

## ***Literature Review Pada Teknik Pendeteksi Ambiguitas Leksikal dalam Software Requirements Specification***

**Iqbal Ramadhani Mukhlis**

*Program Studi Informatika, Fakultas Teknik dan Desain,  
Universitas Hayam Wuruk Perbanas  
Jl. Nginden Semolo No.34-36, Nginden Jangkungan, Kec. Sukolilo,  
Kota Surabaya, Jawa Timur  
[iqbal.ramadhani@perbanas.ac.id](mailto:iqbal.ramadhani@perbanas.ac.id)*

### ***Abstrak***

*Sejumlah studi mencatat bahwa presentase kumulatif kegagalan sebuah proses pengembangan perangkat lunak disebabkan oleh kegagalan mengantisipasi spesifikasi kebutuhan. Salah satu kegagalan tersebut adalah dengan adanya ambiguitas dalam dokumen Software Requirements Specification (SRS). Untuk itu proses pemenuhan kebutuhan perangkat lunak menjadi salah satu proses terpenting dalam pembuatan perangkat lunak. Mendeteksi ambiguitas dari software requirements specification dengan cara manual merupakan cara yang cenderung menghasilkan banyak error, banyak memakan waktu, dan membutuhkan banyak biaya. Ada banyak teknik dan pendekatan yang dapat digunakan untuk mendeteksi ambiguitas, tentu saja teknik tersebut memiliki alur yang berbeda satu dengan lainnya. Pada penelitian ini, penulis mengangkat topik tentang teknik pendeteksi ambiguitas leksikal dalam dokumen Software Requirements Specifications (SRS). Peneliti mengkaji jurnal - jurnal dari penelitian terdahulu untuk merujuk pada suatu kesimpulan tentang teknik-teknik yang digunakan dalam mendeteksi ambiguitas leksikal. Pada penelitian ini mengacu pada beberapa sudut pandang, yaitu banyaknya tahapan dan step yang digunakan pada setiap teknik, faktor yang berperan dalam pengimplementasian teknik tersebut dan juga tingkat keakuratan dari hasil teknik yang digunakan. Hasil dari penelitian ini menyatakan bahwa teknik Natural language Processing merupakan teknik yang paling akurat digunakan dalam pendeteksi ambiguitas leksikal pada dokumen SRS dalam pemenuhan kebutuhan perangkat lunak.*

***Kata Kunci***— Ambiguitas leksikal, Dokumen SRS, Ontologi, Natural language Processing

### ***Abstract***

*A number of studies note that the cumulative percentage of failure of a software development process is caused by failure to anticipate requirements specifications. One of these failures is the ambiguity in the Software Requirements Specification (SRS) document. For this reason, the process of meeting software requirements is one of the most important processes in making software. Detecting ambiguity of software requirements specifications manually is a method that tends to produce a lot of errors, is time consuming, and requires a lot of money. There are many techniques and approaches that can be used to detect ambiguity, of course these techniques have different paths from one another. In this study, the author raised the topic of Lexical ambiguity detection techniques in the Software Requirements Specifications (SRS) document. The researcher reviewed the journals of previous research to refer to a conclusion about the techniques used in detecting Lexical ambiguity. This study refers to several points of view, namely the number of stages and steps used in each technique, the factors that play a role in the implementation of the technique and also the level of accuracy of the results of the techniques used. The results of this study indicate that the Natural language Processing technique is the most accurate*

*technique used in detecting Lexical ambiguity in SRS documents in meeting software requirements.*

**Keywords**— *Lexical ambiguity, SRS Document, ontologi, Natural language Processing*

## 1. PENDAHULUAN

Ambiguitas atau disebut dengan ketaksaan adalah kondisi yang muncul dimana pendengar atau pembaca sulit untuk mengerti dan menangkap pengertian yang dibaca atau yang didengar. Jerrold J Katz (2008) mengatakan “The phenomenon of semantic ambiguity, that is, multiplicity of senses versus uniqueness of sense-for example, the fact that the words “button”, “ball”, “foot”, “pipe” have more than one sense and the fact that the sentences “There’s no school anymore”, “I’ve found the button”, “Take your pick” have two or more senses” (Parera, 2004). Ambiguitas dapat terjadi dalam kajian ilmu sintaksis, semantik dan leksikal. Ambiguitas sintaksis adalah ambiguitas yang berhubungan dengan struktur kalimat (Grammar), sedangkan ambiguitas semantik adalah ambiguitas yang berhubungan dengan ilmu semantik, ambiguitas ini sering disebut dengan ambiguitas leksikal. Ambiguitas semantik terjadi ketika sebuah kata atau konsep mempunyai sifat atau pemakaian makna berdasarkan pada pemakaian kata-kata yang meluas atau kata-kata yang tidak resmi. Menurut Kempson yang dikutip oleh Ullmann menyebutkan ada tiga jenis ambiguitas, yaitu ketaksaan fonetik, yaitu ambiguitas pada tataran fonologi (fonetik), muncul akibat berbaurnya bunyi-bunyi bahasa yang dilafalkan terlalu cepat dan mengakibatkan keragu-raguan (Ullmann, 1967). Jenis ambiguitas selanjutnya adalah ambiguitas leksikal. Ambiguitas leksikal adalah kata yang bermakna lebih dari satu, ambiguitas ini muncul juga karena memiliki homonim, homonim adalah kata-kata yang sama bunyinya, namun berbeda maknanya.

Ambiguitas juga banyak ditemui pada saat penyusunan dokumen SRS, yaitu pada fase pengumpulan kebutuhan pada *software requirements*. Fase yang penting pada saat pembuatan SRS (*Software Requirements Specification*) adalah mengerti apa kebutuhan yang dibutuhkan oleh *user*, dan *developer* mengartikan kebutuhan tersebut dengan tepat. Terdapatnya ambiguitas dalam *software requirements* memiliki dampak yang besar untuk desain, sistem yang akan dibangun dan hasil akhir dari *software* tersebut. Ambiguitas yang sering muncul pada *software requirements* adalah ambiguitas jenis leksikal atau biasa disebut dengan *lexical ambiguity*. *Lexical ambiguity* adalah ketika kata-kata dalam pemenuhan kebutuhan perangkat lunak memiliki lebih dari satu arti atau bisa disebut dengan suatu kata yang memiliki banyak arti. Hal yang perlu dilakukan untuk mengatasi ambiguitas tersebut adalah dengan mencari cara untuk dapat mendeteksi. Mendeteksi ambiguitas dari *software requirements* dengan cara manual merupakan cara yang cukup membosankan, banyak memakan waktu, cenderung menghasilkan banyak *error*, dan membutuhkan banyak biaya. Ada banyak teknik dan pendekatan yang dapat digunakan untuk mendeteksi ambiguitas, tentu saja teknik tersebut memiliki alur yang berbeda satu dengan lainnya, artinya masing-masing teknik memiliki batasan tersendiri untuk mengatasi jenis ambiguitas tertentu.

