

PEMERIKSAAN KEMIRIPAN DIAGRAM KASUS PENGUNAAN TERHADAP SKENARIO

Reza Fauzan¹, Djoko Pramono²

¹ Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

² Fakultas Teknologi Informasi, Universitas Brawijaya

Email: fauzan.reza12@mhs.if.its.ac.id¹, djoko12@mhs.if.its.ac.id²

ABSTRAK

Pada saat ini diagram kasus penggunaan digunakan dalam pendefinisian kebutuhan sistem. Sebelum pembangunan diagram kasus penggunaan tersebut, analis membuat sebuah skenario agar konsumen lebih mengerti tentang sistem yang akan diberikan kepada mereka. Kemudian analis melakukan analisa kembali pada kebutuhan sistem untuk membuat diagram kasus penggunaan. Ketika analis melakukan analisa kebutuhan sistem pada waktu yang berbeda, analis dimungkinkan membuat analisa yang berbeda. Makalah ini bertujuan untuk mengecek kemiripan hasil analisa kebutuhan sistem ketika pembuatan skenario dan kasus penggunaan. Makalah ini mengajukan langkah kerja dari ekstraksi metadata skenario dan kasus penggunaan dan kemudian dicari kemiripan konteks antar keduanya menggunakan Wordnet. Luaran yang dihasilkan dari sistem yang dibangun adalah nilai kemiripan dari setiap kasus penggunaan terhadap ekstraksi metadata skenario. Dengan melakukan penelitian ini, sistem diharapkan dapat membantu analis sistem dalam pengecekan hasil analisa yang mereka lakukan.

Kata Kunci: Kasus Penggunaan, Skenario, Pemrosesan Bahasa Alami, Wordnet, Kemiripan.

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 dibangun sebuah diagram kasus penggunaan, seorang analis membuat skenario terlebih dahulu agar tidak terjadi kesalahpahaman yang terjadi 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, dalam skenario tertulis 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.

Sekarang ini, ada kemungkinan analisa kebutuhan sistem dilakukan oleh beberapa orang dalam satu proyek. Hal ini menyebabkan ada perbedaan pemahaman dalam melakukan analisa

kebutuhan sistem yang akan menyebabkan kerancuan dalam pengembangan perangkat lunak. Dari permasalahan tersebut, sistem ini dibangun agar dapat menyelesaikan perbedaan pemahaman berdasarkan konteks dari kasus penggunaan dengan skenario yang telah dibangun.

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 bebrapa parameter yang telah ditentukan. Berikut adalah parameter pengukuran kualitas dari skenario adalah [2]:

- Mudah diakses dan mudah dimengerti
- Penting, berharga, menarik, atraktif
- Kritis dan menantang
- Sering digunakan dan menghindari hal-hal di luar kebiasaan
- 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 [3]. 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* [4].

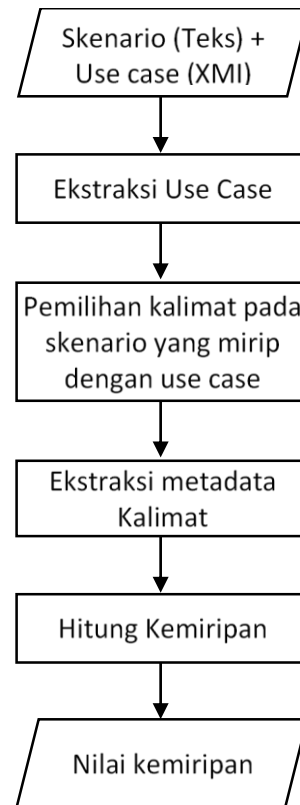
StanfordNLP dalam penelitian ini difokuskan untuk melakukan *tagging*. *Tagging* adalah penandaan kata dalam kalimat untuk menentukan labelnya dengan menggunakan *Part of Speech (POS)* dari model *Pen Tree Bank*.

