

MENDETEKSI REDUNDANSI PASANGAN KALIMAT DALAM DOKUMEN SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK DENGAN PENDEKATAN KEMIRIPAN SEMANTIK

Enny Dwi Oktaviani^{a,1,*}, Licantik^{b,2}

^a Universitas Palangka Raya, Kampus Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia

^b Universitas Palangka Raya, Kampus Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia

¹ enny@it.upr.ac.id*; ² licantikmeteh@gmail.com;

* corresponding author

ARTICLE INFO

Keywords

Software Requirements
Specification Document
Redundancy
Semantic
Sentences

ABSTRACT

Software Requirements Specification Document must be complete, unambiguous and correct because an incorrect Software Requirements Specification Document can lead to failure in software development. Software Requirements Specification document must contain non-redundant sentences. Meyer's Seven Sins describes several errors in Software Requirements Specification Document, one of which is redundancy which is a variant of noise. Redundancy is the repetition of the same information, but using different terms or phrases, thus giving the impression that new information is available. This study analyzes the performance of the framework and tools for detecting redundancy in sentence pairs in the software requirements specification document using the WordNet-based semantic similarity method. This study tries to develop the framework and tools that have been proposed in previous studies, namely by detecting redundancy in sentence pairs without fact extraction. The proposed approach and evaluation process uses Kappa values to determine whether the performance of the framework and assistive tools can be used to detect redundancy in sentence pairs properly.

The method using literature study, data collection, problem analysis, making a framework, making auxiliary tools, and testing. The test uses the Kappa value with two scenarios, from the results of testing the framework and assistive tools can be used to detect redundancy in the Software Requirements Specification Document with a Kappa value of 0.777 with the interpretation of the kappa value is a substantial agreement at a threshold of 0.73.

1. Pendahuluan

Kebutuhan perangkat lunak adalah kondisi atau kemampuan yang harus dimiliki perangkat lunak untuk memenuhi apa yang harus diisyaratkan atau diinginkan oleh pemakai. Salah satu tahap rekayasa kebutuhan adalah spesifikasi kebutuhan. Spesifikasi kebutuhan merupakan suatu proses memformalisasikan sekumpulan kebutuhan dari suatu sistem yang hendak dibangun ke dalam suatu dokumen. Salah satu dokumen yang dihasilkan oleh rekayasa kebutuhan adalah Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).

Karakteristik dokumen spesifikasi kebutuhan perangkat lunak atau dokumen SKPL yang baik sesuai dengan IEEE (1998) [1] adalah benar, tidak ambigu, lengkap, konsisten, terurut berdasarkan kepentingannya atau kestabilannya, dapat diverifikasi, dapat dimodifikasi dan dapat ditelusuri. Sedangkan bentuk kalimat-kalimat dalam dokumen SKPL adalah berupa kalimat pernyataan. Dokumen SKPL yang tidak lengkap, salah, dan ambigu dapat menyebabkan kegagalan dalam pengembangan perangkat lunak [5]. Spesifikasi kebutuhan adalah tahapan kritis dalam SDLC, maka dalam pembuatannya dokumen SKPL harus hati-hati agar menghasilkan perangkat yang berkualitas perangkat [2]. Kesalahan yang terjadi pada rekayasa kebutuhan akan mempengaruhi kegagalan

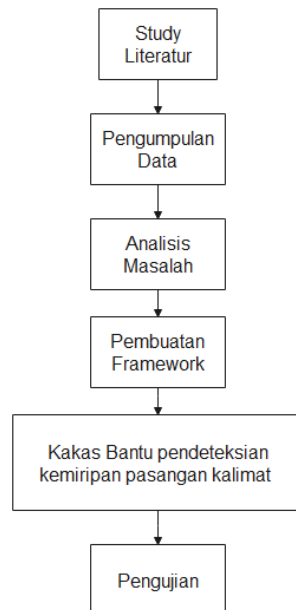
produk perangkat lunak. Tidak peduli seberapa bagus tahap selanjutnya [3]. [6] Membahas pentingnya dampak pemeriksaan formal dari spesifikasi persyaratan untuk Jaminan Kualitas Perangkat Lunak dan peninjauan spesifikasi persyaratan. Dalam penelitian [7] menjelaskan pendekatan otomatis untuk mendeteksi spesifikasi kebutuhan perangkat lunak yang ambigu. Makalah ini mempelajari sekumpulan sembilan algoritma klasifikasi dari komunitas pembelajaran mesin dan mengevaluasi algoritma mana yang berkinerja terbaik untuk mendeteksi spesifikasi kebutuhan perangkat lunak yang ambigu. Sedangkan [8] melakukan penelitian untuk mendeteksi dan menghapus informasi redundansi menggunakan algoritma graph-based representation di level sub-snippet pada dokumen berita