Melihat dari permasalahan mengenai ambiguitas yang sering ditemui pada saat proses pembuatan dokumen SRS tersebut, pada *literature review paper* ini akan dipaparkan mengenai teknik-teknik dalam mendeteksi ambiguitas jenis leksikal dalam *software requirements*. Adapun menurut Geet Sandhu et. Al. (Sandhu & Sikka, 2015), pendekatan yang dapat digunakan untuk mendeteksi ambiguitas jenis umum dalam *software requirements* adalah teknik dengan konsep inti dari ontologi, teknik dengan konsep inti dari *Natural language Processing* dan *Object Oriented Analysis*, dan teknik dengan konsep inti dari *Natural language Processing*. Menurut Unnati S.Shah et. Al. (2015) pendekatan yang

dapat digunakan untuk mendeteksi ambiguitas adalah dengan menggunakan teknik *Natural language Processing*, *Checklist Based Inspection Approach*, *Style Guides*, *Controlled Language*, *Knowledge Based Approach*, dan *Heuristic Based Approach*. Namun tentu tidak semua pendekatan tersebut dapat mendeteksi ambiguitas jenis leksikal.

Dalam penulisan *literature review* ini penulis mencoba untuk membandingkan metode yang digunakan dalam mendeteksi ambiguitas leksikal dalam *software requirements specification*. Metode yang digunakan adalah metode berbasis ontologi, *Natural language Processing* dan satu metode yang digunakan adalah metode yang diadopsi dari penelitian Md. Rizwan Beg, Dr. Qamar Abbas, Alok Joshi (2008) dengan *paper* yang berjudul “A Method to Deal with the Type of *Lexical ambiguity* in a *Software Requirement Specification Document*”. Pada *paper* tersebut dipaparkan mengenai 2 fase yang memiliki 8 tahapan dalam mendeteksi ambiguitas leksikal. *Literature review* kali ini akan membandingkan metode yang satu dengan lainnya berdasarkan hasil penelitian dari masing-masing metode pada studi kasus yang terdahulu.

Penulisan *literature review* ini bertujuan untuk mengkaji beberapa *paper* yang terkait dengan leksikal ambiguitas dan metode yang digunakan dalam mendeteksinya. Sehingga hasil akhir yang diharapkan dari adanya pengkajian ini adalah dapat diambil kesimpulan untuk metode atau teknik yang paling tepat dalam mendeteksi ambiguitas leksikal. Selain itu, pengkajian ini juga bertujuan untuk dapat meningkatkan performa efektifitas dan efisiensi dalam penyusunan dokumen *Software Requirements Specification (SRS)*. Kontribusi yang diberikan oleh penulis adalah untuk mendukung atau menguatkan persepsi dari penelitian-penelitian yang telah dilakukan sebelumnya.

Adapun metode yang akan dikaji pada *literature review* ini adalah metode ontologi based Framework yang diperoleh dari hasil penelitian MPS Bhatia. et.al., metode *Natural language Processing (NLP)* yang diperoleh dari hasil penelitian Sandhu. et.al., dan metode yang diperoleh dari hasil penelitian Md. Rizwan Beg. et. al.

### 1.1. Metode Ontology Based Framework

Menurut Bhatia, M.P.S., et.al., pada penelitian sebelumnya (Bhatia, Kumar, Beniwal, 2016), metode ontologi adalah sebuah kerangka kerja serta implemetasi dari pendekatan yang digunakan untuk mendeteksi ambiguitas dalam dokumen SRS dengan menggunakan aplikasi ontologi domain yang dibantu oleh *Natural language Processing (NLP)* dan teknik *Semantic Web*. Pada penelitian Bhatia, M.P.S., et.al (Bhatia, Kumar, Beniwal, 2016) metode *ontology based framework* ini terdiri dari dua fase, yaitu fase pengembangan (*development phase*) dan fase penalaran (*reasoning phase*). Berikut ini adalah penjelasan dari masing-masing fase tersebut.

#### 1.1.1 Fase Pengembangan (Development Phase)

Pada fase pengembangan, alur prosesnya adalah dengan mengembangkan aplikasi ontologi domain yang mewakili semua pengetahuan domain dengan mengambil SRS sebagai *input*. Proses pengembangan ontologi dengan otomatis menggunakan teknik *Natural language Processing (NLP)* untuk mengetahui syarat dan sinonim multibahasa. Kemudian pada tahap selanjutnya, aplikasi ontologi domain akan mendeskripsikan berbagai hubungan yang akan muncul antara konsep berbeda yang berkaitan dengan domain. Konsep ini mewakili model abstrak dari domain. Lebih lanjut, aplikasi ontologi domain akan memiliki deskripsi konsep, hirarki konsep, hubungan antara konsep yang berbeda, hirarki dari hubungan, skema aksiomatik dan aksioma umum. Aksioma dideskripsikan dengan pengetahuan yang tidak dapat diungkapkan dengan mudah hanya dengan bantuan komponen yang sudah ada. Setelah memiliki aksioma umum, itu berarti spesifikasi kebutuhan telah diubah menjadi definisi formal (bentuk yang dimengerti oleh mesin) dalam aplikasi ontologi domain.

### 1.1.2 Fase Penalaran (*Reasoning Phase*)

Aplikasi ontologi domain dibangun pada fase lebih awal, kemudian akan digunakan oleh penalaran semantik untuk menyimpulkan kesimpulan logis dari aksioma. Lalu peneliti akan memiliki semua ketersediaan informasi dalam bentuk yang mudah dimengerti oleh mesin, yang mana nantinya akan dimiliki satu-satunya interpretasi untuk setiap kebutuhan, sehingga dapat menghilangkan ambiguitas dalam SRS.

Menurut Bhatia, M.P.S. (Bhatia, Kumar, Beniwal, 2016), Teknik mendeteksi ambiguitas leksikal menggunakan pendekatan ontologi tidak sepenuhnya otomatis dan membutuhkan intervensi manusia. Oleh karena itu, tingkat integritas dalam spesifikasi kebutuhan perangkat lunak juga tergantung pada domain. *Framework* ontologi memudahkan menghilangkan ambiguitas yang signifikan yaitu ambiguitas pragmatis, ambiguitas semantik, *vagueness* dan *generality*, serta ambiguitas *language error* untuk meningkatkan spesifikasi kebutuhan yang membantu dalam pengembangan perangkat lunak yang berkualitas.

