian ini.

#### 4.1.2. Pakar atau ahli

Penelitian ini dilakukan oleh dua orang peneliti baik sebagai pakar penyusun deskripsi kasus penggunaan maupun sebagai pengguna atau penguji kakas bantu. Dua orang pakar tersebut masih atau sedang mengambil program pasca sarjana di bidang Rekayasa Perangkat Lunak. Oleh karena itu kedua orang tersebut dianggap mempunyai kemampuan dan pengalaman di bidang pemodelan kasus penggunaan di dalam pengembangan perangkat lunak dan pengetahuan mereka tentang Bahasa Inggris sangat memadai dengan nilai TOEFL ITP lebih dari 500.

#### **4.1.3. Proses Pencarian**

Sekitar 20 sistem yang terbagi menjadi 5 domain telah dibuat guna menguji metode penelitian yang diusulkan sebagaimana terdapat pada Tabel 4.1.1 dan Tabel 4.1.2. Pengguna atau pakar penguji membuat beberapa kombinasi kueri kasus penggunaan seperti pada Tabel 4.1.3 kemudian memasukkan dan mengeksekusi kueri kasus penggunaan tersebut ke dalam sistem kakas bantu. Kueri dijalankan atau dieksekusi pada 2 macam kondisi dataset yaitu dataset yang homogen dan dataset yang heterogen. Hasil kueri atau jawaban yang didapat kemudian diidentifikasi secara detil untuk menghitung indeks Kappa terhadap hasil yang didapat oleh ahli yang lain.

Tujuan dari dibuatnya penggolongan domain yang berbeda yaitu untuk mengidentifikasi atau membandingkan tingkat efektifitas dari metode yang diusulkan dengan metode yang lainnya.

#### **4.1.4. Kueri Kasus Penggunaan**

Pakar membuat total sekitar 20 pernyataan kueri baik kueri berdasar berdasar struktur kasus penggunaan untuk semua subjek domain: A, B, C, D dan E sesuai pada Tabel 4.1.2. Di dalam suatu kueri kasus penggunaan dapat berisi kueri lengkap untuk semua komponen kasus penggunaan, kombinasi beberapa komponen atau minimal satu komponen saja yang diisikan. Kueri tersebut dieksekusi pada dataset dengan dua domain yang berbeda, domain yang homogen dan domain yang heterogen.

Itulah keempat variabel kontrol yang dijadikan acuan di dalam tahap pengujian. Pakar yang ditunjuk dapat melakukan kueri kasus pengguna berdasar variabel kontrol tersebut. Dalam penelitian ini ditunjuk dua pakar untuk melakukan pengujian. Masing-masing pakar diharuskan mengeksekusi 20 kueri dan memberikan evaluasi terhadap hasil kueri yang didapatkan. Kueri-kueri tersebut dieksekusi menggunakan suatu kakas bantu berbasis web yang khusus dibangun untuk memfasilitasi proses pengujian tersebut.



## 4.2 Hasil Pengujian

### 4.2.1. Pengujian Pertama

Pengujian pertama dilakukan untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem dengan hasil yang didapat dari para pakar menggunakan indeks Kappa. *Threshold* yang digunakan yaitu pada rentang 0,1 sampai dengan 0,9.

Tabel 4.2.1.1 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.1 sampai dengan no.6

No.Kueri dan Nilai Kappa						
Threshold	1	2	3	4	5	6
0,1	-0,973	-0,885	-0,950	-0,941	-0,949	-0,895
0,2	-0,837	-0,808	-0,899	-0,941	-0,949	-0,895
0,3	0,559	0,553	-0,536	-0,827	-0,949	-0,895
0,4	0,754	0,839	0,121	-0,039	-0,949	-0,507
0,5	0,863	0,911	0,436	0,468	-0,831	-0,142
0,6	0,896	0,942	0,727	0,720	-0,193	0,000
0,7	0,972	0,942	0,869	0,985	0,469	0,619
0,8	0,987	0,942	0,966	0,971	0,735	0,929
0,9	0,987	0,942	0,983	0,971	0,957	0,947

Tabel 4.2.1.2 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.7 sampai dengan no.13

No.Kueri dan Nilai Kappa							
T	7	8	9	10	11	12	13
0,1	-0,966	-0,959	-0,835	-0,830	0,826	-0,931	-0,931
0,2	-0,966	-0,959	-0,835	-0,580	0,845	-0,931	-0,931
0,3	-0,966	-0,919	-0,835	-0,448	0,901	-0,931	-0,931
0,4	-0,966	-0,791	-0,567	0,093	0,953	-0,431	-0,931
0,5	-0,966	0,444	-0,127	0,836	0,969	-0,183	-0,638
0,6	-0,697	0,629	0,176	0,877	0,969	-0,100	0,003
0,7	0,102	0,858	0,690	0,918	0,954	0,252	0,403
0,8	0,723	0,990	0,865	0,924	0,970	0,946	0,947
0,9	0,966	0,979	0,917	0,914	0,970	0,966	0,966

Tabel 4.2.1.3 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.14 sampai dengan no.20

No.Kueri dan Nilai Kappa							
T	14	15	16	17	18	19	20
0,1	-0,931	-0,949	-0,975	-0,975	-0,971	-0,975	-0,981
0,2	-0,931	-0,949	-0,975	-0,975	-0,971	-0,975	-0,868
0,3	-0,883	-0,949	-0,975	-0,975	-0,971	-0,975	0,050
0,4	-0,080	-0,545	-0,975	-0,975	-0,971	-0,975	0,050
0,5	0,160	-0,131	-0,975	-0,874	-0,971	-0,975	0,413
0,6	0,198	0,243	-0,975	0,280	-0,971	-0,558	0,648
0,7	0,570	0,485	-0,404	0,987	-0,944	0,754	0,693
0,8	0,964	0,848	0,657	1.000	-0,173	0,987	0,951
0,9	0,966	0,966	0,961	1.000	0,888	1.000	1.000

#### 4.2.2. Pengujian Kedua

Pengujian kueri deskripsi kasus penggunaan dengan kombinasi masukan komponen deskripsi kasus penggunaan yang berbeda-beda. Ada dua deskripsi Kasus Penggunaan yang digunakan dalam skenario uji coba yang kedua ini yaitu “Withdraw Cash” dan “Program Switch”.

Tabel 4.2.2.1 Deskripsi Kasus Penggunaan “Withdraw Cash”

<b>Test Case ID:</b>	1
<b>System Name:</b>	Financial or Banking System (eg. ATM)
<b>Use Case Name:</b>	Withdraw Cash
<b>Actors:</b>	Customer
<b>Description:</b>	This use case describes how the Bank Customer uses the ATM to withdraw money to his/her bank account
<b>Preconditions:</b>	There is an active network connection to the Bank. The ATM has cash available.
<b>Postconditions:</b>	The user has received their cash and the internal logs have been updated.
<b>Normal Flow:</b>	1. The use case begins when Bank Customer inserts their Bank Card. 2. Use Case: Validate User is performed.

	<p>3. The ATM displays the different alternatives that are available on this unit. In this case the Bank Customer always selects Withdraw Cash.</p> <p>4. The ATM prompts for an account.</p> <p>5. The Bank Customer selects an account.</p> <p>6. The ATM prompts for an amount.</p> <p>7. The Bank Customer enters an amount.</p> <p>8. Card ID, PIN, amount and account is sent to Bank as a transaction. The Bank Consortium replies with a go/no go reply telling if the transaction is ok.</p> <p>9. Then money is dispensed.</p> <p>10, The Bank Card is returned.</p> <p>11. The receipt is printed.</p> <p>12. The use case ends successfully.</p>
<b>Alternative Flows:</b>	
<b>Exceptions:</b>	
<b>Includes:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

Tabel 4.2.2.2 Hasil Pengujian pada *use case* “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah Result Set	Rata-rata Nilai Kesamaan
1	√							142	0,879139
2	√	√						143	0,728252
3	√	√	√					142	0,628841
4	√	√	√	√				143	0,517633
5	√	√	√	√	√			143	0,435039
6	√	√	√	√	√	√		67	0,284859
7	√	√	√	√	√	√	√	78	0,279775

Tabel 4.2.2.3 Deskripsi Kasus Penggunaan “Program Switch”

<b>Test Case ID:</b>	-
<b>System Name:</b>	
<b>Use Case Name:</b>	Program Switch
<b>Actors:</b>	Home owner or Programmer
<b>Description:</b>	Homeowner or programmer uses central control or PC program to set the behavior of individual light switches in home.
<b>Preconditions:</b>	Homeowner / Programmer has account on system
<b>Postconditions:</b>	Switch is programmed
<b>Normal Flow:</b>	<p>5.0 Program Switch at Central Control Unit</p> <ol style="list-style-type: none"> <li>1. Push program button</li> <li>2. Enter passcode</li> <li>3. Select room switch</li> <li>4. Select light banks to assign to switch</li> <li>5. Add effects, such as dimming and timing</li> <li>6. Save selection</li> </ol>
<b>Alternative Flows:</b>	<p>5.1 Program Switch at PC</p> <ol style="list-style-type: none"> <li>1. Open Lumenations program</li> <li>2. Input authentication</li> <li>3. Select programming mode</li> <li>4. Select switch to program from list</li> <li>5. Select available light banks from list</li> <li>6. Add dimming and timing effects</li> <li>7. Save programming</li> </ol>
<b>Exceptions:</b>	
<b>Includes:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

Tabel 4.2.2.4 Hasil Pengujian pada *use case* “Program Switch” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah Result Set	Rata-rata Nilai Kesamaan
1	√							142	0,857331
2	√	√						142	0,762236
3	√	√	√					142	0,65222
4	√	√	√	√				142	0,538716
5	√	√	√	√	√			142	0,454539
6	√	√	√	√	√	√		129	0,427066
7	√	√	√	√	√	√	√	121	0,361189

#### 4.2.3. Pengujian Ketiga

Pengujian kedua yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kata-kata yang mempunyai kesamaan semantik di salah satu komponen *use case description*.

Tabel 4.2.3.1 Beberapa contoh kata-kata yang mempunyai kesamaan semantik.

No	Kumpulan Kata
1	Recipe, formula
2	Money, fund, cash, coin
3	Select, pick
4	Withdraw, take
5	weapon, gun, pistol
6	Purchase, buy
7	Client, user, customer

Tabel 4.2.3.2 Hasil kueri semantik dengan menggunakan kata frase “Pick gun” pada komponen nama *use case*.

Nama Kasus Penggunaan	Aktor	Nilai Kesamaan
<i>Select weapon</i>	Player	1
Program Switch	Home owner or Programmer	0,915154
Find a Book	Customer	0,893591

Add course block	study administrator	0,876727
Remote Programming	Lumenations Services	0,866828
Select Performance	Venue Manager	0,856239
Add Book to Shopping Cart	Customer	0,846452
Select Payment Method	Employee	0,837092
Place Local Call	Caller and Callee	0,83051
Initiate Emergency Receiver	Resident	0,826237

Tabel 4.2.3.3 Hasil kueri non-semantik(*String-based similarity-dice coefficient*) dengan menggunakan kata frase “Pick gun” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Check Out	Customer	0,02963
Check out	Customer	0,02963
Ticket Print	End User Customer	0,02469
Stock Product	Stock Clerk	0,02339
Check In Customer	Hotel Counter Staff	0,01932
Add course block	study administrator	0,01932
Check Out Customer	Hotel Counter Staff	0,01852
Check Order Status	Customer	0,01852
Check order status	Customer	0,01852
Check room availability	Potential Guest	0,01587

Tabel 4.2.3.4 Hasil kueri semantik dengan menggunakan kata frase “Get coin” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Withdraw Cash	Customer	0,894362
Transfer Money	Customer	0,883077
Transfer Money	customer	0,883077
Deposit Money	Customer	0,856384
Establish Cash Flow Schedule	Operations manager	0,849773
Edit Profile	customer	0,838265
Create Charge Code	Administrative User	0,836104
Write Customer Review	Customer	0,811509
Generate Report	Printer	0,807165

Create Employee	Administrative User	0,804943
-----------------	---------------------	----------

Tabel 4.2.3.5 Hasil kueri non-semantik (*String-based similarity-dice coefficient*) dengan menggunakan kata frase “Get coin” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Ticket Print	End User Customer	0,03704
Enroll in courses	Applicant Administrator	0,02899
Set Limits	Trading Manager	0,02778
Set Timing Sequence	Home owner Programmer	0,02778
Set Up Users	Operations manager	0,02614
Manage Account	Customer	0,02222
Register course	study administrator	0,02116
Login	Customer Customer Rep	0,0202
Get Call History	Customer	0,0202
Add course block	study administrator	0,01932

Tabel 4.2.3.6 Hasil kueri semantik dengan menggunakan kata frase “Add Formula” pada komponen nama *use case*

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Add Recipe	User	0,967743
Add courses to study year	study administrator	0,85786
Add course block	study administrator	0,841139
Update Account	Accounting System	0,839413
Schedule Event	Venue Manager	0,812303
List Transactions	customer	0,80347
Cancel Event	Venue Manager	0,801246
Ticket Print	End User Customer	0,793081
Record Time	Employee	0,787899
Give Product Information	Inventory System	0,786366

Tabel 4.2.3.7 Hasil kueri non-semantik dengan menggunakan kata “Add Formula” pada komponen nama *use case*

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Schedule Performance	Venue Manager	0,03065

Perform Search	Customer	0,02899
Reschedule Performance	Venue Manager	0,02867
Cancel Performance	Venue Manager	0,02778
Search for Order	Customer Customer Rep	0,02667
Select Performance	Venue Manager	0,02469
Add Recipe	User	0,02339
Record Time	Employee	0,02222
Enter Payment Information	Customer	0,0202
Give Product Information	Inventory System	0,0202

### 4.3 Analisa dan Evaluasi Hasil Pengujian

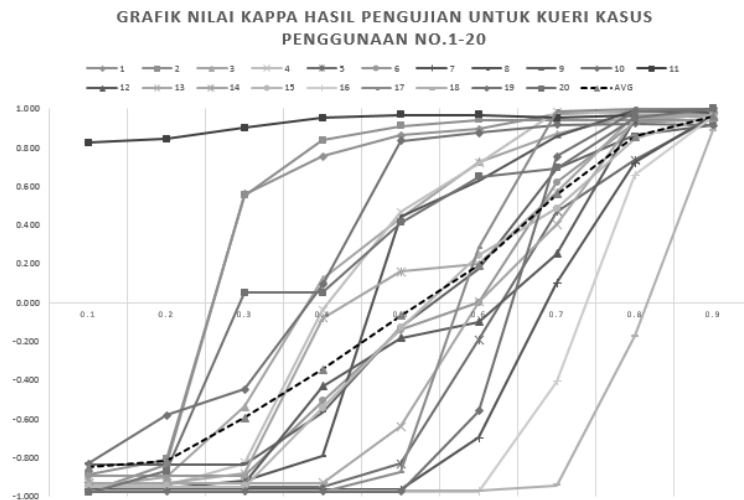
Dari hasil pengujian penelitian tersebut pada Bab 4.2 kemudian kita analisa menggunakan pengukuran indeks Kappa guna mengetahui atau mengukur tingkat kesepakatan dari dua pengamat atau ahli terhadap karakteristik yang menjadi perhatian penelitian.

#### 4.3.1. Analisa Hasil Pengujian Pertama

Tabel 4.3.1.1 Nilai *threshold* dan rata-rata nilai indeks Kappa dari keseluruhan data uji kueri deskripsi kasus penggunaan.