Meyer, 1985 menemukan tujuh kesalahan dalam dokumen spesifikasi kebutuhan perangkat lunak yang disebut dengan *Meyer's seven sins*. *Meyer's seven sins* adalah tujuh kesalahan yang menyebabkan seorang spesifikator salah dalam menganalisis sebuah dokumen spesifikasi kebutuhan. Tujuh kesalahan tersebut adalah *Noise (redundancy, remorse)*, *Silence*, *Overspecification*, *contradiction*, *ambiguity*, *forward reference*, *wishful thinking* [4]. Kesalahan-kesalahan dalam *Meyer's seven sins* tersebut dapat merusak kualitas kebutuhan perangkat lunak. [9] telah melakukan penelitian untuk yang membahas mengenai kesalahan dalam *Meyer's seven sins* yaitu masalah redundansi yang merupakan varian dari *noise*. Masalah redundansi ini juga menjadi fokus pada penelitian yang dilakukan oleh [9] dengan mengusulkan suatu framework untuk mendeteksi redundansi pada pasangan kalimat yang diekstraksi menjadi pasangan fakta atau frase. Berdasarkan penelitian yang dilakukan oleh [9] pasangan frase memiliki nilai redundansi jika pasangan frase memiliki nilai kemiripan semantik yang tinggi dan sebaliknya tidak redundansi jika memiliki nilai kemiripan semantik yang rendah.

Penelitian ini merupakan kelanjutan penelitian sebelumnya [11] yang mengusulkan suatu solusi berupa framework pendeteksian redundansi yang bertujuan untuk mendeteksi redundansi dalam dokumen SKPL pada pasangan kalimat dengan menguji framework yang diusulkan dengan menggunakan kakas bantu menghitung kemiripan pasangan kalimat pada dokumen SKPL. Proses pengujian dan evaluasi menggunakan nilai Kappa dan interpretasi nilai kappa untuk mengetahui kekuatan proporsi kesepakatan antara yang diajukan dengan ahli bahasa

2. Metodologi Penelitian

Metode pada penelitian ini terdiri dari beberapa tahap seperti yang ditunjukkan pada oleh gambar 1



Gambar 1. Langkah-Langkah Penelitian

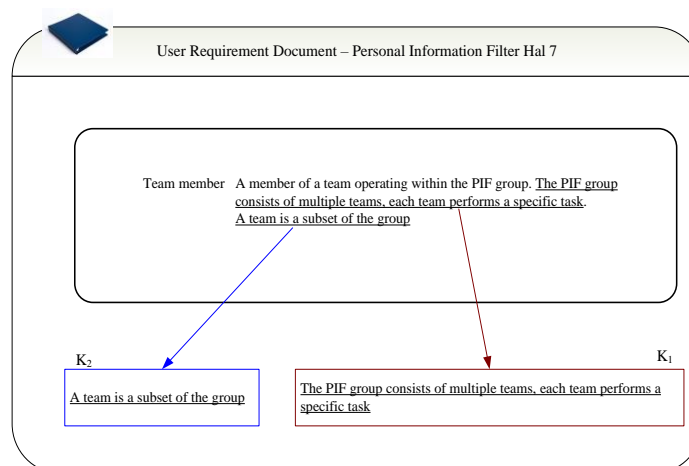
2.1 Studi Literatur

Tahap ini merupakan tahap mencari referensi dan teori yang relevan dengan permasalahan yang ada. Referensi di dapat dari jurnal, buku, dan referensi lainnya yang terkait dengan penelitian ini.

2.2 Pengumpulan Data

Tahap ini merupakan tahap mengumpulkan dataset yang diperlukan untuk penelitian ini. Data set pada penelitian ini terdiri dari 205 pasangan kalimat yang diambil dari 8 dokumen SKPL.

Cara mengambil pasangan kalimat dalam dokumen SKPL dapat dilihat pada ilustrasi yang ditunjukkan pada gambar 2.

