

Kakas Bantu Sistem Rekomendasi Kebutuhan Perangkat Lunak

Hendy Dwi Harfianto¹, Daniel Oranova Siahaan², dan Chastine Fatichah³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: ¹hendy12@mhs.if.its.ac.id, ²daniel@if.its.ac.id, ³chastine@if.its.ac.id

Abstrak—Rekayasa kebutuhan merupakan tahap awal sebelum suatu perangkat lunak dibangun yang terbagi menjadi lima tahap penting, yaitu elisitasi kebutuhan, analisis kebutuhan, spesifikasi kebutuhan, verifikasi kebutuhan, dan manajemen kebutuhan. Pada tahap elisitasi kebutuhan, berbagai kebutuhan dari para pemangku kepentingan dikumpulkan untuk ditangani menggunakan solusi berbasis komputer. Tahap elisitasi termasuk tahap yang sulit dalam rekayasa kebutuhan perangkat lunak yang disebabkan oleh tiga masalah, yaitu masalah cakupan, masalah pemahaman, dan masalah perubahan. Untuk memudahkan perekayasa kebutuhan dalam mengelaborasi kebutuhan suatu perangkat lunak, sebuah sistem rekomendasi kebutuhan perangkat lunak mampu memberikan solusi atas permasalahan-permasalahan tersebut. Sistem rekomendasi diharuskan untuk bisa merekomendasikan kebutuhan-kebutuhan perangkat lunak yang sesuai dengan ruang lingkup perangkat lunak yang akan dibangun. Algoritma FP-Growth dapat digunakan sebagai metode dalam proses perekomendasi dengan mengidentifikasi asosiasi antar kebutuhan perangkat lunak. Penemuan asosiasi antar kebutuhan ini dapat membantu perekayasa kebutuhan dalam mempertimbangkan kebutuhan lain yang diinginkan. Untuk mendukung bekerjanya algoritma FP-Growth, diperlukan dataset untuk membentuk kemungkinan kandidat kombinasi antar kebutuhan perangkat lunak yang akan diproses dengan beberapa pustaka pemrosesan teks secara semantik.

Kata Kunci—*Elisitasi Kebutuhan Perangkat Lunak, Sistem Rekomendasi, Association Rule Mining dengan Algoritma FP-Growth, Stanford CoreNLP, WordNet.*

I. PENDAHULUAN

Sebelum suatu perangkat lunak dibangun, langkah awal yang harus dilakukan yaitu rekayasa kebutuhan perangkat lunak. Rekayasa kebutuhan terbagi menjadi lima tahap penting, yaitu elisitasi kebutuhan, analisis kebutuhan, spesifikasi kebutuhan, verifikasi kebutuhan, dan manajemen kebutuhan. Pada tahap elisitasi kebutuhan, berbagai kebutuhan dari para pemangku kepentingan dikumpulkan untuk ditangani menggunakan solusi berbasis komputer. Tahap elisitasi termasuk tahap yang sulit dalam rekayasa kebutuhan perangkat lunak yang disebabkan oleh tiga masalah, yaitu masalah cakupan, masalah pemahaman, dan masalah perubahan [1]. Ketiga masalah tersebut muncul karena adanya isu-isu terkait elisitasi kebutuhan seperti pemangku kepentingan yang seringkali tidak mengetahui apa yang diinginkan dan mengungkapkan keinginannya dengan kalimat umum dan istilah-istilah pada bidang pekerjaannya, perbedaan permintaan dari pemangku kepentingan yang beragam, dan

faktor politik serta lingkungan bisnis ekonomi yang bersifat dinamis [1].