Nilai Threshold	Rata-rata nilai koefisien Kappa
0,1	-0,849
0,2	-0,817
0,3	-0,595
0,4	-0,345
0,5	-0,066
0,6	0,191
0,7	0,559
0,8	0,856
<b>0,9</b>	<b>0,962</b>



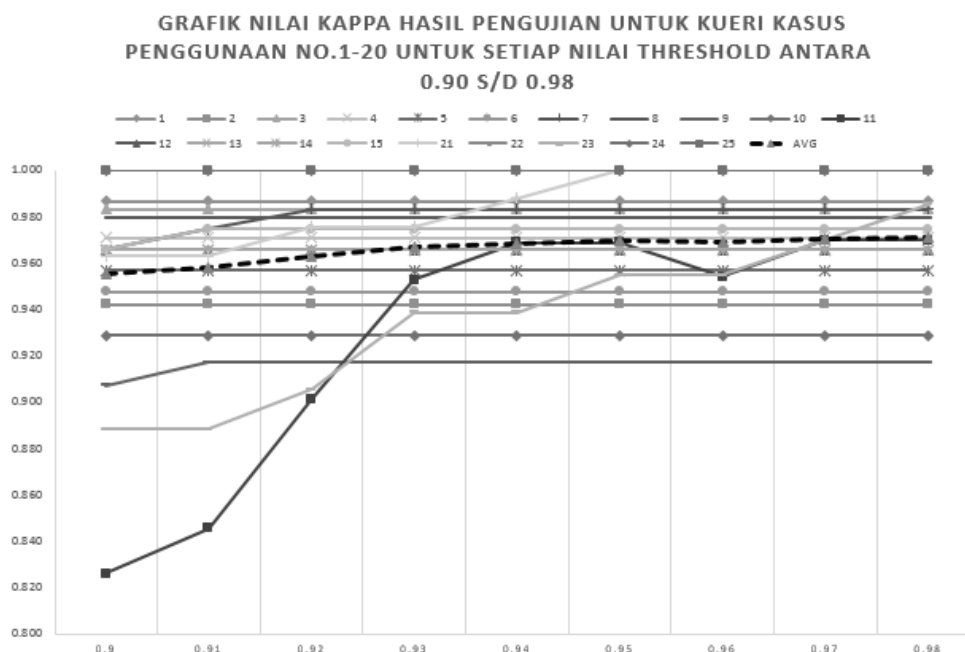


Gambar 4.3.1 Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,1 sampai dengan 0,9.

Hasil pengujian pertama terlihat pada Tabel 4.3.2 yaitu didapat suatu nilai rata-rata indeks Kappa keseluruhan dari hasil kueri kasus penggunaan yaitu sebesar 0,962 dengan nilai *threshold* ideal yang didapat yaitu sebesar 0,9. Nilai ini berarti bahwa metode yang diusulkan mempunyai tingkat kesepakatan terhadap pakar yang tinggi.

Tabel 4.3.1.2 Rata-rata Kappa untuk setiap nilai *threshold* antara 0,90 sampai dengan 0,98.

Threshold (T)	Rata-rata Kappa
0,90	0,955
0,91	0,958
0,92	0,962
0,93	0,967
0,94	0,968
0,95	0,969
0,96	0,969
0,97	0,970
0,98	0,971



Gambar 4.3.2 Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,1 sampai dengan 0,9.

#### 4.3.2. Analisa Hasil Pengujian Kedua

Pengujian kedua dilakukan dengan tujuan untuk mengetahui sejauh mana pengaruh kombinasi masukan pada kueri kasus penggunaan terhadap nilai *similarity*/kesamaan yang dihasilkan. Di pengujian ini diambil contoh salah satu kasus penggunaan yaitu “Withdraw Cash”. Kasus Penggunaan ini kemudian dijadikan kueri dengan kombinasi masukan kueri seperti yang ditampilkan pada Tabel 4.3.4.

Dari hasil pengujian pada Tabel 4.3.4 terlihat bahwa untuk memperoleh hasil kueri yang akurat, diperlukan isian kueri yang lengkap. Semakin lengkap masukan komponen kueri yang dimasukkan hasil yang didapatkan semakin akurat. Hal ini ditunjukkan dengan nilai rata-rata kesamaan yang semakin kecil.

Dari contoh kasus penggunaan “Withdraw Cash”. Jika kita hanya mengisikan komponen aktor saja maka didapat nilai rata-rata kesamaan yang dengan nilai 0,88. Hal ini berbanding terbalik jika kita memasukkan kueri secara lengkap yaitu didapatkan nilai rata-rata kesamaan sebesar 0,28.

Tabel 4.3.2.1 Hasil Pengujian pada use case “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah	Rata-rata
								Result	Nilai
								Set	Kesamaan
1	√							142	0,879139
2	√	√						143	0,728252
3	√	√	√					142	0,628841
4	√	√	√	√				143	0,517633
5	√	√	√	√	√			143	0,435039
6	√	√	√	√	√	√		67	0,284859
7	√	√	√	√	√	√	√	78	0,279775

Pada contoh yang lain yaitu dengan menggunakan kasus penggunaan “Program Switch” juga dihasilkan hasil yang serupa. Pada Tabel 4.2.5 terlihat bahwa semakin kita mengisi lengkap komponen kueri yang ada maka hasil kueri yang kita dapatkan akan semakin akurat. Hal ini ditunjukkan dengan nilai rata-rata kesamaan yang dihasilkan semakin kecil.

### 4.3.3. Analisa Hasil Pengujian Ketiga

Pengujian ketiga ini, yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kata-kata yang mempunyai kesamaan semantik. Pengujian ini dilakukan untuk mengetahui perbandingan antara hasil kueri dengan menggunakan kesamaan semantik dan non-semantik.

Berdasarkan hasil pengujian pada Tabel 4.2.8 pada contoh kata “Pick gun” terlihat bahwa kata “Pick gun” mempunyai kesamaan semantik dengan kata “Select weapon” dengan nilai kesamaan yang didapat yaitu 1. Hal ini menunjukkan bahwa hasil pencarian yang didapatkan relevan dengan input yang dimasukkan. Sebaliknya hasil pencarian atau kueri non-semantik tidak dapat menghasilkan luaran yang sesuai atau dengan kata lain data yang didapatkan tidak relevan sebagaimana ditunjukkan pada Tabel 4.2.9.

Hasil yang sama juga didapatkan pada contoh kata “Get coin”. Pada Tabel 4.2.10 terlihat hasil kueri dengan nilai kesamaan tertinggi 1 yaitu “Withdraw cash”.

Sedangkan pada hasil kueri non-semantik pada Tabel 4.2.11, kata “Withdraw cash” tidak didapatkan. Secara semantik kata “Get” memiliki kemiripan dengan kata “Withdraw” sedangkan kata “Coin” memiliki kemiripan makna dengan kata “Cash”. Hal ini menunjukkan bahwa penggunaan aspek semantik di dalam proses temu kembali deskripsi kasus penggunaan lebih akurat dibandingkan dengan non-semantik (*String-based similarity*).

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di dalam pengembangan perangkat lunak, banyak permasalahan yang berakar pada keterbatasan pemahaman pengembang di dalam memahami kebutuhan pengguna terhadap perangkat lunak yang dibangun. Hal ini bisa disebabkan oleh berbagai faktor diantaranya yaitu keterbatasan data dan informasi yang didapatkan pada waktu proses pengumpulan, penganalisaan, penspesifikasian, verifikasi dan validitas kebutuhan dari perangkat lunak yang hendak dibangun (Siahaan, 2012). Oleh karena itu proses tersebut memerlukan usaha, waktu dan biaya tidak sedikit.

Pada saat ini, terdapat persaingan yang sangat ketat diantara para pengembang di industri perangkat lunak. Para pengembang mencari cara untuk mengurangi biaya proses pengembangan perangkat lunak sebanyak mungkin guna menekan biaya produksi sehingga memperoleh keuntungan dalam pasar.

Oleh karena itu penggunaan kembali komponen perangkat lunak seperti pada penelitian (Mcclure, 1997) dan (Krueger, 1992) dinilai lebih efektif memberikan dampak positif. Untuk mengurangi biaya dan alokasi waktu pada tahap ini berbagai pemodelan kebutuhan digunakan oleh para analis. Diantaranya yaitu melalui pemodelan kasus penggunaan seperti pada penelitian (Gomaa, 2000). Kasus penggunaan dipilih karena merupakan diagram yang seringkali digunakan dan dianggap efektif membantu di dalam proses analisa kebutuhan perangkat lunak. Dengan alasan keterbatasan waktu, sering kali para analis memanfaatkan kasus penggunaan yang pernah mereka buat untuk merancang kasus penggunaan atau spesifikasi kebutuhan yang baru. Dalam hal ini mereka menggunakan konsep *reuse* atau *information retrieval* di dalam analisa kebutuhan. Akan tetapi di dalam penerapannya timbul permasalahan diantaranya yaitu bagaimana mengukur tingkat kesamaan diantara kasus penggunaan.

## 1.2 Penelitian Terdahulu

Berbagai penelitian telah dilakukan terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak ini yaitu diantaranya yang dilakukan oleh (M. C. Blok, and J. L. Cybulski, 1998), (Saeki, Reusing use case descriptions for requirements specification: towards use case patterns, 1999), (Saeki, Patterns and aspects for use cases reuse techniques for use case descriptions, 2000), (H. G. Woo and W. N. Robinson, 2002), (Marir & Haouam, 2004), (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) dan (Srisura & Daengdej, 2010). Namun dari hasil penelitian-penelitian tersebut masih terdapat beberapa kekurangan seperti; proses yang kompleks, respon yang lambat dari *retrieval system* dan adanya proses yang masih dilakukan secara manual. Di dalam penelitiannya, (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) mengusulkan pendekatan *software requirement specification retrieval* dalam bentuk deskripsi kasus penggunaan menggunakan struktur *use case* dan komputasi kesamaan antara *term of use case query* dan *use case* di dalam repository.

Pada penelitian yang lain, (Srisura & Daengdej, 2010) mengusulkan menggunakan metode *case-based reasoning* di dalam menyelesaikan permasalahan pemanggilan kembali pengalaman-pengalaman sebelumnya untuk mendukung penggunaan *use case diagram*. Untuk membuat perangkat lunak yang baru, spesifikasi kebutuhan yang sudah ada/lama tersimpan di dalam suatu repository diambil kembali berdasarkan kesamaan diagram kasus penggunaan. Diagram kasus penggunaan dan spesifikasi kebutuhan yang memiliki tingkat kesamaan yang paling tinggi akan dijadikan sebagai pedoman atau acuan guna menentukan spesifikasi perangkat lunak yang baru. Akan tetapi di dalam penelitian tersebut masih terdapat kekurangan terkait mekanisme pengukuran kesamaan actor dan use case. Pencocokan kata masih menggunakan terbatas pada “*exactly matching word*” atau pencocokan kata yang sama persis.

(H. G. Woo and W. N. Robinson, 2002) mengusulkan teknik otomatisasi yang berfungsi untuk membantu proses penggunaan kembali diagram kasus penggunaan. Mereka mengembangkan kaskas bantu yang disebut “ScenAsst” untuk mengumpulkan dan mendapatkan kembali use case secara praktis. ScenAsst mengubah isi dari *use case* ke dalam bentuk graph, meng-clusternya dan

menyimpannya ke dalam sebuah koleksi *use case*. Di dalam proses *retrieval*, query dari user juga diubah ke dalam bentuk graph. Derajat kesamaan antara *query user* dan *graph* yang dihasilkan di dalam koleksi penyimpanan dibandingkan dengan menggunakan algoritma “SUBDUE”. Teknik ini menghasilkan respon proses yang lama karena faktor kompleksitas. (M. C. Blok, and J. L. Cybulski, 1998).

### **1.3 Perumusan Masalah**

Pada bagian ini akan dipaparkan mengenai permasalahan yang akan menjadi perhatian dalam penelitian ini. Penjabaran akan dibagi menjadi 3 bagian yakni pembahasan input dan output, garis permasalahan yang hendak diselesaikan dalam penelitian ini dan disertai dengan penentuan terhadap batasan masalah, sehingga penelitian lebih terarah.

#### **1.3.1. Masukan dan Luaran**

Masukan dari implementasi metode yang diusulkan yaitu berupa *query use case* dalam format *UC description*. Sedangkan luaran yang dihasilkan berupa urutan ranking UC pada *UC Repository* berdasar parameter dan threshold *query UC* yang dicari.

#### **1.3.2. Permasalahan**

Penelitian sebelumnya (Udomchaiorn, Prompoon, & Kanongchaiyos, 2006) menerangkan bahwa masih terdapat permasalahan terkait metode *Use Case Retrieval* yaitu pada kurangnya tingkat akurasi pada mekanisme pengukuran kesamaan kata dan kalimat dan metode pembobotan fitur yang masih dilakukan *manual* oleh pengguna. Oleh karena itu, penelitian ini mengusulkan solusi terhadap permasalahan tersebut yaitu berupa optimasi pada pengukuran kesamaan diagram *use case* dengan memasukkan aspek kesamaan semantik atau arti pada kata, frase dan kalimat menggunakan pendekatan *knowledge-based similarity* dan pembobotan fitur secara otomatis sesuai tingkat atau nilai akumulasi kesamaan fitur secara keseluruhan.

Beberapa permasalahan dalam penelitian ini dirumuskan sebagai berikut:

1. Bagaimana menambahkan aspek perhitungan kesamaan kata atau kalimat secara semantik atau arti kata di dalam temu kembali kasus penggunaan?
2. Bagaimana mengevaluasi atau mengukur tingkat akurasi dan efektifitas dari metode pengukuran kesamaan kasus penggunaan?

### 1.3.3. Batasan Masalah

Dalam penelitian ini, batasan masalah yang dibahas diuraikan sebagai berikut:

1. Bahasa yang digunakan dalam pemodelan kasus penggunaan adalah Bahasa Inggris.
2. Objek yang diteliti yaitu deskripsi kasus penggunaan dalam bentuk teks.

### 1.3.4. Sumber Data

Kasus penggunaan yang digunakan dalam penelitian ini sebagian besar berasal dari proyek-proyek akademik dan contoh-contoh pemodelan kasus penggunaan.

Tabel 1.3.4.1 Sumber Data

Jenis Sumber	Referensi
<i>Text book</i>	<ol style="list-style-type: none"> <li>1. (Siahaan, 2012)</li> <li>2. UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, Martin Fowler , 2003</li> <li>3. Managing Software Requirements: A Use Case Approach, Second Edition, Dean Leffingwell, Don Widrig , 2003</li> <li>4. UML Weekend Crash Course Thomas A. Pender, 2002</li> <li>5. Use Case Modeling, Kurt Bittner, Ian Spence , 2002</li> <li>6. Fast Track UML 2.0, Kendall Scott, Apress, 2004</li> <li>7. Enterprise Java with UML, CT Arlington, 2001</li> </ol>



	8. Use Case Driven Object Modeling with UML Theory and Practice, Doug Rosenberg, Matt Stephens, apress, 2007
--	--

Sumber data disimpan di dalam repositori dalam bentuk dokumen terstruktur (*structured document*).

Langkah-langkah ekstraksi dataset:

- Dataset dari *text book*, proses ekstraksi dilakukan secara manual dengan cara memasukkan deskripsi kasus penggunaan yang bersesuaian dengan menggunakan UML Modelling tool.
- Setelah semua data terekam dalam bentuk diagram kasus penggunaan beserta deskripsinya, data tersebut kemudian diekspor ke dalam bentuk file XMI.
- Diagram kasus penggunaan dalam format XMI tersebut kemudian di-*import* ke dalam repository basis data guna pengolahan atau proses berikutnya.

#### 1.4 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah untuk mengoptimasi atau meningkatkan akurasi temu kembali dokumen spesifikasi kebutuhan perangkat lunak melalui deskripsi kasus penggunaan pada penelitian sebelumnya oleh (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006).

Manfaat dari penelitian ini, yaitu dihasilkannya solusi yang lebih optimal dalam proses temu kembali spesifikasi kebutuhan perangkat lunak melalui konsep penggunaan kembali (*reuse*) deskripsi kasus penggunaan sehingga didapatkan efisiensi dan efektifitas waktu proses rekayasa kebutuhan.

#### 1.5 Kontribusi Penelitian

Adapun kontribusi dalam penelitian ini yaitu:

1. Penambahan perhitungan kesamaan kata dan kalimat secara semantik menggunakan *text similarity based on semantic nets* pada *use case description* untuk meningkatkan tingkat akurasi.

2. Pengembangan kakas bantu dan repositori kasus penggunaan untuk mempercepat dan mempermudah pengujian metode yang diusulkan.

## **BAB 2**

### **KAJIAN PUSTAKA**

Dalam subbab ini diuraikan kajian pustaka penelitian terkait perhitungan kesamaan semantik dan struktur pada kasus penggunaan di dalam analisa spesifikasi kebutuhan perangkat lunak.

#### **2.1 Pemodelan Kasus Penggunaan**

Melalui pemodelan Kasus Penggunaan (*Use-Case Modeling*), kebutuhan dasar aktor diidentifikasi dan bagaimana kebutuhan tersebut akan dipenuhi oleh sistem dijelaskan. Aktor merepresentasikan kelas pengguna dari sistem. Setiap kelas pengguna memiliki alasan yang spesifik mengapa ia menggunakan sistem. Alasan ini juga mewakili suatu *goal* dari setiap seorang pengguna ketika ia menggunakan sistem. Hubungan interaksi antara aktor dan sistem inilah yang kita sebut sebagai kasus penggunaan atau *Use Cases*. Kasus penggunaan menjelaskan fungsi-fungsi atau layanan-layanan yang harus dilakukan sistem dari perspektif aktor-aktor tertentu. Kasus-kasus penggunaan tersebut menangkap kebutuhan-kebutuhan dan membantu melacak realisasi dari kebutuhan tersebut selama proses pengembangan perangkat lunak (Siahaan, 2012).

