

REKOMENDASI PERBAIKAN PERNYATAAN KEBUTUHAN YANG RANCU DALAM SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK MENGGUNAKAN TEKNIK BERBASIS ATURAN

Depandi Enda¹, Daniel Siahaan²

^{1,2}Institut Teknologi Sepuluh Nopember
Email: ¹depandienda@gmail.com, ²daniel@if.its.ac.id

(Naskah masuk: 22 Januari 2018, diterima untuk diterbitkan: 23 Mei 2018)

Abstrak

Tahap awal dalam pengembangan perangkat lunak ialah menelusuri, mengumpulkan dan menyajikan segala kebutuhan pengguna ke dalam sebuah dokumen spesifikasi kebutuhan perangkat lunak (SKPL). Latar belakang akademik yang beragam, pengalaman yang berbeda, dan keterbatasan pengetahuan yang dimiliki oleh perekayasa kebutuhan memungkinkan adanya kesalahan dalam pembuatan dokumen SKPL. Salah satu kesalahan yang sering muncul pada sebuah dokumen SKPL ialah terdapatnya penggunaan kata-kata yang rancu. Hal ini tentunya dapat menyebabkan kesalahan penafsiran dan kesulitan dalam memahami kebutuhan perangkat lunak yang hendak dibangun bagi pemangku kepentingan dalam proses pengembangan perangkat lunak. Penelitian ini bertujuan mengusulkan sebuah pendekatan untuk memberikan rekomendasi perbaikan pernyataan kebutuhan perangkat lunak yang rancu. Adapun metode yang diusulkan adalah teknik berbasis aturan dengan menggunakan model bahasa *n-gram*. Realibilitas metode usulan dievaluasi menggunakan indeks statistik Gwet's AC1. Hasil analisis metode rekomendasi yang diusulkan memiliki tingkat proporsi kesepakatan yang lebih baik dibandingkan dengan metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-gram*. Metode rekomendasi yang diusulkan memiliki nilai indeks statistik Gwet's AC1 tertinggi sebesar 0.5263 dengan tingkat proporsi kesepakatan sedang.

Kata kunci: Metode Rekomendasi, Pernyataan Kebutuhan Perangkat Lunak, Kata Rancu, Teknik Berbasis Aturan, Indeks Statistik Gwet's AC1

RECOMMENDATION OF REPAIRING AMBIGUOUS REQUIREMENT STATEMENT IN SOFTWARE REQUIREMENT SPECIFICATION USING RULE BASED TECHNIQUE

Abstract

The first stage in software development is to investigate, collect and provide all user requirements into a software requirements specification document (SRS's). Diverse academic background, different experiences, and the limitations of knowledge possessed by the requirement engineer make possible mistakes in the creation of SRS's documents. One of the most common mistakes in SRS's document is the use of ambiguous words. This can certainly lead to misinterpretation and difficulties in understanding the software requirement that stakeholders to built in the software development process. The purpose of this research is to build an approach that gives recommendation improvement of ambiguous software requirement statement. The proposed method is a rule-based technique using *n-gram* language model. The reliability of the proposed method is evaluated using Gwet's AC1 statistical index. The analysis results of the proposed recommendation method have a better level of agreement proportion than the recommendation method using the *n-gram* frequency-based statistical technique. The proposed recommendation method has the highest Gwet's AC1 statistic value of 0.5263 with a moderate agreement proportion rate.

Keywords: Recommendation Method, Software Requirement Statement, Ambiguous Word, Rule Based Technique, Gwet's AC1 Statistics Index

1. PENDAHULUAN

Tahap awal dalam pengembangan perangkat lunak ialah menelusuri, mengumpulkan dan menyajikan segala kebutuhan pengguna kedalam sebuah dokumen spesifikasi kebutuhan perangkat lunak (SKPL). Dokumen spesifikasi kebutuhan

perangkat lunak memiliki peranan yang sangat penting dalam menghasilkan perangkat lunak yang berkualitas tinggi. Sebuah dokumen spesifikasi kebutuhan yang berkualitas sebaiknya memiliki beberapa karakteristik seperti ketepatan, tidak rancu, kelengkapan, konsisten, diurutkan berdasarkan

kepentingan dan stabilitas, terverifikasi, dan mudah dilacak apabila terdapat modifikasi dokumen spesifikasi di masa yang akan datang dengan menyatakan dimana persyaratan telah dirujuk (Siahaan, 2012).

Menelusuri dan mengidentifikasi sebuah kebutuhan adalah suatu hal yang rumit untuk dilakukan. Selalu terdapat kesalahan interpretasi kebutuhan antara pelanggan dan pengembang. Hal ini dapat menurunkan kualitas dokumen spesifikasi kebutuhan perangkat lunak. Kegiatan ini melibatkan proses penerjemahan informasi dari satu bentuk ke bentuk yang lainnya yaitu, dari kebutuhan pelanggan ke dalam bentuk arsitektur rancangan kode (Pressman, 2008). Aktivitas ini biasanya dilakukan melalui komunikasi dua arah antara pelanggan dan pengembang baik secara lisan maupun tulisan yang dalam pelaksanaannya mungkin terdapat kesalahan penerjemahan kebutuhan (Siahaan, 2012). Hal ini tentunya memberikan tantangan tersendiri kepada pengembang dalam menyusun dan menyajikan dokumen spesifikasi kebutuhan yang berkualitas dan minim terhadap kesalahan.

Berbagai aspek latar belakang pendidikan yang beragam dan keterbatasan pengetahuan yang dimiliki oleh tim penyusun kebutuhan perangkat lunak memungkinkan adanya kesalahan didalam pembuatan dokumen spesifikasi kebutuhan perangkat lunak. Untuk itu, seorang tenaga ahli dibutuhkan untuk memvalidasi dan menganalisis kesalahan-kesalahan yang terdapat pada dokumen spesifikasi kebutuhan perangkat lunak. Tentunya hal ini dapat menambah biaya dan waktu pengerjaan proyek perangkat lunak. Di sisi lain pengembang memiliki sumber daya yang terbatas.