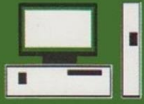


Gambar 2. Mengambil pasangan kalimat dalam dokumen SKPL

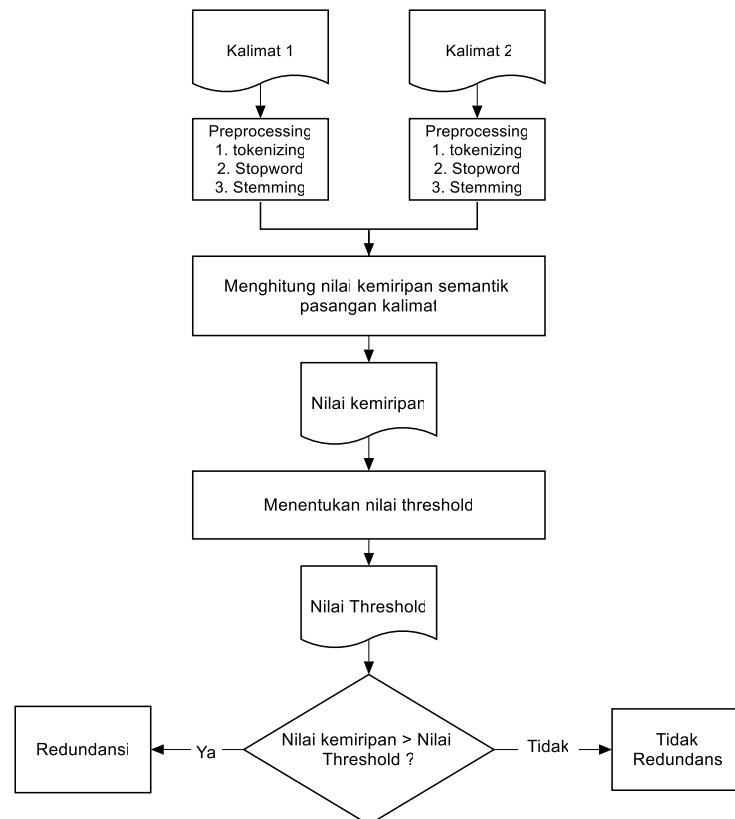
2.3 Analisis Masalah

Tahap ini merupakan tahap untuk mempelajari, memahami, menemukan dan menganalisis lebih lanjut mengenai masalah yang terdapat dalam dokumen-dokumen SKPL

2.4 Framework



Framework yang telah dibuat pada penelitian [11] ditunjukkan pada gambar 3



Gambar 3. Framework Pendeteksian Redundansi pada Pasangan Kalimat Dalam Dokumen SKPL [11]

Sebagai contoh tahap pertama dari framework adalah menginputkan pasangan kalimat yang diambil dari *user requirement document Personal Information Filter* versi 2.0.0 halaman 7.

$K_1 =$ "The PIF group consists of multiple teams, each team performs a specific task"

$K_2 =$ "A team is a subset of the group".

Tahap berikutnya adalah melakukan prapemrosesan yang terdiri dari tahap tokenisasi, penghapusan stopwords, dan stemming. Hasil dari prapemrosesan pasangan kalimat adalah :

Tokenize :

K_1 : "The" "PIF" "group" "consists" "of" "multiple" "teams" "each" "team" "performs" "a" "specific" "task"

K_2 : "A" "team" "is" "a" "subset" "of" "the" "group"

Stopwords removal :

K_1 : pif group consists multiple teams team performs specific task

K_2 : team subset group

Stemming :

K_1 : group consist multiple team team perform specific task

K_2 : team subset group

Tahap ketiga adalah menghitung nilai kemiripan semantik pasangan kalimat K_1 dan K_2 . Kemiripan pasangan kalimat dihitung menggunakan metode kemiripan semantik berbasis WordNet. Penelitian [10] dan [11] juga menggunakan metode ini. WordNet adalah basis data leksikal untuk bahasa Inggris yang merupakan hasil penelitian dari Princeton University dalam bidang

komputasional linguistik. Basis data leksikal adalah kumpulan data yang menyimpan informasi relasi semantik antar sinonim set (*synset*). Setiap synset terdiri dari serangkaian kata-kata yang bersinonim, dan rujukan yang menjelaskan kaitan antar satu synset dengan synset lainnya. Dengan demikian kata-kata yang muncul dalam sebuah synset adalah kata-kata yang dianggap dapat dipakai sebagai kata pengganti

Tabel 1 menunjukkan matriks kemiripan semantic pasangan kalimat K1 dan K2

Tabel 1. Matriks Kemiripan Semantik Pasangan Kalimat K1 Dan K2

	K2	team	subset	group
K1				
Group		0,89	0,8	1
Consist		0	0,29	0
Multiple		0,33	0,38	0,4
Team		1	0,5	0,89
Team		1	0,5	0,89
Perform		0,29	0	0,33
Specific		0,4	0,46	0,5
task		0,38	0,43	0,46