##### **2.1.1. Komponen-komponen dalam Kasus Penggunaan**

Dari mengetahui semua komponen yang ada di dalam kasus penggunaan, pengembang dapat mengetahui siapa saja aktor-aktor yang berhubungan dengan sistem dan apa saja yang dilakukan sistem. Berikut ini adalah komponen-komponen yang terdapat pada pemodelan kasus penggunaan:

###### **a. Aktor**

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah seseorang atau sesuatu yang dapat berinteraksi dengan sistem, yaitu siapa dan apa yang menggunakan sistem. Aktor mempresentasikan peran atau profesi, bukan pemakai individu dari sistem. Aktor memiliki nama. Nama yang dipilih seharusnya menyatakan peran atau profesi aktor.

## b. Kasus Penggunaan

Kasus penggunaan (*Use-Case*) merupakan deskripsi dari interaksi antara aktor dan sistem yang ditujukan untuk memenuhi kebutuhan pengguna. Suatu kasus penggunaan merupakan deskripsi langkah-langkah interaksi antara aktor dan sistem untuk mencapai hasil tertentu. Kasus penggunaan seringkali merepresentasikan fungsional yang dapat dilakukan oleh aktor dari sistem. Kasus penggunaan juga merefleksikan tujuan atau goal seorang aktor ketika menggunakan sistem (Siahaan, 2012).

## c. Relasi dalam Kasus Penggunaan

Relasi digunakan sebagai penghubung antar aktor dengan kasus penggunaan. Jenis relasi dijelaskan pada sub bab berikutnya.

### 2.1.2. Relasi Dalam Diagram Kasus Penggunaan

#### a. Asosiasi

Asosiasi, yaitu hubungan statis antar elemen. Umumnya menggambarkan element yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain.

#### b. Generalisasi dan spesialisasi

Spesialisasi, yaitu hubungan hirarkis antar elemen. Elemen dapat diturunkan dari elemen lain dan mewarisi semua atribut dan metode elemen asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari elemen yang diwarisinya. Kebalikan dari spesialisasi adalah generalisasi.

#### c. Ketergantungan atau *Dependency*

Ketergantungan adalah suatu jenis hubungan yang menandakan bahwa satu elemen, atau kelompok elemen, bertindak sebagai klien tergantung pada unsur lain atau kelompok elemen yang berlaku sebagai penyalur. Ini merupakan suatu hubungan lemah yang menandakan bahwa jika penyalur klien diubah maka klien secara otomatis akan terpengaruh oleh perubahan tersebut. Ini merupakan suatu hubungan searah. Ketergantungan terbagi menjadi dua, yaitu *include* dan *extend*. *Include* adalah relasi dimana kasus penggunaan yang di-*include*-kan adalah bagian dari proses kasus penggunaan utama. Sedangkan *extend* adalah relasi yang

memungkinkan suatu kasus penggunaan memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh kasus penggunaan lainnya.

## 2.2 Deskripsi Kasus Penggunaan

Deskripsi Kasus Penggunaan merupakan suatu dokumen yang mendeskripsikan detail-detail dari kasus penggunaan di dalam diagram kasus penggunaan dan ditulis dalam bentuk bahasa alamiah. Komponen-komponen dasar dari deskripsi kasus penggunaan yang mencakup isi-isi penting dan biasa digunakan terdiri dari sembilan komponen yaitu:

1. *Use Case Name: Use case name* merupakan nama dari kasus penggunaan. Setiap kasus penggunaan harus mempunyai nama guna membedakan antara kasus penggunaan satu dengan yang lainnya.
2. *Objective*: Merupakan suatu elemen yang mendeskripsikan tujuan dari kasus penggunaan.
3. *Actor*: Merupakan seseorang atau sesuatu/peran diluar sistem yang berinteraksi dengan sistem.
4. *Relationship*: Merupakan koneksi semantik antar elemen-elemen model. Di dalam deskripsi kasus penggunaan, hubungan terdiri dari *association*, *include*, *extend* dan *generalization*. Komponen-komponen hubungan mengandung nama kasus penggunaan dari kasus penggunaan yang terkait.
5. *Pre-Condition: Precondition* adalah sebuah konstrain yang harus bernilai *true* ketika kasus penggunaan dipanggil.
6. *Post-Condition: Postcondition* merupakan konstrain yang harus bernilai *true*/benar ketika kasus penggunaan berakhir.
7. *Normal flow of events*: Merupakan elemen dari kasus penggunaan yang menjelaskan implementasi secara normal tanpa adanya *error*/kesalahan.
8. *Subflow*: Dibeberapa kasus, aliran normal dapat didekomposisi menjadi beberapa subflow.
9. *Alternative or exceptional flow of events*: Merupakan elemen dari kasus penggunaan yang menjelaskan implementasi alternatif.

## 2.3 Sistem Temu Kembali Informasi

*Information Retrieval System* (IRS) merupakan sistem yang digunakan untuk menemukan kembali (*retrieve*) dokumen yang relevan terhadap kebutuhan *user* dari suatu kumpulan informasi berdasarkan kata kunci atau *keywords* atau *query* dari *user*. Selain menemukan dokumen yang relevan terhadap *query*, IRS juga melakukan perankingan terhadap hasil pencarian tersebut. Suatu dokumen yang memiliki ranking yang lebih tinggi daripada dokumen lain akan dianggap lebih relevan terhadap *query* (M. J. McGill and G. Salton, 1983).

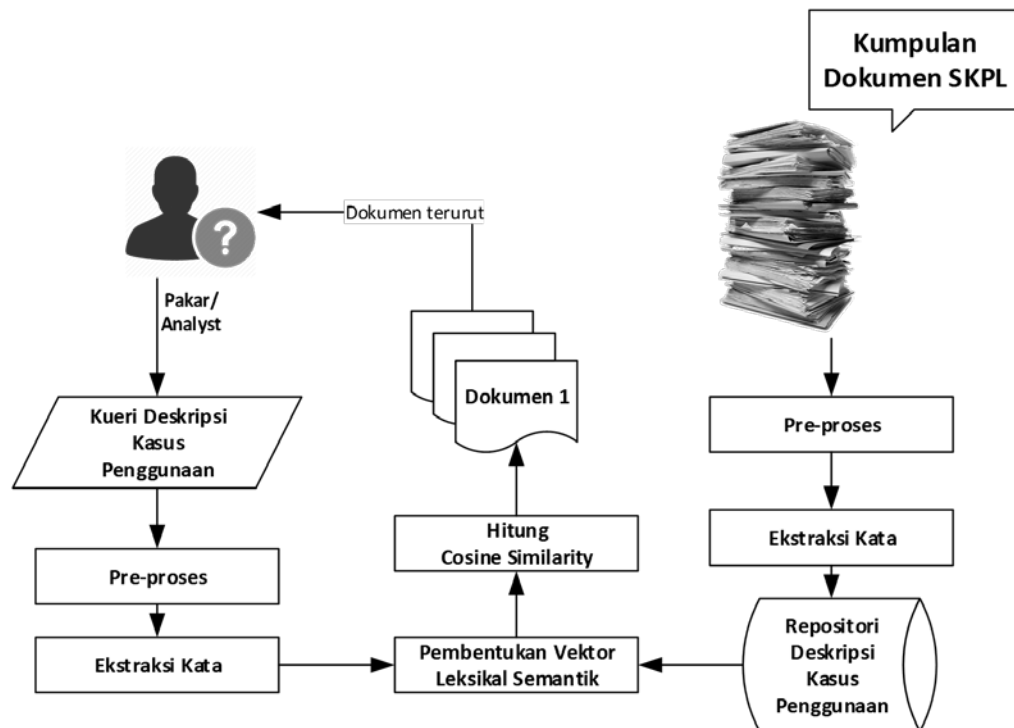
### 2.3.1. Cara Kerja Sistem Temu Kembali Informasi

Model sistem temu kembali informasi menentukan detail sistem temu kembali. Informasi yaitu meliputi representasi dokumen maupun kueri, fungsi pencarian (*retrieval function*) dan notasi kesesuaian (*relevance notation*) dokumen terhadap kueri. Salah satu model sistem temu kembali informasi yang paling awal digunakan adalah model *boolean*. Model boolean mempresentasikan dokumen sebagai suatu himpunan kata-kunci (*set of keywords*). Sedangkan kueri direpresentasikan sebagai ekspresi *boolean*. Kueri dalam ekspresi *boolean* merupakan kumpulan kata kunci yang saling dihubungkan melalui operator *boolean* seperti *AND*, *OR* dan *NOT* serta menggunakan tanda kurung untuk menentukan *scope* operator. Hasil pencarian dokumen dari model *boolean* adalah himpunan dokumen yang relevan.

Gambar 2.3.1 menunjukkan proses temu kembali secara umum.

1. *Text Operations* (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam kueri maupun dokumen dalam pentransformasian dokumen atau kueri menjadi *terms index* (indeks dari kata-kata).
2. *Query Formulation* (formulasi terhadap kueri) yang memberi bobot pada indeks kata-kata kueri.
3. *Ranking*, mencari dokumen-dokumen yang relevan terhadap kueri dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan kueri.
4. *Indexing*, membangun data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen, sistem temu balik informasi menerima kueri dari pengguna, kemudian melakukan perankingan terhadap pada koleksi

berdasarkan kesesuaiannya dengan kueri. Hasil perangkaian yang diberikan kepada pengguna merupakan dokumen yang sistem, relevan dengan kueri, namun relevansi dokumen terhadap suatu kueri merupakan penilaian pengguna yang subyektif dan dipengaruhi banyak faktor.



Gambar 2.3.1 Cara kerja sistem temu kembali informasi

#### 2.4 Pemrosesan Bahasa Alamiah (*Natural Language Processing*)

Pemrosesan bahasa alamiah adalah metode yang digunakan untuk melakukan proses komputasi dari bahasa sehingga terciptanya interaksi antara manusia dengan komputer menggunakan bahasa alamiah. Salah satu alat yang digunakan dalam pemrosesan Bahasa alamiah adalah StanfordNLP. StanfordNLP adalah sistem yang dikembangkan oleh The Stanford NLP Group dengan fokus penelitian pada *sentence understanding*, *probabilistic parsing and tagging*, *biomedical information extraction*, *grammar introduction*, *word sense disambiguation*, dan *automatic question answering* (nlp.stanford.edu, 2013).

### 2.4.1 Ekstraksi Fakta

Ekstraksi fakta bertujuan untuk mengambil fakta-fakta yang ada pada setiap kalimat yang diinputkan dengan cara memotong kalimat menjadi klausa-klausa. Kalimat adalah satuan bahasa berupa kata atau rangkaian kata yang dapat berdiri sendiri dan menyatakan makna yang lengkap. Kalimat mengungkapkan pikiran yang utuh, baik dengan cara lisan maupun tulisan. Proses untuk ekstraksi fakta dilakukan dengan dua tahap yaitu POS tagging dan Parsing.

### 2.4.2 Part of Speech Tagging

*Part of Speech Tagging* (POS Tagging) adalah proses untuk mendeteksi *part of speech* (*verb*, *adverb*, *personal pronoun*) dari tiap token dengan menggunakan kamus *tag* dan model latin. Keluaran dari proses POS tagging menggunakan model *Penn Treebank*. POS tagging memberikan label kata secara otomatis pada suatu kalimat. POS tagging dapat diklasifikasikan menjadi delapan jenis yaitu *noun* (kata benda), *pronoun* (kata ganti), *adjective* (kata sifat), *verb* (kata kerja), *adverb* (kata keterangan), *preposition* (kata depan), *conjunction* (kata penghubung) dan *interjection* (kata seru).

Tabel 2.4.2.1 Penandaan dari POS Tagging

Tag	Deskripsi	Tag	Deskripsi
CC	Coordinating conjunction	UH	Interjection
CD	Cardinal number	VBD	Verb, past tense
DT	Determiner	VBZ	Verb, 3rd ps.sing. present
EX	Existential there	MD	Modal
FW	Foreign word	WDT	Wh-determiner
VB	Verb, base form	NN	Noun, singular or mass
IN	Preposition/subordinate conjunction	WP	Wh-pronoun
JJ	Adjective	NNP	Proper noun, singular
VBG	Verb, gerund/present participle	WPS	Possesive wh-pronoun
JJR	Adjective, comparative	NNPS	Proper noun, plural
VBN	Verb, past participle	NNS	Noun, plural
JJS	Adjective, superlative	PDT	Predeterminer



VBP	Verb, non-3rd ps. sing. Present	POS	Possesive ending
LS	List item marker	PRP	Personal pronoun
RP	Particle	PRP\$	Possesive pronoun
SYM	Symbol	RB	Adverb
TO	To	RBR	Adverb, comparative
RBS	Adverb, superlative	WRB	Wh-adverb

### 2.4.3 Penguraian Kalimat (*Parsing*)

Penguraian Kalimat (*Parsing*) adalah suatu proses menguraikan kalimat berdasarkan struktur kalimat dan tata bahasa yang menggunakan *Penn TreeBank* sebagai dasarnya. Gramatika yang dipakai juga sangat berkaitan dengan proses penguraian kalimat apa yang digunakan. Penguraian kalimat dilakukan dengan cara memecah-mecah suatu rangkaian masukan yang akan menghasilkan suatu pohon urai yang akan digunakan pada tahap analisis semantik.

Tabel 2.4.3.1 Label pada proses penguraian

Label	Deskripsi
ADJP	Adjective Phrase
ADVP	Adverb Phrase
CONJP	Conjunction Phase
INTJ	Interjection
LST	List Maker
PP	Prepositional Phrase
PRT	Particle
SBAR	Clause Introduced by a Subordinating conjunction
UCP	Unlike Coordinate Phrase
VP	Verb Phrase
NP	Noun Phrase

### 2.4.4 Tokenisasi

Tokenisasi adalah tahap pemotongan kata berdasarkan tiap kata yang menyusunnya. Sebelum melakukan tokenisasi, setiap huruf dalam dokumen harus diubah ke dalam huruf kecil. Hanya huruf „a” sampai dengan „z” yang diterima.

Karakter-karakter tertentu seperti digit, angka, tanda hubung, dan tanda baca akan dihilangkan. Berikut contoh hasil proses tokenisasi:

K1 : “The”, “system”, “must”, “be”, “able”, “to”, “receives”, “many”, “request”, “in”, “a”, “time”.

#### 2.4.5 Penghapusan *Stopword*

*Stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Pendekatan *bag-of-words* merupakan penyederhanaan asumsi yang digunakan pada pemrosesan bahasa alami dan temu kembali informasi. Dalam pendekatan ini, sebuah teks (misalnya sebuah kalimat atau sebuah dokumen) dipresentasikan sebagai sebuah koleksi kalimat yang tidak berurutan, serta tidak mengindahkan tata bahasa dan urutan kata.

*Stopwords* perlu dihapus karena dua alasan, yaitu menghemat tempat dan waktu pencarian, serta untuk menghindari kesalahan pemaknaan oleh *parser*. Sebagai contoh, “karena” dan ”sebab” adalah dua *stopword* yang memiliki makna yang sama.

Tabel 2.4.5.1 Contoh *stopword* dalam bahasa Inggris

<b><i>Stopword</i> dalam bahasa Inggris</b>
<i>a, about, above, after, again, all, am, before, being, cannot, could, during, each, few, has, have, her, here, how, I, into, is, let’s, me, more, dan lain-lain.</i>

Tahap penghapusan *stopwords* atau penyaringan adalah tahap mengambil kata-kata penting dari hasil tokenisasi. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) arau *wordlist* (menyimpan kata penting). Contoh hasil penghapusan *stopword* dapat dilihat di Tabel 2.4.4.

Tabel 2.4.5.2 Contoh penghapusan *stopword*

<b>Teks Masukan</b>	<b>Teks Luaran</b>
“An”, “administrator”, “has”, “the”, “task,”, ”of”, “configuring”, “and “, “maintaining”, “the”, “system”	Administrator task configuring maintaining system

### 2.4.6 Stemming

Stemming adalah tahap mencari akar kata dari tiap kata hasil penyaringan. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama. Algoritma *stemming* yang digunakan adalah *wordnet stemmer*. Algoritma ini merupakan algoritma berbasis kamus yang menggunakan kamus *wordnet* sebagai acuannya. *Stemming* dilakukan berdasarkan aturan yang telah ditentukan. Bila kata hasil *stemming* tersebut ada di dalam kamus *wordnet*, maka kata yang ditemukan pada kamus *wordnet* yang menjadi keluaran dari *WordNet stemmer*. Namun, jika kata tersebut tidak ditemukan dalam kamus *wordnet*, maka kata yang dimasukkan akan dianggap sebagai luaran dari *Wordnet Stemmer*. Aturan-aturan *stemming* untuk tiap *part of speech* dalam *wordnet stemmer* dapat dilihat pada Tabel 2.4.5.

