

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

Ferdika Bagus Pristiawan Permana¹⁾, Daniel Oranova Siahaan²⁾

¹⁾²⁾*Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111*

¹⁾ferdika14@mhs.if.its.ac.id

²⁾daniel@if.its.ac.id

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. Makalah 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—*Temu Kembali Kasus Penggunaan, Kesamaan Semantik, Sistem Temu Kembali Informasi*

Abstract— In software industry, there are tremendous number of software requirements documents in effect of large scale of software development. These 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 paper presents a methodology to retrieve software requirements documents using structured base of use case description semantic similarity computation. Each element of query use case description will be calculated with the use case description in the collections or repository. The semantic similarity score is used to rank the use case in the collections and to retrieve 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 scenarios are presented to investigate the ideal threshold value, the retrieval result differences with previous approach and to find out the effect of various combinations of structural query to the retrieval result.

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

Article history:

Received 14 July 2016; Received in revised form 30 July 2016 & 2 August 2016; Accepted 2 August 2016;

Available online 28 October 2016

I. PENDAHULUAN

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, analisis, 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 yaitu mengurangi biaya dan alokasi waktu pada tahap pengembangan perangkat lunak. Salah satu tahap dalam pengembangan perangkat lunak yaitu tahap pemodelan atau rekayasa kebutuhan. Berbagai pemodelan kebutuhan digunakan oleh para analis. Diantaranya yaitu melalui pemodelan kasus penggunaan seperti pada penelitian (Gooma, 2006). 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 analisis kebutuhan. Akan tetapi di dalam penerapannya timbul permasalahan diantaranya yaitu bagaimana mengukur tingkat kesamaan diantara kasus penggunaan. (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) mengusulkan metode pengukuran kesamaan kasus penggunaan melalui perhitungan tingkat kesamaan dari deskripsi kasus penggunaan.

Namun metode yang digunakan masih terbatas pada kesamaan *string* kata yang sama persis. Metode tersebut belum mampu mencakup atau mendeteksi kata-kata yang mempunyai kesamaan makna atau arti kata (semantik).

Berbagai penelitian telah dilakukan terkait penggunaan kembali spesifikasi kebutuhan perangkat lunak ini yaitu diantaranya yang dilakukan oleh (Blok & Cybulski, 1998), (Saeki, 1999), (Saeki, 2000), (Woo & 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 repositori.

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 diagram kasus

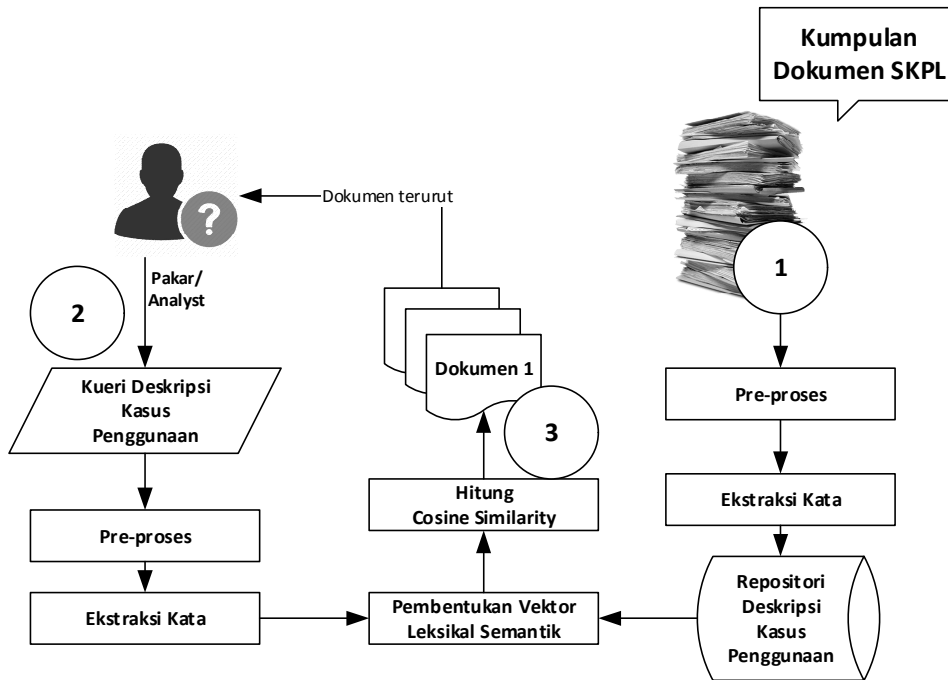
penggunaan. Untuk membuat perangkat lunak yang baru, spesifikasi kebutuhan yang sudah ada atau lama tersimpan di dalam suatu repositori diambil kembali berdasarkan kesamaan diagram kasus penggunaan. Diagram kasus penggunaan dan spesifikasi kebutuhan yang memiliki tingkat kesamaan yang paling tinggi akan 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 aktor dan kasus penggunaan. Pencocokan kata masih menggunakan terbatas pada *“exactly matching word”* atau pencocokan kata yang sama persis.