### 1.2. Metode NLP (*Natural Language Processing*)

Pada jurnal yang ditulis oleh Sandhu et.al. (Sandhu & Sikka, 2015), Popescu et.al., (Popescu, et.al, 2007) mempresentasikan pendekatan dengan 3 tahapan yang didukung dengan *prototype tool* yang bernama Dowser untuk mengidentifikasi ambiguitas leksikal dalam *natural language* yang tertulis pada dokumen SRS. Tahap pertama *tool* melakukan penguraian (*parsing*) kalimat pada *natural language* SRS dengan menggunakan tata bahasa yang dibatasi. Kemudian pada tahap kedua, kelas, metode, variabel dan asosiasi dari model analisis berorientasi objek dari sistem yang ditentukan dibuat berdasarkan hubungan yang dipaparkan oleh parser. Pada tahap ketiga, representasi diagram dari model yang diperoleh disiapkan untuk diperiksa inkonsistensi dan ambiguitasnya oleh seorang *reviewer*.

Penelitian dari Kiyavitskaya.,et.al. (Kiyavitskaya, et.al, 2007) yang dikutip oleh Sandhu, G., et.al. (Sandhu & Sikka, 2015) mempresentasikan pendekatan yang berbasis *2-step tool* atau berbasis *tools* dengan 2 tahapan untuk mengidentifikasi dan mengukur ambiguitas leksikal dalam NL dokumen SRS. Sandhu membuat nama permisalan dari dua tahap *tools* tersebut dengan tahap dengan *tool* 1 disebut sebagai T1, dan tahap dengan *tool* kedua disebut sebagai T2. *Tool* T1 merupakan *prototype* yang dibuat oleh *shell scripts* yang memanggil perintah yang berbasiskan kajian umum. Kajian umum tersebut akan menganalisis dokumen SRS secara leksikal, morfologis, sintaksis dan semantik, yang mana menghasilkan grafik sebagai keluaran, lalu grafik ditambahkan ke sistem jaringan semantik. Tahap ini atau T1 memberikan tindakan untuk mengidentifikasi pernyataan yang ambigu. *Tool* T2 merupakan *prototype* untuk mengidentifikasi contoh dalam kalimat di mana ambiguitas terjadi dari hasil yang diperoleh dari T1. T2 mengidentifikasi semua uraian kalimat yang ada dan semua kemungkinan atau makna dari kata tertentu. Proses penguraian kalimat tersebut juga sesuai dengan persyaratan apakah kalimat tersebut merupakan kalimat yang mengandung kata ambigu atau tidak. Namun keputusan terakhir tetap ada pada *human expert* atau ahlinya (Castaneda, et.al., 2010).

### 1.3. Metode yang Diadopsi Md.Ridzwan Beg, Dr. Qamar Abbas, Alok Joshi

Menurut Md. Ridzwan Beg.,et.al [6] dalam ambiguitas leksikal, sebuah kata memiliki beberapa arti. Hal ini dapat dibagi menjadi beberapa bagian sesuai dengan ejaan, pengucapan dan asal katanya; yang mana sesuai dengan kategori homonim, heteronym, homograf dan homofon. Namun penulis tidak mempertimbangkan homofon dalam penelitiannya. Metode yang digunakan terbagi menjadi dua fase, fase pertama yaitu memeriksa apakah kebutuhan yang tertulis sudah valid, kemudian fase kedua adalah memeriksa apakah terdapat ambiguitas leksikal dalam kebutuhan tertulis tersebut.

Untuk kedua fase tersebut, telah dirancang sebuah algoritma tunggal, yang mana diasumsikan bahwa kebutuhan tertulis dalam dokumen SRS merupakan *natural language* (bahasa alami) dan kebutuhan tersebut tidak terlalu panjang. Dibuat dalam format satu baris dan kalimatnya adalah kalimat yang bertipe deklaratif. Fase ini memiliki 8 tahapan yang akan dipaparkan secara detail untuk mendeteksi ambiguitas leksikal dalam dokumen SRS.

Md. Rizwan Beg et.al. juga menggunakan kamus dan membandingkan kata-kata yang tertulis dalam satu baris dengan menggunakan kamus dan mengekstrak bagian-bagian dari bahasa yang selanjutnya disimpan dalam struktur data tertentu. Modul untuk pola kata kerja dirancang sesuai dengan pola yang diucapkan atau ditulis dalam *natural language*. Setelah mengidentifikasi pola kata kerja dalam kebutuhan tersebut, penggabungan dan perbandingan dengan pola yang tersimpan sudah selesai. Jika struktur data sama dengan modul yang telah dibuat sebelumnya, maka kebutuhan pada dokumen SRS sudah terbilang valid. Selanjutnya dilakukan pemeriksaan kebutuhan untuk mendeteksi ambiguitas dan melihat per satu kalimat, apakah ada kata yang memiliki lebih dari satu kategori kata dari kata yang sudah tersimpan sebelumnya dalam struktur data, sehingga hasilnya dapat berupa kalimat ambigu atau kalimat yang formatnya tidak tepat dengan arti yang tepat. Setelah menerapkan dua fase tersebut, maka didapatkan kalimat yang di dalamnya terdapat ambiguitas leksikal atau tanpa ambiguitas.

Berikut ini adalah ke delapan tahapan dalam fase yang telah dijelaskan sebelumnya.