Tabel 2.4.6.1 Aturan untuk *noun* pada *WordNet Stemmer*

Part of Speech	Suffix	Replacement
Noun	S	NULL
	ses	s
	xes	x
	ches	ch
	shes	sh
	Ies	y
	Ves	f
	Zes	z
	Men	man

Untuk setiap *part of speech noun*, bila kata masukan memiliki akhiran seperti pada kolom *suffix*, maka akhiran awal akan diganti dengan akhiran seperti pada kolom *replacement*. Contoh *stemming* untuk kata benda dapat dilihat pada Tabel 2.4.6.

Tabel 2.4.6.2 Contoh stemming untuk kata benda

Kata Masukan	Kata Luaran
Application	application
Taxes	Tax

Ashes	Ash
Lives	Life
Women	Woman

Tabel 2.4.6.3 Aturan untuk verb pada WordNet *Stemmer*

Part of Speech	Suffix	Replacement
Verb	s	Null
	ies	Y
	es	e
	es	Null
	ed	e
	ing	e
	ing	Null

Untuk setiap *part of speech* kata kerja, bila kata masukan memiliki akhiran seperti pada kolom *suffix*, maka akan di akhiran awal akan diganti dengan akhiran seperti pada kolom *replacement*. Contoh stemming untuk *verb* dapat dilihat di tabel 2.4.8

Tabel 2.4.6.4 Contoh *stemming* untuk kata kerja

Kata Masukan	Kata Luaran
Performs	Perform
Clustering	Cluster
Worked	Work
Living	Live
Presenting	Present

## 2.5 Teknik Pengukuran Kesamaan Kata dan Kalimat

Pengukuran derajat atau tingkat kesamaan antar kata, kalimat, paragraf dan dokumen merupakan komponen yang penting di dalam berbagai disiplin ilmu seperti: *information retrieval*, *document clustering*, *word-sense disambiguation*, *automatic essay scoring*, *short answer grading*, *machine translation* dan *text*

*summarization*. Berikut beberapa teknik atau metode perhitungan tingkat kesamaan teks yang dihasilkan dari beberapa penelitian terakhir:

Tabel 2.4.6.1 Beberapa pendekatan atau metode pengukuran kesamaan teks.

No	Kategori	Sub-Kategori	Nama Metode
1	<i>String-Based Similarity</i> : Pengukuran kesamaan berdasarkan pada rangkaian atau urutan <i>string</i> dan komposisi karakter.	<i>Character-Based Similarity</i>	LCS
			Dameraue-Levenshtein
			Jaro
			Jaro-Winkler
			Needleman-Wunsch
			Smith-Waterman
			N-gram
		<i>Term-Based</i>	Blok Distance
			Cosine Similarity
			Dice's Coefficient
			Euclidean Distance
			Jaccard Similarity
			Matching Coefficient
Overlap Coefficient			
2	<i>Corpus-Based Similarity</i> : Pengukuran kesamaan semantic berdasar informasi yang didapatkan dari basis data <i>corpora</i> . (misal: <i>Brown Corpus</i> )	HAL	
		LSA	GLSA
		ESA	CL-ESA
		PMI-IR	SOC-PMI
		NGD	
		DISCO	DISCO1
			DISCO2
3	<i>Knowledge-Based Similarity</i> : Pengukuran kesamaan semantik berdasar tingkat kesamaan antar kata-kata menggunakan informasi yang didapatkan dari <i>semantic network</i> seperti; <i>WordNet</i> .	<i>Similarity</i>	<i>Information Content</i>
			<ul style="list-style-type: none"> <li>• res</li> <li>• lin</li> <li>• jcn</li> </ul>
			<i>Path Length</i>
			<ul style="list-style-type: none"> <li>• lch</li> <li>• wup</li> <li>• path</li> </ul>
		<i>Relatedness</i>	hso
			lesk
		vector	
4	<i>Hybrid Similarity</i>	<i>Semantic Text Similarity (STS)</i>	

### 2.5.1 String-Based Similarity

*String-Based Similarity* adalah pengukuran kesamaan string yang beroperasi pada urutan string dan komposisi karakter. Sebuah metrik *string* adalah sebuah metrik yang mengukur tingkat kesamaan maupun ketidaksamaan antara dua buah teks *string* untuk pencocokan atau perbandingan secara akurat. *String-Based*

*Similarity* terbagi menjadi dua kategori atau tipe yaitu *Character-Based* dan *Term-Based Similarity*.

*Character-Based Similarity Measures*:

- Algoritma *Longest Common SubString* (LCS) memperhitungkan kesamaan antara dua buah string berdasar pada panjang *contiguous chain* dari karakter-karakter yang muncul pada kedua string.
- *Damerau-Levenshtein* mendefinisikan jarak antara dua buah string dengan menghitung jumlah minimum operasi yang dibutuhkan untuk mengubah satu string ke string yang lain. Dimana sebuah operasi dapat berupa penyisipan, penghapusan atau pergantian dari sebuah karakter atau *transposition* dari dua *adjacent* karakter.

*Term-Based Similarity Measures*:

- *Cosine similarity* adalah metode pengukuran kesamaan antara dua buah vektor dari sebuah *inner product space* yang mengukur *cosinus* sudut dari kedua vektor tersebut.
- *Jaccard similarity* dihitung sebagai jumlah *terms* yang di-*share* diatas atau berbanding jumlah semua terms unik yang terdapat pada kedua buah string.

### 2.5.2 Corpus-Based Similarity

*Corpus-Based Similarity* merupakan metode pengukuran kesamaan teks secara semantik yang menentukan kesamaan antar kata berdasar informasi yang didapatkan dari *corpora* yang besar. Sebuah *Corpus* adalah sekumpulan koleksi yang besar dari teks tertulis maupun lisan yang digunakan untuk penelitian bahasa.

- *Latent Semantic Analysis* (LSA) merupakan teknik *Corpus-Based similarity* yang paling populer. LSA mengasumsikan bahwa kata-kata yang memiliki kesamaan atau kedekatan arti akan muncul pada potongan-potongan teks yang sama.
- *Extracting DIStributionally similar word using Cooccurrence* (DISCO) merupakan kesamaan terdistribusi antar kata yang mengasumsikan kata yang

mempunyai kesamaan arti muncul pada konteks yang sama. Koleksi teks yang besar dianalisa secara statistik untuk memperoleh kesamaan yang terdistribusi.

### 2.5.3 Knowledge-Based Similarity

*Knowledge-Based Similarity* merupakan salah satu metode pengukuran kesamaan secara semantik yang berbasis pada pengidentifikasian derajat kesamaan antar kata menggunakan informasi yang didapatkan dari *semantic networks*. *WordNet* merupakan salah satu bentuk *semantic network* yang populer digunakan pada area *knowledge-based similarity*.

### 2.5.4 Hybrid Similarity Measures

*Metode hybrid similarity* menggunakan atau mengkombinasikan beberapa metode pengukuran kesamaan secara bersamaan.

- *Semantic Text Similarity (STS)*

Metode ini menentukan tingkat kesamaan dua buah teks berdasar kombinasi dari informasi semantik dan informasi sintaks/tata bahasa. Di dalam metode ini terdapat 2 metode *mandatory* (*string similarity* dan *semantic word similarity*) dan satu fungsi *optional* (*common word order similarity*).

## 2.6 Wordnet

*Wordnet* adalah basis data leksikal untuk bahasa Inggris yang merupakan hasil penelitian dari Princeton University dalam bidang komputasi linguistik. Basis data leksikal adalah kumpulan data yang menyimpan informasi relasi semantik antar sinonim set (*synset*). Dua jenis relasi antar *synset* terdiri dari leksikal dan semantik. Relasi leksikal merupakan relasi berdasarkan pada susunan kata sedangkan semantik adalah relasi berdasarkan pada arti kata. *Wordnet* memiliki empat tipe *part of speech* (POS) yaitu *noun* (N), *verb* (V), *adjective* (Adj) dan *adverb* (Av). *Wordnet* juga memiliki beberapa relasi semantik, yaitu sinonim, antonym, hiponim, meronym, troponim dan entailment (Milner, 1995).

Hal yang paling penting dalam *wordnet* adalah kemiripan makna karena penentuan hubungan antar bentuk kata merupakan prasyarat untuk merepresentasikan makna dalam matriks leksikal. Dua buah kata dianggap

bersinonim jika penggantian kata tersebut oleh kata yang lain tidak akan merubah makna kalimat. Jarang sekali ditemukan kata yang bersinonim dan dapat digunakan dalam konteks yang sama. Kata “mobil” dan “truk” memiliki makna serupa tetapi keduanya memiliki perbedaan konteks penggunaan.

### 2.7 Koefisien Kappa

Koefisien Kappa sering digunakan untuk mengukur kesepakatan dari dua pengamat terhadap karakteristik yang menjadi perhatian penelitian. Koefisien Kappa hanya diterapkan pada hasil pengukuran data kualitatif.

Tabel 2.5.4.1 Interpretasi nilai koefisien Kappa

Nilai K	Keeratan Kesepakatan ( <i>Strength of agreement</i> )
< 0,20	Rendah (Poor)
0,21 – 0,40	Lumayan (Fair)
0,41 – 0,60	Cukup (Moderate)
0,61 – 0,80	Kuat (Good)
0,81 – 1.00	Sangat Kuat (Very good)

Berikut ini adalah contoh kasus perhitungan menggunakan koefisien Kappa menurut (Gwet, 2002).

Tabel 2.5.4.2 Hasil eksperimen E1

Pengamat B	Pengamat A		
	“1”	“2”	Total
“1”	40	9	49
“2”	6	45	51
Total	46	54	100

Dengan  $e(\gamma)$  merupakan derajat probabilitas *change-agreement* dan  $P_1$  merepresentasikan aproksimasi kemungkinan bahwa pengamat (A atau B) mengklasifikasikan sebuah objek ke dalam kategori “1”.

$$e(\gamma) = 2 P_1(1 - P_1) \dots\dots\dots (1)$$

$$P_1 = \frac{(A1+B1)/2}{N} \dots\dots\dots (2)$$



$$Kappa(AC1) = \frac{p-e(\gamma)}{1-e(\gamma)} \dots\dots\dots (3)$$

$$p = \frac{A+D}{N} \dots\dots\dots (4)$$

Dari rumus atau persamaan (1), (2), (3) dan (4) didapat hasil perhitungan Kappa sebagai berikut:

$$e(\gamma) = 2 \left( \frac{46 + 49}{2 \times 100} \right) \left( 1 - \frac{46 + 49}{2 \times 100} \right) = 0,49875$$

Sehingga didapatkan koefisien Kappa (AC1) sebesar:

$$AC1 = \frac{0,85 - 0,49875}{1 - 0,49875} = 0,7008$$

## 2.8 Milestone Penelitian Terkait Penggunaan Kembali Spesifikasi Kebutuhan Perangkat Lunak

Tabel 2.5.4.1 Milestone penelitian terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak.

No	Tahun	Judul Penelitian	Penjelasan
1	1999	M. C. Blok, and J. L. Bybulski, "Reusing UML Specifications in a Constrained Application Domain", Proceeding of the 5th Asia Pacific Software Engineering Conference (APSEC),1998.	Merupakan metode pertama dari <i>use case retrieval</i> yang mengukur kesamaan diagram <i>sequence</i> dari akumulasi alur aksi diagram (diagram <i>event flow accumulation</i> ).
2	2002	H. G. Woo, and W. N. Robinson, "Reuse of Scenario Specifications Using an Automated Relational Learner: A Lightweight Approach", Requirement Engineering Conference (SAIS), 2002.	Mengusulkan teknik otomatisasi dalam perhitungan tingkat kesamaan diagram <i>sequence</i> . Diagram <i>sequence</i> ini kemudian ditransformasi ke dalam bentuk <i>graph</i> kemudian dilakukan pencocokan <i>graph</i> guna mengukur tingkat kesamaan diagram.
3	2006	A. Udomchaiporn, N. Prompoon, and P.Kanongchaiyos, "Software	Mengusulkan metode retrieval menggunakan <i>Vector Space Model</i>

		Requirements Retrieval Using Use Case Terms and Structure Similarity Computation”, The 13th Conference in Software Engineering (APSEC), Dec 2006, p.113-120	(VSM) untuk mengukur tingkat kesamaan <i>use case description</i> . Keterbasan: <ul style="list-style-type: none"> <li>• Perhitungan kesamaan kata tidak memperhitungkan kesamaan semantik.</li> <li>• Pemberian bobot untuk masing-masing elemen struktur kasus penggunaan masih dilakukan secara manual oleh pengguna. Sehingga ketidaktepatan pemberian bobot sangat mempengaruhi akurasi pada hasil retrieval.</li> </ul>
4	2006	S. Suksaard, and N. Prompoon, “Use Case Retrieval Using Use Case Clustering and User Relevance Feedback”, The 3rd Joint Conference on Computer Science of Software Engineering, 2006	Mengusulkan ide tambahan untuk meningkatkan metode <i>use case retrieval</i> yang diusulkan oleh Udomchiporn’s dengan menggunakan teknik relevansi <i>feedback</i> .
5	2007	Use Case Extractor: XML Parser for Automated Extraction and Storage of Use Case Diagram	Mengusulkan kakas bantu untuk <i>Use Case Extractor</i> dengan tujuan: <ul style="list-style-type: none"> <li>• Membaca dan menyimpan data <i>use-case</i> ke dalam basis data.</li> <li>• Otomatisasi evaluasi diagram <i>use-case</i></li> </ul>
6	2009	Ilyas, M., & Kung, J. (2009). A Similarity Measurement Framework for Requirements Engineering. Fourth International Multi-Conference on Computing in the Global Information	Mengusulkan suatu kerangka kerja (SimReq) guna penggunaan kembali <i>design</i> dan <i>code</i> . Metodologi: <ul style="list-style-type: none"> <li>• Persiapan Data</li> </ul>

		Technology (pp. 31-35). IEEE Computer Society.	<p>Analisa kelengkapan spesifikasi kebutuhan dan analisis <i>lexical</i>.</p> <ul style="list-style-type: none"> <li>• Pengukuran Kesamaan Spesifikasi Kebutuhan</li> <li>• Penggunaan Kembali Rancangan dan Kode Sumber.</li> </ul>
7	2010	Srisura, B., & Daengdej, J. (2010). Retrieving Use Case Diagram With Case-based Reasoning Approach. Journal of Theoretical and Applied Information Technology, 19(2), 68-78.	<p>Mendapatkan kembali diagram-diagram yg mirip dengan membandingkan dua dimensi: dimensi use case dan <i>actor-text-based</i> format, dan dimensi hubungan – <i>structure-based</i> format, dan kemudian merankingnya.</p> <p>Skor yang sesuai dan bobot yg masuk akal dari setiap dimensi dihitung secara teliti berdasarkan <i>the nearest neighbor matching</i> dan <i>ranking</i> untuk menentukan <i>use case diagram</i> yang paling mirip.</p>
8	2012	Morales, B. B., Sergio, C., & Clunie, C. (2012, January). Reuse of Use Cases Diagrams: An Approach based on Ontologies and Semantic Web Technologies. IJCSI International Journal of Computer Science Issues, 9(1), 24-29.	<p>Mengusulkan sebuah kaskas bantu untuk menggunakan kembali diagram kasus penggunaan dengan menyimpan informasi ke dalam Ontologi (OWL) dan penggunaan teknologi situs semantik.</p> <p>Metodologi:</p> <ul style="list-style-type: none"> <li>• Pembuatan diagram kasus penggunaan dan meng-eksportnya ke dalam format XMI.</li> <li>• Menyimpan informasi-informasi pada diagram kasus</li> </ul>

			<p>penggunaan ke dalam Ontologi OWL.</p> <ul style="list-style-type: none"> <li>• Menyimpan informasi Ontologi OWL ke dalam DBMS MySQL</li> <li>• Mencari dan mendapatkan kembali diagram kasus penggunaan dalam format XML.</li> </ul>
9	2013	Reza Fauzan “Rekomendasi Kasus Penggunaan Berdasarkan Skenario Naratif Menggunakan Teknologi Semantik”	<p>Ekstraksi metadata dari scenario naratif yg memiliki potensi menjadi use case.</p> <p>Metodologi yang dilakukan:</p> <ol style="list-style-type: none"> <li>1. Analis</li> <li>2. Skenario</li> <li>3. Ekstraksi Metadata(StanfordNLP)</li> <li>4. Pencarian dlm repository use case</li> </ol>

## 2.9 Sistem Temu Kembali Kebutuhan Perangkat Lunak Menggunakan Perhitungan Kesamaan Struktur Dan Istilah Dalam Kasus Penggunaan

Pada penelitian (Udomchaiyorn, Prompoon, & Kanongchaiyos, 2006) diusulkan suatu metode penggunaan kembali spesifikasi kebutuhan perangkat lunak melalui perhitungan kesamaan struktur kasus penggunaan. Pendekatan ini diharapkan dapat menjadi solusi untuk mengurangi biaya dan waktu pada setiap pengembangan perangkat lunak.