Berbagai macam teknik elisitasi kebutuhan yang pada umumnya banyak dipakai oleh perekayasa kebutuhan seperti wawancara, kuesioner, observasi, analisis prosedur, pengamatan dokumen, etnografi, *sampling*, *brainstorming*, *prototyping*, *Joint Application Design (JAD)*, *goal-based methods*, dan *scenario-based methods* [1]. Namun, teknik-teknik tersebut memakan banyak biaya dan waktu yang cukup lama dalam pelaksanaannya. Contohnya saja seperti, teknik etnografi yang menghabiskan banyak waktu bagi perekayasa kebutuhan untuk mengamati dan menganalisis secara langsung bagaimana pemangku kepentingan bekerja di lapangan dan teknik wawancara yang membutuhkan waktu serta biaya yang tidak sedikit dalam pelaksanaan wawancaranya maupun pelatihan peningkatan kualitas si pewawancaranya [1]. Pemilihan teknik elisitasi sangatlah bergantung kepada waktu, sumber daya yang tersedia, dan jenis informasi yang perlu digali.

Untuk memudahkan perekayasa kebutuhan dalam mengelaborasi kebutuhan suatu perangkat lunak, sebuah kakas bantu sistem rekomendasi kebutuhan perangkat lunak mampu memberikan solusi atas permasalahan-permasalahan tersebut. Sistem rekomendasi diharuskan untuk bisa merekomendasikan kebutuhan-kebutuhan perangkat lunak yang sesuai dengan ruang lingkup perangkat lunak yang akan dibangun. Algoritma FP-Growth dapat digunakan sebagai metode dalam proses perekomendasi dengan mengidentifikasi asosiasi antar kebutuhan perangkat lunak, sebagai contoh, menemukan asosiasi bahwa suatu perangkat lunak memiliki kebutuhan A dan juga memiliki kebutuhan B yang terkait dengan kebutuhan A [2]. Penemuan asosiasi antar kebutuhan ini dapat membantu perekayasa kebutuhan dalam mempertimbangkan kebutuhan lain yang diinginkan. Untuk mendukung bekerjanya algoritma FP-Growth, diperlukan *dataset* untuk membentuk kemungkinan kandidat kombinasi antar kebutuhan perangkat lunak yang akan diproses dengan beberapa pustaka pemrosesan teks untuk memecah kalimat menjadi bagian-bagian kata berdasarkan jenisnya (*Part-of-Speech*) dan mengelompokkan kata berdasarkan kemiripan maknanya secara semantik.

Dengan demikian, sistem rekomendasi kebutuhan perangkat lunak ini diharapkan mampu merekomendasikan perekayasa kebutuhan dalam menentukan kebutuhan yang sesuai dengan ruang lingkup perangkat lunak guna meminimalkan usaha dalam pelaksanaan elisitasi kebutuhan perangkat lunak.

II. DASAR TEORI

A. Elisitasi Kebutuhan Perangkat Lunak

Elisitasi merupakan aktivitas awal dalam proses rekayasa kebutuhan perangkat lunak (*Software Requirements Engineering*). Sebelum kebutuhan dapat dianalisis, dimodelkan, atau ditetapkan, kebutuhan harus dikumpulkan terlebih dahulu melalui proses elisitasi. Elisitasi kebutuhan adalah sekumpulan aktivitas yang ditujukan untuk menemukan kebutuhan suatu sistem melalui komunikasi dengan pelanggan, pengguna sistem dan pihak lain yang memiliki kepentingan dalam pengembangan sistem [1]. Elisitasi kebutuhan bertujuan untuk mengetahui masalah apa saja yang perlu dipecahkan, mengenali batasan-batasan sistem, mengenali siapa saja para pemangku kepentingan, dan mengenali tujuan dari sistem yaitu sasaran-sasaran yang harus dicapai. Secara umum, model proses elisitasi terdiri dari empat aktivitas, yaitu [1]:

1. Penemuan Kebutuhan (*Requirement Discovery*)
Proses ini merupakan proses interaksi dengan para pemangku kepentingan sistem untuk mengumpulkan kebutuhan mereka. Ranah kebutuhan dari para pemangku kepentingan dan dokumentasi juga didapatkan selama aktivitas ini. Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah perencanaan sistem.
2. Pengelompokkan dan Pengorganisasian Kebutuhan (*Requirement Classification and Organization*)
Aktivitas ini mengoleksi kebutuhan yang belum terstrukturkan, mengelompokkan kebutuhan yang saling terkait, dan kemudian mengorganisasikannya ke dalam kelompok yang koheren. Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah penetapan tujuan dari masing-masing pengguna.
3. Prioritas dan Negosiasi Kebutuhan (*Requirements Prioritization and Negotiation*)
Konflik kebutuhan yang timbul akibat banyaknya pemangku kepentingan yang dilibatkan tidak dapat dielakkan. Kegiatan ini difokuskan pada prioritas kebutuhan dan menemukan serta memecahkan konflik kebutuhan melalui proses negosiasi. Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah analisis resiko dari masing-masing kebutuhan, yang meliputi penilaian resiko serta identifikasi kontrol yang dapat diterapkan untuk mereduksi resiko dari setiap kebutuhan.
4. Dokumentasi Kebutuhan (*Requirements Documentation*)
Kebutuhan didokumentasikan dan menjadi masukan untuk siklus spiral selanjutnya (iteratif). Dari setiap siklus tersebut akan dihasilkan suatu dokumen kebutuhan formal dan nonformal. Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah validasi dan pengembangan sistem.

Berikut ini merupakan langkah-langkah dalam mengelisisasi kebutuhan [1]:

1. Identifikasi orang-orang yang akan membantu menemukan kebutuhan dan memahami organisasi mereka, menilai kelayakan bisnis dan teknis untuk sistem yang diusulkan.

2. Menentukan lingkungan teknis (misalnya, komputasi arsitektur, sistem operasi, kebutuhan telekomunikasi) ke mana sistem atau produk akan ditempatkan.
3. Identifikasi ranah permasalahan, yaitu karakteristik lingkungan bisnis yang spesifik ke ranah aplikasi.
4. Menentukan satu atau lebih metode elisitasi kebutuhan, misalnya wawancara, *Focus Group Discussion* (FGD), dan pertemuan tim.
5. Meminta partisipasi dari banyak orang sehingga dapat mereduksi dampak dari kebutuhan yang bias yang teridentifikasi dari sudut pandang yang berbeda dari pemangku kepentingan dan mengidentifikasi alasan untuk setiap kebutuhan yang dicatat.
6. Mengidentifikasi kebutuhan yang ambigu dan menyelesaikannya.
7. Membuat skenario penggunaan untuk membantu pelanggan/pengguna mengidentifikasi kebutuhan utama.

B. Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang bertujuan untuk memperkirakan informasi yang menarik bagi penggunanya dan juga membantu pengguna dalam memutuskan informasi apa saja yang akan diambilnya. Sistem rekomendasi diperlukan adanya preferensi atau profil pengguna dalam menentukan pilihan dari sekian banyak *item* yang ada sesuai dengan kebutuhan pengguna. Profil pengguna umumnya didasarkan pada menarik atau tidaknya suatu informasi yang dibuat oleh pengguna. Sistem rekomendasi menawarkan kemungkinan dari penyaringan informasi pribadi sehingga hanya informasi yang sesuai dengan kebutuhan dan preferensi pengguna yang akan ditampilkan di sistem dengan menggunakan sebuah teknik atau model rekomendasi [3]. Metode atau pendekatan yang dipilih pada sistem rekomendasi bergantung pada permasalahan yang akan diselesaikan, teknik rekomendasi yang berbeda-beda digunakan untuk aplikasi yang berbeda, dasar dari suatu tujuan dan objektif dari sebuah aplikasi. Beberapa metode sistem rekomendasi, yaitu [4]:

- a. *Content-Based*, bekerja dengan cara mencari *item* lain yang mirip dengan *item* yang disukai oleh pengguna berdasarkan informasi atau teks dari setiap *item*.
- b. *Collaborative Filtering*, memprediksi kegunaan *item* berdasarkan penilaian pengguna sebelumnya.
- c. *Knowledge-Based*, merekomendasikan berdasarkan pengetahuan tentang bagaimana *item* tertentu dapat memenuhi kebutuhan pengguna.
- d. *Hybrid Based*, kombinasi dari beberapa metode yang ada dalam pengembangan sistem rekomendasi.