Kemudian berdasarkan matriks kemiripan semantic, nilai-nilai kemiripan tiap kata dihitung menggunakan metode heuristic cepat. Dari hasil perhitungan dengan menggunakan metode heuristic cepat, diperoleh nilai kemiripan pasangan kalimat K1 dan K2 adalah 0,707

$$Sim(Z^A, Z^C) = \frac{\sum(\text{nilai maksimal tiap baris}) + \sum(\text{nilai maksimal tiap kolom})}{\text{jumlah baris dan jumlah kolom}} \quad (1)$$

$$Sim(K1, K2) = \frac{2,8+4,98}{11}$$

$$Sim(K1, K2) = 0,707$$

Tahap ke empat adalah menentukan nilai *threshold* berdasarkan nilai kemiripan pasangan kalimat. Nilai *threshold* ditentukan dengan menggunakan nilai kemiripan pasangan kalimat yang diperoleh pada tahap ketiga. Skenario untuk menentukan nilai *threshold* adalah dengan mengamati nilai kesamaan pasangan kalimat dalam dataset kemudian melakukan pengujian terhadap beberapa nilai *threshold* untuk mendapatkan tingkat akurasi tinggi dari nilai kappa . Memilih nilai ambang batas yang tepat akan meningkatkan akurasi dan kualitas nilai kappa.

Tahap terakhir adalah menentukan apakah pasangan kalimat redundansi atau tidak. Jika nilai kemiripan lebih besar dari nilai *threshold*, maka pasangan kalimat adalah redundansi, dan jika lebih kecil, maka pasangan kalimat tidak redundansi.

2.5 Menghitung Kemiripan Pasangan Kalimat Dengan Kakas Bantu Pendeteksian Redundansi Pasangan Kalimat

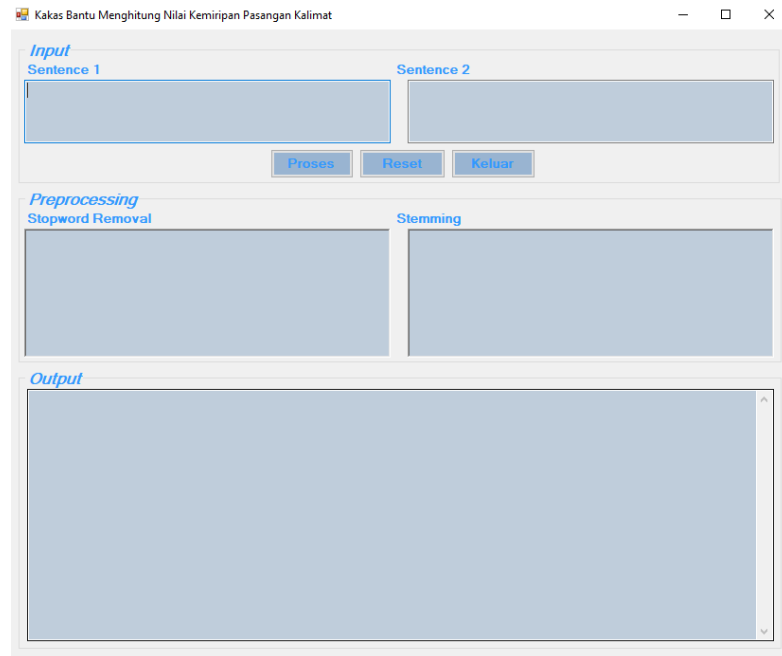
Pasangan kalimat akan dihitung kemiripannya menggunakan kakas bantu menghitung kemiripan pasangan kalimat dalam dokumen spesifikasi kebutuhan perangkat lunak .

2.6 Pengujian

Tahap ini untuk menguji Framework dan Kakas Bantu yang dibuat. Pengujian dan evaluasi dalam penelitian ini menggunakan Nilai Kappa dan interpretasi nilai kappa. Nilai-nilai Kappa menunjukkan proporsi kesepakatan antara dua penguji, dalam hal ini seorang ahli bahasa Inggris untuk menguji keakuratan redundansi data sebagai tes pertama dan Kakas Bantu yang diusulkan sebagai tes kedua. Nilai-nilai Kappa digunakan untuk menentukan apakah Kakas Bantu yang diusulkan dapat diandalkan. Dalam penelitian ini, pengujian untuk mendeteksi redundansi diambil dari delapan SRS dan 205 pasangan kalimat

3. Hasil Dan Pembahasan

Kakas bantu menghitung kemiripan pasangan kalimat ditunjukkan pada gambar 4.