Kesalahan yang sering muncul pada sebuah dokumen spesifikasi kebutuhan perangkat lunak yang ditulis menggunakan bahasa alamiah ialah terdapatnya penggunaan kata-kata yang rancu, hal ini tentunya dapat menyebabkan kesalahan penafsiran dan sangat sulit untuk dipahami oleh pihak yang terlibat langsung dalam proses pengembangan perangkat lunak (Bussel, 2009). Walau bagaimanapun kesalahan ini dapat memberikan dampak secara signifikan dalam proses pengembangan pada tahap selanjutnya. Salah satu dampak yang disebabkan oleh penggunaan kata rancu pada dokumen spesifikasi kebutuhan perangkat lunak ialah sering terjadinya konflik antara pelanggan dan pengembang, terutama dalam memahami dan memverifikasi pencapaian pengembang terhadap kebutuhan yang telah ditetapkan sejak awal. Semakin terlambat kesalahan pada kebutuhan terdeteksi, maka semakin besar biaya dan waktu yang dibutuhkan untuk mengatasinya. Pada sisi pengembang, spesifikasi kebutuhan yang rancu akan mendorong pengembang untuk membuat banyak asumsi-asumsi terkait kebutuhan pengguna, namun hal ini juga dapat menimbulkan resiko apabila dilakukan. Oleh karena

itu, sebuah pendekatan untuk mengidentifikasi dan mengurangi penggunaan kata-kata yang berpotensi rancu pada dokumen spesifikasi kebutuhan perangkat lunak sangat diperlukan.

Beberapa pendekatan yang digunakan untuk mengidentifikasi kerancuan yang terdapat pada sebuah dokumen spesifikasi kebutuhan perangkat lunak telah dikaji oleh beberapa peneliti antara lain telah dibahas oleh Muliawan *et al.*, (2011) mengusulkan sebuah pendekatan untuk mendeteksi kerancuan pada pernyataan kebutuhan perangkat lunak yang ditulis menggunakan bahasa alamiah, dimana pendekatan yang diusulkan menggunakan teknik pemrosesan bahasa alamiah dengan menerapkan aturan SMART (*Specific, Measureable, Attainable, Reliazable, and Trackable*). Vimalraj dan Seema (2016) juga membahas pendekatan untuk mengidentifikasi kerancuan kebutuhan perangkat lunak menggunakan acuan makna kata multibahasa. Adapun kerancuan yang teridentifikasi dikategorikan ke dalam kerancuan leksikal, sintaksis dan semantik. Bhatia *et al.*, (2016) mengusulkan sebuah pendekatan berbasis ontologi untuk mendeteksi kerancuan pada dokumen spesifikasi perangkat lunak. Sabriye dan Zainon (2017) mengembangkan sebuah kerangka kerja yang dapat mengidentifikasi kerancuan dalam dokumen spesifikasi perangkat lunak menggunakan teknik pemrosesan bahasa alamiah.

Dari beberapa penelitian sebelumnya yang telah dipaparkan hanya berfokus pada pendekatan yang digunakan untuk mengidentifikasi kerancuan pada dokumen spesifikasi kebutuhan perangkat lunak. Sehingga topik dalam pengembangan metode rekomendasi perbaikan pernyataan kebutuhan yang rancu menjadi topik yang menarik untuk dikaji. Adapun penelitian ini bertujuan mengusulkan sebuah pendekatan untuk memberikan rekomendasi perbaikan pernyataan kebutuhan yang rancu dalam spesifikasi kebutuhan perangkat lunak. Penelitian ini juga melakukan kajian terhadap beberapa penelitian terkait yang membahas beberapa pendekatan yang digunakan untuk memberikan rekomendasi pada ruang lingkup perbaikan kesalahan tata bahasa dalam bahasa Inggris. Kajian ini dilakukan sebagai referensi untuk membangun pendekatan rekomendasi perbaikan pernyataan kebutuhan yang rancu.

2. PENELITIAN TERKAIT

Teknik pemeriksaan dan perbaikan kesalahan tata bahasa umumnya dapat digolongkan menjadi tiga pendekatan yaitu pendekatan berbasis pencocokan pola, berbasis statistik dan berbasis aturan. Pendekatan berbasis statistik telah dibahas oleh Henrich dan Reuter (2009) yang mengusulkan sebuah metode dan kaskas bantu untuk memeriksa dan memperbaiki kesalahan tata bahasa yang independen terhadap jenis bahasa berdasarkan pendekatan statistik. Bahasa yang digunakan sebagai

bahasa uji coba penelitian ialah bahasa Inggris dan bahasa Jerman. Hasil evaluasi menunjukkan bahwa pendekatan Henrich dan Reuter (2009) dapat bekerja untuk bahasa yang berbeda walaupun ketepatan pemeriksaan tata bahasa bervariasi, alasannya adalah karena perbedaan kekayaan morfologis bahasa. Keterbatasan metode ini ialah kuantitas dan kualitas data latih sangat menentukan performa metode. Metode perbaikan yang diusulkan oleh Henrich dan Reuter (2009) menunjukkan bahwa masih banyak terdapat masalah dalam menemukan semua kesalahan gramatikal. Henrich dan Reuter (2009) mengatasi masalah tersebut dengan mengusulkan sebuah gagasan yang mengkombinasikan dua pendekatan yaitu pendekatan berbasis statistik dan aturan (*hybrid*). Teknik *hybrid* akan bekerja dengan sangat baik dalam kasus bahasa yang tergantung pada satu bahasa tertentu (bahasa yang dependen), misalnya sebuah kasus permasalahan yang diselesaikan di batasi hanya menggunakan bahasa Inggris (Henrich dan Reuter, 2009).

Teknik perbaikan kesalahan tata bahasa berbasis statistik lainnya juga di bahas oleh Athanaselis *et al.*, (2011) yang mengusulkan sebuah metode dalam memperbaiki kesalahan tata bahasa dalam urutan kata menggunakan model bahasa statistik *bigram* dan *trigram* yang diekstrak menggunakan sebuah korpus yang besar yaitu *British National Corpus* (BNC). Penelitian Athanaselis *et al.*, (2011) menunjukkan hasil yang cukup baik dalam mendeteksi dan memperbaiki kesalahan tata bahasa dalam urutan kata menggunakan model bahasa *bigram* dan *trigram*. Wu *et al.*, (2013) mengusulkan sebuah metode berdasarkan cara kerja mesin terjemahan untuk mengoreksi kesalahan serial gramatikal pada kalimat tertentu dalam penulisan tata bahasa. Evaluasi terhadap serangkaian kalimat pada korpus latih menunjukkan bahwa metode ini dapat memperbaiki kesalahan serial dengan cukup baik menggunakan model bahasa *trigram*.