C. Association Rule Mining dengan Algoritma FP-Growth

Algoritma *FP-Growth* merepresentasikan transaksi dengan menggunakan struktur *FP-Tree* yang dibentuk oleh sebuah akar yang diberi label *null* atau \emptyset [2]. Algoritma ini biasanya diterapkan untuk *market-based analysis* yaitu mencari aturan asosiasi apabila suatu pembeli membeli barang A dan pasti akan membeli barang B. Sebagai contoh, terdapat aturan asosiasi terhadap produk roti, dengan selai, margarin, dan coklat meses. *FP-Tree* akan semakin efektif apabila

semakin banyak pula *item* yang sama pada setiap *itemset*. Proses algoritma ini terdiri dari beberapa tahap, yaitu:

1. Analisis *frequent item*, atau analisis pola frekuensi tinggi untuk mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support (threshold)*. Apabila terdapat *item* yang tidak *frequent* atau yang memiliki tingkat kemunculan yang sedikit, maka *item* tersebut akan dihapus. Nilai *support* dapat dihitung dengan persamaan 1 dimana $JT(A)$ merupakan jumlah transaksi yang terdapat pada *item* A dan T merupakan jumlah transaksi.

$$Support(A) = \frac{JT(A)}{T} \times 100\% \quad (1)$$

2. Pembangunan *FP-Tree*, diawali dengan pembangunan *tree* pada setiap *itemset* yang telah diurutkan berdasarkan nilai *support*. Pembangunan *tree* diawali dengan *prefix* atau awalan yang sama dari setiap *itemset*. Apabila terdapat *itemset* yang memiliki *prefix* berbeda, maka *itemset* berikutnya dibangun pada lintasan yang berbeda.
3. Pencarian *Conditional Pattern Base*, dapat dicari dengan menelusuri *item* dengan nilai *support* terendah dimulai dari titik paling bawah sampai ke atas dari suatu *FP-Tree*.
4. Pencarian *Conditional FP-Tree*, dapat dicari dengan membangun *tree* yang didapatkan dari *Conditional Pattern Base* untuk setiap *item* berdasarkan frekuensi *item* yang muncul.
5. Penemuan aturan asosiasi, didapatkan dari seluruh *Conditional FP-Tree* yang telah dibangun untuk setiap *item*. *Item* yang memiliki nilai *support* yang tinggi tidak memiliki aturan asosiasi.

D. Stanford CoreNLP

Stanford CoreNLP merupakan suatu kerangka terpadu dalam menerapkan sekelompok analisis linguistik pada teks. Stanford CoreNLP menyediakan satu *set* alat analisis bahasa alami yang dapat memberikan bentuk dasar kata, kata berdasarkan pada pengucapannya, pemilahan kata berdasarkan subjek, objek, maupun kata kerjanya, dan sebagainya. Stanford CoreNLP mengintegrasikan banyak alat pada Stanford NLP *tools* seperti *Part-of-Speech (POS) tagger*, *Entity Recognizer (NER)*, *parser*, *dependency parser*, sistem resolusi *coreference*, analisis sentimen, dan *pattern learning tools*. Stanford CoreNLP tersedia untuk pengembangan perangkat lunak yang ditulis dengan bahasa pemrograman Java, Python, Ruby, Pearl, Javascript, F#, bahasa .NET, dan Java *Virtual Machine* lainnya [5].

E. WordNet

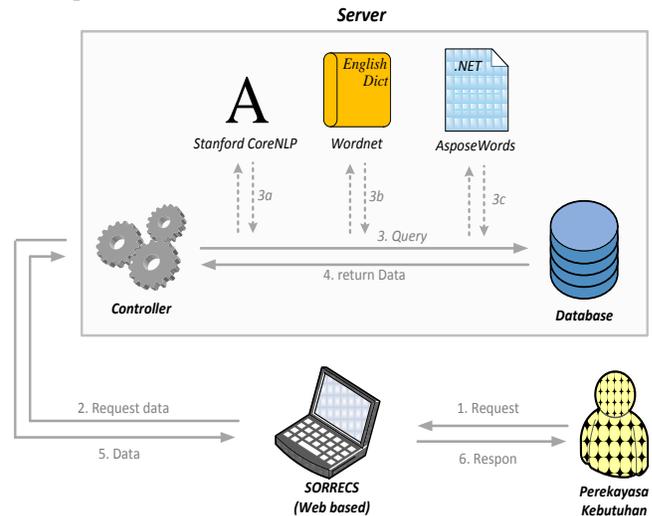
WordNet merupakan *database* leksikal bahasa Inggris yang berisi kata-kata yang tersebar berdasarkan tipenya, seperti kata objek, sifat, kerja, keterangan, dsb. Masing-masing kata dikelompokkan menjadi satu kesatuan sinonim kognitif atau yang biasa disebut dega *synset* yang membedakan antar tipe kata lain. *Synsets* dikelompokkan berdasarkan arti konsep semantik dan relasi leksikal. WordNet sangat mirip dengan database thesaurus lainnya yang mengelompokkan kata berdasarkan tipenya atau huruf alpabetnya, namun yang membedakan yaitu WordNet juga mempertimbangkan masalah maksud kata (*sense of words*). Sebagai hasilnya, antar kata berhubungan dengan kata lain