Konsep Sistem Temu Kembali Informasi diimplementasikan dalam penelitian tersebut, yaitu meliputi empat langkah utama:

### 1) *Automatic Indexing*:

- a. Parsing isi dari *use case*; memisahkan setiap kata di dalam *use case*.
- b. Menghapus atau menghilangkan kata-kata yang termasuk dalam *stop list*.

c. *Stemming word* ke dalam bentuk *grammatical root*. Salah satu model yang sering digunakan dalam *words stemming* yaitu metode yang dikembangkan oleh (Porter, 1980).

d. Menghilangkan *high frequency words*.

2) *Term weighting system*

Merupakan suatu proses untuk menentukan bobot pada masing-masing *term*. Pada saat ini terdapat banyak sekali metode pembobotan. Akan tetapi pada penelitian tersebut menggunakan metode yang disebut *Inverse Document Frequency (IDF)*.

3) *Vector Similarity Computation*

Metode perhitungan kesamaan vektor yang digunakan dalam penelitian tersebut yaitu *Dice's coefficients*.

4) *Retrieval evaluation*

Tujuan dari evaluasi yaitu untuk mengukur tingkat efektifitas dan efisiensi dari sebuah sistem temu kembali dan penyimpanan informasi. Untuk proses evaluasi, terdapat tiga metrik yang populer; *recall*, *precision* dan *harmonic mean*. Berikut penjelasan singkat ketiga metrik tersebut:

- *Recall* mendefinisikan proporsi dari dokumen yang diterima kembali dan relevan terhadap semua dokumen relevan di dalam koleksi.
- *Precision* adalah proporsi dari dokumen yang relevan dan diterima kembali terhadap semua dokumen yang diterima kembali di dalam koleksi.
- *Harmonic mean* merupakan pengukuran tunggal yang mengkombinasikan antara *recall* dan *precision*.

Kueri dan Proses Temu Kembali

Pertama, pengguna membangkitkan atau menentukan sebuah kasus penggunaan sebagai kueri beserta bobot masing-masing elemen struktur kasus penggunaan. Nilai bobot merupakan *integer* dalam interval 1-5.

Tabel 2.5.4.1 Form Masukan Kueri Kasus Penggunaan

No	Elemen Deskripsi Kasus Penggunaan	Teks Kueri	Bobot (1-5)
1	Name		
2	Objective		

3	Actor			
4	Relationship	Association		
5		Include		
6		Extend		
7		Generalization		
8	Precondition			
9	Postcondition			
	Flow of Events			
10		Normal Flow		
11		SubFlow		
12	Alternative or Exceptional Flow of events			

Tingkat atau nilai kesamaan antara masing-masing elemen pada kueri kasus penggunaan dan elemen kasus penggunaan pada koleksi dihitung dengan persamaan berikut:

$$\text{Similarity}(E_{mi}, E_{mj}) = \frac{2[\sum_{k=l}^t (\text{Term}_{mik} W_{ik}) \text{Term}_{mjk}]}{\sum_{k=l}^t \text{Term}_{mik} + \sum_{k=l}^t \text{Term}_{mjk}} \dots\dots\dots (2.1)$$

$E_{mi}$  = elemen m dari *use case* i.

$E_{mj}$  = elemen m dari *use case* j.

$W_{ik}$  = nilai bobot IDF *term* k pada *use case* i.

$\text{Term}_{mik}$  = 1 jika *term* k muncul pada elemen m dari *use case* i, sebaliknya 0

$\text{Term}_{mjk}$  = 1 jika *term* k muncul pada elemen m dari *use case* j, sebaliknya 0

Sedangkan nilai kesamaan kueri kasus penggunaan dengan kasus penggunaan dalam koleksi dihitung berdasar persamaan (2.2).

$$\text{Similarity}(UC_i, Query_j) = \frac{\sum_{m=l}^e [\text{Similarity}(E_{mi}, E_{mj}) \cdot WE_m]}{\text{TotalWeight}} \dots\dots\dots (2.2)$$

$\text{Similarity}(UC_i, Query_j)$  = Skor tingkat kesamaan antara kueri *use case* dari pengguna dan masing-masing *use case* di dalam repositori.

$\text{Similarity}(E_{mi}, E_{mj})$  = Skor kesamaan masing-masing elemen

$WE_m$  = Bobot setiap elemen yang ditentukan secara otomatis berdasar nilai tingkat kesamaan total dari masing-masing elemen kasus penggunaan.

Total Weight = total akumulasi bobot pada semua komponen.

Hasil koleksi kasus penggunaan (*Topmost T similarity score*) kemudian ditampilkan kepada pengguna aplikasi.

Hipotesis dari eksperimen penelitian tersebut adalah temu kembali kasus penggunaan dengan struktur kasus penggunaan lebih efektif daripada temu kembali melalui kata kunci pada kasus penggunaan. Berdasarkan nilai *precision*, nilai *precision improvement* (P+) adalah terdapat pada rating tinggi untuk semua subjek sample. Nilai rata-rata P+ (3) dari (1) dan P+(3) dari (2) adalah 55,78% dan 44,21%.

Namun demikian, pendekatan yang dilakukan pada penelitian tersebut masih mempunyai keterbatasan. Pertama, proses temu kembali hanya memperhitungkan pada kesamaan kata, tetapi tidak memperhitungkan kesamaan semantik. Keterbatasan yang kedua yaitu pengguna masih harus menentukan bobot untuk masing-masing elemen secara manual. Jika pengguna memilih bobot yang tidak sesuai maka rata-rata *recall* menjadi lebih rendah.

## **2.10 Arsitektur dan Teknologi Kakas Bantu Pendukung**

Guna mendukung pengujian dan evaluasi dari metode yang diusulkan dalam tesis ini, dibuat suatu kakas bantu berbasis web. Kakas bantu ini terbagi menjadi tiga komponen utama yaitu;

1. Repositori Kasus Penggunaan. Repositori ini dibuat disimpan dalam bentuk basis data. Adapun DBMS yang digunakan yaitu MySQL.
2. Teknologi *Java web service* digunakan untuk melakukan proses perhitungan kesamaan semantik antara dua buah kata atau kalimat. Pemrosesan *Natural Language Processing* (NLP) juga dilakukan pada bagian ini. Teknologi Java dipilih karena mempunyai keunggulan dalam hal kecepatan proses eksekusi dan kehandalan dalam pemrosesan bahasa alamiah.
3. Aplikasi *web-client* menggunakan teknologi PHP: semua masukan maupun luaran dari dan ke pengguna ditangani dalam bagian ini.
  - Masukan: berupa form isian kueri deskripsi kasus penggunaan
  - Luaran: berupa daftar deskripsi kasus penggunaan yang didapatkan kembali.

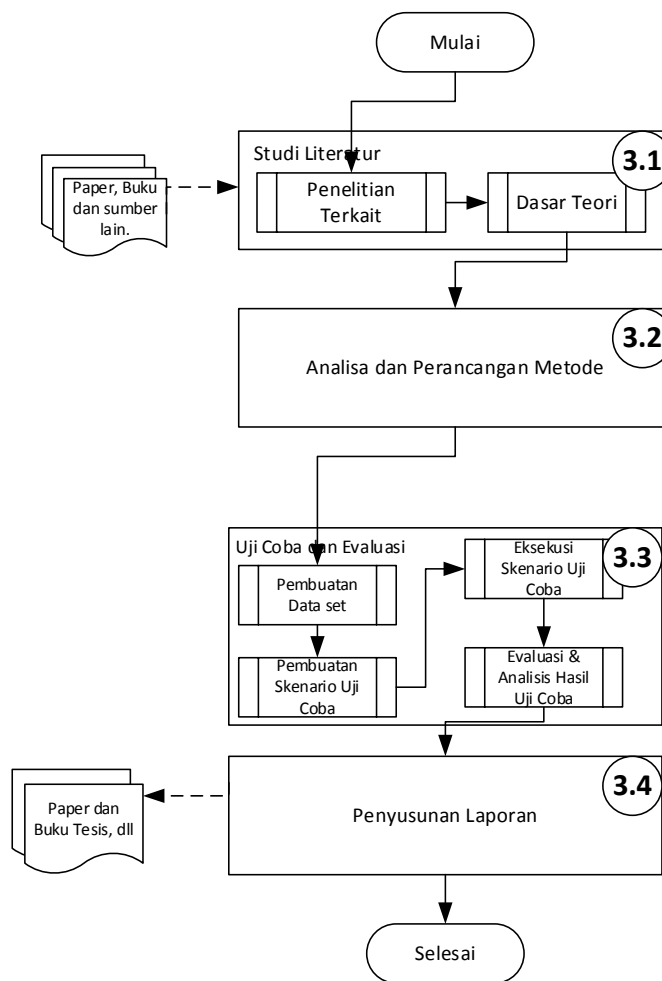
[Halaman ini sengaja dikosongkan]



## BAB 3

### METODE PENELITIAN

Secara umum, penelitian ini diawali dengan studi literatur yang meliputi penelitian terkait, dasar teori dan analisa. Kemudian dilanjutkan dengan perancangan metode yang meliputi perumusan masalah, usulan metode penyelesaiannya serta diakhiri dengan uji coba dan evaluasi metode. Sedangkan penulisan laporan tesis dimulai pada awal sampai akhir penelitian. Secara lebih detil, penelitian ini dirancang dengan urutan sebagai berikut:



Gambar 3.22.10.1 Alur metodologi penelitian

### 3.1 Studi Literatur

Studi literatur merupakan tahap awal dari penelitian. Pada tahap ini dilakukan studi literatur berkaitan dengan metode-metode dan dasar teori yang

digunakan dalam penelitian, meliputi konsep Kasus Penggunaan, Sistem Temu Kembali Informasi, Metode perhitungan kesamaan teks, dan penelitian-penelitian terkait permasalahan *Use Case Retrieval*. Pada Bab 2 dibahas lebih detil mengenai hasil dari studi literatur yang telah dilakukan.

### 3.2 Analisa dan Perancangan Metode

Pada tahap ini dilakukan analisa menyeluruh mengenai perancangan solusi metode yang dibangun untuk menyelesaikan permasalahan dalam penelitian meliputi: rancangan detil langkah-langkah yang akan ditempuh dan kriteria-kriteria yang akan digunakan. Metode yang diusulkan meliputi tiga tahapan. Gambar 3.2.2 menggambarkan alur proses metode *Use Case Retrieval* yang diusulkan.

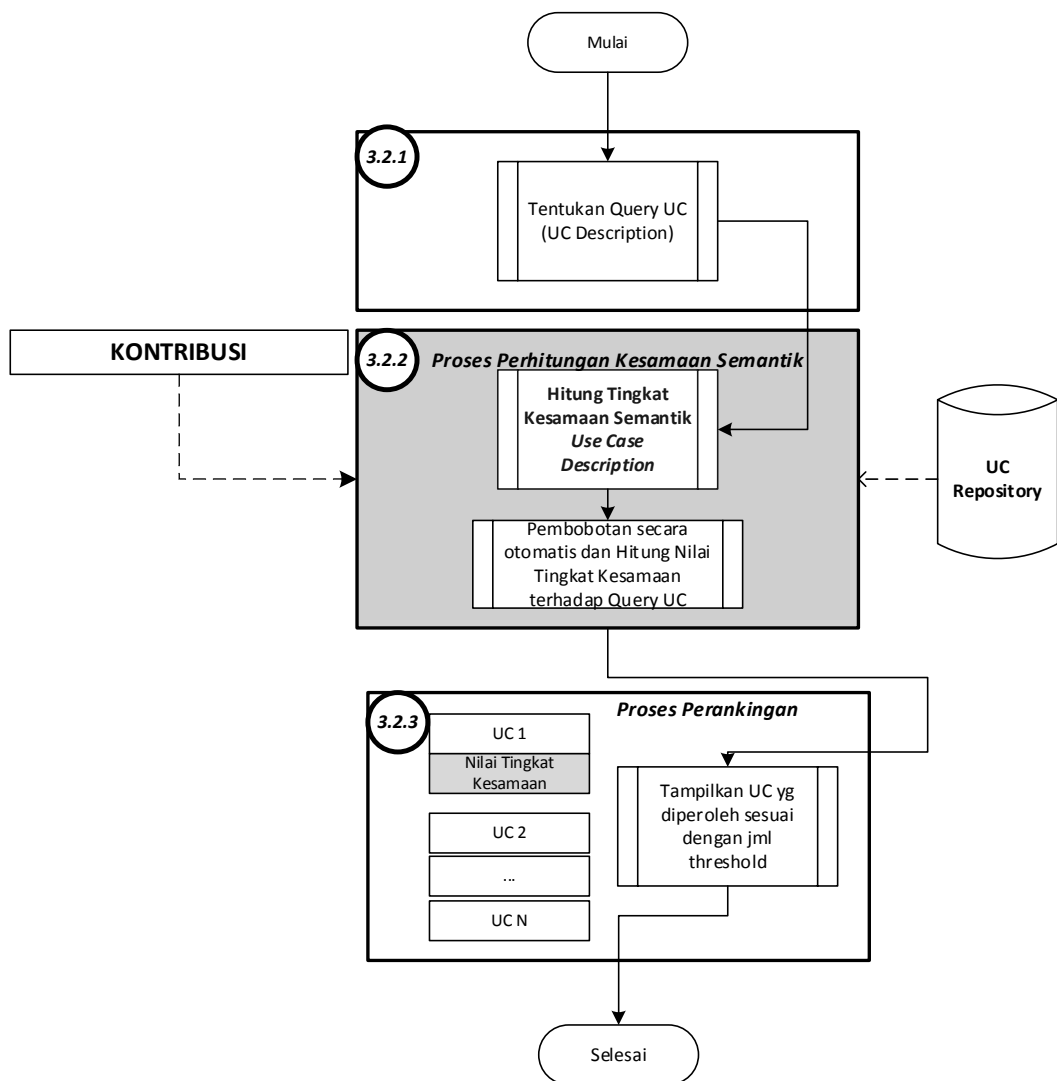
#### 3.2.1 Penentuan Kueri Kasus Penggunaan

Pertama-tama, pengguna menentukan sebuah kasus penggunaan sebagai kueri dengan mengisi form deskripsi kasus penggunaan seperti pada Tabel 3.2.1

Tabel 3.2.1.1 Form kueri berdasar struktur deskripsi kasus penggunaan

No	<i>Use Case Description Elements</i>		<i>Query</i>
1	Name		
2	Objective/Description		
3	Actor		
4	Trigger		
5	Precondition		
6	Postcondition		
7	Flow of Events		
		Normal/Basic Flow Events	
		Alternative or Exceptional Flow of events	
8	Special Requirement		
9	Assumptions		
10	Includes		

Pada form diatas (Tabel 3.2.1.1) terdiri dari sepuluh elemen deskripsi kasus penggunaan yang harus diisikan oleh pengguna.

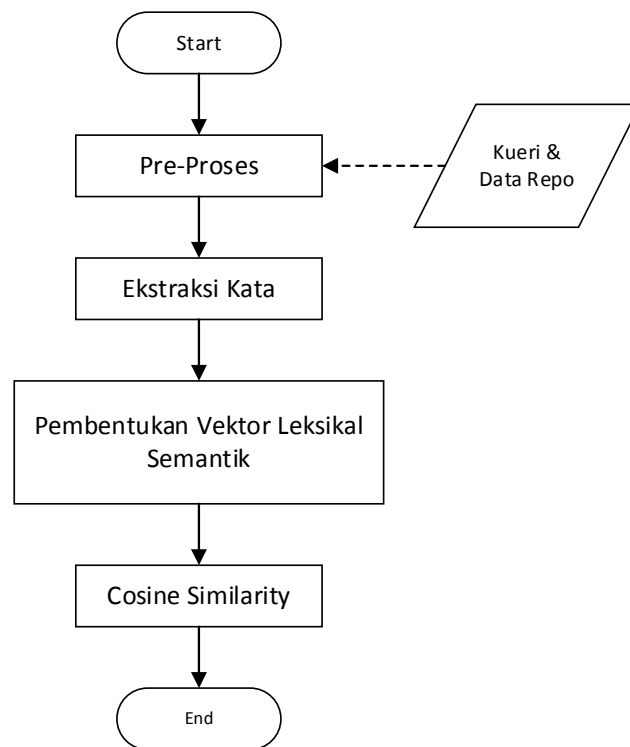


Gambar 3.2.2 Alur proses temu kembali Kasus Penggunaan

### 3.2.2 Perhitungan Kesamaan Semantik

Setelah kueri deskripsi kasus penggunaan ditentukan, proses selanjutnya yaitu melakukan perhitungan kesamaan kata secara semantik antara kueri deskripsi kasus penggunaan dan deskripsi kasus penggunaan di dalam repositori. Dalam proses ini, kami mengusulkan menggunakan *Wordnet* dalam melakukan proses perhitungan kesamaan semantik (Liu & Wang, 2013).