2.4 Wordnet

Wordnet merupakan basis data leksikal elektronik bahasa Inggris yang tersedia secara online. Dalam format elektronik membuat wordnet lebih mudah digunakan dan dimanipulasi. Dalam wordnet kata kerja, kata benda, kata keterangan dan kata sifat dikelompokkan dalam suatu set sinonim yang disebut *synonym set (synsets)*. *Synsets* merupakan unit terkecil dalam wordnet. Dalam satu *synsets* terdapat makna kata dan sinonim dari kata tersebut. Dalam kamus tradisional kata disusun secara alfabetik sehingga kata dalam satu kelompok bisa memiliki makna yang tidak berhubungan satu dengan lainnya. Wordnet menyusun kata berdasar makna, sehingga kata yang berdekatan akan memiliki makna yang berhubungan. Satu *synsets* memiliki hubungan secara semantik dengan *synsets* lainnya [5].

3. PERANCANGAN SISTEM

Mekanisme untuk menghasilkan nilai luaran yang diinginkan diperlukan empat proses utama. Berdasarkan proses-proses tersebut, sistem ini diharapkan dapat memberikan nilai kemiripan berdasarkan konteks antara kasus penggunaan dan skenario. Skenario yang diberikan haruslah berupa teks dan berbahasa Inggris, sedangkan kasus



Gambar 1. Perancangan Sistem

penggunaan yang dimasukkan harus sesuai dengan format umum diagram kasus penggunaan. Penggambaran dari desain sistem digambarkan pada Gambar 1.

3.1 Ekstraksi Diagram Kasus Penggunaan

Diagram kasus penggunaan yang digunakan dalam penelitian ini dalam format XML Metadata Interchange (XMI). XMI merupakan standar yang digunakan oleh Object Management Group (OMG) untuk bertukar informasi pada level metadata. Arsitektur XMI memudahkan pertukaran informasi antar aplikasi dari platform yang berbeda. XMI digunakan sebagai media pertukaran antara perangkat pemodelan seperti UML ke perangkat pembangkit kode sumber. Format XMI sudah didukung banyak perangkat lunak editor UML. Beberapa perangkat pembuat diagram UML seperti ArgoUml, StarUml dan RationalRose sudah menyediakan fasilitas untuk menyimpan diagram UML dalam format XMI.

Diagram use case yang sudah dalam format XMI diparsing menggunakan *library* Java untuk operasi parsing XML yaitu *SAXParser*. Untuk menangani proses parsing, dibuat satu class *XmiHandler* yang merupakan turunan dari class *DefaultHandler* dengan melakukan *override* pada method *startElement*. Pada *SAXParser* proses parsing dilakukan secara

sekuensial. *Override* method *startElement* dan lakukan seleksi pada parameter *localName* untuk nilai-nilai *Actor*, *UseCase*, *Include*, *Extend* dan *Association*. Hasil parsing XMI ditampung dalam class *UseCase* yang merupakan model dalam java yang nantinya akan digunakan dalam proses perhitungan kemiripan terhadap hasil ekstraksi skenario.

XMI memuat kasus penggunaan, aktor dan hubungannya dalam struktur seperti ditunjukkan pada Kode Sumber 1. Setiap simpul yang mewakili kasus penggunaan, aktor dan hubungannya memiliki identitas yang diwakili atribut *xmi.id* dalam struktur XMI.

```
<XMI.content>
<UML:Model xmi.id = '...' name = 'Hotel Model' >
  <UML:Namespace.ownedElement>
    <UML:Actor xmi.id = '...' name='...' />
    <UML:UseCase xmi.id = '...' name= '...' />
    <UML:Association xmi.id = '...' >
  </UML:Association>
  <UML:Include xmi.id = '...' >
  </UML:Include>
  </UML:Namespace.ownedElement>
</UML:Model>
</XMI.content>
```

Kode Sumber 1

Dalam XMI kasus penggunaan, hubungan antara aktor dan kasus penggunaan digambarkan dalam suatu asosiasi. Struktur XMI dari asosiasi antara suatu kasus penggunaan dan aktor ditunjukkan dalam Kode Sumber 2.

```
<UML:Association xmi.id = '8000:00000...' >
<UML:Association.connection>
  <UML:AssociationEnd>
    <UML:AssociationEnd.participant>
      <UML:Actor xmi.idref = '...' />
    </UML:AssociationEnd.participant>
  </UML:AssociationEnd>
  <UML:AssociationEnd xmi.id = '...' >
    <UML:AssociationEnd.participant>
      <UML:UseCase xmi.idref = '...' />
    </UML:AssociationEnd.participant>
  </UML:AssociationEnd>
</UML:Association.connection>
</UML:Association>
```

Kode Sumber 2

Dalam struktur tersebut ditunjukkan bahwa suatu asosiasi terdiri dari dua simpul *associationEnd*. Satu simpul merujuk pada kasus penggunaan sedangkan

simpul satunya merujuk pada aktor. Setiap simpul menunjuk *xmi.id* dari kasus penggunaan dan aktor yang dituju.

