

IDENTIFIKASI TITIK PERCABANGAN PADA DESKRIPSI TEKSTUAL *USE CASE* MENGGUNAKAN PENGENALAN ENTITAS BERNAMA DENGAN METODE ASSOCIATION RULES MINING

Mirnasari Dewi, Indra Budi dan Petrus Mursanto
Fakultas Ilmu Komputer Universitas Indonesia
Kampus UI Depok 16424 Indonesia
Telp/Fax: 021-/863419/021-7863415
Email: mirma101@mhs.cs.ui.ac.id, indrasanto@cs.ui.ac.id

Abstrak

Pengenalan entitas bernama dapat digunakan dalam software engineering yaitu pada deskripsi tekstual *use case* untuk mengidentifikasi titik-titik percabangan yang terjadi pada *use case*. Titik-titik percabangan ini dapat digunakan untuk membuat *use case scenario* secara otomatis. Pendekatan yang digunakan adalah machine learning dengan metode association rule mining. Berdasarkan hasil eksperimen diperoleh F-measure sebesar 96,34%.

Kata Kunci: pengenalan entitas bernama, *use case scenario*, machine learning, association rules mining.

1. PENDAHULUAN

Pengujian (*testing*) merupakan tahapan yang cukup penting dalam teknologi perangkat lunak (*software engineering*). Tahapan *testing* diperlukan dalam semua pengembangan perangkat lunak, dalam pengembangannya perangkat lunak tidak dapat dikatakan bekerja dengan baik bila tahap *testing* belum dilakukan. Penundaan *testing* sampai semua pengembangan selesai dilakukan mempunyai resiko tinggi [7].

Testing harus dilakukan secara sistematis agar tidak ada kasus *testing* yang terlewat, sehingga didapatkan hasil yang baik. Untuk menjaga agar semua tahapan *testing* diujikan, perlu disusun *test case*. *Test case* adalah kumpulan input yang akan diuji, kondisi yang harus dieksekusi dan hasil yang diharapkan, yang dikembangkan untuk tujuan tertentu, misalnya untuk menjalankan jalan (*path*) khusus pada program atau memeriksa pemenuhan kebutuhan tertentu [7]. *Test case* yang dimaksud adalah *test case* untuk pengujian kebutuhan fungsionalitas sistem.

Saat ini *test case* masih disusun secara manual, penyusunan *test case* dilakukan dengan melihat apakah fungsi *use case* yang diinginkan telah berhasil diimplementasikan dengan benar. Dalam panduan yang diterbitkan Rational Software, disebutkan bahwa *test case* dapat dihasilkan dari *use case* [7]. Untuk

mendapatkan gambaran keseluruhan tentang cara kerja *use case*, dibutuhkan penjelasan dari *use case* tersebut. *Use case* digunakan untuk komunikasi *end user*. Penjelasan dari *use case* biasanya ditulis dalam bentuk tekstual. Meski dapat juga ditulis dengan menggunakan *flow chart*, *sequence chart*, atau bahasa pemrograman, namun *user* lebih memahami deskripsi tekstual, oleh karena itu deskripsi tekstual sederhana adalah pilihan terbaik [3].

Langkah awal yang dibutuhkan untuk menyusun *test case* dari *use case* adalah mendapatkan semua *use case scenario*-nya. Setelah *use case scenario*-nya didapatkan maka langkah selanjutnya adalah menyusun *test case*. *Use case scenario* dapat dibuat secara manual atau otomatis. Untuk membuat *use case scenario* secara otomatis, dibutuhkan informasi yang menjadi titik-titik percabangan dalam *flow*. Tujuan dari penelitian ini adalah menunjukkan bahwa pengenalan entitas bernama dapat digunakan dalam domain *software engineering* yaitu pada deskripsi tekstual *use case*. Pengenalan entitas bernama ini digunakan untuk mendapatkan titik-titik percabangan sebagai langkah awal membuat *use case scenario*.

Selanjutnya paper ini disusun dengan sistematika sebagai berikut. Bagian kedua menjelaskan tentang *use case scenario*. Bagian ketiga menjelaskan pengenalan entitas bernama, kelas entitas, dan *association rule*.