Pendekatan berbasis aturan telah dibahas oleh Naber (2003) yang mengembangkan sebuah metode dan kaskas bantu yang dapat memeriksa kesalahan tata letak dan tata bahasa dalam kalimat bahasa Inggris menggunakan teknik berbasis aturan. Untuk meningkatkan akurasi metode yang diusulkan, Naber (2003) menggunakan sebuah korpus yang besar yaitu *British National Corpus* untuk memastikan bahwa sistem yang dibangun tidak melaporkan terlalu banyak kesalahan pada kalimat yang benar, sehingga akurasi sistem dapat meningkat. Teknik berbasis aturan yang diusulkan Naber (2003) dapat digunakan untuk mengekspresikan aturan yang menggambarkan kesalahan pada tingkat frasa dan tidak hanya pada tingkat kata. Sehingga metode ini menjadi salah satu metode yang dapat diandalkan dalam memeriksa kesalahan tata bahasa (Naber, 2003). Peneliti lainnya yang juga membahas teknik berbasis aturan dalam

memeriksa kesalahan tata bahasa yaitu Singh *et al.*, (2016) yang mengembangkan sebuah metode untuk memperbaiki kesalahan ejaan menggunakan teknik rekomendasi berbasis frekuensi dan memeriksa kesalahan tata bahasa menggunakan teknik berbasis aturan. Adapun pemeriksaan tata bahasa yang diusulkan hanya berfokus pada mendeteksi dan memperbaiki aturan *tenses* yang salah dalam bahasa Inggris.

Dari beberapa penelitian yang telah dijabarkan serta beberapa pendekatan solusi dari permasalahan yang telah dibahas, maka pada penelitian ini akan mengusulkan sebuah pendekatan untuk memberikan rekomendasi perbaikan pernyataan kebutuhan yang rancu menggunakan teknik berbasis aturan (Naber, 2003) (Singh *et al.*, 2016). Dimana teknik berbasis aturan adalah salah satu teknik yang dapat diandalkan dalam memberikan rekomendasi perbaikan kesalahan tata bahasa. Untuk menambah performa metode pemberian rekomendasi yang diusulkan, penelitian ini menggunakan teknik *hybrid* yang di gagas oleh Henrich dan Reuter (2009) dalam menutupi keterbatasan dan kelemahan dari teknik berbasis statistik.

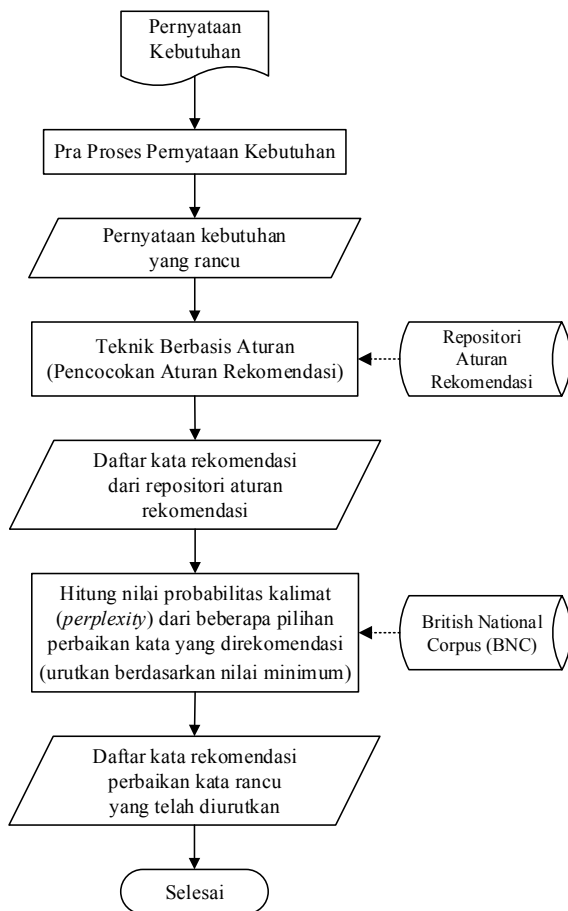
Teknik *hybrid* bekerja dengan cara mengkombinasikan dua teknik yaitu teknik berbasis aturan dan teknik berbasis statistik *n-gram* dalam meningkatkan kinerja metode. Teknik berbasis statistik *n-gram* berguna untuk menentukan pilihan kandidat kata yang akan direkomendasikan. Model bahasa *n-gram* yang digunakan pada penelitian ini terdiri dari dua model yaitu model bahasa *bigram* dan *trigram* yang menunjukkan kinerja yang cukup baik dalam memberikan rekomendasi kesalahan tata bahasa Athanaselis *et al.*, (2011) dan Wu *et al.*, (2013). Sedangkan untuk meningkatkan kinerja model bahasa *n-gram* Athanaselis *et al.*, (2011) dan Wu *et al.*, (2013) memanfaatkan sebuah korpus yang besar sebagai kamus kata referensi pencarian pasangan *n-gram*. Sehingga pada penelitian ini juga akan memanfaatkan sebuah korpus yang besar yaitu *British National Corpus* dalam meningkatkan kinerja metode.

Pendekatan yang diusulkan diharapkan dapat mengurangi kesalahan khususnya penggunaan kata-kata yang rancu pada sebuah pernyataan kebutuhan perangkat lunak dengan menggantikan peran seorang tenaga ahli dalam menganalisis kebutuhan.

3. METODE

Metode rekomendasi yang diusulkan pada penelitian ini adalah teknik berbasis aturan, dimana teknik ini menggunakan model bahasa *bigram* dan *trigram* untuk menghitung nilai probabilitas pernyataan kebutuhan (*perplexity*) sebagai acuan perbaikan dari daftar urutan kata yang akan direkomendasikan. Tahapan metode rekomendasi yang diusulkan pada penelitian ini terdiri dari 3 tahap yaitu, tahap pertama pemrosesan awal pernyataan kebutuhan, tahap kedua pencocokan aturan

rekomendasi menggunakan teknik berbasis aturan dan tahap ketiga menghitung nilai probabilitas pernyataan kebutuhan (*perplexity*) dari beberapa pilihan kata rekomendasi. Adapun mekanisme solusi yang diusulkan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Alur kerja metode rekomendasi yang diusulkan

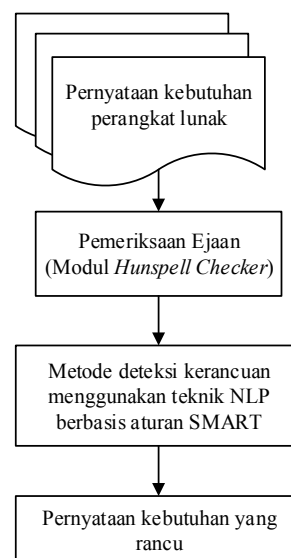
Secara terperinci penjelasan alur kerja metode rekomendasi yang diusulkan pada penelitian ini akan dijabarkan pada pembahasan sub bagian berikut.