Struktur XMI dari hubungan *include* antar kasus penggunaan ditunjukkan pada Kode Sumber 3. Pada struktur tersebut hubungan *include* memiliki dua simpul yaitu *Include.base* dan *Include.addition*. *Include.base* merujuk pada *xmi.id* dari kasus penggunaan sumber hubungan *include*. Sedangkan *Include.addition* merujuk pada kasus penggunaan yang menjadi tujuan hubungan *include*.

```
<UML:Include xmi.id = '...' >
  <UML:Include.addition>
    <UML:UseCase xmi.idref = '...' />
  </UML:Include.addition>
  <UML:Include.base>
    <UML:UseCase xmi.idref = '...' />
  </UML:Include.base>
</UML:Include>
```

Kode Sumber 3

```
<UML:Extend xmi.id = '...' >
  <UML:Extend.base>
    <UML:UseCase xmi.idref = '...' />
  </UML:Extend.base>
  <UML:Extend.extension>
    <UML:UseCase xmi.idref = '...' />
  </UML:Extend.extension>
  <UML:Extend.extensionPoint>
    <UML:ExtensionPoint xmi.idref = '...' />
  </UML:Extend.extensionPoint>
</UML:Extend>
```

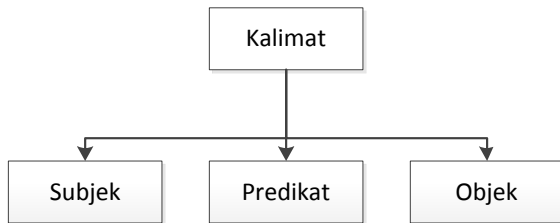
Kode Sumber 4

Struktur XMI dari hubungan *extend* antar kasus penggunaan ditunjukkan pada Kode Sumber 4. Pada struktur tersebut hubungan *extend* memiliki dua simpul yaitu *Extend.base* dan *Extend.extension*. *Extend.base* merujuk pada *xmi.id* dari kasus penggunaan sumber hubungan *extend*. Sedangkan *Extend.extension* merujuk pada *xmi.id* kasus penggunaan yang menjadi tujuan hubungan *extend*.

3.2 Pemilihan Kalimat yang Mirip dengan Kasus Penggunaan

Kalimat yang akan dipilih adalah kalimat yang ada pada skenario. Pemilihan kalimat diperlukan dalam mencari kalimat yang memiliki kemiripan dengan kasus penggunaan yang telah dimasukkan karena tidak semua kalimat yang akan diproses untuk dicari kesamaannya.

Mekanisme pemilihan kalimat dilakukan menggunakan *Stanford Natural Language*



Gambar 2. Taksonomi Metadata Kalimat

Processing (NLP). Pertama dilakukan proses tokenisasi pada skenario untuk memisahkan kalimat-kalimat dalam skenario. Proses ini menghasilkan kalimat-kalimat yang berdiri sendiri. Kemudian dilakukan pemberian label (*tagging*) pada setiap kata dalam kalimat. Setelah setiap kata diberikan label, sistem dapat menentukan kata kerja pada setiap kalimat dengan mengenali label seperti VB, VBP, dan sejenisnya. Kemudian kata kerja pada setiap kalimat diambil untuk dicari kemiripan dengan kata kerja pada kasus penggunaan. Ketika sistem telah menemukan kemiripan kata kerja pada salah satu kata kerja dalam skenario, maka kalimat yang di dalamnya terdapat kata kerja tersebut akan diambil dan kalimat lain diabaikan.