(Woo & Robinson, 2002) mengusulkan teknik otomatisasi yang berfungsi untuk membantu proses penggunaan kembali diagram kasus penggunaan. Mereka mengembangkan kakas bantu yang disebut *“ScenAsst”* untuk mengumpulkan dan mendapatkan kembali kasus penggunaan secara praktis. ScenAsst mengubah isi dari use case ke dalam bentuk graph, meng-clusternya dan menyimpannya ke dalam sebuah koleksi kasus penggunaan. Di dalam proses retrieval, *query* dari pengguna juga diubah ke dalam bentuk graph. Derajat kesamaan antara *query* pengguna dan grafik yang dihasilkan di dalam koleksi penyimpanan dibandingkan dengan menggunakan algoritma *“SUBDUE”*. Teknik ini menghasilkan respon proses yang lama karena faktor kompleksitas (Blok & Cybulski, 1998).

Penelitian ini bertujuan untuk melengkapi kekurangan-kekurangan dari penelitian oleh (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) tersebut yaitu dengan menambahkan pendeteksian arti kata semantik pada pencocokan deskripsi kasus penggunaan pada proses penggunaan kembali komponen perangkat lunak (*software reuse*). Hipotesis awal dari penelitian ini yaitu dengan penambahan proses perhitungan kesamaan semantik pada kata atau kalimat akan mampu meningkatkan efektifitas dari proses temu kembali kasus penggunaan.

II. METODE YANG DIUSULKAN

Metode temu kembali dokumen spesifikasi perangkat lunak melalui pengukuran kesamaan deskripsi kasus penggunaan yang diusulkan dalam makalah ini terdiri dari tiga proses utamayaitu pembuatan repositori Deskripsi Kasus Penggunaan, melakukan proses temu kembali, yang didalamnya terdapat proses perhitungan kesamaan semantik dan yang terakhir proses evaluasi. Metode tersebut disusun berdasarkan beberapa metode ada sebelumnya diantaranya yaitu (Liu & Wang, 2013), (Li, McLean, Bandar, O'Shea, & Crockett, 2006), (Slimani, 2013) dan (Li, Tian, Ye, & Cai, 2010).



Gambar 1. Alur proses temu kembali Kasus Penggunaan

A. Repositori Deskripsi Kasus Penggunaan

Sesuai Gambar 1 nomor 1, proses pertama kali yang dilakukan yaitu pembuatan repositori Deskripsi Kasus Penggunaan. Proses ini meliputi beberapa pre-proses sebagai berikut ini:

- Parsing isi dari Kasus Penggunaan dan melakukan proses tokenisasi.
- Menghilangkan kata-kata yang termasuk dalam *stop list* bahasa Inggris. Daftar stop list didapatkan dari <http://www.cs.cmu.edu/~mccallum/bow/>.
- Melakukan proses stemming untuk setiap kata
- Melakukan proses POS Tagging pada setiap kata. Proses ini bersifat optional dalam arti tergantung kebutuhan. Jika kita ingin membandingkan semua kata maka proses ini tidak diperlukan. Sebaliknya, jika kita ingin membandingkan jenis kata tertentu semisal: Kata benda atau kata kerja maka proses ini diperlukan
- Menyimpan Deskripsi Kasus Penggunaan (setelah dilakukan pre-proses) ke dalam repositori.