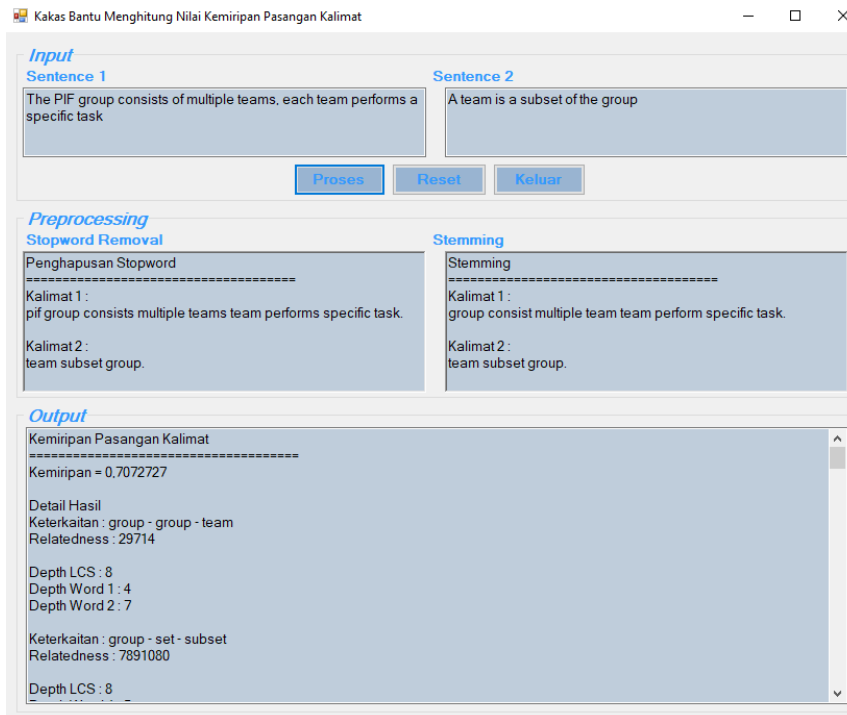
Gambar 4. Kakas Bantu Menghitung Kemiripan Pasangan Kalimat

Pada kakas bantu akan dicoba diinputkan :

Kalimat 1 = *“The PIF group consists of multiple teams, each team performs a specific task”*

Kalimat 2 = *“A team is a subset of the group”*.

Hasil kemiripan pasangan kalimat tersebut dengan menggunakan kakas bantu adalah 0,707 sesuai dengan perhitungan manual yang dilakukan pada bab sebelumnya. Gambar 5 menunjukkan hasil perhitungan kemiripan pasangan kalimat 1 dan 2 dengan menggunakan kakas bantu.



Gambar 5. Hasil perhitungan kemiripan pasangan kalimat menggunakan kakas bantu

Pada bagian Output kakas bantu terlihat detail perhitungan kemiripan pasangan kalimat adalah sebagai berikut :

Kemiripan Pasangan Kalimat

```
=====
Kemiripan = 0,7072727
```

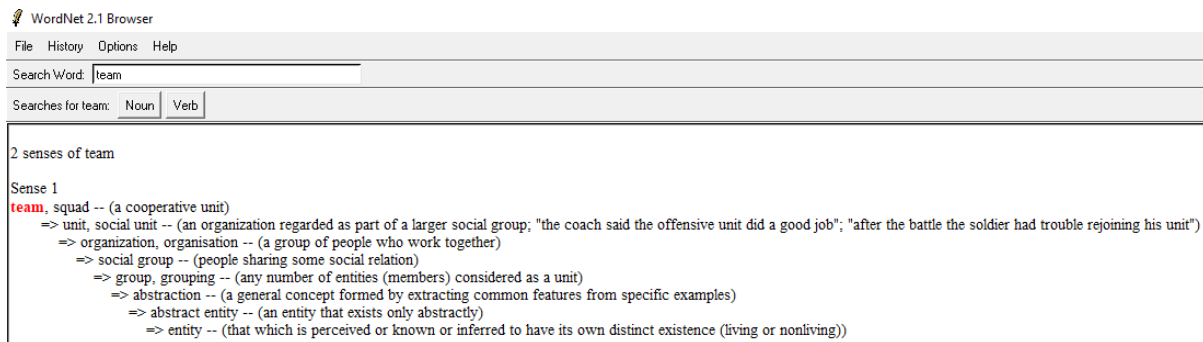
Pada bagian output akan dihitung berdasarkan matriks keterkaitan antar kata pada pasangan kalimat. Misalkan pada kata **specific** dan kata **group** setelah dicek dalam kamus WordNet maka Tidak ada keterkaitan pada kata tersebut, sehingga keterkaitan bernilai -1.

```
Keterkaitan : specific - Tidak Ada Keterkaitan - group
Relatedness : -1
Depth LCS : -1
Depth Word 1 : -1
Depth Word 2 : -1
```