3.1. Pemrosesan Awal Pernyataan Kebutuhan

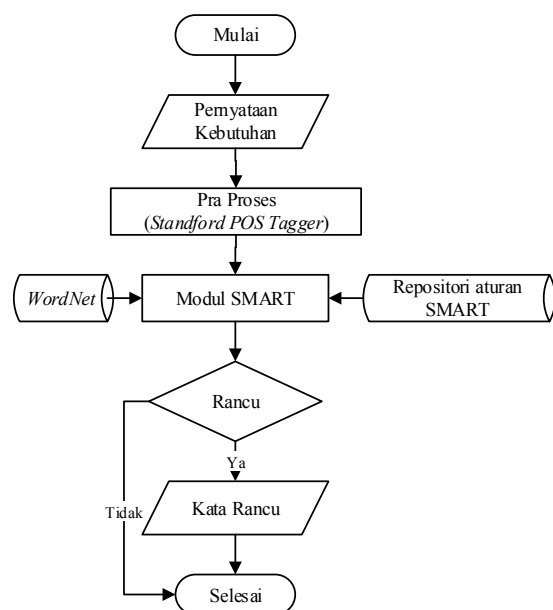
Data yang digunakan sebagai data masukan ialah pernyataan kebutuhan perangkat lunak. Pernyataan kebutuhan terlebih dahulu akan dilakukan pemrosesan awal yaitu pemeriksaan kesalahan ejaan dan pendeteksian kerancuan. Hal ini bertujuan untuk mendapatkan pernyataan kebutuhan perangkat lunak yang rancu dan tidak terdapat kesalahan ejaan. Pernyataan kebutuhan yang rancu akan digunakan sebagai data untuk membuat repositori aturan rekomendasi dan sebagai data pengujian pada penelitian ini. Adapun tahapan pemrosesan awal pernyataan kebutuhan dapat dilihat pada Gambar 2 berikut.

Modul yang digunakan untuk memeriksa kesalahan ejaan pada pernyataan kebutuhan ialah modul *Hunspell Checker*. Modul ini bersifat terbuka dan telah banyak digunakan pada beberapa aplikasi

pengolah kata seperti *LibreOffice* dan *OpenOffice*. Modul ini bekerja menggunakan teknik kesamaan *n-gram*, berbasis aturan dan kamus pengucapan. Hasil dari modul ini ialah daftar kata yang terdapat kesalahan ejaan beserta rekomendasi perbaikannya. Selanjutnya, setelah kata yang salah ejaan diperbaiki langkah berikutnya adalah mendeteksi kerancuan pada pernyataan kebutuhan perangkat lunak. Adapun metode yang digunakan untuk mendeteksi kerancuan pada penelitian ini adalah teknik NLP berbasis aturan SMART (Muliawan *et al.*, 2011).



Gambar 2. Tahapan pemrosesan awal pernyataan kebutuhan



Gambar 3. Metode identifikasi kata rancu

Pada Gambar 3 dapat diketahui bahwa tahapan awal dari metode identifikasi kerancuan yang digunakan pada penelitian ini ialah melakukan pemrosesan terhadap data input yang berupa

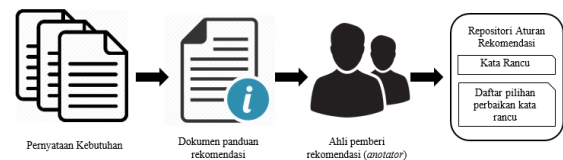
pernyataan kebutuhan. Kemudian pustaka *Stanford POS Tagger* digunakan untuk menandai tiap bagian kata atau *Part of Speech* (POS) pada pernyataan kebutuhan agar bisa di proses lebih lanjut oleh Modul SMART. Sebuah repositori aturan SMART dibuat menggunakan basis data *MySQL*. Kata-kata yang bersifat rancu beserta POS-nya akan disimpan dalam repositori ini yang nantinya digunakan oleh Modul SMART dalam proses pemilihan aturan yang tepat dalam rangka menganalisis kerancuan sebuah pernyataan kebutuhan. Selanjutnya pustaka *WordNet* bertugas untuk memberikan rekomendasi berupa sinonim dari kata-kata yang bersifat rancu yang ada di dalam repositori aturan SMART. Modul SMART adalah modul utama yang bertugas untuk menganalisis kerancuan setiap pernyataan kebutuhan yang telah diproses sebelumnya. Proses analisis dibantu oleh repositori aturan SMART dan pustaka *WordNet*. Modul SMART memiliki 6 tugas utama, yaitu memilih aturan yang tepat, mendeteksi frasa rancu, mendeteksi kerancuan tanpa bantuan sinonim dari *WordNet*, menanyakan sinonim kata rancu pada *WordNet*, mendeteksi kerancuan dengan bantuan sinonim dari *WordNet*, dan mencetak hasil analisis yang disertai dengan catatan mengapa sebuah pernyataan kebutuhan dinilai rancu dengan menunjukkan bagian kata mana yang bersifat rancu. Hasil dari metode deteksi kerancuan ialah apakah pernyataan kebutuhan rancu ataupun tidak.

Jika kata rancu ditemukan pada pernyataan kebutuhan maka proses akan berlanjut pada proses pemberian rekomendasi perbaikan kata rancu, dimana kata rancu yang terdeteksi dan hasil penandaan tiap bagian kata pada pernyataan kebutuhan akan dijadikan sebagai input untuk tahapan selanjutnya pada penelitian ini.