3.3 Ekstraksi Kalimat

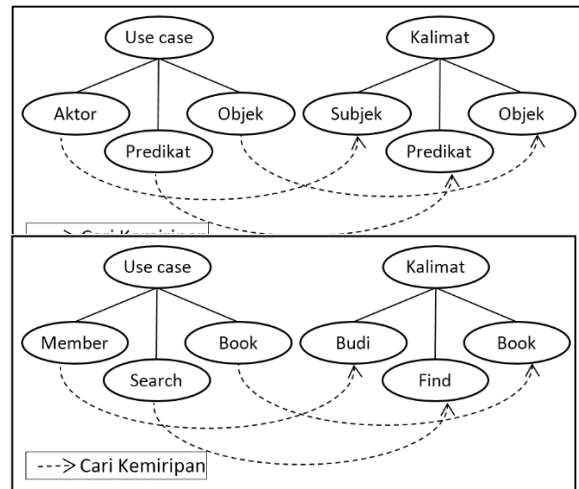
Kalimat yang telah dipilih dari proses sebelumnya akan diekstrak kata-kata penting yang diperlukan dalam proses perhitungan kemiripan antar kasus penggunaan dengan kalimat dalam skenario. Setiap kalimat memiliki struktur taksonomi metadata yang akan diperjelas pada Gambar 2.

Proses pengambilan subjek adalah dengan mengambil kata benda yang terletak sebelum kata kerja. Kata benda memiliki label “NN” jika menggunakan *StanfordNLP*. Subjek diambil agar dapat mengetahui siapa aktor yang berperan dalam kasus penggunaan yang mirip. Dan untuk proses pengambilan predikat adalah dengan langsung mengambil kata kerja yang mirip dengan kata kerja pada kasus penggunaan. Dan untuk pengambilan objek adalah mengambil kata benda (NN) pada kalimat yang menyertai kata kerja yang mirip dengan kata kerja pada kasus penggunaan. Metadata objek diambil agar dapat memperjelas maksud dari kata kerja dalam metadata predikat karena dalam satu kasus penggunaan memiliki kata kerja dan objeknya secara terkait.

3.4 Perhitungan Kemiripan

Untuk menghitung kemiripan kata berdasarkan konteks digunakan wordnet sebagai alat bantu. Wordnet dapat memberikan kedekatan antar dua sinset/kata. Dalam implementasinya, penulis

menggunakan *library Jaws* pada Java untuk menggunakan wordnet. Gambar 3 menjelaskan



Gambar 4. Contoh Pencarian Kemiripan

proses perhitungan kemiripan antar metadata.

Setelah dicari kemiripan antar metadata, akan dikalkulasikan semua hasil dari kemiripan dengan bobot masing-masing pada setiap submetadata. Pencarian nilai bobot didapatkan dari pemberian beberapa jenis bobot yang berbeda. Kemudian bobot terbaiklah yang akan digunakan dalam sistem. Bobot terbaik akan didapatkan dari hasil pengujian setiap bobot menggunakan kappa statistik [7]. Setelah kemiripan pada satu kasus penggunaan terhadap kalimat dalam skenario dilakukan kalkulasi, akan dilihat apakah kemiripan tersebut sudah melewati *threshold* atau tidak. Jika melewati *threshold* maka dapat dikatakan kasus penggunaan tersebut memiliki kesamaan terhadap skenario yang dibangun. Berikut adalah contoh perhitungan yang dilakukan:

Kasus penggunaan: member → search book

Kalimat: Budi find the book using the computer.

Dari Gambar 4 dapat dihasilkan :

Member → Budi = 0

Search → Find = 1

Book → Book = 1

Total =

$(0 \times \text{bobotSubjek}) + (1 \times \text{bobotPredikat}) + (1 \times \text{bobotObjek})$

Dari contoh yang diatas dapat dihitung bahwa kasus penggunaan tersebut mirip dengan kalimat yang ada dalam skenario karena luaran yang diberikan jika telah melebihi batas (*threshold*) yang telah ditentukan.