B. Proses Temu Kembali (Retrieval Process)

Pada Gambar 1 nomor 2 yaitu proses yang dilakukan terhadap kueri Kasus Penggunaan yang dimasukkan oleh Pakar selaku pengguna sistem. Pada dasarnya pre-proses kueri Kasus Penggunaan sama dengan yang dilakukan pada proses nomor 1. Perbedaannya yaitu pada proses nomor 1 luaran pre-proses Deskripsi Kasus Penggunaan langsung

disimpan ke dalam repositori sedangkan pada proses nomor 2 luaran hasil proses langsung dibandingkan atau dihitung kesamaannya dengan masing-masing data yang ada di dalam repositori.

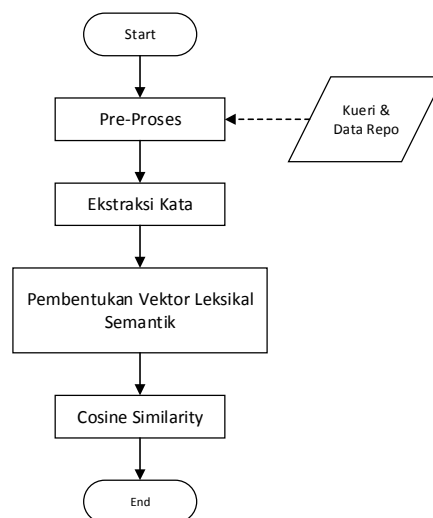
$$UCSimilarity(Query, Repo) = \frac{\sum Similarity(EQ_i, ER_i)}{n} \quad (1)$$

dimana,

EQ_i : Elemen ke i pada kueri Kasus Penggunaan

ER_i : Elemen ke i pada kueri Kasus Penggunaan pada repositori.

n : Jumlah elemen Deskripsi Kasus Penggunaan yang diisikan sebagai Kueri.



Gambar 2. Alur proses perhitungan kesamaan semantik

Gambar 2 menjelaskan proses perhitungan kesamaan semantik pada kata atau kalimat (Liu &

Wang, 2013). Gambar 2 merupakan subproses dari proses pada Gambar 1 nomor 3.

Proses perhitungan kesamaan semantik terbagi menjadi empat bagian utama (Liu & Wang, 2013):

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

C. Evaluasi

Kami tidak dapat melakukan evaluasi perbandingan dengan metode sebelumnya (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) dikarenakan data set yang digunakan tidak sama persis. Sebagai alternatif, untuk mengukur sampai sejauh mana tingkat efektivitas metode yang diusulkan, digunakan pengukuran koefisien Kappa (Hasil dari sistem dengan penilaian Pakar).

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.

Berikut inipada Tabel 1. adalah contoh kasus perhitungan menggunakan koefisien Kappa menurut (Gwet, 2002).

TABEL 1. HASIL PERCOBAAN 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".

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

$$P_1 = \frac{(A1+B1)/2}{N} \quad (3)$$

$$Kappa(AC1) = \frac{p-e(\gamma)}{1-e(\gamma)} \quad (4)$$

$$p = \frac{A+D}{N} \quad (5)$$

Dari rumus atau persamaan Eq. 2, Eq. 3, Eq. 4 dan Eq. 5 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:

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

III. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan hal-hal terkait pengujian metode yang diusulkan pada makalah ini yang meliputi lingkungan pengujian, skenario pengujian dan hasil pengujian.

