

REKOMENDASI KASUS PENGGUNAAN BERDASARKAN SKENARIO NARATIF MENGGUNAKAN TEKNOLOGI SEMANTIK

Reza Fauzan¹⁾, Daniel Oranova Siahaan²⁾, Nurul Fajrin Ariyani³⁾

^{1,2,3)} Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

¹⁾ fauzan.reza12@mhs.if.its.ac.id, ²⁾ daniel@if.its.ac.id, ³⁾ nurulfajrin@if.its.ac.id

ABSTRAK

Pada saat ini diagram kasus penggunaan sering digunakan dalam pendefinisian kebutuhan sistem. Sebelum pembangunan diagram kasus penggunaan tersebut, seorang sistem analis membuat sebuah skenario agar konsumen lebih mengerti tentang sistem yang akan diberikan kepada mereka. Kemudian seorang analis melakukan analisa kembali pada kebutuhan sistem untuk membuat diagram kasus penggunaan. Makalah ini bertujuan untuk mengajukan metode baru yang bersifat penelitian awal dengan melakukan analisa secara manual dari setiap langkah yang diajukan. Makalah ini mengajukan suatu langkah kerja dalam melakukan generasi sebuah skenario bebas menjadi metadata-metadata yang memiliki potensi menjadi kasus penggunaan. Setelah dilakukan ekstraksi metadata dari skenario, metadata tersebut akan dicari kemiripannya dengan kasus penggunaan yang ada dalam repositori sehingga diharapkan dapat meningkatkan relevansi dari metadata yang telah digenerasi tersebut. Luaran yang dihasilkan adalah rekomendasi kasus penggunaan dari hasil ekstraksi pada skenario dan pencarian kesamaan dengan kasus penggunaan pada repositori. Dengan melakukan ekstraksi pada skenario bebas, sistem diharapkan dapat membantu dalam pemberian rekomendasi diagram kasus penggunaan dan dapat meningkatkan penggunaan ulang dari kasus penggunaan sebelumnya dengan menggunakan repositori.

Kata Kunci: Kasus Penggunaan, Skenario, Pemrosesan Bahasa Alami, Semantik, Rekayasa Kebutuhan

1. PENDAHULUAN

Pembangunan sebuah diagram kasus penggunaan (*use case diagram*) merupakan teknik yang populer digunakan dalam rekayasa kebutuhan [1]. Diagram kasus penggunaan dianggap dapat mewakili kebutuhan utama dari sistem yang akan dibangun. Karena itu, diagram kasus penggunaan dapat digunakan untuk memahami bagaimana kebutuhan sistem dan bagaimana sistem seharusnya bekerja.

Sebelum membangun sebuah diagram kasus penggunaan, seorang analis membuat skenario terlebih dahulu agar tidak terjadi kesalahpahaman antara klien dengan pihak pengembang. Skenario adalah sebuah cerita atau narasi yang menjelaskan alur dari sistem tersebut. Akan tetapi skenario bukan deskripsi spesifikasi teknis dari suatu produk perangkat lunak. Skenario memiliki beberapa komponen utama, yaitu tujuan, ruang lingkup, sudut pandang pemangku kepentingan, visualisasi, singkat, rekursif, dekomposisi, dan penyempurnaan [2].

Dari sebuah skenario, seorang analis dapat dengan mudah menganalisa apa saja diagram kasus penggunaan yang akan dibuat. Hal ini dikarenakan, pada skenario dituliskan siapa saja aktor yang berperan di dalamnya dan apa saja hal yang dilakukan oleh aktor itu sendiri. Karena itu, ada

keterkaitan yang kuat antara skenario dan diagram kasus penggunaan.

Pada penelitian sebelumnya, terdapat penelitian yang melakukan ekstraksi dari skenario ke diagram kasus penggunaan [3]. Penelitian tersebut menggunakan empat konsentrasi elisitasi, yaitu bisnis, servis, interaksi, dan internal aspek. Pembangunan skenario juga diharuskan sesuai template yang mereka berikan agar mudah dilakukan elisitasinya. Penulisan pada skenario harus mengikuti format yang terdiri dari subjek, predikat, tujuan, dan arahan.