4. MEKANISME PENGUJIAN

Sistem generasi pengecekan kemiripan antara kasus penggunaan dan skenario ini dibangun berdasarkan rancangan sistem yang dibuat dan akan diuji dengan melakukan perbandingan dengan pakar. Hasil sistem yang dibandingkan dengan pakar akan

dibagi menjadi beberapa kelompok sesuai dengan bobot yang diberikan. Pembagian kelompok hasil sistem dilakukan agar dapat menemukan nilai bobot yang terbaik dan dapat menemukan hal apa yang terpenting dalam perbandingan kasus penggunaan terhadap skenario. Asumsi yang diberikan adalah jika ada salah satu kasus penggunaan yang tidak mirip terhadap skenario, maka diagram kasus penggunaan tersebut dianggap tidak mirip sehingga harus dianalisa ulang, baik dari sisi skenario maupun diagram kasus penggunaan.

Pengujian dilakukan menggunakan 20 pasangan dokumen skenario dan diagram kasus penggunaan. Kondisi diagram kasus penggunaan memiliki karakteristik yang berbeda-beda, dari yang memiliki jumlah kasus penggunaan yang kecil sampai besar. Metode pengujian yang digunakan adalah metode kappa statistik. Hasil dari sistem akan dibandingkan dengan dua orang pakar. Pakar mengatakan “mirip” sebanyak 16, dan mengatakan “tidak” sebanyak 4. Berikut adalah nilai bobot yang akan diuji untuk dibandingkan dengan pakar.

- Subjek (0,1), Predikat (0,5), Objek (0,4)
- Subjek (0,2), Predikat (0,4), Objek (0,4)
- Subjek (0,4), Predikat (0,3), Objek (0,3)
- Subjek (0,3), Predikat (0,4), Objek (0,3)
- Subjek (0,5), Predikat (0,4), Objek (0,1)
- Subjek (0,4), Predikat (0,4), Objek (0,2)
- Subjek (0,4), Predikat (0,1), Objek (0,5)
- Subjek (0,4), Predikat (0,2), Objek (0,4)

Tabel 1. Hasil perbandingan 1

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 13 | 3 |
| | Tidak | 0 | 4 |

$$\Pr(a) = \frac{13+4}{20} = 0,85$$

$$\Pr(e) = 0,52 + 0,07 = 0,59$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,85 - 0,59}{1 - 0,59} = 0,634146$$

Tabel 2. Hasil perbandingan 2

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 11 | 5 |
| | Tidak | 0 | 4 |

$$\Pr(a) = \frac{11+4}{20} = 0,85$$

$$\Pr(e) = 0,53$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,85 - 0,53}{1 - 0,53} = 0,468085$$

Tabel 3. Hasil perbandingan 3

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 1 | 15 |
| | Tidak | 0 | 4 |

$$\Pr(a) = \frac{1+4}{20} = 0,25$$

$$\Pr(e) = 0,23$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,25 - 0,23}{1 - 0,23} = 0,025974$$

Tabel 4. Hasil perbandingan 4

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 1 | 15 |
| | Tidak | 0 | 4 |

$$\Pr(a) = \frac{1+4}{20} = 0,25$$

$$\Pr(e) = 0,23$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,25 - 0,23}{1 - 0,23} = 0,025974$$

Tabel 5. Hasil perbandingan 5

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 2 | 14 |
| | Tidak | 0 | 4 |

$$\Pr(a) = \frac{2+4}{20} = 0,3$$

$$\Pr(e) = 0,26$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,3 - 0,26}{1 - 0,26} = 0,054054$$

Tabel 6. Hasil perbandingan 6

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 2 | 14 |
| | Tidak | 0 | 4 |

$$\text{Pr}(a) = \frac{2+4}{20} = 0,3$$

$$\text{Pr}(e) = 0,26$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,3 - 0,26}{1 - 0,26} = 0,054054$$

Tabel 7. Hasil perbandingan 7

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 1 | 15 |
| | Tidak | 0 | 4 |

$$\text{Pr}(a) = \frac{1+4}{20} = 0,25$$

$$\text{Pr}(e) = 0,23$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,25 - 0,23}{1 - 0,23} = 0,025974$$

Tabel 8. Hasil perbandingan 8

| | | Sistem | |
|-------|-------|--------|-------|
| | | Mirip | Tidak |
| Pakar | Mirip | 1 | 15 |
| | Tidak | 0 | 4 |

$$\text{Pr}(a) = \frac{1+4}{20} = 0,25$$

$$\text{Pr}(e) = 0,23$$

Nilai *cohen's kappa* adalah sebagai berikut.

$$k = \frac{0,25 - 0,23}{1 - 0,23} = 0,025974$$

5. ANALISA HASIL UJI COBA

Uji coba dilakukan dengan menggunakan tiga klasifikasi pembobotan. Pembobotan tersebut adalah sebagai berikut.