3.2. Pembuatan Repositori Aturan Rekomendasi

Kata rancu yang terdeteksi pada sebuah pernyataan kebutuhan akan diperbaiki dengan mencocokkan aturan yang terdapat pada sebuah repositori aturan rekomendasi menggunakan metode pencocokan *string*. Dalam repositori aturan rekomendasi terdapat kata-kata rancu beserta beberapa opsi perbaikannya. Dimana untuk membangun sebuah repositori aturan rekomendasi, pada penelitian ini dibutuhkan 2 orang ahli yang menguasai dan memahami bidang rekayasa kebutuhan maupun bidang rekayasa perangkat lunak khususnya dalam ruang lingkup bahasa Inggris. Ahli akan dibekali dokumen panduan untuk memberikan rekomendasi pernyataan kebutuhan yang rancu, dimana pemberian rekomendasi dari kata yang teridentifikasi sebagai kata yang rancu pada pernyataan kebutuhan dilakukan dengan memberikan beberapa pilihan perbaikan kata rancu yang terdeteksi agar menjadi sebuah pernyataan kebutuhan yang tidak rancu. Gambar 4 mengilustrasikan beberapa tahapan yang digunakan dalam pembuatan repositori aturan.

Sumber data yang digunakan untuk membangun repositori aturan rekomendasi ialah beberapa pernyataan kebutuhan yang rancu berasal dari sumber data uji coba beberapa penelitian sebelumnya yang telah membahas tentang deteksi kerancuan pada dokumen spesifikasi kebutuhan perangkat lunak. Adapun dataset pernyataan kebutuhan yang akan digunakan pada penelitian ini terdiri dari 12 dataset yaitu pernyataan kebutuhan pada penelitian Hussain (2007) *ACM's OOPSLA DesignFest®*, spesifikasi kebutuhan *Lift Controller System*, dokumen spesifikasi kebutuhan *Yacht Race Results*, *Batch Poster System*, *Cask Loader Software*, *EVLA Array Operations*, *Large Area Telescope (LAT) Science Analysis Software (SAS) Level III Specification*, *PESA High-Level Trigger Selection*, *Sort Algorithm Demonstration Program*, *New Adelaide Airport System* dan *MCSS System*.



Gambar 4. Tahapan pembuatan repositori aturan rekomendasi

Jumlah data pernyataan kebutuhan yang akan dijadikan sebagai dataset pembuatan repositori aturan rekomendasi adalah sebanyak 96 data pernyataan kebutuhan. Dimana karakteristik pernyataan kebutuhan yang digunakan untuk membangun repositori aturan rekomendasi diupayakan dapat mencakup seluruh aturan yang ada pada kamus aturan kata rancu dari metode deteksi kerancuan. Beberapa sampel pernyataan kebutuhan akan dipilih secara acak dari 12 dataset yang ada agar dapat mencakup seluruh aturan yang ada pada kamus aturan kata rancu. Selanjutnya, ahli akan bertindak sebagai penganalisis kata-kata apa saja yang tepat dan disarankan untuk menggantikan kata rancu yang terdapat pada pernyataan kebutuhan, dimana kemungkinan kata yang direkomendasi bisa saja berjumlah lebih dari satu rekomendasi kata pengganti.

Berikut adalah salah satu contoh pemberian rekomendasi dari ahli dalam memberikan saran perbaikan pernyataan kebutuhan yang rancu pada sampel pernyataan kebutuhan perangkat lunak.

Tabel 1. Contoh Hasil Rekomendasi Perbaikan Pernyataan Kebutuhan Dari Ahli (Anotator)

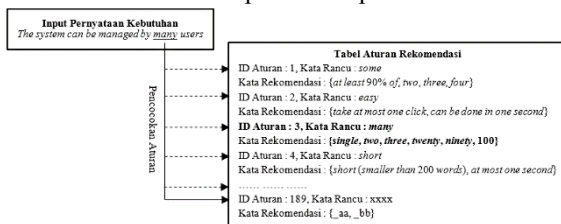
Contoh Pemberian Rekomendasi Dari Ahli	
Pernyataan kebutuhan	The system can be managed by many users
Hasil penandaan tiap bagian kata (POS Tagging)	The_DT system_NN can_MD be_VB managed_VBN by_IN many_JJ users_NNS ._. Deskripsi kerancuan Kata dan tag rancu : 1) many_JJ → users_NNS
Kata rancu (kata yang dicetak tebal)	The system can be managed by many ¹ users
Kandidat kata yang direkomendasi	1) two, three, five, ten, 100

Pada Tabel 1 terdapat satu kata rancu yang terdeteksi pada pernyataan kebutuhan “The system can be managed by many users” dimana kata rancu yang terdeteksi ditandai dengan huruf yang dicetak tebal dan *caption* 1 yaitu kata “**many**”. Annotator memberikan rekomendasi pada bagian kandidat kata rekomendasi. Dimana pada pernyataan kebutuhan “The system can be managed by many users” annotator memberikan rekomendasi pada kata rancu “many” yaitu (two, three, five, ten, 100).

Daftar kata rancu beserta kata rekomendasi yang disarankan oleh ahli akan dikumpulkan dan dimasukkan ke dalam tabel aturan rekomendasi, untuk selanjutnya akan digunakan sebagai repositori aturan rekomendasi perbaikan pernyataan kebutuhan yang rancu menggunakan teknik berbasis aturan.

3.3. Pencocokan Aturan Rekomendasi

Pemrosesan awal dataset menghasilkan keluaran berupa daftar kata rancu yang terdeteksi pada sebuah pernyataan kebutuhan. Tahapan selanjutnya ialah mencocokkan kata rancu yang terdeteksi dengan daftar aturan rekomendasi yang terdapat pada tabel aturan rekomendasi. Gambaran proses pencocokan aturan rekomendasi dari pernyataan kebutuhan “The system can be managed by many users” dengan kata rancu yang ditemukan yaitu kata “many” pada tabel aturan rekomendasi dapat dilihat pada Gambar 5.



Gambar 5. Proses pencocokan aturan rekomendasi

Pada Gambar 5 input dari proses pencocokan aturan rekomendasi ialah pernyataan kebutuhan yang rancu. Kata rancu yang ditemukan pada pernyataan kebutuhan yang telah melewati proses pemrosesan awal pernyataan kebutuhan selanjutnya akan digunakan sebagai kata kunci pencarian untuk mencari aturan yang cocok pada tabel aturan rekomendasi. Adapun jumlah aturan yang terdapat pada tabel aturan rekomendasi berjumlah 189 aturan yang memuat daftar kata rancu beserta daftar kata rekomendasinya. Selanjutnya, proses pencocokan

aturan dilakukan dengan input kata “many” sebagai kata kunci pencarian, hasil dari proses pencocokan aturan pada Gambar 5 ditemukan salah satu aturan yang cocok pada tabel aturan rekomendasi untuk menggantikan kata rancu “many”, yaitu id aturan nomor 3, dengan kata rancu “many” dan daftar kata rekomendasinya adalah {single, two, three, twenty, ninety, 100}. Hasil dari proses pencocokan aturan rekomendasi ialah mengembalikan daftar kata rekomendasi dari kata rancu yang ditemukan pada pernyataan kebutuhan.