1. Tahap 1, ambil kalimat pertama dari dokumen SRS dan simpan kalimat tersebut sebagai *string* pertama dalam struktur data, simpan juga kalimat yang tersisa sebagai *string*.
2. Tahap 2, *scan string* satu per satu dan ekstrak kata-kata dari *string* tersebut dan disimpan dalam struktur data yang lain, lanjutkan tahap ini untuk semua kalimat.
3. Tahap 3, bandingkan masing-masing kata dari kalimat dengan kamus Bahasa Inggris dan ekstrak informasi, lalu isikan informasi tersebut ke dalam struktur data. Pada tahap 3 ini dibuat dalam bentuk tabel.
4. Tahap 4, simpan pola kata pertama untuk kebutuhan sebagai kalimat deklaratif dalam struktur data. Pada tahap 4 ini juga dibuat dalam bentuk tabel yang berisi struktur yang memungkinkan dari suatu kalimat yang diberikan pada SRS sesuai dengan pola kata kerja dalam Bahasa Inggris.
5. Tahap 5, untuk semua kata dalam kalimat, carilah atribut dari pola kata kerja seperti S,V,C, *adjective, preposition, preposition object* mengikuti aturan dari *present* atau *past participle*.
6. Tahap 6, untuk semua kata dalam kalimat, gabungkan nilai dari atribut dan simpan pola dalam struktur data yang berbeda.
7. Tahap 7, bandingkan struktur data yang diperoleh pada tahap 6 dengan semua struktur yang diinginkan pada tabel yang telah dibuat pada tahap 4. Jika itu termasuk ke salah satu struktur, maka kalimat yang diberikan dalam SRS sudah dalam format yang benar. Lanjutkan ke tahap 8 untuk memeriksa kalimat ambiguitas leksikal lain dalam SRS.
8. Tahap 8, untuk semua kalimat dan kata dari kalimat tertentu, jika kata yang didapat lebih dari satu kata dari *speech* seperti yang ditunjukkan pada tabel di tahap 3, maka kalimat tersebut memiliki tipe ambiguitas leksikal, terutama yang berkaitan dengan heteronim, homonim, homograf, periksa juga ke tipe ambiguitas lain.

## 2. METODE PENELITIAN

Pada bagian ini akan dipaparkan mengenai tahapan yang dilakukan dalam pengkajian *paper*. Adapun tahap-tahap dari pengkajian ini mencakup :

### 2.1. Ambiguitas Leksikal

Berdasarkan literatur yang telah direview, ambiguitas merupakan hal yang paling sering ditemui, khususnya ambiguitas leksikal pada dokumen SRS (*Software Requirement Specification*). Ambiguitas leksikal adalah ambiguitas yang berupa sebuah kata dalam suatu kalimat yang memiliki lebih dari satu arti (Beg, et.al., 2008) Terkadang adanya ambiguitas dalam dokumen SRS sering dianggap hal sepele, tanpa disadari hal tersebut sangat berpengaruh dan berdampak besar pada proses pemenuhan kebutuhan perangkat lunak hingga hasil akhir dari *software* yang dibuat. Metode yang digunakan dalam mendeteksi ambiguitas pada dokumen SRS pun sangat beragam. Sejauh ini, banyak metode yang digunakan untuk mendeteksi ambiguitas namun hanya secara umum saja dan belum mengerucut pada ambiguitas jenis leksikal, padahal ambiguitas jenis ini adalah ambiguitas yang paling banyak ditemukan dalam SRS. Melihat dari permasalahan tersebut, tujuan dari pengkajian *paper* ini adalah untuk membahas mengenai teknik pendeteksi ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak dari beberapa metode yang dikaji, adapun metode yang dikaji pada *paper* ini yaitu metode ontologi, metode *Natural language Processing* (NLP), dan metode yang diadopsi dari penelitian Md. Ridzwan Beg, Dr. Qamar Abbas, Alok Joshi.

### 2.2. Mencari Literatur

Pada proses pengumpulan literatur untuk pengkajian *paper* ini, sebagian besar literatur yang didapat diperoleh dari penyedia jurnal ilmiah yang ditunjukkan dalam tabel 1 berikut ini.

**Tabel 1.** Sumber Data Literatur

Sumber	Url
IEEE	ieeexplore.ieee.org
ACM	acm.org
IJARCSSE	ijarcsse.com
Science Direct	sciencedirect.com

Penyedia jurnal ilmiah tersebut menyediakan hampir semua makalah dan hasil dari konferensi. Pencarian literatur dan ulasan dilakukan berdasarkan kata kunci yang telah disesuaikan dengan topik permasalahan yang diangkat pada pengkajian *paper* ini. Pengkajian ini menggunakan kata kunci atau *keyword* berupa "*Lexical ambiguity in software requirement*". Hasil pencarian yang di dapat dengan menggunakan keyword tersebut ditunjukkan pada tabel 2 berikut.

**Tabel 2.** Hasil Pencarian Jurnal

Sumber	Url	Hasil
IEEE	ieeexplore.ieee.org	10 buah
ACM	acm.org	2 buah
Science Direct	sciencedirect.com	7 buah

### 2.3. Pengukuran Kualitas Makalah

Beberapa pertanyaan diusulkan dalam pengkajian ini guna mengukur literatur yang dirujuk. Pertanyaan yang menjadi acuan dalam proses pengukuran kualitas makalah dapat dilihat pada tabel 3 berikut.

**Tabel 3.** Tabel Pertanyaan Pengukuran Kualitas Jurnal

No	Pertanyaan Uji Kualitas	Nilai
Q1	Makalah membahas tentang ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	1
	Makalah membahas tentang ambiguitas (ambiguitas leksikal	0,5

No	Pertanyaan Uji Kualitas	Nilai
	termasuk di dalamnya) dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	
	Makalah tidak membahas tentang ambiguitas dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	0
Q2	Makalah membahas tentang metode dan tahapan ( <i>step by step</i> ) yang digunakan untuk mendeteksi ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	1
	Makalah membahas tentang metode yang digunakan untuk mendeteksi ambiguitas dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	0,5
	Makalah tidak membahas metode maupun tahapan untuk mendeteksi ambiguitas dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	0
Q3	Hasil penelitian menunjukkan hasil yang rinci dari metode yang digunakan untuk mendeteksi ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	1
	Hasil penelitian menunjukkan hasil secara garis besar dari metode yang digunakan untuk mendeteksi ambiguitas dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	0,5
	Hasil penelitian tidak menunjukkan hasil dari metode yang digunakan untuk mendeteksi ambiguitas dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak.	0