Penelitian lainnya adalah dengan membuat sebuah kriteria bagaimana membuat skenario yang baik [4]. Pada penelitian ini digunakan empat kriteria untuk mengukur kualitas dari sebuah skenario. Penelitian ini hanya menunjukkan bagaimana cara membuat skenario yang baik sehingga pembangunan skenario dapat menghasilkan kebutuhan perangkat lunak yang efektif.

Sekarang ini, pembangunan skenario sering memiliki struktur dan karakteristik yang berbeda-beda. Tidak semua orang dapat menuliskan dengan benar sebuah skenario. Pembuatan cerita naratif pada skenario tidak dapat ditentukan oleh sebuah template saja dan harus mengikuti secara keseluruhan template tersebut. Setiap analis membangun sebuah skenario hanya sesuai dengan persepsi mereka tanpa aturan

atau template tertentu. Sehingga muncul pertanyaan bagaimana melakukan generasi sebuah skenario ke diagram kasus penggunaan jika pembuatan skenario tidak sesuai dengan template atau aturan?

Dari permasalahan generasi sebuah skenario ke diagram kasus penggunaan, penulis mencoba membuat sebuah langkah kerja dalam melakukan generasi skenario bebas tanpa menggunakan template. Kemudian hasil generasi tersebut akan dilakukan pencocokkan dengan kasus penggunaan dalam repositori. Hal ini dilakukan agar analisis tidak melakukan pembangunan diagram kasus penggunaan dari awal. Penggunaan diagram kasus penggunaan yang ada dalam repositori dilakukan agar dapat mempercepat pengiriman dokumen kepada klien. Pencarian menggunakan semantik juga diperlukan agar pencocokkan skenario yang dihasilkan menjadi relevan dan dapat memberikan usulan diagram kasus penggunaan yang sesuai dengan skenario yang diinginkan.

2. STUDI PUSTAKA

2.1 Skenario

Skenario adalah suatu cerita atau narasi yang berisi alur dari suatu sistem yang membuat aplikasi jauh lebih hidup [2]. Sebagai contoh mesin ATM, skenario harus dapat menunjukkan kelebihan-kelebihan mesin ATM agar masyarakat lebih memilih menjadi nasabah bank tersebut karena kelebihan-kelebihan yang diberikan mesin ATM tersebut.

Sebuah skenario yang baik adalah skenario yang memenuhi beberapa parameter yang telah ditentukan. Berikut adalah parameter pengukuran kualitas dari skenario [2] :

- a. Mudah diakses dan mudah dimengerti
- b. Penting, berharga, menarik, atraktif
- c. Kritis dan menantang
- d. Sering digunakan dan menghindari hal-hal di luar kebiasaan
- e. Spesifik

2.2 Kasus penggunaan

Diagram kasus penggunaan merupakan salah satu diagram untuk memodelkan aspek perilaku sistem atau digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem [5]. Diagram kasus penggunaan bertujuan untuk mendapatkan pemahaman tentang perangkat lunak yang akan dikembangkan oleh pengembang. Diagram kasus penggunaan juga dapat membantu dalam menyusun kebutuhan sebuah sistem dengan cara mengkomunikasikan rancangan terhadap pelanggan.

2.3 Pemrosesan Bahasa Alami

Pemrosesan bahasa alami adalah metode yang digunakan untuk melakukan proses komputasi

dari bahasa sehingga terciptanya interaksi antara manusia dengan komputer menggunakan bahasa alami. Salah satu alat yang digunakan dalam pemrosesan bahasa alami 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* [6].

StanfordNLP dalam penelitian ini difokuskan untuk melakukan *tagging*. *Tagging* adalah penandaan kata dalam kalimat untuk menentukan labelnya dengan menggunakan *Part of Speech (POS)* dari model *Pen Tree Bank*.

2.4 Ontologi

Ontologi dapat didefinisikan sebagai sebuah struktur hirarki dari istilah untuk menjelaskan sebuah domain yang dapat digunakan sebagai landasan untuk sebuah basis pengetahuan [7]. Ontologi memberikan penjelasan secara eksplisit dari konsep sebagai representasi pengetahuan itu sendiri.