Gambar 3.2.3 menunjukkan proses perhitungan kesamaan semantik pada kalimat atau frasa.



Gambar 3.2.3 Alur metodologi perhitungan kesamaan semantik

Proses perhitungan kesamaan semantik terbagi menjadi empat bagian utama:

1. Pre-Proses  
Meliputi proses transformasi kalimat ke dalam *bag-of-word* atau tokenisasi.
2. Ekstraksi Kata  
Proses pelabelan kata sesuai jenis kata (*Part of Speech Tagging*).
3. Pembentukan Vektor Leksikal Semantik menggunakan wordnet
4. Menghitung nilai Cosine Similarity

Kesamaan semantik antara dua kata  $w_1$  dan  $w_2$  dinotasikan sebagai  $sim(w_1, w_2)$ . Nilai kesamaan ini didapat melalui analisa dari sebuah *lexical knowledge base* seperti *Wordnet* dimana kata-kata diorganisir ke dalam sebuah *synonym sets* (*synsets*).

- Jika kedua target kata adalah identik atau terdapat di dalam synset yang sama maka nilai kesamaannya adalah 1.
- Jika salah satu target kata tidak terdapat di dalam *Wordnet*, maka nilai kesamaannya adalah 0,

- Jika kedua target kata tidak terdapat di dalam synset yang sama tetapi masih di dalam hirarki *Wordnet*, maka nilai kesamaan dihitung berdasarkan relevansi *node* di dalam hirarki.

Berikut dijeaskan contoh proses penghitungan kesamaan kalimat menggunakan dua buah kalimat pada Tabel 3.2.2 diambil dari salah satu elemen deskripsi kasus penggunaan (misal: *use case objective/description* atau tujuan kasus penggunaan), satu *query use case* oleh pengguna dan satu lagi diambil dari *use case* pada *repository*.

Tabel 3.2.2.1 Contoh kalimat-kalimat pada use-case.

Kalimat S <sub>1</sub> ( <i>Query UC Objective</i> )	Kalimat S <sub>2</sub> ( <i>UC Objective Repo</i> )
Consumers would still have to get a descrambling security card from their cable operator to plug into the set.	To watch pay television, consumers would insert into the set a security card provided by their cable service.

Untuk setiap data set, *Wordnet* digunakan sebagai sumber untuk menghitung kesamaan kata. Untuk dapat menggunakan *Wordnet*, pertama kali kita harus mengubah kalimat S<sub>1</sub> dan S<sub>2</sub> ke dalam representasi *bag-of-word*.

$T_1 = \{\text{kata-kata di dalam } S_1\}$

$T_2 = \{\text{kata-kata di dalam } S_2\}$

Kita juga membentuk sebuah superset T, yang merupakan gabungan dari T<sub>1</sub> dan T<sub>2</sub>

$T = T_1 \cup T_2$ .

Tabel 3.2.2.2 *Sentence preprocessing (Bag-of-word)*

Notasi	Kumpulan Kata
T <sub>1</sub>	{"consumer","would","still","have","to","get","a","descramble","security","card","from","their","cable","operator","to","plug","into","the","set"}
T <sub>2</sub>	{"to","watch","pay","television","consumer","would","insert","into","the","set","a","security","card","provide","by","their","cable","service"}
T	{"consumer","would","still","have","get","a","descramble","security","card","from","their","cable","operator","to","plug","into","the","set","watch","pay","television","insert","provide","by","service"}

Langkah selanjutnya yaitu pembentukan vektor.

Tujuan dari pembentukan T, yang mengandung seluruh kata dari T<sub>1</sub> dan T<sub>2</sub> adalah untuk membentuk vector semantik bagi T<sub>1</sub> dan T<sub>2</sub>. Vektor yang diturunkan dari gabungan kumpulan kata disebut lexical semantik vektor yang dinotasikan dengan symbol V<sub>1</sub> dan V<sub>2</sub>. Dimensi dari lexical semantik vektor adalah sesuai dengan jumlah kata-kata pada gabungan kumpulan kata.

Tabel 3.2.2.3 Nilai Masukan dari V<sub>1</sub> dan V<sub>2</sub>

Nilai dari V <sub>1</sub>	Nilai dari V <sub>2</sub>
("a","a"): 1.0,	("a","a"): 1.0,
("by","a"): 0,0,	("by","by"): 1.0,
("cable","cable"): 1.0,	("cable","cable"): 1.0,
("card","card"): 1.0,	("card","card"): 1.0,
("consumer","consumer"): 1.0,	("consumer","consumer"): 1.0,
("descrambling","descrambling"): 1.0,	("descrambling","a"): 0,0,
("from","from"): 1.0,	("from","a"): 0,0,
("get","get"): 1.0,	("get","pay"): 0,807
("have","get"): 1.0,	("have","pay"): 0,836
("insert","insert"): 0,935	("insert","insert"): 1.0,
("into","into"): 1.0,	("into","into"): 1.0,
("operator","operator"): 1.0,	("operator","card"): 0,0,
("pay","have"): 0,836	("pay","pay"): 1.0,
("plug","plug"): 1.0,	("plug","set"): 0,995
("provide","set"): 0,999	("provide","provide"): 1.0,
("security","security"): 1.0,	("security","security"): 1.0,
("service","set"): 0,996	("service","service"): 1.0,
("set","set"): 1.0,	("set","set"): 1.0,
("still","still"): 1.0,	("still","television"): 0,792
("television","cable"): 0,999	("television","television"): 1.0,
("the","the"): 1.0,	("the","the"): 1.0,
("their","their"): 1.0,	("their","their"): 1.0,
("to","to"): 1.0,	("to","to"): 1.0,
("watch","set"): 0,857	("watch","watch"): 1.0,
("would","would"): 1.0	("would","would"): 1.0

Tabel 3.2.2.4 Nilai Vektor dari V<sub>1</sub> dan V<sub>2</sub>

Vektor	Nilai Vektor
V <sub>1</sub>	(1.0 , 0,0 , 1.0 , 1.0 , 1.0 , 1.0 ,1.0 ,1.0 ,1.0 ,0,935 ,1.0 ,1.0 ,0,836 ,1.0 ,0,999 ,1.0 ,0,996 ,1.0 ,1.0 ,0,999 ,1.0 ,1.0 ,1.0 ,0,857 ,1.0)
V <sub>2</sub>	(1.0 ,1.0 ,1.0 ,1.0 ,1.0 ,0,0 ,0,0 ,0,807 ,0,836 ,1.0 ,1.0,0,0 ,1.0 ,0,995 ,1.0 ,1.0 ,1.0 ,1.0 ,0,792 ,1.0 ,1.0 ,1.0 ,1.0,1.0 ,1.0 )

Tabel 3.2.2.5 Nilai Vektor Akhir dari V<sub>1</sub> dan V<sub>2</sub>

Vektor	Nilai Vektor
V <sub>1</sub>	(0,2 ,0,0 ,1.0 ,1.0 ,1.0 ,1.0 ,0,2 ,1.0 ,1.0 ,0,935 ,0,2 ,1.0 ,0,836 ,1.0 ,0,999 ,1.0 ,0,996 ,1.0 ,0,2 ,0,999 ,0,2 ,0,2 ,0,2 ,0,857 ,0,2 )
V <sub>2</sub>	0,2 ,0,2 ,1.0 ,1.0 ,1.0 , 0,0 ,0,0 ,0,807 ,0,836 ,1.0 ,0,2 ,0,0 ,1.0 ,0,995 ,1.0 ,1.0 ,1.0 ,1.0 ,0,792 ,1.0 ,0,2 ,0,2 ,0,2 ,1.0 ,0,2)

Sehingga nilai dari *cosine similarity* antara S<sub>1</sub> dan S<sub>2</sub> dapat dihitung dengan persamaan:

$$\text{Sim}(S_1, S_2) = \frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|} \dots\dots\dots (3.1)$$

$$\text{Sim}(S_1, S_2) = 0,907$$

Khusus untuk elemen deskripsi kasus penggunaan *Normal Flows/Basic Flows*, perhitungan nilai kesamaan juga mempertimbangkan urutan dari aliran deskripsinya. Aliran normal no.1 dari kueri kasus penggunaan dibandingkan dengan aliran normal no.1 deskripsi kasus penggunaan di repositori. Aliran normal no.2 dari kueri deskripsi kasus penggunaan dibandingkan dengan aliran normal no.2 kasus penggunaan di repositori dan seterusnya.

### 3.2.3 Proses Perankingan dan *Threshold*

Langkah terakhir yaitu proses perankingan hasil kueri kasus penggunaan berdasarkan nilai kesamaan atau kemiripan secara semantik. Sebelum melakukan proses perankingan diperlukan pencarian nilai *threshold* yang ideal untuk

menentukan atau memvalidasi data hasil kueri. Nilai *threshold* ini didapat dari hasil pengujian melalui pengukuran indeks Kappa. *Threshold* dengan nilai rata-rata Kappa yang tertinggi dari hasil masing-masing kueri pada rentang 0,1 sampai dengan 0,9 akan diambil sebagai *threshold*.

#### **3.2.4 Skenario Pengujian**

Pengujian dilakukan dengan tiga skenario yang berbeda:

- Pengujian untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem terhadap penilaian pakar menggunakan indeks Kappa.
- Pengujian kueri kasus penggunaan dengan menggunakan kesamaan semantik di salah satu elemen atau item dari *use case description*.
- Pengujian *query use case* dengan kombinasi masukan elemen atau item *use case* yang berbeda-beda.



## **BAB 4**

### **HASIL PENELITIAN DAN PEMBAHASAN**

Di dalam bagian ini dibahas perihal hasil penelitian dan pembahasan yang meliputi beberapa bagian yaitu: pembahasan variabel pengujian, hasil pengujian dan analisa evaluasi hasil pengujian. Lingkungan pengujian terdiri dari empat aspek yaitu meliputi Koleksi Deskripsi Kasus Penggunaan, Pakar atau Ahli, Proses Pencarian dan Kueri Deskripsi Kasus Penggunaan. Hasil pengujian memperlihatkan data statistik yang didapat dari proses eksekusi kueri deskripsi kasus penggunaan menggunakan metode yang berbeda-beda yang dilakukan oleh para ahli.

Tahap yang terakhir yaitu evaluasi meliputi analisa pengukuran hasil kueri deskripsi kasus penggunaan oleh pakar terhadap hasil yang didapatkan oleh sistem. Tujuan utama dari eksperimen adalah untuk menguji asumsi dari penelitian yaitu apakah temu kembali kasus penggunaan berdasar struktur dengan memasukkan aspek semantik lebih efektif dan akurat dibandingkan dengan temu kembali berdasar struktur tanpa memasukkan aspek semantik. Pengukuran indeks Kappa digunakan untuk mengetahui tingkat keakuratan dari pendekatan yang dilakukan pada penelitian ini

#### **4.1 Lingkungan Pengujian**

Eksperimen ini dirancang untuk menghilangkan bias, sehingga mempunyai beberapa faktor yang harus dikontrol. Faktor-faktor ini terbagi menjadi empat variabel yaitu koleksi deskripsi kasus penggunaan, pakar atau ahli, proses pencarian dan kueri deskripsi kasus penggunaan. Berikut penjelasan selengkapnya mengenai empat variabel kontrol tersebut:

##### **4.1.1. Koleksi Deskripsi Kasus Penggunaan**

Sebagaimana dijelaskan di Bab 1, dataset penelitian ini diambil dari beberapa *textbook* seputar pemodelan kasus penggunaan. Dataset dari *textbook* ditulis dan digambarkan ulang ke dalam bentuk diagram kasus penggunaan beserta deskripsinya menggunakan suatu kakas bantu. Total kasus penggunaan yang dapat

diambil sebagai dataset sejumlah 137 kasus penggunaan dari 20 sistem atau projek aplikasi.

Tabel 4.1.1.1 Daftar dataset berdasar nama sistem dan sumber referensi.

No	SID	Sistem	Sumber Referensi
1	TS	Trading System	UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition By Martin Fowler Publisher : Addison Wesley Pub Date : September 15, 2003 ISBN : 0-321-19368-7 Pages : 208
2	TCS	TimeCard System	Enterprise Java with UML by CT Arlington, 2001 John Wiley & Sons, Inc.
3	TRS	Theater Management System	UML Bibble by Tom Pender ISBN:0764526049 John Wiley - Sons 2003 (940 pages)
4	PMS	The project management system	Learning UML By Sinan Si Alhir Publisher : OReilly Pub Date : July 2003 ISBN : 0-596-00344-7 Pages : 252 Slots : 1
5	STS	The simple telephone system	Use Case Modeling By Kurt Bittner, Ian Spence Publisher : Addison Wesley Pub Date : August 21, 2002 ISBN : 0-201-70913-9 Pages : 368
6	BS	The Bookstore	Fast Track UML 2.0 b y Kendall Scott ISBN:1590593200 Apress 2004
7	ACME	the ACME Super ATM	Use Case Modeling By Kurt Bittner, Ian Spence Publisher : Addison Wesley Pub Date : August 21, 2002 ISBN : 0-201-70913-9 Pages : 368
8	PS	Payroll System	<a href="http://faculty.kfupm.edu.sa/ICS/mwaslam/Rich%20Files/doc/Payroll%20System%20Use-Case%20Model.htm">http://faculty.kfupm.edu.sa/ICS/mwaslam/Rich%20Files/doc/Payroll%20System%20Use-Case%20Model.htm</a>
9	OSS	Order Shipment Management System	UML Weekend Crash Course by Thomas A. Pender Published by Wiley Publishing, Inc. Copyright 2002 by Wiley LOC: 2002103278 ISBN: 0-7645-4910-3

10	OPS	Order Processing System	IBM - UNITED <a href="http://www.ibm.com/developerworks/rational/library/content/legacy/parttwo/1000/0670/0670_Schneider_Ch07.pdf">http://www.ibm.com/developerworks/rational/library/content/legacy/parttwo/1000/0670/0670_Schneider_Ch07.pdf</a>
11	OBS	Online Books Store	Enterprise Java with UML by CT Arlington, 2001 John Wiley & Sons, Inc.
12	HOLIS	Home Lighting Automation System	Managing Software Requirements: A Use Case Approach, Second Edition By Dean Leffingwell, Don Widrig Publisher : Addison Wesley Pub Date : May 05, 2003 ISBN : 0-321-
13	CBS	Cinema Booking System	www.itu.dk Course: Modelling and Design (fall 2007) Submitted by: Toomas Kutt(toomas@itu.dk), Fraz Tabassam(tabassam@itu.dk), Jens kaae Christensen(jenschr@itu.dk)
14	ARIS	ARIS System	www.opuscollege.net
15	REMS	Real Estate Management System	Use Cases: Requirements in Context, Second Edition By Daryl Kulak, Eamonn Guiney Publisher : Addison Wesley Pub Date : July 25, 2003 ISBN : 0-321-15498-3 Pages : 272
16	IME	Instant Messaging Encryption	Use Cases: Requirements in Context, Second Edition By Daryl Kulak, Eamonn Guiney Publisher : Addison Wesley Pub Date : July 25, 2003 ISBN : 0-321-15498-3 Pages : 272
17	HR	Hotel Reservation	UML 2 for Dummies by Michael Jesse Chonoles and James A. Schardt ISBN:0764526146 Hungry Minds 2003 (412 pages)
18	HMS	Hotel Management System	Aspect-Oriented Software Development with Use Cases By Ivar Jacobson, Pan-Wei Ng Publisher : Addison Wesley Professional Pub Date : December 30, 2004 ISBN : 0-321-26888-1 Pages : 464
19	HD	HomeDirect	Developing Enterprise Java Applications with J2EE and UML by Khawar Zaman Ahmed, Cary E. Umrysh Paperback: 288 pages ; Dimensions (in inches): 0,66 x

			9.28 x 7.32 Publisher: Addison-Wesley Pub Co; ISBN: 0201738295; 1st edition (December 15, 2001)
20	ERS	e-retail system	Building Web Applications with UML Second Edition By Jim Conallen Publisher : Addison Wesley Pub Date : October 04, 2002 ISBN : 0-201-73038-3 Pages : 496

Tabel 4.1.1.2 Domain kategori Kasus Penggunaan

ID	Domain	Keterangan	Jumlah	
			Sistem	UC
A	Finantial Calculation	Domain sistem terkait permasalahan transaksi keuangan	3	20
B	e-Commerce/Trading System	Domain terkait system transaksi jual beli barang dan jasa.	5	37
C	Information Management System	Domain terkait sistem informasi secara umum atau general	8	54
D	Teaching-Studying System	Domain terkait sistem manajemen kegiatan belajar mengajar secara umum.	1	13
E	Other Unspecified Domain	Domain lainnya diluar domain No 1 hingga 4.	3	13
Jumlah			20	137

Pada Tabel 4.1.2 menunjukkan pembagian domain sistem serta sebaran jumlah sistem dan kasus penggunaan untuk masing-masing domain pada dataset. Terdapat 20 sistem dengan total kasus penggunaan 137 yang diambil dari 20 sumber *textbook* berbahasa inggris.