A. Lingkungan Pengujian

1) Dataset Koleksi Deskripsi Kasus Penggunaan

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 dan kemudian di-*import* ke dalam basis data MariaDB. Total kasus penggunaan yang dapat diambil sebagai dataset sejumlah 137 kasus penggunaan dari 20 sistem atau projek aplikasi.

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 *English as a Foreign Language* (EFL-ITS) lebih dari 500.

3) Proses Pencarian

Sekitar 20 sistem yang terbagi menjadi 5 domain telah dibuat guna menguji metode penelitian yang diusulkan. Pengguna atau pakar penguji membuat beberapa kombinasi kueri kasus penggunaan 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.

Pembagian 5 domain tersebut terdiri atas *Finantial Calculation*, *e-Commerce/Trading System*, *Information Management System*, *Teaching-Studying System*, dan *Other*

4) *Kueri Deskripsi Kasus Penggunaan*

Pakar membuat total sekitar 20 pernyataan kueri baik kueri berdasar berdasar struktur kasus penggunaan untuk semua subjek domain. 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.

B. *Skenario Pengujian*

Pengujian dilakukan dengan tiga skenario yang berbeda:

1) *Skenario Pertama*

Pengujian untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem terhadap penilaian pakar menggunakan indeks Kappa.

2) *Skenario Kedua*

Pengujian *query use case* dengan kombinasi masukan elemen atau item *use case* yang berbeda-beda.

3) *Skenario Ketiga*

Pengujian kueri kasus penggunaan dengan menggunakan kesamaan semantik di salah satu elemen atau item dari *use case description*.

C. *Hasil Pengujian*

Berikut ini adalah hasil pengujian metode berdasarkan skenario pengujian yang telah disusun sebelumnya:

1) *Pengujian Pertama*

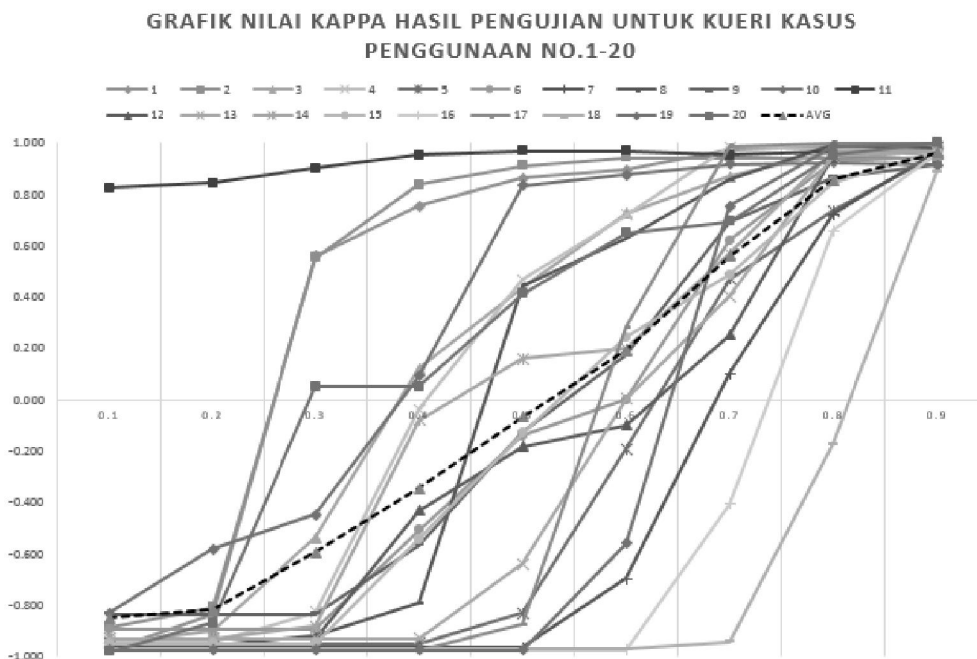
Pengujian pertama dilakukan untuk mengetahui atau mencari nilai *threshold* yang ideal dengan cara mengukur hasil yang didapat oleh sistem dengan hasil yang didapat dari para pakar menggunakan indeks Kappa. *Threshold* yang digunakan yaitu pada rentang 0,1 sampai dengan 0,9 (Gambar 3).