2.5 SUMO

Suggested Upper Merged Ontology (SUMO) adalah sebuah ontologi yang dibangun dari 1000 konsep dan 4000 definisi dari pernyataan [8]. SUMO dikembangkan dengan berbagai tahap dan eksperimen sehingga dapat membuatnya menjadi sebuah ontologi standar. Selain itu, SUMO merupakan pengembangan dari *Mid-Level Ontology (MILO)* dan sejumlah domain ontologi. Salah satu keunggulan dari SUMO adalah SUMO dapat digunakan dalam bentuk sub-ontologi saja. Sehingga penulis tidak perlu mengambil semua bagian dari ontologi, penulis hanya mengambil bagian ontologi yang diperlukan saja.

Salah satu kelebihan lain dari SUMO adalah tersedianya pemetaan dari kata-kata yang ada di dalam *WordNet* [9][10]. *WordNet* adalah sebuah kosakata yang digunakan sebagai standar dari pemrosesan bahasa alami. *WordNet* mendefinisikan sebuah kata menjadi sebuah konsep sehingga dapat berhubungan dengan kata yang lain seperti sebuah kata "pergi" memiliki sinonim kata "berangkat".

Di dalam SUMO, SUMO tidak merepresentasikan kata tersebut sebagai kata, tetapi SUMO merepresentasikan kata sebagai kelas. Kelas-kelas yang ada di dalam SUMO direlasikan dengan kelas yang lain melalui tiga jenis relasi. Relasi yang digunakan di dalam SUMO adalah *equivalent*, *instance*, dan *subsumption*.

2.6 Semantik

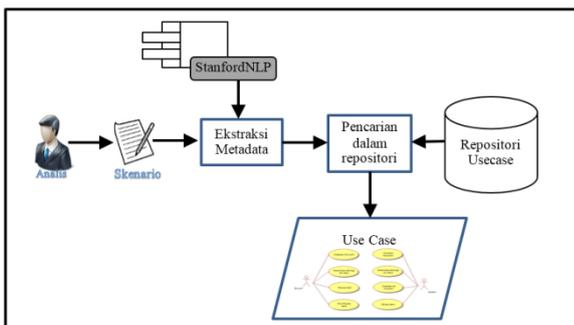
Semantik adalah ilmu tentang makna kata dan kalimat; pengetahuan mengenai seluk-beluk dan pergeseran arti kata; bagian struktur bahasa

yang berhubungan dengan makna ungkapan atau struktur makna suatu wicara. Jaringan semantik adalah sebuah jaringan yang mewakili hubungan antar konsep. Jaringan semantik biasa digunakan sebagai bentuk representasi pengetahuan. Jaringan semantik berupa grafik berarah atau tidak yang terdiri dari simpul dan garis.

Teknologi semantik yang digunakan pada penelitian ini adalah dengan menggunakan RDF sebagai basis data. Penggunaan RDF sebagai basis data karena lebih mudah beradaptasi terhadap perubahan. RDF dapat menambahkan atribut baru pada basis data dan tidak menyebabkan perubahan pada data yang lain. Sebagai contoh, jika pada awalnya dalam metadata hanya memiliki *actor*, *action*, dan *object* saja, dan ternyata pada akhirnya mungkin ada penambahan atribut lain seperti *direction*, *detail*, dan lain sebagainya tidak akan menyebabkan perubahan pada data sebelumnya yang tidak memiliki atribut tersebut. Selain itu, RDF juga dapat memberikan integritas data yang lebih baik dari pada basis data relasional. Sebagai contoh, ketika ada penambahan atribut pada contoh sebelumnya, basis data relasional akan menggagap data yang telah dimasukkan sebelumnya pada atribut tersebut menjadi *null*. Sedangkan pada RDF, RDF menggagap data tersebut memang tidak memiliki atribut tersebut dan tidak akan merubah data sebelumnya.