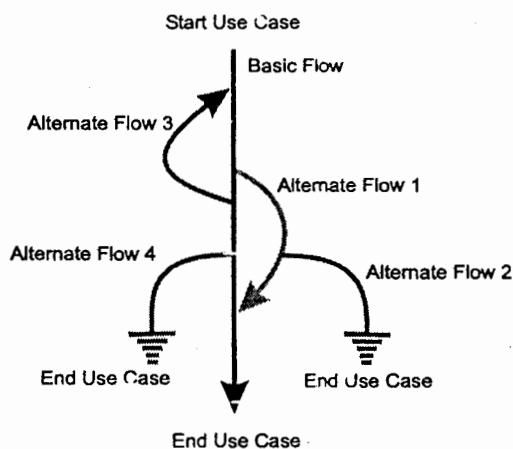
Bagian keempat menjelaskan arsitektur sistem. Bagian kelima menjelaskan eksperimen yang dilakukan serta kesimpulan dijelaskan pada bagian akhir paper ini.

2. USE CASE SCENARIO

Jim Heumann mendefinisikan bahwa “*Use case scenario is an instance of a use case, or a complete path through the use case*” [7]. Dari sebuah *use case* dapat dibuat beberapa *use case scenario*. Dalam deskripsi tekstual *use case* dijelaskan berbagai keterangan, seperti deskripsi singkat tentang *use case* (*brief description*), kondisi awal (*precondition*) yang harus dimiliki, kondisi akhir (*postcondition*) yang akan dicapai dan alur dari kejadian (*flow of events*).

Untuk menghasilkan *use case scenario*, bagian yang digunakan adalah *flow of events*. *Flow of events* terdiri dari dua bagian, yaitu *basic flow* dan *alternate flow*. *Basic flow* menggambarkan alur yang seharusnya terjadi bila *use case* berjalan dengan normal, sedangkan *alternate flow* menggambarkan percabangan yang terjadi dari alur normal. Sebuah *use case* memiliki sebuah *basic flow* dan dapat tidak memiliki *alternate flow*.

Gambar 1 merupakan gambar *flow* pada sebuah *use case*, yang memiliki percabangan pada *basic flow*.



Gambar 1. Basic and Alternate Flows of Events from Use Case

Tabel 1 merupakan hasil *use case scenario* yang didapat dari Gambar 1.

Dari 8 *use case scenario* yang terdapat dalam Tabel 1, tiap *use case scenario* melalui *basic flow*, setelah itu dapat bercabang ke

alternate flow sesuai dengan titik percabangannya.

Tabel 1. Hasil Use Case Scenario

Scenario 1	Basic Flow			
Scenario 2	Basic Flow	Alternate Flow 1		
Scenario 3	Basic Flow	Alternate Flow 1	Alternate Flow 2	
Scenario 4	Basic Flow	Alternate Flow 3		
Scenario 5	Basic Flow	Alternate Flow 3	Alternate Flow 1	
Scenario 6	Basic Flow	Alternate Flow 3	Alternate Flow 1	Alternate Flow 2
Scenario 7	Basic Flow	Alternate Flow 4		
Scenario 8	Basic Flow	Alternate Flow 3	Alternate Flow 4	

3. PENGENALAN ENTITAS BERNAMA

Named entity recognition atau pengenalan entitas bernama merupakan tugas dasar dalam sistem ekstraksi informasi. Tugas pengenalan entitas bernama biasanya meliputi pengenalan entitas nama (nama orang, nama organisasi, dan nama lokasi), pengenalan ekspresi waktu (tanggal, jam, dan durasi), dan pengenalan ekspresi angka (uang, persentase, ukuran, dan kardinal) [5]. Kelas entitas umum yang biasa digunakan dalam pengenalan entitas nama adalah PERSON, ORGANIZATION, LOCATION, pengenalan ekspresi waktu adalah DATE, TIME, DURATION, dan pengenalan ekspresi angka adalah MONEY, PERCENT, MEASURE, CARDINAL [4].

3.1. Kelas Entitas

Dalam penelitian ini, pengenalan entitas bernama digunakan untuk menemukan alur yang terdapat pada *alternate flow* dalam *use case*. Karena tujuan dan domain dokumen masukan berbeda, maka dibutuhkan kelas entitas baru yang sesuai dengan domain. Setelah mempelajari berbagai dokumen yang berisikan deskripsi tekstual dari *use case*, kami menentukan ada empat kelas entitas yang dapat mewakili semua alur yang terjadi pada percabangan dari *basic flow* dan *alternate flow*.

Kelas-kelas entitas yang digunakan dalam penelitian ini adalah sebagai berikut:

1. START

Kelas START menyimpan informasi bahwa objek Flow-nya merupakan awal alur dimulai.

2. **NEXT**
 Kelas NEXT menyimpan informasi objek Flow-nya merupakan tujuan alur.
3. **ALL**
 Kelas ALL menyimpan informasi bahwa objek Flow-nya merupakan tujuan alur dari semua objek Flow.
4. **END**
 Kelas END menyimpan informasi bahwa objek Flow-nya merupakan akhir dari alur.