Hasil pengujian pertama terlihat pada Tabel 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 2. NILAI THRESHOLD DAN RATA-RATA NILAI INDEKS KAPPA DARI KESELURUHAN DATA UJI KUERI DESKRIPSI CASUS PENGGUNAAN

Threshold	Rata-rata 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
0,9	0,962

Gambar 4 dan 5 menunjukkan grafik nilai Kappa untuk setiap kueri pada *threshold* di rentang 0.9 sampai dengan 0.98. Sedangkan Tabel 3 menunjukkan nilai rata-rata Kappa pada setiap rentang *threshold*. Dari Tabel 3 dan 4 dapat dilihat bahwa *threshold* ideal yang didapatkan yaitu sebesar 0,98.



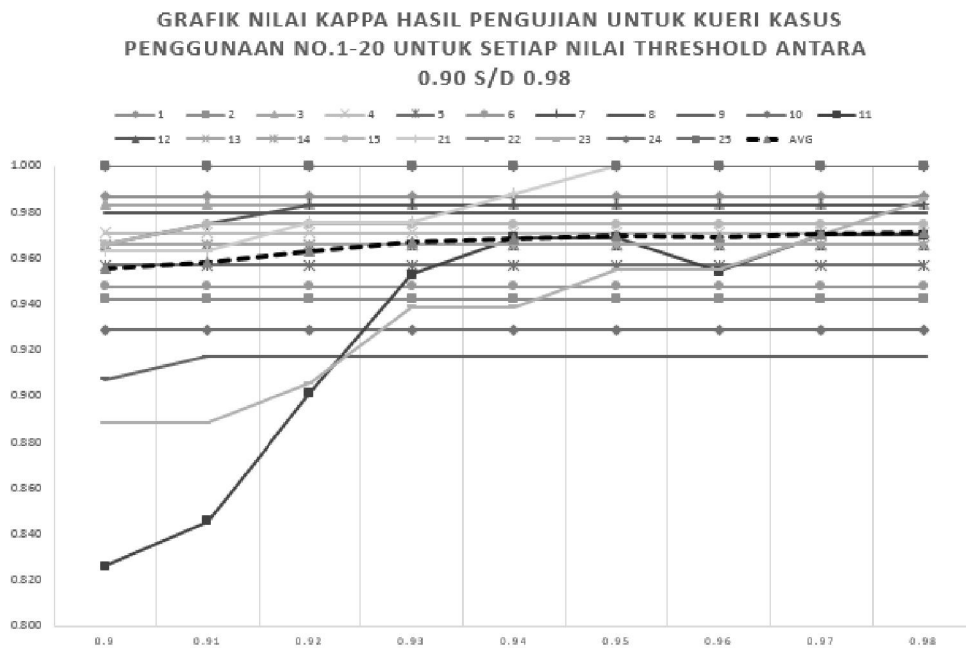
Gambar 3. Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,1 sampai dengan 0,9