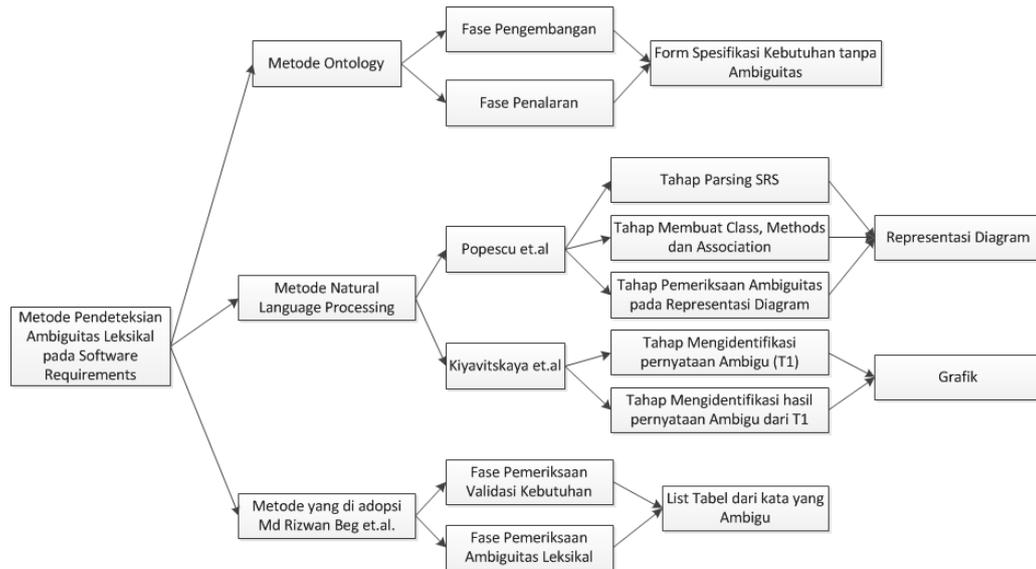
Makalah yang dipilih untuk dijadikan literatur berdasarkan pada topik dan metode yang digunakan untuk mendeteksi ambiguitas leksikal dalam dokumen SRS pada proses kebutuhan perangkat lunak Rentang nilai yang diberikan untuk setiap makalah sebesar 0 sampai dengan 1.

#### 2.4. Analisa

Pada tahap analisa, pengkajian ini memetakan hasil kajian jurnal yang sesuai dengan topik yang diangkat.

Pada gambar 1 ditampilkan skema hasil dari kajian makalah yang telah direview. Skema tersebut menjelaskan bahwa ada 3 teknik atau metode yang dapat digunakan untuk mendeteksi ambiguitas leksikal pada SRS. Metode tersebut adalah metode ontologi, metode *Natural language Processing* dan metode yang diadopsi oleh Md Rizwan Beg et.al. Masing-masing metode memiliki tahap/fase yang berbeda, begitupun dengan output yang dihasilkan. Metode ontologi memiliki 2 fase, dan hasil akhir dari metode ontologi ini berupa form spesifikasi kebutuhan yang sudah bebas dari ambiguitas.

Pada metode *Natural language Processing*, didapatkan 2 pendekatan berbasis NLP yang dapat digunakan untuk mendeteksi ambiguitas leksikal. Pendekatan pertama didapatkan dari hasil penelitian Popescu et.al yang memperkenalkan 3 tahap dalam mendeteksi ambiguitas. Hasil akhir dari pendekatan ini berupa representasi diagram. Pendekatan selanjutnya adalah hasil dari penelitian yang dilakukan oleh Kiyavitskaya et.al, yang mana pendekatan ini terdiri dari 2 tahap dan hasil akhir dari pendekatan ini berupa grafik.



**Gambar 1.** Skema Pemetaan Hasil Kajian

Pada metode yang diadopsi oleh Md Rizwan Beg et.al., dikatakan jika terdapat 2 fase dalam mendeteksi leksikal ambiguitas. Kedua fase tersebut terdiri dari 8 tahapan dan hasil akhir dari metode ini berupa list tabel yang didalamnya berisi kata-kata yang mengandung leksikal ambiguitas.

### 3. HASIL DAN PEMBAHASAN

Berdasarkan kajian yang telah dilakukan, dihasilkan sebuah pemahaman mengenai teknik yang dapat digunakan untuk mendeteksi ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak. Pada tabel 4 dijelaskan hasil pengukuran kualitas dari makalah yang diperoleh.

**Tabel 4.** Tabel Hasil Perhitungan Pengukuran Kualitas Makalah

Jurnal	Nama Pengarang	Tahun	Q1	Q2	Q3	Rata - Rata
1	MPS Bhatia, Akshi Kumar, Rohit Beniwal	2016	1	1	1	1.00
2	Md. Rizwan Beg, Dr. Qamar Abbas, Alok Joshi	2008	1	1	1	1.00
3	Robert B. Rowen	1990	0.5	1	1	0.83
4	Kanwal Daud Gill, Athar Mohsin Zaidi, Arif Raza, Muhammad Muneeb Kiani	2014	0.5	1	1	0.83
5	Neil Maiden	2012	0.5	1	1	0.83
6	Castaneda, Ballejos, Caliusco, Galli	2010	1	1	0.5	0.83
7	Shaokun Fan, Zhimin Hua, Veda C. Storey, Leon Zhao	2016	0.5	1	1	0.83
8	Unnati S.Shah, Devesh C.Jinwala	2015	0.5	1	1	0.83
9	Miles Osborne and C.K. MacNish	1996	0	1	1	0.67
10	Popescu, Rugaber, Medvidovic, Berry	2007	0.5	1	0.5	0.67
11	Geet Sandhu, Dr. Sunil Sikka	2015	0.5	1	0.5	0.67
12	Erik Kamsties, Daniel M. Berry,	2001	1	0.5	0.5	0.67

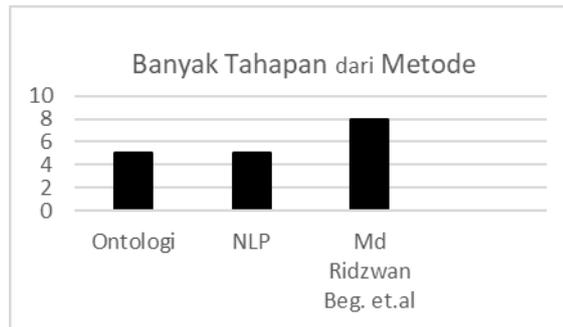
Jurnal	Nama Pengarang	Tahun	Q1	Q2	Q3	Rata - Rata
	Barbara Paech					
13	Ashfa UMBER, Imran Sarwar Bajwa	2011	0	1	1	0.67
14	Nadzeya Kiyavitskaya, Nicola Zeni, Luisa Mich, Daniel M. Berry	2007	0.5	0.5	0.5	0.50
15	Hazlina Haron, Abdul Azim Abdul Ghani, Hazliza Haron	2015	1	0.5	0	0.50
16	Fabian Pittke, Henrik Leopold, Jan Mendling	2015	1	0.5	0	0.50
17	Shilpi Singh	2015	0.5	0.5	0.5	0.50
18	Christina Unger, Phillipp Cimiano	2013	0.5	0.5	0	0.33
19	Omar Al-Harbi, Shaidah Jusoh, Norita Norwawi	2012	0.5	0	0	0.17
<b>Rata-Rata</b>			0.60	0.78	0.63	

Dari hasil dari perhitungan uji kualitas makalah berdasarkan pertanyaan yang sudah ditetapkan, penulis memilih jurnal yang memiliki nilai rata-rata dari range 0.67 – 1. Jurnal yang memiliki nilai rata-rata dibawah *range* yang telah ditentukan, tidak dikaji lebih dalam pada *literature review* kali ini. Hasil dari perhitungan menyatakan bahwa sebagian besar makalah membahas tentang metode dan tahapan (*step by step*) yang digunakan untuk mendeteksi ambiguitas leksikal dalam dokumen SRS pada proses pemenuhan kebutuhan perangkat lunak. Hal ini dibuktikan dengan perolehan perhitungan rata-rata nilai kualitas makalah dari ketiga pertanyaan diatas menghasilkan nilai Q2 yang paling tinggi, yaitu sebesar 0.78.