3. PERANCANGAN SISTEM

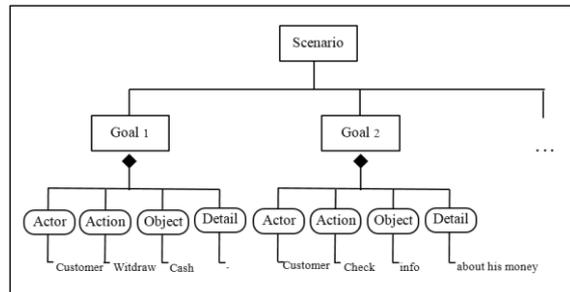
Mekanisme untuk menghasilkan diagram kasus penggunaan yang sesuai dengan sistem diperlukan dua proses utama. Proses pertama adalah dengan melakukan ekstraksi metadata dari skenario. Proses selanjutnya adalah dengan mencocokkan hasil ekstraksi metadata tersebut dengan kasus penggunaan yang ada di dalam repositori. Berdasarkan dua proses tersebut, sistem ini diharapkan dapat memberikan kasus penggunaan yang sesuai dengan skenario yang diberikan. Skenario yang diberikan haruslah berupa teks dan berbahasa Inggris. Penggambaran dari desain sistem akan digambarkan pada Gambar 1.



Gambar 1. Perancangan Sistem

3.1 Pemodelan Metadata Skenario

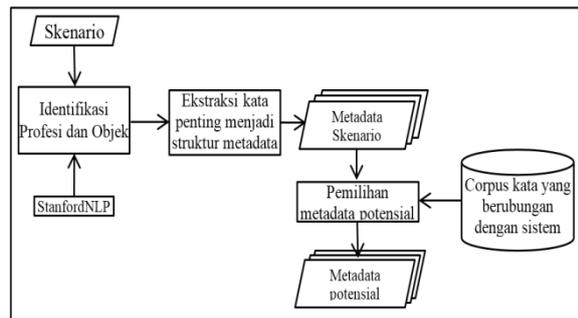
Pemodelan metadata skenario digunakan untuk mendeskripsikan hubungan antar data dalam skenario dengan menggunakan sebuah taksonomi skenario yang dapat dilihat pada Gambar 2.



Gambar 2. Model Metadata Skenario

3.2 Ekstraksi Metadata Skenario

Ekstraksi metadata *actor*, *action*, dan *object* akan diidentifikasi dalam setiap kalimat. Kalimat yang baik adalah kalimat yang minimal memiliki subjek dan predikat. Akan tetapi, dalam satu kalimat seringkali memiliki beberapa subjek, predikat, dan objek. Hal tersebut disebabkan karena adanya klausa kalimat. Dengan adanya klausa kalimat, diperlukan pemilihan kata-kata penting yang saling berhubungan antara subjek, predikat, dan objek. Dalam penentuan kasus penggunaan, subjek dan predikat saja tidak mencukupi, dibutuhkan objek sebagai keterangan dari predikatnya. Kemudian metadata dalam kalimat diambil dengan mekanisme sesuai pada Gambar 3.

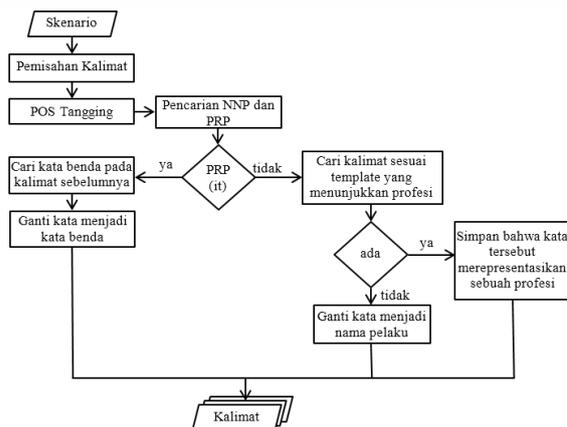


Gambar 3. Ekstraksi Metadata Skenario

3.2.1 Identifikasi Profesi dan Objek

Syarat dari sebuah skenario yang baik adalah harus spesifik, salah satunya harus spesifik dalam penamaan pelaku yang berhubungan dengan sistem, seperti dia harus memiliki nama, pekerjaan, alamat, dan sebagainya. Selain itu, dalam skenario banyak ditemukan kata ganti orang atau kata ganti benda yang berinteraksi dengan pelaku tersebut. Sedangkan

dalam diagram kasus penggunaan, aktor yang diperlukan adalah berbentuk profesi dari pelaku, sebagai contoh *customer*, *administrative officer*, dan lain-lain. Dari hal tersebut, diperlukan pengenalan semua kata ganti orang dan nama orang dengan profesi orang tersebut. Selain itu, kata ganti benda yang ditemukan seperti “*it*” juga harus diubah agar setelah ekstraksi klausa, kalimat tetap dapat diketahui pelaku berinteraksi dengan apa. Mekanisme identifikasi profesi aktor digambarkan pada Gambar 4.