Tabel 4.1.1.3 Sebaran kueri deskripsi kasus penggunaan berdasar kondisi setiap fitur/atribut deskripsi kasus penggunaan.

QID	Name	Actor	Desc.	Pre-Cond	Post-Cond	Norm Flw	Alt Flw	Assm	Inc	Note	Trigg
1	√	√	√	√	√	√					
2	√	√	√	√	√	√					

3	√	√	√	√		√					
4	√	√	√	√	√	√					
5	√	√									
6	√		√								
7	√										
8	√	√				√					
9	√	√									
10	√		√								
11						√					
12	√	√	√	√	√	√					
13	√		√			√					
14	√					√					
15	√	√	√			√					
16	√	√		√		√					
17	√	√		√		√					
18	√	√		√		√					
19	√	√		√		√					
20	√	√		√		√					
	19	14	9	10	4	15	0	0	0	0	0

Terdapat sekitar 20 kueri deskripsi kasus penggunaan yang disusun oleh pakar yang turut berpartisipasi dalam penelitian ini.

#### 4.1.2. Pakar atau ahli

Penelitian ini dilakukan oleh dua orang peneliti baik sebagai pakar penyusun deskripsi kasus penggunaan maupun sebagai pengguna atau penguji kakas bantu. Dua orang pakar tersebut masih atau sedang mengambil program pasca sarjana di bidang Rekayasa Perangkat Lunak. Oleh karena itu kedua orang tersebut dianggap mempunyai kemampuan dan pengalaman di bidang pemodelan kasus penggunaan di dalam pengembangan perangkat lunak dan pengetahuan mereka tentang Bahasa Inggris sangat memadai dengan nilai TOEFL ITP lebih dari 500.

#### **4.1.3. Proses Pencarian**

Sekitar 20 sistem yang terbagi menjadi 5 domain telah dibuat guna menguji metode penelitian yang diusulkan sebagaimana terdapat pada Tabel 4.1.1 dan Tabel 4.1.2. Pengguna atau pakar penguji membuat beberapa kombinasi kueri kasus penggunaan seperti pada Tabel 4.1.3 kemudian memasukkan dan mengeksekusi kueri kasus penggunaan tersebut ke dalam sistem kaskas bantu. Kueri dijalankan atau dieksekusi pada 2 macam kondisi dataset yaitu dataset yang homogen dan dataset yang heterogen. Hasil kueri atau jawaban yang didapat kemudian diidentifikasi secara detil untuk menghitung indeks Kappa terhadap hasil yang didapat oleh ahli yang lain.

Tujuan dari dibuatnya penggolongan domain yang berbeda yaitu untuk mengidentifikasi atau membandingkan tingkat efektifitas dari metode yang diusulkan dengan metode yang lainnya.

#### **4.1.4. Kueri Kasus Penggunaan**

Pakar membuat total sekitar 20 pernyataan kueri baik kueri berdasar berdasar struktur kasus penggunaan untuk semua subjek domain: A, B, C, D dan E sesuai pada Tabel 4.1.2. Di dalam suatu kueri kasus penggunaan dapat berisi kueri lengkap untuk semua komponen kasus penggunaan, kombinasi beberapa komponen atau minimal satu komponen saja yang diisikan. Kueri tersebut dieksekusi pada dataset dengan dua domain yang berbeda, domain yang homogen dan domain yang heterogen.

Itulah keempat variabel kontrol yang dijadikan acuan di dalam tahap pengujian. Pakar yang ditunjuk dapat melakukan kueri kasus pengguna berdasar variabel kontrol tersebut. Dalam penelitian ini ditunjuk dua pakar untuk melakukan pengujian. Masing-masing pakar diharuskan mengeksekusi 20 kueri dan memberikan evaluasi terhadap hasil kueri yang didapatkan. Kueri-kueri tersebut dieksekusi menggunakan suatu kaskas bantu berbasis web yang khusus dibangun untuk memfasilitasi proses pengujian tersebut.

## 4.2 Hasil Pengujian

### 4.2.1. Pengujian Pertama

Pengujian pertama dilakukan untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem dengan hasil yang didapat dari para pakar menggunakan indeks Kappa. *Threshold* yang digunakan yaitu pada rentang 0,1 sampai dengan 0,9.

Tabel 4.2.1.1 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.1 sampai dengan no.6

No.Kueri dan Nilai Kappa						
Threshold	1	2	3	4	5	6
0,1	-0,973	-0,885	-0,950	-0,941	-0,949	-0,895
0,2	-0,837	-0,808	-0,899	-0,941	-0,949	-0,895
0,3	0,559	0,553	-0,536	-0,827	-0,949	-0,895
0,4	0,754	0,839	0,121	-0,039	-0,949	-0,507
0,5	0,863	0,911	0,436	0,468	-0,831	-0,142
0,6	0,896	0,942	0,727	0,720	-0,193	0,000
0,7	0,972	0,942	0,869	0,985	0,469	0,619
0,8	0,987	0,942	0,966	0,971	0,735	0,929
0,9	0,987	0,942	0,983	0,971	0,957	0,947

Tabel 4.2.1.2 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.7 sampai dengan no.13

No.Kueri dan Nilai Kappa							
T	7	8	9	10	11	12	13
0,1	-0,966	-0,959	-0,835	-0,830	0,826	-0,931	-0,931
0,2	-0,966	-0,959	-0,835	-0,580	0,845	-0,931	-0,931
0,3	-0,966	-0,919	-0,835	-0,448	0,901	-0,931	-0,931
0,4	-0,966	-0,791	-0,567	0,093	0,953	-0,431	-0,931
0,5	-0,966	0,444	-0,127	0,836	0,969	-0,183	-0,638
0,6	-0,697	0,629	0,176	0,877	0,969	-0,100	0,003
0,7	0,102	0,858	0,690	0,918	0,954	0,252	0,403
0,8	0,723	0,990	0,865	0,924	0,970	0,946	0,947
0,9	0,966	0,979	0,917	0,914	0,970	0,966	0,966

Tabel 4.2.1.3 Nilai Kappa hasil pengujian untuk kueri kasus penggunaan no.14 sampai dengan no.20

No.Kueri dan Nilai Kappa							
T	14	15	16	17	18	19	20
0,1	-0,931	-0,949	-0,975	-0,975	-0,971	-0,975	-0,981
0,2	-0,931	-0,949	-0,975	-0,975	-0,971	-0,975	-0,868
0,3	-0,883	-0,949	-0,975	-0,975	-0,971	-0,975	0,050
0,4	-0,080	-0,545	-0,975	-0,975	-0,971	-0,975	0,050
0,5	0,160	-0,131	-0,975	-0,874	-0,971	-0,975	0,413
0,6	0,198	0,243	-0,975	0,280	-0,971	-0,558	0,648
0,7	0,570	0,485	-0,404	0,987	-0,944	0,754	0,693
0,8	0,964	0,848	0,657	1.000	-0,173	0,987	0,951
0,9	0,966	0,966	0,961	1.000	0,888	1.000	1.000

#### 4.2.2. Pengujian Kedua

Pengujian kueri deskripsi kasus penggunaan dengan kombinasi masukan komponen deskripsi kasus penggunaan yang berbeda-beda. Ada dua deskripsi Kasus Penggunaan yang digunakan dalam skenario uji coba yang kedua ini yaitu “Withdraw Cash” dan “Program Switch”.

Tabel 4.2.2.1 Deskripsi Kasus Penggunaan “Withdraw Cash”

<b>Test Case ID:</b>	1
<b>System Name:</b>	Financial or Banking System (eg. ATM)
<b>Use Case Name:</b>	Withdraw Cash
<b>Actors:</b>	Customer
<b>Description:</b>	This use case describes how the Bank Customer uses the ATM to withdraw money to his/her bank account
<b>Preconditions:</b>	There is an active network connection to the Bank. The ATM has cash available.
<b>Postconditions:</b>	The user has received their cash and the internal logs have been updated.
<b>Normal Flow:</b>	1. The use case begins when Bank Customer inserts their Bank Card. 2. Use Case: Validate User is performed.



	<p>3. The ATM displays the different alternatives that are available on this unit. In this case the Bank Customer always selects Withdraw Cash.</p> <p>4. The ATM prompts for an account.</p> <p>5. The Bank Customer selects an account.</p> <p>6. The ATM prompts for an amount.</p> <p>7. The Bank Customer enters an amount.</p> <p>8. Card ID, PIN, amount and account is sent to Bank as a transaction. The Bank Consortium replies with a go/no go reply telling if the transaction is ok.</p> <p>9. Then money is dispensed.</p> <p>10, The Bank Card is returned.</p> <p>11. The receipt is printed.</p> <p>12. The use case ends successfully.</p>
<b>Alternative Flows:</b>	
<b>Exceptions:</b>	
<b>Includes:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

Tabel 4.2.2.2 Hasil Pengujian pada *use case* “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah Result Set	Rata-rata Nilai Kesamaan
1	√							142	0,879139
2	√	√						143	0,728252
3	√	√	√					142	0,628841
4	√	√	√	√				143	0,517633
5	√	√	√	√	√			143	0,435039
6	√	√	√	√	√	√		67	0,284859
7	√	√	√	√	√	√	√	78	0,279775

Tabel 4.2.2.3 Deskripsi Kasus Penggunaan “Program Switch”

<b>Test Case ID:</b>	-
<b>System Name:</b>	
<b>Use Case Name:</b>	Program Switch
<b>Actors:</b>	Home owner or Programmer
<b>Description:</b>	Homeowner or programmer uses central control or PC program to set the behavior of individual light switches in home.
<b>Preconditions:</b>	Homeowner / Programmer has account on system
<b>Postconditions:</b>	Switch is programmed
<b>Normal Flow:</b>	<p>5.0 Program Switch at Central Control Unit</p> <ol style="list-style-type: none"> <li>1. Push program button</li> <li>2. Enter passcode</li> <li>3. Select room switch</li> <li>4. Select light banks to assign to switch</li> <li>5. Add effects, such as dimming and timing</li> <li>6. Save selection</li> </ol>
<b>Alternative Flows:</b>	<p>5.1 Program Switch at PC</p> <ol style="list-style-type: none"> <li>1. Open Lumenations program</li> <li>2. Input authentication</li> <li>3. Select programming mode</li> <li>4. Select switch to program from list</li> <li>5. Select available light banks from list</li> <li>6. Add dimming and timing effects</li> <li>7. Save programming</li> </ol>
<b>Exceptions:</b>	
<b>Includes:</b>	
<b>Assumptions:</b>	
<b>Notes and Issues:</b>	

Tabel 4.2.2.4 Hasil Pengujian pada *use case* “Program Switch” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah Result Set	Rata-rata Nilai Kesamaan
1	√							142	0,857331
2	√	√						142	0,762236
3	√	√	√					142	0,65222
4	√	√	√	√				142	0,538716
5	√	√	√	√	√			142	0,454539
6	√	√	√	√	√	√		129	0,427066
7	√	√	√	√	√	√	√	121	0,361189

#### 4.2.3. Pengujian Ketiga

Pengujian kedua yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kata-kata yang mempunyai kesamaan semantik di salah satu komponen *use case description*.

Tabel 4.2.3.1 Beberapa contoh kata-kata yang mempunyai kesamaan semantik.

No	Kumpulan Kata
1	Recipe, formula
2	Money, fund, cash, coin
3	Select, pick
4	Withdraw, take
5	weapon, gun, pistol
6	Purchase, buy
7	Client, user, customer

Tabel 4.2.3.2 Hasil kueri semantik dengan menggunakan kata frase “Pick gun” pada komponen nama *use case*.

Nama Kasus Penggunaan	Aktor	Nilai Kesamaan
<i>Select weapon</i>	Player	1
Program Switch	Home owner or Programmer	0,915154
Find a Book	Customer	0,893591

Add course block	study administrator	0,876727
Remote Programming	Lumenations Services	0,866828
Select Performance	Venue Manager	0,856239
Add Book to Shopping Cart	Customer	0,846452
Select Payment Method	Employee	0,837092
Place Local Call	Caller and Callee	0,83051
Initiate Emergency Receiver	Resident	0,826237

Tabel 4.2.3.3 Hasil kueri non-semantik(*String-based similarity-dice coefficient*) dengan menggunakan kata frase “Pick gun” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Check Out	Customer	0,02963
Check out	Customer	0,02963
Ticket Print	End User Customer	0,02469
Stock Product	Stock Clerk	0,02339
Check In Customer	Hotel Counter Staff	0,01932
Add course block	study administrator	0,01932
Check Out Customer	Hotel Counter Staff	0,01852
Check Order Status	Customer	0,01852
Check order status	Customer	0,01852
Check room availability	Potential Guest	0,01587

Tabel 4.2.3.4 Hasil kueri semantik dengan menggunakan kata frase “Get coin” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Withdraw Cash	Customer	0,894362
Transfer Money	Customer	0,883077
Transfer Money	customer	0,883077
Deposit Money	Customer	0,856384
Establish Cash Flow Schedule	Operations manager	0,849773
Edit Profile	customer	0,838265
Create Charge Code	Administrative User	0,836104
Write Customer Review	Customer	0,811509
Generate Report	Printer	0,807165

Create Employee	Administrative User	0,804943
-----------------	---------------------	----------

Tabel 4.2.3.5 Hasil kueri non-semantik (*String-based similarity-dice coefficient*) dengan menggunakan kata frase “Get coin” pada komponen nama *use case*.

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Ticket Print	End User Customer	0,03704
Enroll in courses	Applicant Administrator	0,02899
Set Limits	Trading Manager	0,02778
Set Timing Sequence	Home owner Programmer	0,02778
Set Up Users	Operations manager	0,02614
Manage Account	Customer	0,02222
Register course	study administrator	0,02116
Login	Customer Customer Rep	0,0202
Get Call History	Customer	0,0202
Add course block	study administrator	0,01932

Tabel 4.2.3.6 Hasil kueri semantik dengan menggunakan kata frase “Add Formula” pada komponen nama *use case*

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Add Recipe	User	0,967743
Add courses to study year	study administrator	0,85786
Add course block	study administrator	0,841139
Update Account	Accounting System	0,839413
Schedule Event	Venue Manager	0,812303
List Transactions	customer	0,80347
Cancel Event	Venue Manager	0,801246
Ticket Print	End User Customer	0,793081
Record Time	Employee	0,787899
Give Product Information	Inventory System	0,786366

Tabel 4.2.3.7 Hasil kueri non-semantik dengan menggunakan kata “Add Formula” pada komponen nama *use case*

<b>Nama Kasus Penggunaan</b>	<b>Aktor</b>	<b>Nilai Kesamaan</b>
Schedule Performance	Venue Manager	0,03065

Perform Search	Customer	0,02899
Reschedule Performance	Venue Manager	0,02867
Cancel Performance	Venue Manager	0,02778
Search for Order	Customer Customer Rep	0,02667
Select Performance	Venue Manager	0,02469
Add Recipe	User	0,02339
Record Time	Employee	0,02222
Enter Payment Information	Customer	0,0202
Give Product Information	Inventory System	0,0202

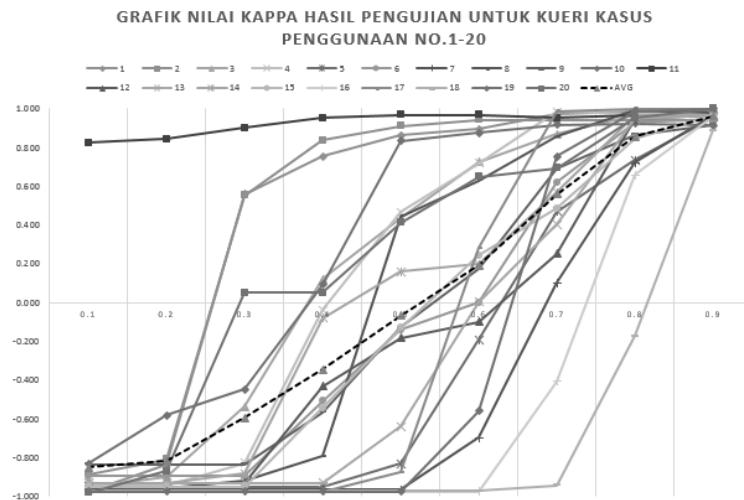
### 4.3 Analisa dan Evaluasi Hasil Pengujian

Dari hasil pengujian penelitian tersebut pada Bab 4.2 kemudian kita analisa menggunakan pengukuran indeks Kappa guna mengetahui atau mengukur tingkat kesepakatan dari dua pengamat atau ahli terhadap karakteristik yang menjadi perhatian penelitian.