Pada pembahasan selanjutnya akan dihitung tingkat keakuratan dari masing-masing metode, yang mana dalam perhitungan ini digunakan 2 dasar penilaian yang dapat dijadikan tolak ukur dalam menghitung tingkat keakuratan dari masing-masing metode. Dasar penilaian itu adalah banyak tahapan dari suatu metode dan banyaknya faktor yang berperan serta dalam metode. Penulis memilih tolak ukur penilaian dari banyak tahapan dan dari faktor yang berperan dalam metode tersebut, karena dari semua makalah yang telah di kaji, sebagian besar metode yang memiliki hasil pendeteksian yang akurat (di hasil akhir tidak ada ambiguitas sama sekali) adalah metode yang memiliki banyak tahapan dalam mengidentifikasi dokumen SRS dan juga memiliki banyak elemen/faktor yang ikut berperan dalam metode tersebut. Elemen atau faktor yang dimaksud adalah adanya bantuan dari aplikasi, *tools*, mesin, dan juga dari *human expert* atau seorang ahli.

Jadi pada perhitungan tingkat keakuratan kali ini penulis menjumlahkan berapa banyak tahap yang ada pada masing-masing metode, lalu menjumlahkannya dengan berapa banyak faktor yang ikut berperan dalam masing-masing metode dan hasil akhirnya adalah rata-rata dari jumlah kedua tolak ukur tersebut. Hasil dari rata-rata tersebut yang akan menunjukkan tingkat keakuratan dari masing-masing metode. Berikut ini adalah perhitungan dari masing-masing tahapan, elemen serta tingkat keakuratan dari masing-masing metode yang dikaji.

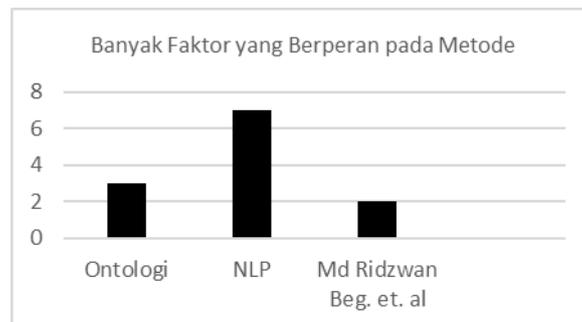
### 3.1. Perhitungan Tahapan dari Masing-masing Metode



**Gambar 2.** Grafik Perhitungan Banyak Tahapan dari Metode

Pada gambar 2 ditunjukkan hasil perhitungan mengenai banyaknya tahapan dari metode ontologi, NLP, dan Md Ridzwan Beg et.al. Metode ontologi memiliki 2 fase, yaitu fase pengembangan dan penalaran. Fase pengembangan memiliki 3 tahap didalamnya dan fase penalaran memiliki 2 tahap didalamnya, jadi banyaknya tahapan dalam metode ontologi adalah 5 tahapan. Metode NLP memiliki 2 jenis pendekatan, yaitu pendekatan hasil penelitian dari Popescu et.al., dan Kiyavitskaya et.al. pendekatan yang diteliti oleh Popescu et.al. memiliki 3 tahap di dalamnya dan pendekatan yang diteliti oleh Kiyavitskaya et.al. memiliki 2 tahap di dalamnya. Jadi banyaknya tahapan dalam metode NLP adalah 5 tahapan. Metode terakhir, yaitu metode yg diadopsi dari Md Ridzwan Beg et. al memiliki 2 fase, yang mana dalam 2 fase tersebut terdapat 8 tahapan. Jadi dapat disimpulkan dari hasil perhitungan banyak tahapan yang dimiliki oleh masing-masing metode tersebut, tahapan yang paling banyak terdapat pada metode yang diadopsi dari Md Ridzwan Beg et. al, yaitu 8 tahapan.

### 3.2. Perhitungan Faktor yang Berperan dari Masing-masing Metode

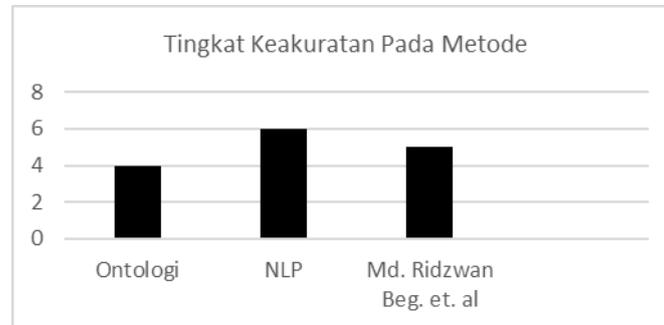


**Gambar 3.** Grafik perhitungan banyak faktor yang berperan pada metode

Pada gambar 3 ditunjukkan hasil perhitungan mengenai banyaknya faktor/elemen yang berperan dalam proses pendeteksian pada masing-masing metode. Pada metode ontologi, faktor yang berperan dalam mendeteksi ambiguitas terdiri dari 3 elemen, yaitu teknik NLP, aplikasi ontologi domain, dan pemeriksaan akhir oleh seorang ahli. Sedangkan pada metode NLP, faktor yang berperan dalam mendeteksi ambiguitas terdiri oleh 7 elemen, yaitu *prototype tool* yang bernama *dowser*, seorang *parser*, seorang *reviewer*, *prototype shell scripts*, sistem jaringan semantik, *prototype tool* untuk mengidentifikasi ambiguitas dan seorang *human expert* atau ahli. tingkat keakuratan dari metode *Natural language Processing* (NLP). Pada metode yang diadopsi oleh Md Ridzwan et. al., faktor yang berperan dalam mendeteksi ambiguitas terdiri dari 2 elemen, yaitu menggunakan algoritma tunggal yang diciptakan sendiri oleh peneliti dan menggunakan bantuan kamus Bahasa Inggris. Metode ini masih terbilang metode yang manual, karena tidak memerlukan bantuan

aplikasi atau *tool* khusus. Jadi dapat disimpulkan dari hasil perhitungan banyak faktor/elemen yang berperan pada masing-masing metode tersebut, elemen yang paling banyak berperan terdapat pada metode yang NLP, yaitu terdapat 7 elemen.