TABEL 3 RATA-RATA KAPPA UNTUK SETIAP NILAI THRESHOLD ANTARA 0,90 SAMPAI DENGAN 0,98

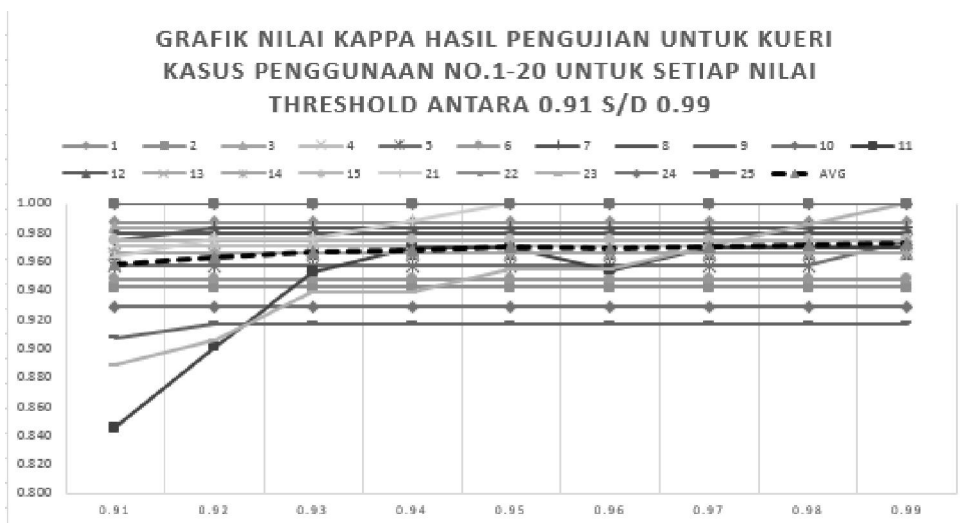
Threshold (T)	Rata-rata Kappa
0,90	0,955
0,91	0,958
0,92	0,962
0,93	0,967
0,94	0,968
0,95	0,969
0,96	0,969
0,97	0,970
0,98	0,971

TABEL 4 RATA-RATA KAPPA UNTUK SETIAP NILAI THRESHOLD ANTARA 0,90 SAMPAI DENGAN 0,98

Threshold (T)	Rata-rata Kappa
0,91	0,957
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
0,99	0,973



Gambar 4. Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,9 sampai dengan 0,98



Gambar 5. Grafik nilai Kappa untuk setiap nilai *threshold* antara 0,91 sampai dengan 0,99

TABEL 5. NILAI KAPPA PADA THRESHOLD ANTARA 0,91 SAMPAI DENGAN 0,99 PADA KUERI NO.1 HINGGA 10

T	Kappa									
	1	2	3	4	5	6	7	8	9	10
0.91	0.987	0.942	0.983	0.971	0.957	0.947	0.974	0.979	0.907	0.929
0.92	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.93	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.94	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.95	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.96	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.97	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.98	0.987	0.942	0.983	0.971	0.957	0.947	0.983	0.979	0.917	0.929
0.99	0.987	0.942	0.983	0.971	0.974	0.947	0.983	0.979	0.917	0.929

TABEL 6. NILAI KAPPA PADA THRESHOLD ANTARA 0,91 SAMPAI DENGAN 0,99 PADA KUERI NO.11 HINGGA 20

T	Kappa										Rata-rata
	11	12	13	14	15	16	17	18	19	20	
0.91	0.845	0.966	0.966	0.966	0.974	0.963	1.000	0.888	1.000	1.000	0.957
0.92	0.901	0.966	0.966	0.966	0.974	0.976	1.000	0.905	1.000	1.000	0.962
0.93	0.953	0.966	0.966	0.966	0.974	0.976	1.000	0.939	1.000	1.000	0.967
0.94	0.969	0.966	0.966	0.966	0.974	0.988	1.000	0.939	1.000	1.000	0.968
0.95	0.969	0.966	0.966	0.966	0.974	1.000	1.000	0.955	1.000	1.000	0.969
0.96	0.954	0.966	0.966	0.966	0.974	1.000	1.000	0.955	1.000	1.000	0.969
0.97	0.970	0.966	0.966	0.966	0.974	1.000	1.000	0.970	1.000	1.000	0.970
0.98	0.970	0.966	0.966	0.966	0.974	1.000	1.000	0.985	1.000	1.000	0.971
0.99	0.970	0.966	0.966	0.966	0.974	1.000	1.000	1.000	1.000	1.000	0.973

Pada Tabel 5 dan Tabel 6 terlihat sebaran nilai Kappa hasil pengujian pada masing-masing kueri yaitu no.1 hingga 20 pada *threshold* rentang 0,91 hingga 0,99. Pada kedua tabel tersebut terlihat memang rata-rata Kappa semakin naik berbanding lurus dengan nilai *threshold*. Namun jika kita lebih detil Kappa pada masing-masing hasil kueri, semakin tinggi nilai *threshold*, nilai kappa tidak semuanya ikut semakin tinggi (contoh pada Kappahasil kueri no.1, 2, 3, 4, 6, 8, 10, 12, 13, 14, 15, 17, 19, 20). Fenomena ini dipengaruhi dari beberapa faktor yaitu diantaranya kualitas atau variasi komponen dari masing-masing kueri, jumlah data yang ada dalam repositori dan pakar atau ahli.