Berikut merupakan contoh kalimat yang terdapat dalam *use case* beserta kelas entitasnya.

"In Alternate Flow 2 - Check Validation, if the user identification and password combination is not valid, the system responds by notifying the user that the combination was invalid. User can choose login again and rejoins Basic Flow Step 1. Or user can choose cancel and use case terminates."

Istilah "Alternate Flow" memiliki kelas entitas START, "Basic Flow Step" memiliki kelas entitas NEXT, dan "use case terminates" memiliki kelas entitas END.

3.2. Association Rule

Terdapat dua pendekatan besar dalam pengembangan sistem pengenalan entitas bernama, yaitu *rules-based approach* dan *machine learning approach*. Dalam pendekatan *rules-based*, pengenalan entitas bernama dilakukan dengan membuat *rules* (aturan-aturan) secara manual. Sedangkan dalam pendekatan *machine learning* dilakukan pembelajaran pada sistem sehingga sistem memiliki pengetahuan untuk melakukan klasifikasi atau pengenalan terhadap entitas bernama.

Association rule merupakan salah satu metode dalam pendekatan *machine learning*. *Association rule* dipilih sebagai metode dalam penelitian ini, karena melihat definisi *association rule* dapat digunakan untuk pencarian pola dalam menentukan kelas entitas pada pengenalan entitas bernama [2].

Menurut [1], *Association rule* adalah relasi dalam bentuk $X \Rightarrow Y$, dimana X dan Y adalah kumpulan *item* dari kumpulan data yang menjadi fokus perhatian dan $X \cap Y = \emptyset$. Setiap *association rule* dinyatakan dengan sebuah nilai *support* dan nilai *confidence*.

$support = \frac{X \cup Y}{N}$	$confidence = \frac{X \cup Y}{ X }$
--------------------------------	-------------------------------------

Dimana N adalah jumlah total rekord, |A| menyatakan jumlah rekord yang mengandung seluruh *item* pada himpunan A. Nilai *support* adalah rasio dari jumlah *item* pada X dan Y terhadap jumlah total *item* pada *dataset*. Nilai *confidence* adalah rasio jumlah *item* pada X dan Y terhadap jumlah *item* pada X.

Dalam pengenalan entitas bernama, *association rule* dapat digunakan untuk menentukan kelas entitas, dari sejumlah pola sintak dan fitur dari istilah (*term*). Pola tersebut diperoleh melalui tahap pelatihan (*training*) dari sejumlah koleksi dokumen yang dilatih (*training set of documents*). Dalam penelitian ini digunakan pembelajaran terarah (*supervised training*), dengan menggunakan dokumen yang telah diberi label kelas entitas. Sesuai dengan definisi *association rule* dimana didefinisikan sebagai $X \Rightarrow Y$, himpunan X dan Y dapat dideskripsikan dengan istilah, urutan istilah, fitur dan kelas entitas, dan Y merupakan kelas entitas yang akan diprediksi.

Dalam penelitian ini, himpunan X dideskripsikan dengan istilah (*term*) dan Y merupakan kelas entitas yang akan diprediksi. *Rule* yang digunakan dinyatakan dalam:

$$\langle t_1, t_2 \rangle \Rightarrow nc_2, (support, confidence)$$

Dimana t_1 dan t_2 adalah istilah (*term*) yang berpasangan, t_1 muncul lebih dahulu dari t_2 , dan nc_2 adalah kelas entitas dari istilah t_2 . nc_2 dapat mempunyai nilai START, NEXT, END atau ALL.

Misalkan pada dokumen *training* terdapat kalimat:

"In <FLOW TYPE="START">Alternate Flow</FLOW> 2 - Check Validation, if the user identification and password combination is not valid, the system responds by notifying the user that the combination was invalid."

Berdasarkan kalimat tersebut sistem dapat membuat *rule*, yaitu:

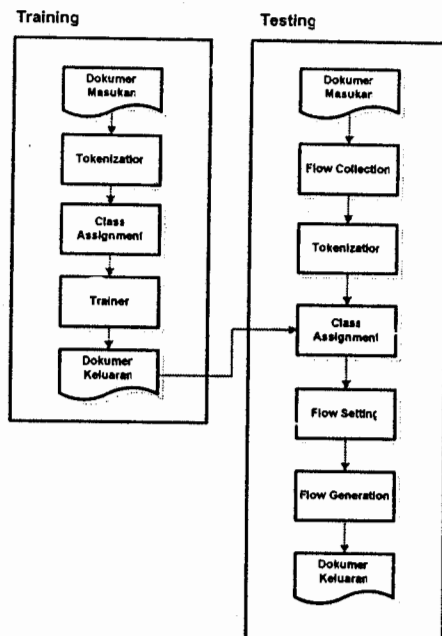
$\langle \text{In, 'Alternate Flow'} \rangle \Rightarrow \text{START}$
 dengan nilai *support* dan *confidence* tergantung kepada jumlah kemunculan ekspresi 'In Alternate Flow' dan jumlah kemunculan istilah 'Alternate Flow' yang

dilabelkan dalam kelas entitas tersebut (START) pada dokumen *training*.