Gambar 4. Identifikasi Profesi dan Objek

Pada Gambar 4, terdapat bagian yang menunjukkan pencarian profesi berdasarkan template. Template ditemukan dari kalimat-kalimat yang telah ditemukan sebelumnya yang mengidentifikasi profesi seseorang. Berikut adalah template yang dihasilkan dari kalimat-kalimat yang telah ditemukan :

1. [nama]NNP + is + DT + ... + [profesi]NN
2. [nama]NNP + is + ... + [profesi]NN
3. PRP + is + DT + ... + [profesi]NN
4. PRP + is + ... + [profesi]NN
5. [nama]NNP + , + DT + ... + [profesi]NN
6. [nama]NNP + as + DT + ... + [profesi]NN
7. DT + ... + [profesi]NN + named + [nama]NNP
8. [nama]NNP + who + worked + as + ... + [profesi]NN

Ket :

- ... hanya dapat diisi dengan kata sifat (JJ) atau dikosongkan
- Jika terdapat kata sifat sebelum kata benda, maka keduanya akan menjadi profesi.

3.2.2 Ekstraksi Kata Penting Menjadi Metadata

Kata-kata penting yang dicari akan dijadikan sebagai *actor*, *action*, dan *object* yang ada di dalam metadata. Sedangkan untuk detail berisi informasi sebagai pelengkap dari sebuah objek, detail dapat berupa keterangan dari objek tersebut. Proses ekstraksi langsung mengambil NN/NNP sebagai

actor, VB/VBZ/VBD/VBG/VBN/VBP sebagai *action*, dan NN/NNS yang menyertai *action* sebagai *object*. Sedangkan detail adalah informasi lain yang melengkapi objek dengan kata-kata penunjang seperti *from*, *about*, *of*, *in*, *to*, *with*, *for*, dan keterangan yang lain.

Dalam proses ekstraksi metadata, proses tersebut harus melewati beberapa tahap proses awal terlebih dahulu agar hanya kata-kata yang penting saja yang diambil. Proses awal tersebut adalah dengan melakukan seleksi :

- Jika dalam beberapa kata setelah kata kerja ditemukan kata kerja kembali, maka kata kerja yang pertama diabaikan, contoh : *he wants to check the cash.*
- Jika setelah kata kerja tidak memiliki kata benda lagi, maka kata kerja tersebut akan diabaikan.
- Jika setelah kata kerja langsung ada anak kalimat, maka kata kerja tersebut diabaikan, contoh : *She thinks that she might make a video.*
- Jika setelah kata kerja memiliki banyak objek, maka dia akan dibagi menjadi beberapa metadata, contoh : *doctor types the diseases and patient's condition.*
Metadata yang dihasilkan adalah : *doctor/actor types/action diseases/object, doctor/actor types/action patient's condition/object.*
- Jika dalam anak kalimat tidak memiliki aktor, maka aktor akan diambil dari induk kalimat, kecuali jika kondisi determiner tersebut adalah “*which*” atau “*that*” (khusus untuk *that*, jika dalam anak kalimat tersebut tidak memiliki aktor), maka yang diambil dari induk kalimat adalah objeknya.
- Jika menemukan kalimat pasif, maka objeknyalah yang akan jadi aktor, dan aktor menjadi objek.

3.2.3 Pemilihan Metadata Potensial

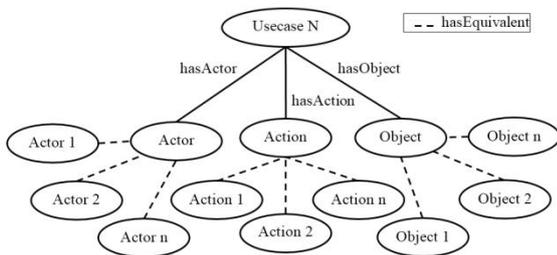
Metadata yang dihasilkan pada fase sebelumnya memiliki banyak hasil yang tidak relevan untuk berpotensi menjadi kasus penggunaan. Karena itu, metadata yang tidak relevan akan dihilangkan. Mekanisme pemilihan metadata yang relevan adalah dengan mencocokkan kata-kata yang ada di dalam metadata dengan korpus kata yang berhubungan dengan sistem. Jika dalam satu metadata tidak memiliki kata-kata yang ada di dalam korpus, maka metadata tersebut akan dihilangkan.