- Menyetarakan semua parameter.
- Menitikberatkan predikat dan objek.
- Menitikberatkan subjek dan predikat.
- Menitikberatkan subjek dan objek.

Dari hasil uji coba tersebut, penulis melakukan analisa terhadap tiga parameter utama, yaitu subjek, predikat, dan objek yang dibandingkan. Dalam perbandingan subjek dari skenario terhadap aktor yang ada dalam kasus penggunaan, terdapat banyak perbedaan antara keduanya. Sebagai contoh di salah satu skenario memberikan contoh bahwa pelaku utama dalam skenario tersebut adalah seorang mahasiswa. Sedangkan dalam diagram kasus penggunaan adalah seorang pelanggan. Sehingga jika sistem memberikan bobot yang besar terhadap subjek, maka hasil kemiripannya kecil sedangkan profesi dalam skenario sangat beragam.

Dalam perbandingan predikat pada skenario dengan predikat pada kasus penggunaan, penulis mendapatkan banyak kemiripan antar keduanya. Hal ini disebabkan predikat menunjukkan kebutuhan dari sistem yang dibangun. Sehingga jika sistem membuat bobot yang lebih besar terhadap predikat, sistem akan memberikan hasil kemiripan yang relevan atas tujuan dari sistem.

Parameter terakhir yang dibandingkan adalah objek dari skenario dan kasus penggunaan. Dari hasil perhitungan kappa dengan mengkombinasikan bobot besar terhadap predikat dan objek, sistem dapat memberikan hasil yang baik. Hal ini disebabkan karena objek memberikan pelengkap terhadap predikat sehingga memberikan kebutuhan utama dari sistem secara utuh.

6. KESIMPULAN

Penelitian ini dilakukan untuk menentukan kemiripan antara diagram kasus penggunaan terhadap skenario yang telah dibangun sebelumnya. Proses yang dilakukan adalah dari ekstraksi kasus penggunaan, kemudian ekstraksi skenario. Kemudian hasil ekstraksi antar keduanya akan dicari kemiripannya menggunakan bantuan *Wordnet*.

Hasil pengujian menggunakan kappa statistik memberikan nilai terbaik *cohen's kappa* sebesar 0,63 pada pembobotan pertama. Nilai tersebut membuktikan bahwa sistem yang telah dibangun dapat digunakan untuk mengetahui kemiripan antar diagram kasus penggunaan terhadap skenario. Dari nilai pembobotan tersebut, didapatkan bahwa bobot untuk subjek sangat kecil dibandingkan predikat dan objek.

Akan tetapi, sistem ini memiliki kekurangan yang ditemukan selama pengujian. Beberapa kasus penggunaan tidak dapat ditemukan kemiripannya dalam skenario karena tidak ada penjelasan secara tekstual di skenario. Skenario hanya memberikan pemahaman yang mengharuskan sistem menyimpulkan atau mengambil makna dari satu kalimat atau lebih. Pada penelitian selanjutnya diharapkan sistem dapat memahami makna dari satu kalimat atau lebih dengan mengambil kesimpulan di dalamnya.

7. 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] Jurafsky, Dan, 2013. The Stanford Natural Language Processing Group, <URL:<http://nlp.stanford.edu>>.
- [4] Fellbaum, C. (1998). "WordNet: An Electronic Lexical Database." Cambridge: MIT Press.
- [5] Lesmana, Riyan Adi. (2012). "Building Web Based Indonesian WordNet." Departemen Ilmu Komputer, Fakultas MIPA, Institut Pertanian Bogor.
- [6] Alspaugh, T. A., & Anton, A. I. (2008). "Scenario support for effective requirements". Information and Software Technology 50, 198-220.
- [7] Carletta, Jean. (1996) "Assessing agreement on classification tasks: The kappa statistic." Computational Linguistics, 22(2), pp. 249–254.