4. ARSITEKTUR SISTEM

Secara garis besar, proses pada sistem dapat dibedakan menjadi dua. Proses pertama adalah menentukan kelas entitas dengan pengenalan entitas bernama, sedangkan proses kedua adalah pembuatan *use case scenario*.

Alur proses sistem dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur Sistem Pengenalan Entitas Bernama

Pada tahapan *training*, dokumen masukan yang telah diberi label kelas entitas akan diuraikan menjadi token-token oleh modul Tokenization. Setelah itu modul Class Assignment akan menyimpan kelas entitas sebagai atribut pada token yang diberi label kelas entitas tersebut. Modul Trainer akan mengumpulkan semua *rule* dan akan menghitung *support* dan *confidence* sesuai dengan metode *association rule*. Pada akhir tahapan *training* akan dihasilkan dokumen keluaran yang berisi token dan nilai *support* dan *confidence*. Dokumen keluaran dari tahapan *training* ini akan dipergunakan dalam tahapan *testing*.

Pada tahapan *testing*, dokumen masukan berupa deskripsi tekstual *use case*. Modul Flow Collection akan mengumpulkan informasi *flow* yang terdapat dalam dokumen masukan. Deskripsi dalam dokumen masukan

kemudian akan diuraikan menjadi token-token oleh modul Tokenization. Setelah itu modul Class Assignment akan memberikan kelas entitas yang bersesuaian dengan dokumen hasil *training*. Modul Flow Setting akan menyimpan informasi titik-titik percabangan. Terakhir modul Flow Generation akan menghasilkan *use case scenario*.

5. EKSPERIMEN

Hasil eksperimen yang akan dijelaskan dalam paper ini hanya mencakup pengenalan entitas bernama untuk mendapatkan titik-titik percabangan dalam *flow*.

5.1. Karakteristik Dokumen

Dokumen yang digunakan dalam uji coba adalah dokumen *use case* yang terdiri dari 50 dokumen. Dokumen ini ditulis dalam format XML (lihat Gambar 3) dengan mengikuti *template use case specification* dari Rational Unified Process (RUP). Penulisan deskripsinya juga mengikuti contoh yang ada dalam contoh *use case specification* yang diberikan oleh Rational Unified Process [8].

Dokumen yang digunakan untuk *training* berjumlah 30 dokumen yang mengandung 75 kelas START, 14 kelas NEXT, 24 kelas END dan 1 kelas ALL. Dokumen yang digunakan untuk *testing* berjumlah 20 dokumen yang mengandung 57 kelas START, 12 kelas NEXT, 13 kelas END dan 3 kelas ALL. Dokumen tersebut sebagian besar diambil dari RUP (Rational Unified Process) dan *website* Victoria University of Wellington [9] dalam format XML.

5.2. Evaluasi Kinerja

Dalam proses pengenalan entitas bernama, kinerja sistem diukur berdasarkan nilai tiga parameter yaitu *recall*, *precision*, dan *F-measure*. Parameter ini diperkenalkan oleh MUC (Message Understanding Conference) dalam [6]. *Recall* menyatakan jumlah pengenalan entitas bernama bernilai benar yang dilakukan sistem dibagi dengan jumlah entitas bernama yang seharusnya dapat dikenali sistem. *Precision* dihitung dari jumlah pengenalan bernilai benar oleh sistem dibagi dengan jumlah keseluruhan pengenalan yang dilakukan oleh sistem. *F-measure* merupakan nilai yang mewakili keseluruhan kinerja sistem dan merupakan penggabungan nilai *recall* dan