Sedangkan kata **team** dan **group** memiliki keterkaitan. Dalam kamus WordNet, team dan group merupakan satu hirarki taksonomi hiponim.

```
Keterkaitan : team - group - group
Relatedness : 29714
Depth LCS : 8
Depth Word 1 : 7
Depth Word 2 : 4
```

Keterkaitan kata **team** dan **group** dalam kamus WordNet dapat dilihat pada gambar 6.



Gambar 6. Keterkaitan kata **Team** dan **Group** Dalam Kamus Basis Data Leksikal WordNet Panjang Naskah (Heading 2)

3.1 Pengujian

Tahapan pengujian menggunakan skenario dalam dua tahap untuk menguji kinerja dari *framework* yang ada. Tahap pertama dengan menguji 205 pasang kalimat dari 8 dokumen bahasa Inggris pada nilai threshold dari rentang 0,1 – 1. Berdasarkan scenario tahap pertama, maka diperoleh nilai kappa tertinggi yaitu 0,749 pada threshold 0,7. Kemudian pada tahap kedua, pengujian akan dilakukan pada rentang nilai 0,66 – 0,73 dimana pada rentang tersebut terdapat nilai threshold 0,7 (nilai threshold yang didapat pada scenario tahap pertama dengan nilai kappa tertinggi), ini bertujuan untuk melihat lebih detail nilai threshold.

Nilai Kappa digunakan untuk mengetahui apakah kerangka kerja dapat diandalkan dan dapat digunakan untuk mendeteksi redundansi pada pasangan kalimat. Disamping itu, subyektivitas dari seorang ahli sebagai penguji pertama adalah pertimbangan lain mengapa indeks Kappa digunakan untuk menganalisis hasil pengujian. Nilai kappa menunjukkan proporsi kesepakatan antara dua penguji, dalam hal ini seorang ahli di bidang bahasa Inggris, untuk menguji akurasi data redundansi sebagai penguji pertama dan kerangka kerja sebagai penguji kedua. Susunan data untuk menghitung nilai Kappa ditunjukkan pada Tabel 2. dan interpretasi nilai Kappa ditunjukkan pada Tabel 3.

Tabel 2. Susunan Data untuk Menghitung Nilai Kappa

	Pengamat 1		Total
	ya	tidak	
Pengamat 2	Ya	b	m_1
	tidak	d	m_0
Total	n_1	n_0	n

$$p_o = \frac{a+d}{n}$$

$$p_e = \left[\left(\frac{n_1}{n} \right) \times \left(\frac{m_1}{n} \right) \right] + \left[\left(\frac{n_0}{n} \right) \times \left(\frac{m_0}{n} \right) \right]$$

$$Kappa, K = \frac{(p_o - p_e)}{(1 - p_e)}$$

Nilai a merupakan nilai dimana pengamat 1 dan pengamat 2 sepakat mengatakan “ya”, dan nilai d merupakan nilai dimana pengamat 1 dan pengamat 2 sepakat mengatakan “tidak”. Sedangkan nilai b dan c adalah nilai dimana pengamat 1 dan 2 tidak memiliki kesepakatan.

Tabel 3. Interpretasi Nilai Kappa

Nilai Indeks Kappa	Proporsi Kesepakatan
< 0	Rendah (<i>less than chance agreement</i>)
0.01 – 0.20	Sedikit (<i>slight agreement</i>)
0.21 – 0.40	Cukup (<i>fair agreement</i>)
0.41 – 0.60	Sedang (<i>moderate agreement</i>)
0.61 – 0.80	Banyak (<i>substantial agreement</i>)
0.81 – 1	Hampir Sempurna (<i>almost perfect agreement</i>)

3.1.1 Skenario Pengujian Tahap Pertama

Pada tahap pertama akan dilakukan pengujian dengan rentang 0,1 – 1. Contoh susunan data untuk menghitung nilai Kappa dengan nilai threshold 0,7 pada skenario tahap pertama dapat dilihat pada Tabel 4.