yang dapat berbeda tipenya berdasarkan semantik untuk menghindari disambiguitas. WordNet banyak dipakai untuk membangun aplikasi berbasis pemrosesan teks dikarenakan WordNet memiliki jumlah kelompok kata yang lebih dari 117000 *synsets* [6].

III. ANALISIS DAN PERANCANGAN

A. Deskripsi Umum Sistem

Kakas bantu sistem rekomendasi kebutuhan perangkat lunak yang diberi nama “SORRECS” yang merupakan singkatan dari *Software Requirements Recommendation System* ini merupakan kakas bantu yang ditujukan bagi para perekayasa kebutuhan untuk melakukan penggalian kebutuhan suatu perangkat lunak. Perekayasa kebutuhan juga dapat memilih kebutuhan perangkat lunak yang lain yang direkomendasikan oleh sistem. Sistem dirancang untuk mampu beroperasi dan bekerja tanpa adanya bantuan administrator (*machine learning*). Fungsi utama dari sistem ini adalah pengelolaan kebutuhan proyek perangkat lunak. Arsitektur perangkat lunak dari kakas bantu SORRECS dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Perangkat Lunak

B. Proses Pembentukan Aturan Asosiasi

Untuk membentuk aturan asosiasi dari kebutuhan perangkat lunak, diperlukan suatu *dataset* yang berupa URD (*User Requirement Document*) yang bertujuan untuk membangun kumpulan-kumpulan kebutuhan perangkat lunak yang terkait untuk direkomendasi. Tabel 1 berisi kumpulan dataset yang digunakan beserta sumber website dataset tersebut didapat.

Tabel 1. Dokumen Dataset URD

No.	Nama Dokumen	Sumber Dokumen
1.	CheatCReq	http://www.csee.wvu.edu/~cukic/.../CheatCheckDes.doc
2.	Deliverable_II	http://folk.uio.no/jumal/www_docs_I/N219/Deliverables/Deliverable_II.doc
3.	Home Subscriber Server_SRS_Document	http://www.imszone.org/.../HomeSubscriberServer_SRS_Document.doc
4.	Detailed_System_Design_Specification_Final	http://www.chart.state.md.us/.../SystemDesignSpecificationFinal.doc
5.	libra_srs	http://www.cloudbus.org/libra/doc/libra

	a_srs.doc
6.	PINES Reports Software Requirements Specification http://pines.georgialibraries.org/
7.	SRS_Outline0517 osucse772socialnetworking.googlecode.com
8.	SRSEExample-webapp https://www.cse.msu.edu/~cse435/Handouts/SRSEExample-webapp.doc
9.	srsv6 https://www.scribd.com/.../aniket
10.	ST3SRS www.courses.utep.edu/.../ST3SRS.doc