precision dalam sebuah nilai. Nilai *recall*, *precision*, dan *F-measure* dinyatakan dalam persen. Semakin tinggi prosentase ketiga nilai tersebut menunjukkan semakin baiknya kinerja sistem pengenalan entitas bernama.

```
<Use-Case_Specification>
  <Use-Case_Name>
    <Brief_Description>
    </Brief_Description>
  </Use-Case_Name>
  <Flow_of_Events>
    <Basic_Flow>
      <BF ID="" TITLE="">
      </BF>
      <BF ID="" TITLE="">
      </BF>
    </Basic_Flow>
    <Alternative_Flows>
      <AF ID="" TITLE="">
      </AF>
      <AF ID="" TITLE="">
      </AF>
    </Alternative_Flows>
  </Flow_of_Events>
  <Special_Requirements>
    <First_Special_Requirement>
    </First_Special_Requirement>
  </Special_Requirements>
  <Preconditions>
    <Precondition_One>
    </Precondition_One>
  </Preconditions>
  <Postconditions>
    <Postcondition_One>
    </Postcondition_One>
  </Postconditions>
  <Extension_Points>
    <Name_of_Extension_Point>
    </Name_of_Extension_Point>
  </Extension_Points>
</Use-Case_Specification>
```

Gambar 3. Format XML Dokumen Masukan

5.3. Hasil Eksperimen dan Analisis

Dari 20 dokumen *testing* yang diujikan, kami mendapatkan nilai *recall* sebesar 92,941%, *precision* sebesar 100%, dan *F-measure* sebesar 96,341%. Tabel 2 merupakan rincian dari *recall* dan *precision* tiap kelas entitas.

Tabel 2. Hasil Pengenalan Entitas Bernama

Kelas Entitas	REC	PRE
START	100 %	100 %
NEXT	83,33 %	100 %
END	92,30 %	100 %
ALL	0 %	0 %
TOTAL		
Recall	Precision	F-Measure
92,94 %	100 %	96,34 %

Berdasarkan hasil pada Tabel 2, sistem menghasilkan nilai *precision* 100%, kelas entitas yang ada dalam dokumen *testing* dapat dikenali dengan benar. Namun nilai *recall* 92,94%, hal ini terjadi karena sistem tidak berhasil mengenali beberapa entitas, entitas tersebut muncul dalam dokumen kunci namun tidak terdapat dalam dokumen *response*. Kesalahan seperti ini dapat diperkecil dengan cara melakukan *training* dengan menambahkan dokumen *training* sehingga menghasilkan *rule* yang lebih lengkap.

6. KESIMPULAN dan TUGAS SELANJUTNYA

Pengenalan entitas bernama dapat digunakan dalam domain *software engineering*, yaitu pada deskripsi tekstual *use case* untuk mengidentifikasi titik-titik percabangan yang terdapat dalam *flow of events*. Dalam penelitian ini, diperkenalkan kelas entitas yang sesuai dengan domain dokumen masukan, yaitu START, NEXT, END dan ALL. Kelas-kelas entitas tersebut dapat mewakili percabangan yang terjadi dalam *use case*. Berdasarkan hasil eksperimen diperoleh bahwa sistem dapat mengenali titik-titik percabangan dengan *F-measure* sebesar 96,34%

Setelah mendapatkan titik-titik percabangan, proyek selanjutnya dari penelitian ini adalah bagaimana menentukan *use case scenario* berdasarkan titik-titik percabangan pada *use case*.

DAFTAR PUSTAKA

- [1] Agrawal, Rakesh., Imielinski, Tomasz., Swami, Arun. *Mining Association Rules between Sets of Items in Large Databases*. Washington, D.C: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993.
- [2] Budi, Indra dan S. Bressan. *Association Rules Mining for Name Entity Recognition*. Roma: Proceeding of 2003 WISE Conference. 2003.
- [3] Cockburn, Alistair. *Writing Effective Use Cases*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
- [4] Chincor, Nancy. *MUC-7 Information Extraction Task Definition Version 5.1*

- [online]. 1998. Dari: http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ie_task.html. Internet; diakses: 2 April 2005.
- [5] Chincor, Nancy. Brown, Erica. Ferro, Lisa. Robinson, Patty. *Named Entity Task Definition*. Version 1.4. The MITRE Corporation and SAIC. 1999.
- [6] Douthat, A. *The Message Understanding Conference Scoring Software User's Manual*. Proceedings of the 7th Message Understanding Conference (MUC-7). 1998.
- [7] Heumann, Jim. *The Rational Edge: Generating Test Cases From Use Cases* [online]. 2001. Dari: <http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf>; Internet; diakses: 10 September 2004.
- [8] Rational Unified Process. *Guidelines: Test Case*. Rational Software Corporation. 2003.
- [9] University of Wellington. *Use Case Analysis With Narrative Semiotics* [online]. 2004. Dari: <http://www.vuw.ac.nz/lals/research/usecase/usecases.aspx>; Internet; diakses: 21 September 2004.