#### 4.3.1. Analisa Hasil Pengujian Pertama

Tabel 4.3.1.1 Nilai *threshold* dan rata-rata nilai indeks Kappa dari keseluruhan data uji kueri deskripsi kasus penggunaan.

Nilai Threshold	Rata-rata nilai koefisien Kappa
0,1	-0,849
0,2	-0,817
0,3	-0,595
0,4	-0,345
0,5	-0,066
0,6	0,191
0,7	0,559
0,8	0,856
<b>0,9</b>	<b>0,962</b>

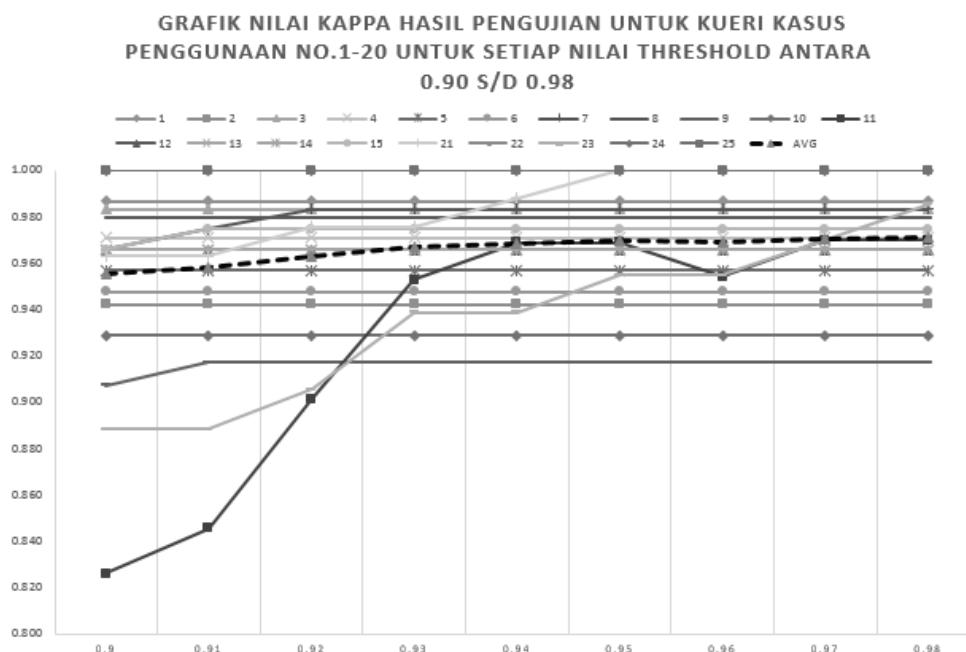


Gambar 4.3.1 Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,1 sampai dengan 0,9.

Hasil pengujian pertama terlihat pada Tabel 4.3.2 yaitu didapat suatu nilai rata-rata indeks Kappa keseluruhan dari hasil kueri kasus penggunaan yaitu sebesar 0,962 dengan nilai *threshold* ideal yang didapat yaitu sebesar 0,9. Nilai ini berarti bahwa metode yang diusulkan mempunyai tingkat kesepakatan terhadap pakar yang tinggi.

Tabel 4.3.1.2 Rata-rata Kappa untuk setiap nilai *threshold* antara 0,90 sampai dengan 0,98.

Threshold (T)	Rata-rata Kappa
0,90	0,955
0,91	0,958
0,92	0,962
0,93	0,967
0,94	0,968
0,95	0,969
0,96	0,969
0,97	0,970
0,98	0,971



Gambar 4.3.2 Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,1 sampai dengan 0,9.

#### 4.3.2. Analisa Hasil Pengujian Kedua

Pengujian kedua dilakukan dengan tujuan untuk mengetahui sejauh mana pengaruh kombinasi masukan pada kueri kasus penggunaan terhadap nilai *similarity*/kesamaan yang dihasilkan. Di pengujian ini diambil contoh salah satu kasus penggunaan yaitu “Withdraw Cash”. Kasus Penggunaan ini kemudian dijadikan kueri dengan kombinasi masukan kueri seperti yang ditampilkan pada Tabel 4.3.4.

Dari hasil pengujian pada Tabel 4.3.4 terlihat bahwa untuk memperoleh hasil kueri yang akurat, diperlukan isian kueri yang lengkap. Semakin lengkap masukan komponen kueri yang dimasukkan hasil yang didapatkan semakin akurat. Hal ini ditunjukkan dengan nilai rata-rata kesamaan yang semakin kecil.

Dari contoh kasus penggunaan “Withdraw Cash”. Jika kita hanya mengisikan komponen aktor saja maka didapat nilai rata-rata kesamaan yang dengan nilai 0,88. Hal ini berbanding terbalik jika kita memasukkan kueri secara lengkap yaitu didapatkan nilai rata-rata kesamaan sebesar 0,28.



Tabel 4.3.2.1 Hasil Pengujian pada use case “Withdraw Cash” dengan kombinasi masukan komponen yang berbeda.

No	Actor	Name	Desc	Pre	Post	BasicFlow	AltFlow	Jumlah	Rata-rata
								Result	Nilai
								Set	Kesamaan
1	√							142	0,879139
2	√	√						143	0,728252
3	√	√	√					142	0,628841
4	√	√	√	√				143	0,517633
5	√	√	√	√	√			143	0,435039
6	√	√	√	√	√	√		67	0,284859
7	√	√	√	√	√	√	√	78	0,279775

Pada contoh yang lain yaitu dengan menggunakan kasus penggunaan “Program Switch” juga dihasilkan hasil yang serupa. Pada Tabel 4.2.5 terlihat bahwa semakin kita mengisi lengkap komponen kueri yang ada maka hasil kueri yang kita dapatkan akan semakin akurat. Hal ini ditunjukkan dengan nilai rata-rata kesamaan yang dihasilkan semakin kecil.

### 4.3.3. Analisa Hasil Pengujian Ketiga

Pengujian ketiga ini, yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kata-kata yang mempunyai kesamaan semantik. Pengujian ini dilakukan untuk mengetahui perbandingan antara hasil kueri dengan menggunakan kesamaan semantik dan non-semantik.

Berdasarkan hasil pengujian pada Tabel 4.2.8 pada contoh kata “Pick gun” terlihat bahwa kata “Pick gun” mempunyai kesamaan semantik dengan kata “Select weapon” dengan nilai kesamaan yang didapat yaitu 1. Hal ini menunjukkan bahwa hasil pencarian yang didapatkan relevan dengan input yang dimasukkan. Sebaliknya hasil pencarian atau kueri non-semantik tidak dapat menghasilkan luaran yang sesuai atau dengan kata lain data yang didapatkan tidak relevan sebagaimana ditunjukkan pada Tabel 4.2.9.

Hasil yang sama juga didapatkan pada contoh kata “Get coin”. Pada Tabel 4.2.10 terlihat hasil kueri dengan nilai kesamaan tertinggi 1 yaitu “Withdraw cash”.

Sedangkan pada hasil kueri non-semantik pada Tabel 4.2.11, kata “Withdraw cash” tidak didapatkan. Secara semantik kata “Get” memiliki kemiripan dengan kata “Withdraw” sedangkan kata “Coin” memiliki kemiripan makna dengan kata “Cash”. Hal ini menunjukkan bahwa penggunaan aspek semantik di dalam proses temu kembali deskripsi kasus penggunaan lebih akurat dibandingkan dengan non-semantik (*String-based similarity*).

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Bab ini membahas tentang kesimpulan hasil penelitian dan saran untuk penelitian selanjutnya dari penulis berdasarkan pada hasil pengujian yang telah dilakukan dalam Bab 4.

#### **5.1 Kesimpulan**

Penelitian ini mengusulkan penambahan aspek semantik di dalam pemrosesan temu kembali kasus penggunaan berdasar struktur deskripsi kasus penggunaan. Teori temu kembali dan penyimpanan informasi dalam bentuk basis data seperti perhitungan tingkat kesamaan (*similarity*) diterapkan dalam proses penyimpanan data dan proses temu kembali. Guna mempermudah dan mempercepat proses evaluasi metode telah dibuat suatu kaskas bantu yang dirancang berdasar skenario pengujian yang telah disusun.

Pengujian terbagi menjadi 3 macam skenario yang berbeda. Pengujian pertama yaitu pengujian untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur kesepakatan hasil yang didapat oleh sistem dengan pakar menggunakan indeks Kappa. Sedangkan pengujian kedua yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kesamaan semantik di salah satu komponen *use case description*. Dan yang terakhir yaitu pengujian kueri deskripsi kasus penggunaan dengan beberapa kombinasi masukan komponen deskripsi *use case* yang berbeda-beda.

Pembuatan dan pengujian metode dilakukan oleh total 2 orang pakar di bidang rekayasa perangkat lunak. Satu pakar membuat kueri kasus penggunaan dan satu pakar lainnya mengeksekusi ketiga skenario berbeda tersebut dan mengamati serta menganalisa hasil yang didapatkan. Beberapa faktor seperti koleksi kasus penggunaan, pakar, proses pencarian dan kueri kasus penggunaan dikontrol dengan tujuan untuk mengurangi sejumlah bias yang terdapat di penelitian ini.

Perhitungan koefisien Kappa dilakukan guna mengetahui tingkat kesepakatan sistem dengan pakar yang mengamati dan melakukan proses pengujian. Dari data statistik hasil pengujian didapat kesimpulan bahwa secara

keseluruhan metode dengan mempertimbangkan arti semantik mempunyai nilai koefisien Kappa rata-rata yaitu sebesar 0,962 dengan nilai *threshold* ideal yang didapat yaitu sebesar 0,9 pada skala 0,1 sampai dengan 0,9. Sedangkan pengujian pada skala rentang 0,9 sampai dengan 0,98 didapatkan rata-rata nilai Kappa tertinggi 0,971 pada nilai *threshold* 0,98. Nilai ini berarti bahwa metode yang diusulkan mempunyai tingkat kesepakatan terhadap pakar yang tinggi. Pada pengujian kedua didapat simpulan bahwa semakin lengkap masukan komponen kueri yang dimasukkan hasil kueri yang didapatkan akan semakin akurat. Hal ini ditunjukkan dengan nilai rata-rata kesamaan yang semakin kecil.

Sedangkan pada pengujian ketiga didapatkan kesimpulan bahwa penggunaan aspek semantik memberikan hasil kueri yang lebih akurat dan relevan dibanding dengan cara pencocokan kata biasa atau non-semantik (*String-based similarity*).

## **5.2 Saran**

Untuk meningkatkan akurasi dan presisi dari perhitungan kesamaan kata atau kalimat, perlu diuji coba lebih lanjut kombinasi *Part-Of-Speech* yang berbeda atau lain pada teks kalimat yang dibandingkan.

- Noun
- Verb
- Adjective
- Modal
- Adverb dan kombinasi POS Tagging lainnya.
- Dan kombinasi jenis kata yang lainnya

## DAFTAR PUSTAKA

- Gomaa, H. (2000). *Designing concurrent, distributed, and real-time applications with UML*. Addison-Wesley.
- Gwet, K. (2002). Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters. *Statistical Methods For Inter Rater Reliability Assessment, I*, 1-5.
- H. G. Woo and W. N. Robinson. (2002). Reuse of scenario specifications using an automated relational learner: a lightweight approach. Requirements Engineering Conference.
- Ilyas, M., & Küng, J. (2009). A Similarity Measurement Framework for Requirements Engineering. *Fourth International Multi-Conference on Computing in the Global Information Technology* (pp. 31-35). IEEE Computer Society.
- Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys*, 24, 131-183.
- Lee, M. C. (2011). A novel sentence similarity measure for semantic-based expert system. *Expert System with Applications*, 38, 6392-6399.
- Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., & Crockett, K. (2006). Sentence Similarity Based on Semantic Nets and Corpus Statistic. *IEEE Transaction on Knowledge and Data Engineering*, 18(8), 1138-1149.
- Liu, H., & Wang, P. (2013). Assessing Sentence Similarity Using WordNet based Word Similarity. *Journal Of Software*, 8(6), 1451-1459.
- M. C. Blok, and J. L. Cybulski. (1998). Reusing UML specifications in a constrained application domain. Asia Pacific Software Engineering Conference.
- M. J. McGill and G. Salton. (1983). *Introduction to modern information retrieval*. McGraw-Hill.
- Marir, F., & Haouam, K. (2004). Rhetorical Structure Theory for content-based indexing and retrieval of web documents. ITRE.
- Mcclure, C. (1997). *Software reuse techniques* (I ed.). Prentice Hall.

- Morales, B. B., Sergio, C., & Clunie, C. (2012, January). Reuse of Use Cases Diagrams: An Approach based on Ontologies and Semantic Web Technologies. *IJCSI International Journal of Computer Science Issues*, 9(1), 24-29.
- Porter, M. F. (1980). An algorithm for suffix stripping, Program. *Automated library and information Systems*, 14, 130-137.
- Saeki, M. (1999). Reusing use case descriptions for requirements specification: towards use case patterns. Asia Pacific Software Engineering Conference.
- Saeki, M. (2000). Patterns and aspects for use cases reuse techniques for use case descriptions. Requirements Engineering Conference.
- Siahaan, D. O. (2012). *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak* (1 ed.). Yogyakarta: ANDI.
- Srisura, B., & Daengdej, J. (2010). Retrieving Use Case Diagram With Case-based Reasoning Approach. *Journal of Theoretical and Applied Information Technology*, 19(2), 68-78.
- Udomchaiporn, A., Prompoon, N., & Kanongchaiyos, P. (2006). Software Requirements Retrieval Using Use Case Terms and Structure Similarity Computation. Kanpur: XIII Asia Pasific Software Engineering Conference.
- Vachharajani, V., & Pareek, J. (2007). Use Case Extractor: XML Parser for Automated Extraction and Storage of Use Case Diagram.

## BIOGRAFI PENULIS



**Ferdika Bagus Pristiawan Permana** adalah nama penulis dari Tesis ini. Penulis adalah anak tunggal dari orang tua Fathur Rochman dan Sukismiswati. Penulis dilahirkan di kota Mojokerto Jawa Timur pada tanggal 5 Agustus 1984. Penulis menempuh pendidikan dimulai dari SDN Gedongan II Mojokerto, melanjutkan ke SLTP Negeri 2 Mojokerto. Pada tahun 1999, penulis pindah ke kota Surabaya untuk menempuh studi sekolah menengah atas di SMU Negeri 6 Surabaya. Penulis kemudian melanjutkan studi S-1 di T.Informatika ITS Surabaya.

Penulis memulai karir pada tahun 2007 sebagai *Software Engineer* di salah satu perusahaan swasta di Jepang. Sekitar kurang lebih 3 tahun, penulis berkecimpung dan menggali pengalaman seputar *Information Communication Technology (ICT)* di negeri sakura.

Kemudian pada tahun 2010, penulis kembali ke tanah air dan bergabung dengan Tim IT di UPT TIK Universitas Brawijaya sejak tahun 2012 hingga sekarang.

Dengan ketekunan, motivasi tinggi untuk terus belajar dan berusaha, penulis telah berhasil menyelesaikan penulisan Tesis ini. Penulis berharap Tesis ini nantinya dapat bermanfaat bagi kita semua.

Akhir kata penulis mengucapkan rasa syukur yang sebesar-besarnya atas terselesaikannya Tesis yang berjudul “Pendekatan Kesamaan Semantik dan Struktur dalam Kasus Penggunaan untuk Mendapatkan Kembali Spesifikasi Kebutuhan Perangkat Lunak”.