3.4. Menghitung Nilai Probabilitas (Perplexity)

Kandidat kata yang direkomendasi akan diurutkan berdasarkan nilai probabilitas kalimat menggunakan teknik statistik model bahasa *n-gram*. Dimana model bahasa *n-gram* yang digunakan pada penelitian ini ialah model bahasa *bigram* dan *trigram*. Hal ini dilakukan untuk memilih kandidat kata yang terbaik untuk menggantikan kata rancu berdasarkan kriteria pilihan kata yang sering muncul pada sebuah korpus.

Untuk mencari probabilitas tiap kalimat menggunakan model statistik *n-gram* maka diperlukan sebuah kamus kata atau korpus yang berguna untuk mencari jumlah frekuensi satu kata (*unigram*), dua kata (*bigram*) dan tiga kata (*trigram*). Korpus yang digunakan pada penelitian ini ialah *British National Corpus* yang terdiri dari sekitar 6,25 juta kalimat dan 100 juta kata, dimana dari 100 juta kata terdapat 90% kata-kata bahasa Inggris yang tertulis dan 10% berasal dari konteks percakapan.

Selanjutnya, setiap kandidat kata rekomendasi akan dijadikan satu kalimat dan akan dihitung nilai probabilitasnya. Tahap awal untuk mencari nilai probabilitas kalimat ialah mencari jumlah frekuensi *unigram* dan nilai probabilitasnya pada korpus. Jika sebuah kalimat dinyatakan dengan $\{w_1, w_2, \dots, w_{i-1}, w_i\}$ dan i adalah jumlah kata dalam kalimat maka nilai probabilitas *unigram* dapat dihitung melalui persamaan berikut :

$$P(w_i) = \frac{N(w_i)}{N} \quad (1)$$

Dimana $P(w_i)$ adalah probabilitas *unigram* kata ke- i (w_i) pada korpus, $N(w_i)$ adalah jumlah kemunculan kata ke- i pada korpus, dan N adalah jumlah seluruh kata yang terdapat pada korpus. Langkah selanjutnya ialah mencari jumlah frekuensi *bigram* dan nilai probabilitasnya pada korpus. Nilai probabilitas pasangan *bigram* dapat dihitung menggunakan persamaan perkiraan kemungkinan maksimum (*maximum likelihood estimation*) berikut :

$$P(w_i | w_{i-1}) = \frac{N(w_{i-1}, w_i)}{N(w_{i-1})} \quad (2)$$

Dimana $P(w_i | w_{i-1})$ adalah probabilitas pasangan *bigram* kata ke- i dan kata ke ($i-1$), $N(w_{i-1}, w_i)$ adalah jumlah kemunculan pasangan *bigram* kata ke- i dan kata ke ($i-1$) pada korpus, dan $N(w_{i-1})$ adalah jumlah

kemunculan *unigram* kata ke $(i-1)$. Sedangkan nilai probabilitas sebuah kalimat dari model statistik *bigram* dapat dihitung menggunakan aturan Chain, aturan Chain dapat dijabarkan melalui persamaan berikut :

$$P(W_1^n) \approx \prod_{i=1}^n P(w_i | w_{i-1}) \quad (3)$$

Dimana $P(W_1^n)$ adalah probabilitas kalimat yang dihitung berdasarkan model statistik *bigram*, dan n adalah jumlah pasangan *bigram* kata dalam sebuah kalimat. Nilai probabilitas pasangan *trigram* kata pada sebuah kalimat dapat dihitung menggunakan persamaan kemungkinan maksimum berikut :

$$P(w_i | w_{i-2}, w_{i-1}) = \frac{N(w_{i-2}, w_{i-1}, w_i)}{N(w_{i-2}, w_{i-1})} \quad (4)$$

Dimana $P(w_i | w_{i-2}, w_{i-1})$ adalah probabilitas pasangan *trigram* kata ke- i , kata ke $(i-1)$ dan kata ke $(i-2)$, $N(w_{i-2}, w_{i-1}, w_i)$ adalah jumlah kemunculan pasangan *trigram* kata ke- i , kata ke $(i-1)$ dan kata ke $(i-2)$ pada korpus, dan $N(w_{i-2}, w_{i-1})$ adalah jumlah kemunculan pasangan *bigram* kata ke $(i-1)$ dan kata ke $(i-2)$. Adapun nilai probabilitas sebuah kalimat dari model statistik *trigram* dapat dihitung menggunakan persamaan kemungkinan maksimum berikut :

$$P(W_1^n) \approx \prod_{i=1}^n P(w_i | w_{i-2}, w_{i-1}) \quad (5)$$

Dimana $P(W_1^n)$ adalah probabilitas kalimat yang dihitung berdasarkan model statistik *trigram*, dan n adalah jumlah pasangan *trigram* kata dalam sebuah kalimat. Berdasarkan aturan Chain yang digunakan untuk menghitung nilai probabilitas kalimat dari model statistik *bigram* maka nilai *perplexity* dari probabilitas kalimat (W) dapat didefinisikan melalui persamaan berikut :

$$Perplexity(W) = \sqrt[n]{\prod_{i=1}^n \frac{1}{P(w_i | w_{i-1})}} \quad (6)$$

Sedangkan nilai *perplexity* dari probabilitas kalimat (W) dari model statistik *trigram* dapat didefinisikan sebagai berikut :

$$Perplexity(W) = \sqrt[n]{\prod_{i=1}^n \frac{1}{P(w_i | w_{i-2}, w_{i-1})}} \quad (7)$$

Setelah menghitung nilai probabilitas kalimat dari setiap kandidat kata yang direkomendasi, tahap selanjutnya ialah mengurutkan nilai probabilitas kalimat (*perplexity*) yang dihasilkan berdasarkan nilai terendah (minimum). Nilai *perplexity* yang minimum sama halnya dengan nilai probabilitas yang maksimum. Sehingga semakin rendah nilai *perplexity* yang diperoleh maka probabilitas kalimat yang dihasilkan akan semakin tinggi. Kandidat kata yang telah diurutkan berdasarkan nilai *perplexity* yang minimum akan ditampilkan sebagai kata rekomendasi perbaikan kata rancu yang lebih disarankan.