Kalimat dari setiap *dataset* dilakukan perbaikan struktur dengan standard kalimat harus memiliki setidaknya satu subjek, satu predikat, dan satu objek. Kalimat juga dilakukan perbaikan dari sisi makna apakah kalimat tersebut logis dan kohesif atau tidak. Dikarenakan setiap kalimat nantinya akan disimpan pada database dan memungkinkan antar kalimat dalam satu dokumen yang sama tidak bisa saling terkait, kalimat juga harus dilakukan perbaikan dari sisi kata ganti orang. Kalimat-kalimat kebutuhan perangkat lunak yang berasal dari dataset kemudian dilakukan proses pemrosesan teks dengan menggunakan pustaka Stanford CoreNLP *Dependency Parser* yang bertujuan untuk menganalisis struktur gramatikal sebuah kalimat, membangun hubungan antar kata utama “*root*” dengan kata-kata yang berhubungan dengan kata utama (subjek, predikat, dan objek) dan *Lemmatization* untuk menyetarakan setiap kata menjadi bentuk tunggalnya. Tanda (*tag*) dari proses *Dependency Parser* yang digunakan untuk sistem ini yaitu *nsubj* dan *conj* untuk kata subjek; *root*, *csubj*, *ccomp*, *xcomp*, *acl*, *advcl*, *conj* untuk kata predikat; dan *dobj*, *iobj*, *nmod*, *nummod*, *appos*, *nsubj*, *nsubjpass*, dan *conj* untuk kata objek; serta *compound* dan *amod* untuk kata pelengkap subjek objek.

Tahap selanjutnya yaitu kata predikat yang didapat dari pemrosesan teks dengan Stanford CoreNLP dilakukan penilaian kemiripan predikat dengan predikat lain berdasarkan maknanya yang telah tersedia dalam database (*dataset*). Penilaian ini menggunakan pustaka Wordnet Penilaian ini menggunakan pustaka Wordnet dari proyek program *WordsMatching*, hasil riset pengembang perangkat lunak NLP asal Australia, Troy Simpson dan Thanh Dao [7]. Modul *WordsMatching* ini bertujuan untuk menghitung secara semantik seberapa mirip makna dari suatu kata dengan kata lain. Nilai paling besar yang dihasilkan berarti pasangan kata tersebut paling mirip. seluruh predikat yang ada di dalam database akan dilakukan penilaian kemiripan semantik satu per satu dan dikelompokkan.

Setelah kelompok predikat didapat, pembentukan aturan-aturan rekomendasi kebutuhan perangkat lunak dengan algoritma FP-Growth dengan kelompok kata predikat. Terdapat modifikasi pada algoritma FP-Growth ini, yaitu perhitungan tidak dilakukan sampai ke tahap pembangunan *Conditional FP-Tree* karena mengingat proses rekomendasi perangkat lunak untuk sistem ini akan melempar kalimat untuk diproses dengan Stanford CoreNLP, dan kata predikat yang didapat akan dilakukan pengelompokkan berdasarkan kemiripan maknanya dan hanya menghasilkan satu kode kelompok, setelah itu akan memanggil aturan yang dibentuk dari FP-Growth berdasarkan kode kelompok yang sama beserta *item* apa saja yang termasuk dalam aturan tersebut. Seluruh aturan-aturan asosiasi yang diperoleh akan disimpan

ke dalam database untuk dilakukan proses penampilan kebutuhan rekomendasi perangkat lunak apabila perekayasa kebutuhan perangkat lunak memasukkan kebutuhan perangkat lunak pada tampilan antarmuka pengguna. Proses pembentukan aturan asosiasi ini dilakukan setiap perekayasa kebutuhan menyimpan seluruh kebutuhan perangkat lunak untuk suatu versi. Dengan kata lain, aturan-aturan asosiasi terbentuk dari seluruh proyek perangkat lunak yang ada berdasarkan versi dokumen yang terbaru.

IV. PENGUJIAN DAN EVALUASI

Pengujian yang dilakukan adalah pengujian terhadap hasil rekomendasi dengan membandingkan hasil rekomendasi dengan hasil analisa manual yang dilakukan oleh perekayasa kebutuhan. Pengujian hasil rekomendasi menggunakan rumus *Cohen Kappa* untuk menghitung akurasi sistem menurut ahli (*expert*). Ahli akan menilai secara subjektif/kualitatif yang hanya memiliki nilai “Ya” (sepakat) atau “Tidak” (tidak sepakat) terhadap data yang diuji. Ahli yang menguji terhadap hasil rekomendasi yaitu perekayasa kebutuhan yang telah berpengalaman dalam membangun suatu perangkat lunak.