Tabel 4. Komposisi Data Untuk menghitung Nilai Kappa dengan Nilai Threshold 0,7 Pada Skenario Pertama

		Ahli Bahasa		
		redundancy	Not redundancy	Total
Kerangka Kerja	redundancy	44 (a)	13 (b)	57 (m1)
	Not redundancy	7 (c)	141 (d)	148 (m0)
	Total	51 (n1)	154 (n0)	205 (n)

Berdasarkan nilai pada tabel 4, maka dapat dihitung nilai kappa untuk threshold 0,7 adalah sebagai berikut :

$$p_o = \frac{a + d}{n}$$

$$p_o = \frac{44 + 141}{205} = 0,902$$

$$p_e = \left[\left(\frac{n_1}{n} \right) \times \left(\frac{m_1}{n} \right) \right] + \left[\left(\frac{n_0}{n} \right) \times \left(\frac{m_0}{n} \right) \right]$$

$$p_e = \left[\left(\frac{51}{205} \right) \times \left(\frac{57}{205} \right) \right] + \left[\left(\frac{154}{205} \right) \times \left(\frac{148}{205} \right) \right] = 0,612$$

$$Kappa, K = \frac{(p_o - p_e)}{(1 - p_e)}$$

$$Kappa, K = \frac{0,902 - 0,612}{(1 - 0,612)} = 0,749$$

Pada Skenario tahap pertama, diperoleh nilai kappa tertinggi 0,749 dengan threshold 0,7

3.1.2 Skenario Pengujian Tahap Kedua

Skenario tahap kedua adalah melakukan percobaan dengan rentang nilai threshold 0,66 – 0,73 dimana pada rentang tersebut terdapat nilai threshold 0,7, yaitu nilai threshold dengan nilai kappa tertinggi pada scenario tahap pertama. Susunan data pada scenario tahap kedua dapat dilihat pada tabel 5.

Tabel 5. Komposisi Data Untuk Menghitung Nilai Kapaa Pada Skenario Kedua

		Ahli Bahasa		
		redundancy	Not redundancy	Total
Kerangka Kerja	redundancy	42 (a)	8 (b)	50 (m1)
	Not redundancy	9 (c)	146 (d)	155 (m0)
	Total	51 (n1)	154 (n0)	205 (n)

Berdasarkan tabel 5, diperoleh perhitungan nilai kappa untuk threshold 0,73 adalah sebagai berikut :

$$p_o = \frac{a + d}{n}$$

$$p_o = \frac{42 + 146}{205} = 0,917$$

$$p_e = \left[\left(\frac{n_1}{n} \right) \times \left(\frac{m_1}{n} \right) \right] + \left[\left(\frac{n_0}{n} \right) \times \left(\frac{m_0}{n} \right) \right]$$

$$p_e = \left[\left(\frac{51}{205} \right) \times \left(\frac{50}{205} \right) \right] + \left[\left(\frac{154}{205} \right) \times \left(\frac{155}{205} \right) \right] = 0,629$$

$$Kappa, K = \frac{(p_o - p_e)}{(1 - p_e)}$$

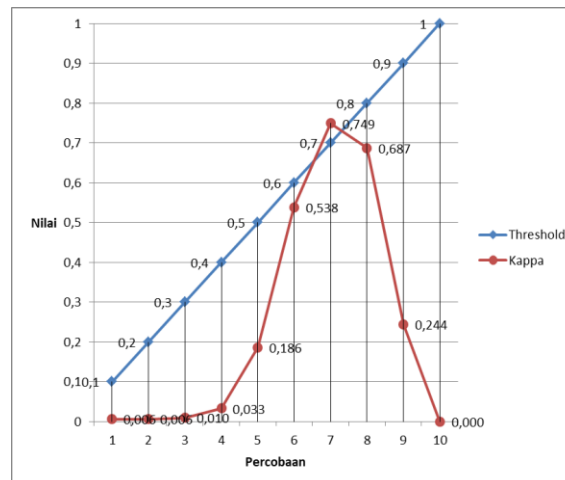
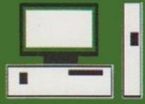
$$Kappa, K = \frac{0,917 - 0,629}{(1 - 0,629)} = 0,777$$

Hasil perhitungan nilai Kappa pada scenario tahap pertama dan scenario tahap kedua dapat dilihat pada tabel 6.