4. UJI COBA DAN ANALISIS

Dataset yang digunakan pada penelitian ini terdiri dari beberapa pernyataan kebutuhan yang rancu yang dikumpul dari beberapa penelitian sebelumnya yang membahas tentang kerancuan pada dokumen spesifikasi kebutuhan perangkat lunak. Dataset diambil dari berbagai domain permasalahan yang berbeda, dimana hal ini diharapkan dapat mengenali berbagai jenis kerancuan yang terjadi pada beberapa domain spesifikasi kebutuhan perangkat lunak. Adapun 12 dataset yang digunakan pada penelitian ini dapat dilihat pada Tabel 2 berikut.

Jumlah pernyataan kebutuhan perangkat lunak yang dikumpulkan dari 12 dataset yaitu berjumlah 647 pernyataan kebutuhan baik ambigu maupun tidak ambigu. Data pengujian yang digunakan pada penelitian ini berjumlah 100 pernyataan kebutuhan yang diambil dari 12 dataset yang merujuk pada Tabel 1. Untuk mengetahui kinerja metode yang diusulkan dapat dilakukan dengan cara membandingkan pengetahuan yang dimiliki oleh ahli dengan keluaran yang dihasilkan oleh metode rekomendasi. Ahli bertindak sebagai penguji pertama, sedangkan kakas bantu bertindak sebagai penguji kedua. Penelitian ini menggunakan teknik statistik Gwet's AC1 untuk mengukur nilai reliabilitas antara metode rekomendasi dan ahli (Gwet, 2002).

Tabel 2. Dataset Penelitian

No	Dataset	Sumber
1	ACM's OOPSLA DesignFest®	(Hussain, 2007)
2	Lift Controller System	(Bray, I.K, 2002)
3	Yacht Race Results (YRR)	(Bray, I.K, 2002)
4	Batch Poster System (BPS)	(Batch Poster System, 2005)
5	Cask Loader Software (CLS)	(CLS, 1999)
6	Data Cycle System (DCS)	(DCS, 2002)
7	EVLA Array Operations (EVLA)	(EVLA, 2003)
8	Large Area Telescope (LAT) Science Analysis Software (SAS) Level III Specification	(LAT, 2002)
9	PESA High-Level Trigger Selection	(PESA, 2001)
10	Sort Algorithm Demonstration Program (SAD)	(Stevenson <i>et al.</i> , 2005)
11	New Adelaide Airport System	(Tjong, 2008)
12	MCSS System	(Tjong, 2008)

Proses uji coba metode rekomendasi dilakukan berdasarkan tiga skenario uji coba yaitu uji coba metode rekomendasi menggunakan teknik berbasis aturan dan model bahasa *bigram*, uji coba metode rekomendasi menggunakan teknik berbasis aturan dan model bahasa *trigram* dan uji coba metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-gram*. Hasil penilaian ahli terhadap data

pengujian selanjutnya akan dibuat tabel kesepakatan antara ahli dan metode rekomendasi pada masing-masing skenario uji coba. Tabel hasil kesepakatan antara ahli dan kakas bantu pada tiga skenario uji coba dari 100 data pengujian yang digunakan pada penelitian ini dapat dilihat pada Tabel 3.

Hasil komparasi antara metode rekomendasi yang diusulkan pada penelitian ini yaitu teknik berbasis aturan menggunakan model bahasa *bigram* dan *trigram* dengan metode rekomendasi lainnya yaitu teknik statistik berbasis frekuensi *n-gram* menunjukkan bahwa metode yang diusulkan memiliki kinerja yang lebih baik dalam memberikan rekomendasi perbaikan pernyataan kebutuhan yang rancu. Hal ini dapat dilihat dari nilai indeks *Kappa* yang dihasilkan dari ketiga skenario hasil uji coba yaitu hasil uji coba metode rekomendasi menggunakan teknik berbasis aturan dan model bahasa *bigram* memiliki nilai indeks *Kappa* sebesar 0.4117 dengan proporsi kesepakatan sedang, hasil uji coba metode rekomendasi menggunakan teknik berbasis aturan dan model bahasa *trigram* memiliki nilai indeks *Kappa* sebesar 0.5263 dengan proporsi kesepakatan sedang, dan nilai indeks *Kappa* dari hasil uji coba metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-gram* yaitu sebesar -0.2844 dengan proporsi kesepakatan rendah.

Tabel 3. Hasil Penilaian Ahli Terhadap Hasil Keluaran Metode Rekomendasi

Penilaian ahli terhadap hasil keluaran metode rekomendasi	Skenario Uji Coba		
	Teknik Berbasis Aturan dan Model <i>Bigram</i>	Teknik Berbasis Aturan dan Model <i>Trigram</i>	Teknik Statistik Berbasis Frekuensi <i>N-Gram</i>
Setuju	60	66	30
Tidak Setuju	40	34	70
Indeks <i>Kappa AC1</i>	0.4117	0.5263	-0.2844

Salah satu faktor yang menyebabkan rendahnya proporsi kesepakatan antara ahli dengan metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-gram* ialah kurang akuratnya daftar kata yang direkomendasi untuk menggantikan kata rancu. Pada beberapa kasus uji coba terdapat hasil kata rekomendasi yang masih mengandung kerancuan, sehingga pilihan kata yang direkomendasi menjadi tidak valid untuk menggantikan kata rancu yang terdeteksi. Disisi lain, metode rekomendasi yang diusulkan pada penelitian ini telah mampu mengatasi permasalahan tersebut dengan menyediakan beberapa pilihan kata rekomendasi (daftar kata rekomendasi yang tidak rancu) dari kata rancu yang terdeteksi. Sehingga daftar pilihan kata yang direkomendasi bisa menjadi lebih valid dalam memberikan pilihan kata rekomendasi.