Matriks tingkat kesepakatan dapat digunakan untuk membantu dalam pengerjaan dengan rumus Kappa yang dapat dilihat pada Tabel 2 dimana Exp merupakan Ahli (*expert*), A merupakan sistem, B merupakan perekayasa kebutuhan, TP (*True Positive*) merupakan jumlah kesepakatan antara A dan B sama-sama sepakat, FP (*False Positive*) merupakan jumlah kesepakatan dimana A sepakat dan B tidak sepakat, FN (*False Negative*) merupakan jumlah kesepakatan dimana A tidak sepakat dan B sepakat, dan TN (*True Negative*) merupakan jumlah kesepakatan antara A dan B sama-sama tidak sepakat.

Tabel 2. Matriks Tingkat Kesepakatan

		B	
		Yes	No
A	Exp	TP	FP
		FN	TN

Dari matriks kesepakatan, rumus Kappa dapat dihitung dengan persamaan 2 dimana K merupakan nilai akurasi Kappa, p_o merupakan nilai relatif kesepakatan antara dua ahli, dan p_e merupakan nilai probabilitas Yes dan No dari kedua ahli.

$$K = \frac{p_o - p_e}{1 - p_e} \quad (2)$$

Pengujian dilakukan dengan dataset dokumen URD dengan mengambil seluruh kalimat-kalimat kebutuhan fungsional dan non fungsionalnya. Skenario yang digunakan untuk pengujian hasil rekomendasi yaitu dengan dokumen URD tentang perangkat lunak “*E-Store Project*” (www.utdallas.edu) yang fungsi utamanya yaitu menangani seluruh data-data produk elektronik dan hiburan keluarga dari suatu perusahaan elektronik di Amerika Serikat serta dokumen URD “*Twitter iPhone Client*” (www.csee.wvu.edu) yang fungsi utamanya yaitu sebagai aplikasi Twitter pada perangkat genggam iPhone.

Untuk kalimat kebutuhan yang pertama dari dokumen *E-Store Project* yaitu kalimat kebutuhan fungsional “*The system*

shall display all the products that can be configured” akan menampilkan 7 kalimat kebutuhan fungsional rekomendasi lain dari 551 kalimat kebutuhan fungsional yang ada di database.

1. *If the product is not in the stock, the system will inform that the product is not available*
2. *The system can analyze the staff productivity and the workflow*
3. *If the system can not update the product, the sales person must check and make manually the sale and report the sale to the sales manager*
4. *The system can produce statistics on sales, products, sales persons and revenues*
5. *The sales manager registers all the necessary information about the product in the system*
6. *The system can register a product*
7. *The sales person looks for a product in the system*

Sehingga, nilai akurasi Kappa untuk kalimat kebutuhan yang pertama dari dokumen *E-Store Project* yaitu sebesar 44%. Untuk dokumen *E-Store Project* secara keseluruhan, tingkat keakuratan total rata-rata yaitu sebesar 47%.

Sedangkan untuk kalimat kebutuhan yang pertama dari dokumen *Twitter iPhone Client* yaitu kalimat kebutuhan fungsional “A user will enter the credentials into the proper fields and the user will be logged into the Twitter system” akan menampilkan 30 kalimat kebutuhan fungsional rekomendasi lain dari 551 kalimat kebutuhan fungsional yang ada di database.

1. *The system provides a user friendly interface for designing the queries against all the record types*
2. *The user selects the option to enter a property list for a course*
3. *User can enter the files to be examined*
4. *If the request contains valid a username and password, and successfully authenticates with the CHART user database that the user has Manage Traffic Event right, the session is marked as CHART*
5. *If the username and password information is missing or if they cannot be authenticated within the CHART user database, the session is marked as ReadOnly*
6. *When user clicks the Go button, user enters county, route type, route number and mile*
7. *When user clicks the Go button, user enters the county only and zoom to the county*
8. *When user clicks the Go button, user enters the county, the route type and the oute number, intersecting street name and then center on the intersection point*
9. *When user clicks the Go button, user enters the county, the route type and the route number and then zoom to the route extent*
10. *For each student, the user can enter the points earned for each assessment unit*
11. *The general user can enter and modify the course information*
12. *The general user can enter and modify the student information or download the student information from the university database*
13. *The general user can enter the information related to an AI*
14. *The general user can enter the student grades related to an AI and enter the formulae for calculating the grades*
15. *The general user can use the system to enter the course outcomes*
16. *The system prompts the user to enter the password*
17. *The user can enter a description of a targeted competency that can be used for manual checking*
18. *The user can enter the values associated with the different types*
19. *The user can set the password for accessing and entering an information on the course*