### 3.3. Perhitungan Keakuratan dari Masing-masing Metode



**Gambar 4.** Grafik Perhitungan Tingkat Keakuratan Pada Metode

Pada gambar 4 ditunjukkan hasil perhitungan mengenai tingkat keakuratan dari masing-masing metode yang telah dikaji. Tingkat keakuratan dari metode ontologi menghasilkan nilai 4, pada metode NLP menghasilkan nilai 6, dan pada metode yang diadopsi pada Md Ridzwan Beg et. al., menghasilkan nilai 5. Jadi dapat disimpulkan dari hasil perhitungan tingkat keakuratan, metode yang memiliki tingkat keakuratan paling tinggi adalah metode NLP.

### 3.4. Pembahasan

Pembahasan kali ini akan menjelaskan mengenai kesimpulan dari hasil pengukuran tahap, faktor yang berperan dan tingkat keakuratan yang dimiliki oleh metode ontologi, NLP dan metode yang diadopsi dari penelitian Md. Ridzwan Beg et.al.

Pada metode ontologi, terdapat 2 fase dalam mendeteksi ambiguitas, yang mana pada masing-masing fase tersebut terdapat 3 dan 2 tahapan untuk mendeteksi ambiguitas. Pendeteksian ambiguitas pada metode ontologi juga tidak dilakukan secara manual, melainkan memerlukan bantuan dari beberapa faktor, diantaranya memerlukan adanya teknik NLP, aplikasi ontologi domain dan pemeriksaan oleh ahli. Dilihat dari masing-masing tahapan yang dilakukan, metode ini masih terbilang belum cukup baik untuk mendeteksi ambiguitas jenis leksikal, karena pada dasarnya metode ini paling banyak digunakan untuk mendeteksi berbagai macam jenis ambiguitas, termasuk ambiguitas leksikal dan juga metode ontologi memang digunakan untuk mendeteksi semua jenis ambiguitas, namun untuk jenis ambiguitas leksikal, metode ini masih terbilang belum menghasilkan hasil akhir yang sempurna atau benar-benar bebas dari ambiguitas. Meskipun demikian, metode ini masih tetap menjadi opsi bagi para peneliti dalam mendeteksi ambiguitas leksikal, karena metode ini cukup sederhana, terlebih lagi dibantu oleh bantuan aplikasi dan juga tidak memerlukan waktu yang terlalu lama untuk mendapatkan hasil akhir. Namun terkadang hasil yang diharapkan tidak terlalu maksimal untuk mendeteksi ambiguitas jenis leksikal. Dilihat dari hasil keakuratan yang dihasilkan oleh grafik mengenai perhitungan tingkat keakuratan di atas, metode ini mendapatkan hasil akhir di angka 4, yang mana nilai tersebut menunjukkan untuk keakuratan dari metode ini tergolong dalam tingkatan cukup akurat.

Pada metode *Natural Language Processing* (NLP), terdapat 2 pendekatan yang berbasis NLP, yang mana pada pendekatan pertama terdapat 3 tahapan dan pada pendekatan kedua terdapat 2 tahapan untuk mendeteksi ambiguitas. Pendeteksian ambiguitas pada metode NLP juga tidak dilakukan secara manual, melainkan memerlukan

bantuan dari beberapa faktor, diantaranya memerlukan adanya *prototype tool*, seorang *parser*, seorang *reviewer*, jaringan sistem semantik dan juga campur tangan seorang ahli. Dilihat dari masing-masing tahapan yang dilakukan, metode ini terbilang baik untuk mendeteksi ambiguitas jenis leksikal karena pengidentifikasian kalimat yang dilakukan pada metode ini mendasar ke arah leksikal, yaitu mencari kata yang memiliki banyak arti. Selain karena faktor banyak digunakan untuk mendeteksi ambiguitas jenis leksikal, dalam metode ini banyak *tools* yang digunakan, maka dari itu tidak heran jika hasil akhir dari metode ini terbilang akurat. Terlebih lagi setelah diproses dengan *tools*, hasil akhir dari metode ini juga diperiksa kembali oleh seorang ahli, jadi hasil yang didapat bebas dari ambiguitas leksikal. Dilihat dari hasil keakuratan yang dihasilkan oleh grafik mengenai perhitungan tingkat keakuratan, metode ini mendapatkan hasil akhir di angka 6, yang mana nilai tersebut menunjukkan untuk keakuratan dari metode ini tergolong dalam tingkatan yang akurat.

Pada metode yang diadopsi dari Ridzwan Beg et.al., terdapat 2 fase dalam mendeteksi ambiguitas leksikal, yang mana dalam 2 fase tersebut terdapat 8 tahapan untuk mendeteksi ambiguitas leksikal. Pendeteksian dengan metode ini termasuk pendekatan yang masih manual, karena tidak memerlukan *tools*, *prototype* ataupun aplikasi khusus dalam proses pendeteksian. Metode ini hanya menggunakan algoritma tunggal yang diciptakan oleh peneliti dan juga memerlukan bantuan kamus Bahasa Inggris untuk proses perbandingan kata-kata yang telah diidentifikasi. Dilihat dari masing-masing tahapan yang dilakukan, metode ini terbilang baik untuk mendeteksi ambiguitas jenis leksikal, karena algoritma yang digunakan oleh peneliti memang diciptakan khusus untuk mendeteksi ambiguitas leksikal. Namun, masih belum ada penelitian lain yang mencoba untuk mengimplementasikan metode ini dalam penelitian mereka, jadi hasil yang didapatkan dari metode ini pun masih belum cukup teruji keakuratannya dan juga karena faktor manual, metode ini membutuhkan waktu yang cukup lama dalam mengidentifikasi masing-masing kalimat serta kata yang ada didalamnya. Dilihat dari hasil keakuratan yang dihasilkan oleh grafik mengenai perhitungan tingkat keakuratan, metode ini mendapatkan hasil akhir di angka 5, yang mana nilai tersebut menunjukkan untuk keakuratan dari metode ini tergolong dalam tingkatan yang cukup akurat.