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” yang diambil dari sumber http://epf.eclipse.org/wikis/openup/core.tech.com.mon.extend_supp/guidances/examples/use_case_spec_CD5DD9B1.html.

TABEL 7. DESKRISI KASUS PENGGUNAAN “WITHDRAW CASH”

Test Case ID	1
System Name	Financial or Banking System (eg. ATM)
Name	Withdraw Cash
Actors	Customer
Description	This use case describes how the Bank Customer uses the ATM to withdraw money to his/her bank account
Pre-conditions	There is an active network connection to the Bank. The ATM has cash available.
Post-conditions	The user has received their cash and the internal logs have been updated.
Normal Flow	1. The use case begins when Bank Customer inserts their Bank Card. 2. Use Case: Validate User is performed. 3. The ATM displays the different alternatives that are available on this unit. In this case the Bank Customer always selects Withdraw Cash. 4. The ATM prompts for an account. 5. The Bank Customer selects an account. 6. The ATM prompts for an amount. 7. The Bank Customer enters an amount. 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. 9. Then money is dispensed. 10. The Bank Card is returned. 11. The receipt is printed. 12. The use case ends successfully.
Alternative Flows	-

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 7.

Dari hasil pengujian pada Tabel 8. 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 8. HASIL PENGUJIAN PADA USE CASE “WITHDRAW CASH” DENGAN KOMBINASI KOMPONEN

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
√							142	0,879
√	√						143	0,728
√	√	√					142	0,628
√	√	√	√				143	0,517
√	√	√	√	√			143	0,435
√	√	√	√	√	√		67	0,284
√	√	√	√	√	√	√	78	0,279

TABEL 9. HASIL PENGUJIAN PADA USE CASE “PROGRAM SWITCH” DENGAN KOMBINASI KOMPONEN

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
√							142	0,851
√	√						142	0,762
√	√	√					142	0,652
√	√	√	√				142	0,538
√	√	√	√	√			142	0,454
√	√	√	√	√	√		129	0,427
√	√	√	√	√	√	√	121	0,361

Keterangan:

- (1) : Actor
- (2) : Use Case Name
- (3) : Description / Objective
- (4) : Pre-Conditions
- (5) : Post-Conditions
- (6) : Basic / Normal Flow
- (7) : Alternatif Flow
- (8) : Jumlah Result set.
- (9) : Rata-rata nilai kesamaan

Pada contoh yang lain yaitu dengan menggunakan kasus penggunaan “Program Switch” juga dihasilkan hasil yang serupa. Pada Tabel 9. 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.

3) Pengujian Ketiga

Pengujian ketiga yaitu pengujian kueri deskripsi kasus penggunaan dengan menggunakan kata-kata yang mempunyai kesamaan semantik (seperti pada Tabel 10) di salah satu komponen *use case description*.

TABEL 10. CONTOH KUMPULAN 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 11. HASIL PENGUJIAN DENGAN KUERI KATA “PICK GUN” DENGAN SEMANTIK

Nama Kasus Penggunaan	Aktor	Nilai Kesamaan
Select weapon	Player	1
Program Switch	Home owner or Programmer	0,915
Find a Book	Customer	0,893
Add course block	administrator	0,876
Remote Programming	Lumenations	0,866
Select Performance	Services	0,856
Add Book to Shopping	Venue Manager	
Cart	Customer	0,846
Select Payment Method	Employee	0,837
Place Local Call	Caller and Callee	0,830
Initiate Emergency	Receiver	0,826
Receiver	Resident	0,826