20. *The user chooses an AI to enter the grades*
21. *The user chooses to enter a description of a targeted competency that can be used for manual checking*
22. *The user does not enter all the required information*
23. *The user enters a formula to calculate the totals for the AIPs and the AI*
24. *The user enters a label and description for each AIP*
25. *The user enters a label and description for each AIP in the subdivision*
26. *The user enters a login name*
27. *The user enters a new value to an existing attribute*
28. *The user enters a password*
29. *The user enters an incorrect login name*
30. *The user enters an incorrect password*

Sehingga, nilai akurasi Kappa untuk kalimat kebutuhan yang pertama dari dokumen *Twitter iPhone Client* yaitu sebesar 44%. Untuk dokumen *Twitter iPhone Client* secara keseluruhan, tingkat keakuratan total rata-rata yaitu sebesar 55%.

V. KESIMPULAN DAN SARAN

Berdasarkan proses perancangan serta pengujian kakas bantu sistem rekomendasi kebutuhan perangkat lunak, dapat diambil kesimpulan sebagai berikut:

1. Sistem dapat digunakan untuk menggali kebutuhan perangkat lunak.
2. Sistem dapat memecah kalimat kebutuhan perangkat lunak untuk diambil kata subjek, predikat, dan objeknya untuk dilakukan proses penilaian kemiripan kata yang hasilnya akan digunakan untuk membentuk aturan asosiasi rekomendasi kebutuhan perangkat lunak.
3. Tidak hanya dapat digunakan untuk merekomendasikan item pada studi kasus *market-based analysis*, *FP-Growth* juga dapat digunakan sebagai algoritma untuk merekomendasikan kebutuhan perangkat lunak.
4. Nilai *minimum support (threshold)* yang digunakan untuk sistem ini pada algoritma *FP-Growth* yaitu sebesar 20% yang merupakan nilai *threshod* terbaik.

Berdasarkan saran-saran untuk pengembangan aplikasi di masa yang akan datang diantaranya sebagai berikut:

1. Pemrosesan kemiripan kata tidak hanya berasal dari kata predikatnya saja, namun juga dapat berasal dari kata subjek dan objeknya dengan suatu algoritma yang dapat memproses topik / maksud tersirat dari suatu kalimat
2. Dataset tersedia untuk seluruh topik perancangan perangkat lunak
3. Bahasa yang disediakan tidak hanya bahasa Inggris, namun juga bahasa Indonesia
4. Menambahkan fitur untuk meng-generate rincian atau alur proses spesifik dari setiap kebutuhan fungsional dan non fungsional

VI. DAFTAR PUSTAKA

- [1] D. Siahaan, "Analisa Kebutuhan dalam Rekayasa Perangkat Lunak," Surabaya, Penerbit ANDI, 2012.
- [2] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, The United States of America: Elsevier, 2006.

- [3] R. Tsalaatsa, E. Santoso and D. E. Ratnawati, "Sistem Rekomendasi Menggunakan Item-Based Clustering Hybrid Method," 2013.
- [4] R. Burke, "Hybrid Recommender System.," *In Proceedings of The Adaptive Web*, 2007.
- [5] The Stanford Natural Language Processing Group, "Natural Language Processing," Stanford University, [Online]. Available: <http://nlp.stanford.edu/>. [Accessed 10 Desember 2015].
- [6] WordNet, "WordNet," Princeton University, [Online]. Available: wordnet.princeton.edu.
- [7] T. Simpson and T. Dao, "WordNet based semantic similarity measurement," 2005.