#### 4. KESIMPULAN

Berdasarkan hasil pengkajian mengenai metode-metode yang digunakan untuk mendeteksi ambiguitas leksikal, dapat disimpulkan bahwa metode yang menghasilkan hasil yang akurat diantara metode ontologi, *Natural language Processing* dan metode yang diadopsi dari penelitian Md. Ridzwan, adalah metode *Natural Language Processing*. Pernyataan ini didukung oleh makalah-makalah yang sudah dikaji, yang mana makalah-makalah tersebut membahas mengenai metode dalam mengatasi ambiguitas leksikal. Pernyataan ini juga didukung oleh perhitungan dari step/tahapan dan elemen-elemen yang ada pada masing-masing metode, karena semakin banyak tahapan dan elemen (*tools*, aplikasi, *prototype*, *expert*, dan lain-lain) yang digunakan dalam metode tersebut, maka semakin akuratlah pendeteksian ambiguitas leksikal yang ada pada dokumen SRS.

#### 5. SARAN

Pada penelitian mendatang, diharapkan akan adanya penelitian yang lebih mengupas tentang teknik pendeteksi ambiguitas leksikal secara lebih luas. Dan juga diharapkan lebih banyak lagi mengupas dan membahas lebih dalam tentang teknik yang digunakan dalam mendeteksi ambiguitas bukan hanya dari ambiguitas leksikal, namun ambiguitas yang lain.

## UCAPAN TERIMA KASIH

Penelitian ini tidak akan berhasil tanpa dukungan berbagai pihak. Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu penelitian ini.

## DAFTAR PUSTAKA

- Katz J.J., et.al., 2008. The Structure of a Semantic Theory. Vol. 39(2), pp 170-210.
- Parera, J.D., 2004. Teori Semantik. Jakarta: Penerbit Erlangga.
- Ullmann, S., 1967. Semantics: An Introduction to the Science of Meaning. Oxford: Basil Blackwell, pp55.
- Sandhu, G., Sikka, S., 2015. State-of-art practices to detect Inconsistencies and Ambiguities from *Software Requirements*. IEEE: International Conference on Computing, Communication and Automation, pp 812-817.
- Shah, U.S., Jinwala, D.C., 2015. Resolving Ambiguities in *Natural language Software Requirements* : A Comprehensive Survey. ACM SIGSOFT *Software Engineering Notes*, Vol. 40(5), pp 1-7.
- Beg, M.R., Abbas, Q., Joshi, A., 2008. A Method to Deal with the Type of *Lexical ambiguity* in a *Software Requirement Specification Document*. IEEE: First International Conference on Emerging Trends in Engineering and Technology. Integral University, India, pp 1212-1215.
- Bhatia, M.P.S., Kumar, A., Beniwal, R., 2016. Ontology based Framework For Detecting Ambiguities in *Software Requirements Specification*. International Conference on Computing for Sustainable Global Development (INDIACom). 978-9-3805-4421-2/16, India.
- Popescu, Rugaber, Medvidovic, Berry, 2007. Reducing Ambiguities Inrequirement Specifications via Automatic Created Object-Oriented Models. Springer-Verlag proceedings of 14th Monterey Workshop on Innovations for requirement analysis: From stakeholders needs to formal designs. CA, USA, ISBN no. 978-3-540-89777-4, September 2007, pp 103-124.
- Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D.M., 2007. *Requirements for tools for ambiguity identification and measurement in Natural Language Requirements Specifications*. Proceedings of workshop on emEngenharia de requisitos. Toronto, Canada, May 2007, pp 197-206.
- Castaneda, Ballejos, Caliusco, Galli, 2010. The Use of Ontologies in *Requirements Engineering*. 1st ed Vol. 10, no. 6. Global Journal of Research in Engineering.
- Osborne, M., MacNish, C.K., 1996. Processing *Natural language Software Requirement Specifications*. IEEE : Proceedings of ICRE'96, pp. 229-236.
- Haron, H., et.al., 2015. A Conceptual Model to Manage *Lexical ambiguity* in Malay Textual *Requirements*. ARPN Journal of Engineering and Applied Sciences Vol. 10, ISSN 1819-6608, pp. 1405-1412.
- Fan, S., et.al., 2016. A Process ontologi Based Approach to Easing Semantic Ambiguity in Business Process Modeling. *Data & Knowledge Engineering*, pp. 1-21.
- Pittke, F., Leopold, H., Mendling, J., 2015. IEEE Transactions on *Software Engineering*, Vol. 41, No.6, pp. 526-544.
- Al-Harbi, O., Jusoh, Shaidah., Norwawi, N., 2012. Handling Ambiguity Problems of *Natural Language Interface for Question Answering*. International Journal of Computer Science Issues, Vol. 9, Issue 3, ISSN (Online): 1694-0814, pp. 17-25.
- Singh, S., Saikia, L.P., 2015. Ambiguity in Requirement Engineering Documents: Importance, Approaches to Measure and Detect, Challenges and Future Scope. International Journal of Advanced Research in Computer Science and *Software Engineering*, Vol. 5, Issue 10, ISSN 2277 128X, pp. 791-798.
- Kamsties, E., Berry, D.M., Paech, B., 2001. Detecting Ambiguities in Requirements Documents Using Inspections. pp. 1-13.
- Unger, C., Cimiano, P., Representing and Resolving Ambiguities in ontologi-based Question Answering.
- Rowen, R.B., 1990. Software Project Management Under Incomplete and Ambiguous Specifications. Ieee Transactions On Engineering Management, Vol. 37, No. I, February 1990. 10 – 21.
- Umber, A., Bajwa, I.S., 2011. Minimizing Ambiguity in Natural language *Software Requirements Specification*. 978-1-4577-1539-6/11/\$26.00 ©2011 IEEE. 102 – 107.

- Maiden, Nell., 2012. Cherishing Ambiguity. The IEEE Computer Society. 0740-7459/12/\$31.00 © 2012 IEEE. 16-17.
- Gill, D.K., et.al., 2014. Semi-Automation for Ambiguity Resolution in Open Source Software Requirements. CCECE. 978-1-4799-3010-9/14/\$31.00 ©2014 IEEE. 1-6.
- Lawrence, B., Jackson, D., 1996. Do You Really Need Formal Requirements ?. IEEE Software.
- Jain R., et.al. 2013. Lexical ambiguity Resolution in Web Search for Phrases. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3, Issue 4, April 2013ISSN: 2277 128X.pp 647 – 651.