TABEL 12. HASIL PENGUJIAN DENGAN KUERI KATA “PICK GUN” NON-SEMANTIK

Nama Kasus Penggunaan	Aktor	Nilai Kesamaan
Check Out	Customer	0,029
Check out	Customer	0,029
Ticket Print	End User	0,024
Stock Product	Customer	0,023
Check In Customer	Stock Clerk	0,019
Add course block	Hotel Counter	0,019
Check Out Customer	Staff	0,018
Check Order Status	study	0,018
Check order status	administrator	0,018
Check room availability	Hotel Counter	0,018
	Staff	0,015
	Customer	
	Customer	
	Potential Guest	

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. Pengujian ketiga ini merupakan usulan perbaikan bagi metode sebelumnya pada

penelitian (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006).

Berdasarkan hasil pengujian pada Tabel 11. pada contoh kasus penggunaan "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 12.

IV. 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 kakas 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 menganalisis 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 penelitian sebelumnya (Udomchaiporn, Prompoon, & Kanongchaiyos, 2006) didapatkan simpulan bahwa template yang efektif dari deskripsi kasus penggunaan terdiri dari tiga komponen saja yaitu *Use Case Name, Objective/Description* dan *Actor*. Untuk mendapatkan hasil temu kembali yang efektif tidak perlu memasukkan kueri pada semua komponen dari deskripsi kasus penggunaan. Namun, pada pengujian kedua penelitian ini didapat simpulan yang berbeda yaitu 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*). Hal ini membuktikan hipotesis awal dari penelitian ini yaitu dengan memasukkan aspek kesamaan semantik dapat meningkatkan efektifitas dari proses temu kembali kasus penggunaan.

V. DAFTAR PUSTAKA

- Blok, M. C., & Cybulski, J. L. (1998). Reusing UML specifications in a constrained application domain. *Asia Pacific Software Engineering Conference* (pp. 196-202). Taipei: IEEE.
- Gooma, H. (2006). Designing concurrent, distributed, and real-time applications with UML. *Proceedings of the 28th international conference on Software engineering* (pp. 1059-1060). New York: ACM.
- Gwet, K. (2002). Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters. *Statistical Methods For Inter Rater Reliability Assessment*, 1, 1-5.
- Krueger, C. (1992). Software reuse. *ACM Computing Surveys*, 24 (2), 131-183.
- Li, H., Tian, Y., Ye, B., & Cai, Q. (2010). Comparison of Current Semantic Similarity Methods in WordNet. *International Conference on Computer Application and System Modeling* (pp. V4-408-V4-411). Taiyuan: IEEE.
- 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.
- Marir, F., & Haouam, K. (2004). Rhetorical Structure Theory for content-based indexing and retrieval of web documents. *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on* (pp. 160-164). London: ITRE.
- McClure, C. (1997). *Software reuse techniques, 1 ed.* Upper Saddle River: Prentice Hall.
- Saeki, M. (2000). Patterns and aspects for use cases reuse techniques for use case descriptions. *4th International Conference on Requirements Engineering* (pp. 62-66). Schaumburg: IEEE.
- Saeki, M. (1999). Reusing use case descriptions for requirements specification: towards use case patterns. *Software Engineering Conference, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific* (pp. 309-316). Takamatsu: IEEE.
- Siahaan, D. O. (2012). *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak* (I ed.). Yogyakarta: ANDI.
- Slimani, T. (2013). Description and Evaluation of Semantic similarity Measures Approaches. *International Journal of Computer Applications* , 80 (10), 25-33.
- 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. *2006 13th Asia Pacific Software Engineering Conference (APSEC'06)* (pp. 113-120). Kanpur: IEEE.
- Woo, H. G., & Robinson, W. N. (2002). Reuse of scenario specifications using an automated relational learner: a lightweight approach. *IEEE Joint International Conference on Requirements Engineering* (pp. 173-180). Essen: IEEE.