Tabel 6. Hasil Perhitungan Nilai Kappa

Skenario Pertama		Skenario Kedua	
Threshold	Kappa	Threshold	Kappa
0,1	0,006	0,66	0,70
0,2	0,006	0,67	0,697
0,3	0,010	0,68	0,713
0,4	0,033	0,69	0,706
0,5	0,186	0,7	0,749
0,6	0,538	0,71	0,742
0,7	0,749	0,72	0,754
0,8	0,687	0,73	0,777
0,9	0,244	0,74	0,759
1	0,000	0,75	0,755

Berdasarkan tabel 6, maka dapat dibuat grafik nilai threshold terhadap nilai kappa yang dilakukan pada scenario tahap pertama dan tahap kedua yang ditunjukkan pada gambar 7.



Gambar 7. Grafik Nilai Threshold Terhadap Nilai Kappa Skenario Tahap Pertama dan Tahap Kedua

Berdasarkan skenario, maka diperoleh nilai kappa tertinggi 0,777 dengan nilai threshold 0,73. Maka berdasarkan tabel 9, kinerja kerangka kerja yang diusulkan terletak pada rentang 0,61 – 0,80 dengan proporsi nilai kappa ada Substantial Agreement, yang artinya banyak kesepakatan antara ahli bahasa dan kerangka kerja yang diusulkan.

Tabel 9. Interpretasi Nilai Kappa

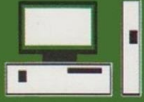
Kappa Index Values	Proportion Agreement
< 0	<i>less than chance agreement</i>
0.01 – 0.20	<i>slight agreement</i>
0.21 – 0.40	<i>fair agreement</i>
0.41 – 0.60	<i>moderate agreement</i>
0.61 – 0.80	<i>substantial agreement</i>
0.81 – 1	<i>almost perfect agreement</i>

4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan dengan dua skenario menggunakan nilai Kappa, maka framework dan kaskas bantu yang diusulkan dapat digunakan untuk mendeteksi redundansi pada pasangan kalimat dalam dokumen spesifikasi kebutuhan perangkat lunak dengan nilai threshold 0,73 dan nilai Kappa 0,777 pada interpretasi nilai Kappa adalah *substantial agreement*. Untuk penelitian lebih lanjut maka kaskas bantu dapat dikembangkan sehingga dapat menerima inputan dokumen spesifikasi kebutuhan perangkat lunak.

Daftar Pustaka

- [1] S. Engineering and S. Committee, IEEE recommended practice for software requirements specifications, vol. 1998. 2011.
- [2] R. Pressman, *Rekayasa Perangkat Lunak 7th ed.* Yogyakarta: Andi, Yogyakarta, 2012.
- [3] S. Nurfauziah, Pendeteksian Ketidakeengkapan Kebutuhan Dengan Teknik Klasifikasi Pada Dokumen Spesifikasi Kebutuhan Perangkat Lunak, vol. 9, no. 2. 2017.
- [4] B. Meyer, "On Formalism in Specification," *IEEE Softw.*, vol. 2, no. 1, pp. 6–26, 1985.
- [5] Umer A and Bajwa I.S, "No Title," in *Minimizing Ambiguity in Natural Language Software Requirements Specification*, 2011, pp. 102–107.
- [6] P. Gupta, "Impact of formal Inspection to Software Requirements Specification For Software Quality Assurance," *Glob. J. Enterp. Inf. Syst.*, vol. 2, no. 2, pp. 11–19, 2010.



- [7] M. H. Osman and M. F. Zaharin, “Ambiguous software requirement specification detection: An automated approach,” *Proc. - Int. Conf. Softw. Eng.*, no. July, pp. 33–40, 2018.
- [8] K. Thadani and K. McKeown, “A framework for identifying textual redundancy,” *Coling 2008 - 22nd Int. Conf. Comput. Linguist. Proc. Conf.*, vol. 1, no. August, pp. 873–880, 2008.
- [9] E. D. Oktaviyani and D. Oranova, “Redundancy Detection In Software Requirements Statements,” (hal 90-91). The 12th International Conference on Electronics, Information, and Communication (ICEIC 2013), Bali, 30 Jan – 2 Feb 2013
- [10] S. Christina and E. D. Oktaviyani, “Identifying the relevant page numbers that referred by the back-of-book index using syntactic similarity and semantic similarity,” in *2017 Second International Conference on Informatics and Computing (ICIC)*, 2017, pp. 1–6.
- [11] E. D. Oktaviyani and Licantik, “Redundancy Detection of Sentence Pairs in the Software Requirements Specification Documents with Semantic Approach,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 879, no. 1, 2020.