3.3 Pemodelan Repositori

Pembangunan ontologi sebagai repositori memiliki dua tahap. Tahap pertama adalah dengan memasukkan kasus penggunaan beserta deskripsinya. Deskripsi kasus penggunaan diperlukan agar dapat memberikan penjelasan lebih detail kepada setiap kasus penggunaan karena ada kemungkinan skenario yang akan menjadi masukan hanya memiliki isi dari

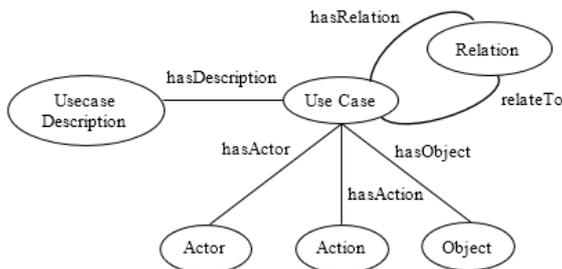
deskripsi kasus penggunaan. Kemudian mencari relasi kata pada individu yang ada di dalam ontologi menggunakan *Suggested Upper Merged Ontology* (SUMO). Kata yang dimasukkan ke dalam ontologi adalah kata yang memiliki relasi “*equivalent mapping*” terhadap kata yang dimasukkan. Kata-kata berelasi yang dicari menggunakan SUMO adalah aktor dari kasus penggunaan, kata kerja dan objek dari kasus penggunaan.

Pencarian relasi kata menggunakan SUMO dapat menghasilkan ontologi yang terdiri dari *goal* yang memiliki *actor*, *verb*, dan *object*. *Actor*, *verb*, dan *object* akan memiliki relasi kata yang sesuai dengan hasil pencarian relasi dan kata pada SUMO yang memiliki relasi “*equivalent mapping*”. Gambar 5 menjabarkan ontologi yang dihasilkan.



Gambar 5. Pemodelan Repositori

Ontologi dalam repositori tidak hanya sebatas seperti pada Gambar 5 saja. Sebuah kasus diagram juga memiliki relasi dengan kasus diagram yang lain seperti relasi *include* dan *extend*. Penambahan kasus penggunaan yang berelasi dimaksudkan agar ketika kasus penggunaan utama ditemukan, maka kasus penggunaan yang berelasi dengannya juga dapat diusulkan sebagai calon kasus penggunaan yang disarankan. Domain ontologi yang akan dibangun digambarkan pada Gambar 6.

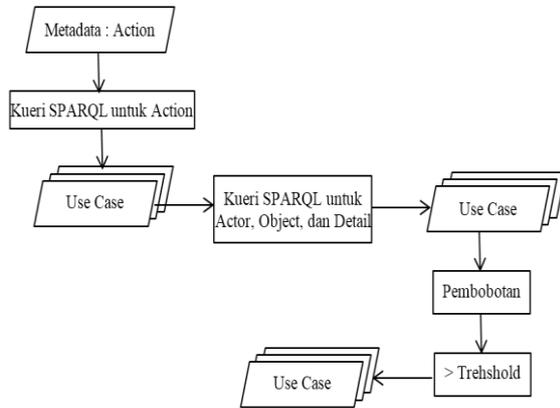


Gambar 6. Domain Ontologi

3.4 Pencarian dalam Repositori

Dalam melakukan pencarian dalam repositori digunakan kueri SPARQL yang dikhususkan dalam pencarian ontologi. Proses pencarian akan mengambil dari setiap metadata pada skenario yang telah diekstraksi sebelumnya. Sebagai contoh, *action* yang ditemukan pada metadata akan

dicari dalam ontologi, kemudian jika ditemukan akan dicari semua kelas yang berhubungan dengan kelas tersebut. Kelas-kelas yang berhubungan akan dicari kesamaannya dengan atribut lain pada metadata seperti *actor*, *object* dan *detail* yang dimiliki. Mekanisme pencarian dalam repositori akan dijelaskan pada Gambar 7.