Berdasarkan hasil uji coba terhadap tiga skenario uji coba dapat diketahui bahwa metode rekomendasi menggunakan teknik berbasis aturan dan model bahasa *trigram* memiliki nilai indeks *Kappa* yang tertinggi yaitu sebesar 0.5263 diantara metode rekomendasi lainnya. Kelebihan metode rekomendasi menggunakan teknik berbasis aturan dengan model bahasa *trigram* ialah metode dapat memberikan urutan pilihan daftar kata rekomendasi yang lebih baik jika dibandingkan dengan metode rekomendasi menggunakan teknik berbasis aturan dengan model bahasa *bigram*. Akan tetapi, juga terdapat kelemahan dari metode teknik berbasis aturan dengan model bahasa *trigram*, dimana cara kerja pengurutan daftar kata hasil rekomendasi sangat bergantung terhadap frekuensi kata pada sebuah korpus. Pada suatu kasus sering dijumpai pasangan *trigram* kata yang tidak terdapat pada sebuah korpus (tidak memiliki frekuensi pada korpus) sehingga menghasilkan nilai probabilitas kalimat yang rendah dan daftar urutan kata rekomendasi menjadi tidak valid pada kasus tersebut.

5. KESIMPULAN

Berdasarkan hasil uji coba dan evaluasi yang telah dilakukan terhadap metode rekomendasi perbaikan pernyataan kebutuhan perangkat lunak yang rancu menggunakan teknik berbasis aturan dengan model bahasa *bigram* dan *trigram* maka dapat disimpulkan bahwa metode rekomendasi yang diusulkan dapat digunakan untuk memberikan rekomendasi perbaikan pernyataan kebutuhan perangkat lunak yang rancu dengan menyediakan pilihan kata rekomendasi yang lebih baik jika dibandingkan dengan metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-gram*. Hal ini dibuktikan dengan hasil uji coba metode rekomendasi yang diusulkan memiliki tingkat proporsi kesepakatan yang lebih baik jika dibandingkan dengan metode rekomendasi menggunakan teknik statistik berbasis frekuensi *n-grams*.

Hasil skenario uji coba menggunakan teknik berbasis aturan dan model bahasa *trigram* memiliki nilai indeks *Kappa* tertinggi, yaitu sebesar 0.5263, dengan proporsi kesepakatan sedang. Hal ini menunjukkan bahwa metode rekomendasi yang diusulkan memiliki performa yang cukup baik dalam memberikan rekomendasi perbaikan pernyataan kebutuhan perangkat lunak yang rancu.

DAFTAR PUSTAKA

- ATHANASELIS, T., MAMOURAS, K., BAKAMIDIS, S. dan DOLOGLOU, I., 2011. a Corpus Based Technique for Repairing Ill-Formed Sentences With Word Order Errors Using Co-Occurrences of N-Grams. *International Journal on Artificial*

- Intelligence Tools*, [daring] 20(3), hal.401–424.
- BATCH POSTER SYSTEM, 2005. Detailed Business Requirements. Technical report, EDS MySC, Malaysia
- BHATIA, M., KUMAR, A. BENIWAL, R., 2016. Ontology based Framework for Detecting Ambiguities in Software Requirement Specification. In: *2016 International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, hal.3572–3575.
- BRAY, I.K., 2002. An Introduction To Requirements Engineering, © Pearson Education Limited
- BUSSEL, D. VAN, 2009. *Detecting ambiguity in requirements specifications*. [daring] Tilburg University.
- CASK LOADER SOFTWARE, 1999. General Requirements Document Draft, © Electric Power Research Institute, Inc.
- GWET, K., 2002. Kappa Statistic is not satisfactory for assessing the extent of agreement between raters. *Statistical Methods For Inter-Rater Reliability Assessment*, [daring] (1), hal.1–5.
- HENRICH, V. & REUTER, T., 2009. *LISGrammarChecker: Language Independent Statistical Grammar Checking*. [daring] Hochschule Darmstadt & Reykjavik University.
- HUSSAIN, H.I., 2007. *Using text classification to automate ambiguity detection in SRS documents*. [daring] Concordia University.
- MULIAWAN, I.W., SIAHAAN, D.O., SARWOSARI, 2011. Analisis ambiguitas kebutuhan perangkat lunak berdasarkan acuan. In: *Prosiding Seminar Nasional Manajemen Teknologi XIV*. Surabaya: Program Studi MMT-ITS, Surabaya 23 Juli 2011, hal.C28(1-7).
- NABER, D., 2003. *A Rule-Based Style and Grammar Checker*. [daring] Bielefeld University.
- NELBACH, F.J., 2002. Software Requirements Document For the Data Cycle System (DCS) Of The SOFIA Project, Universities Space Research Association.
- PRESSMAN, R. S., 2008. Software Engineering, A Practioner Approach, Eight Edition, Mc Graw Hill
- R. DUBOIS, 2002. Gamma-ray Large Area Space Telescope (GLAST) Large area telescope (LAT) Science Analysis Software Level III Specification, Technical report, GE- 0000X-DO.
- R. MOESER, & P. PERLEY, 2003. Expanded Very Large Array (EVLA) operations interface, software requirements. Technical report, EVLA-SW-003 Revision: 2.5
- S. GEORGE, 2001. PESA high-level trigger selection software requirements. Technical report, Centre for Particle Physics at Royal Holloway University.
- SABRIYE, A. & ZAINON, W., 2017. A Framework For Detecting Ambiguity In Software Requirement Specification. In: *2017 8th International Conference on Information Technology (ICIT)*. Penang: IEEE, hal.209–213.
- SIAHAAN, D. O., 2012. Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak, Edisi 1. Yogyakarta : Andi
- SINGH, S.P., KUMAR, A., SINGH, L., BHARGAVA, M., GOYAL, K., SHARMA, B., 2016. Frequency based spell checking and rule based grammar checking. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, hal.4435–4439.
- STEVENSON, M, M. HARTLEY, H. IACOVOU, A. TAN, L. PHAN, 2005. Software Requirements Specification for Sort Algorithm Demonstration Program, Version 2.4, SDPM
- TJONG, S.F., 2008. *Avoiding ambiguity in requirements specifications*. *Faculty of Engineering & Computer Science*, .
- VIMALRAJ, T.J. & SEEMA, B., 2016. Identification of Ambiguity in Requirement Specification using Multilingual Word Sense. 5(6), hal.386–388.
- WU, J., CHANG, J. & CHANG, S.J., 2013. Correcting Serial Grammatical Errors based on N-grams and Syntax. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 18, Number 4, December 2013-Special Issue on Selected Papers from ROCLING XXV*, 18(4), hal.31–44.

Halaman ini sengaja dikosongkan