Gambar 7. Pencarian dalam Repositori

Untuk menghitung kedekatan antara metadata skenario terhadap kasus penggunaan di dalam repositori akan dilakukan pembobotan pada setiap atribut (*actor*, *action*, *obj*, dan *detail*).

4. MEKANISME PENGUJIAN

Sistem generasi kasus penggunaan dari skenario bebas dikembangkan berdasarkan rancangan sistem yang dibuat dan akan diuji dengan melakukan perbandingan dengan seorang ahli. Pengujian dilakukan dengan melakukan perhitungan nilai *precision* (*P*) dan *recall* (*R*). Tabel 1 menjelaskan perbandingan antara *precision* dan *recall*.

Tabel 1. Precision dan Recall

		Identifikasi diagram kasus penggunaan secara manual	
		Relevan (<i>R</i>)	Tak Relevan (\bar{R})
Identifikasi diagram kasus penggunaan oleh sistem	Ditemukan (<i>D</i>)	DR	$D\bar{R}$
	Tak ditemukan (\bar{D})	$\bar{D}R$	$\bar{D}\bar{R}$

Hasil implementasi diharapkan memberikan nilai presisi dan recall yang baik dalam memberikan rekomendasi kasus diagram pada kasus tersebut.

5. KESIMPULAN

Makalah ini bertujuan untuk melakukan eksperimen awal apakah metode yang diajukan dapat untuk dilanjutkan pada fase implementasi. Percobaan awal dilakukan pada 10 skenario naratif yang memiliki kasus yang berbeda dan repositori yang berisi kasus penggunaan yang serupa dengan kasus pada skenario. Dari hasil penelitian awal yang dilakukan secara manual ini, metode tersebut dapat memberikan rekomendasi kasus penggunaan dari skenario yang dimasukkan berdasarkan kasus penggunaan pada repositori. Akan tetapi hasil yang direkomendasikan masih tergantung pada kelengkapan kasus penggunaan pada repositori. Kemudian ditemukan beberapa kendala ketika pemrosesannya. Kendala pertama adalah terlalu banyaknya variasi dari skenario yang dibuat sehingga tidak semua jenis skenario dapat dikenali dalam pengidentifikasian profesi dari aktor. Kendala berikutnya adalah besarnya ontologi yang dihasilkan melalui SUMO sehingga jika dilakukan proses komputasi, maka akan memakan waktu yang lama dalam pemrosesannya.

6. DAFTAR PUSTAKA

- [1] Regnell, B., Kimbler, K., & Weslen, A. (1995). Improving the use case driven approach to requirements engineering. *Second IEEE International Symposium on Requirement Engineering (RE'95)*. Hal. 1-8. York.
- [2] Siahaan, D. (2012). *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak*. Yogyakarta: ANDI.
- [3] Kim, J., Park, S., & Sugumaran, V. (2006). "Improving use case driven analysis using goal and scenario authoring: A linguistic-based approach". *Data & Knowledge Engineering* 58, 21-46.
- [4] Alspaugh, T. A., & Anton, A. I. (2008). "Scenario support for effective requirements". *Information and Software Technology* 50, 198-220.
- [5] Hariyanto, B. (2004). *Sistem Manajemen Basis Data*. Bandung: Informatika.
- [6] Jurafsky, Dan, 2013. *The Stanford Natural Language Processing Group*, <URL:http://nlp.stanford.edu>.
- [7] Bernaras, A., Laresgoiti, I., & Corraera, J. (1996). "Building and Reusing Ontologies for Electrical Network Applications". 12th European conference on Artificial Intelligence, 298-302.
- [8] Pease, A., Niles, I., & Li, J. (2002). "Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications". *The AAAI Workshop on Ontologies and the SW*. Canada.
- [9] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.
- [10] Niles, I., & Pease, A. (2003). "Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology". *The IEEE International Conference on IKE*, (hal. 412-416).
- [11] Borst, W. N. (1997). *"Construction of Engineering Ontologies for Knowledge Sharing and Reuse"*. Netherland: SIKS The Dutch Graduate School.
- [12] Munawar. (2005). *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- [13] Pramudiono. (2006). *Model Pencarian pada Mesin Pencari*. Bandung.