

PERINGKASAN TEKS MULTI-DOKUMEN BERDASARKAN METODE *SENTENCE EXTRACTION* DAN *WORD SENSE DISAMBIGUATION*

Khairiyah Nur Aisyah¹⁾, Syadza Anggraini²⁾, dan Agus Zainal Arifin³⁾

^{1, 2, 3)} Departemen Informatika, Institut Teknologi Sepuluh Noverber
Surabaya, 60111 Indonesia

e-mail: khairiyahaisyah16@gmail.com¹⁾, anggrainisasa@gmail.com²⁾, agusza@cs.its.ac.id³⁾

ABSTRAK

Memahami makna utama yang terkandung dalam beberapa dokumen tentu tidak mudah dan membutuhkan waktu yang cukup lama. Menanggapi masalah tersebut, penelitian terkait peringkasan teks dokumen secara otomatis menjadi perhatian khusus dalam beberapa tahun terakhir. Penelitian ini mengusulkan metode peringkasan teks multi-dokumen yang dapat meningkatkan relevansi antar kalimat dengan menggunakan metode sentence extraction dan word sense disambiguation. Metode sentence extraction yang digunakan didasarkan pada sentence distribution dan part of speech (POS) tagging. Berdasarkan pengujian peringkasan teks dengan metode yang diusulkan, nilai rata-rata ROUGE-1 adalah 0,712, 0,163, 0,247 pada recall, precision, f-measure secara berurutan. Sedangkan hasil pengujian peringkasan teks multi-dokumen tanpa menggunakan word sense disambiguation mendapatkan nilai rata-rata ROUGE-1 sebesar 0,685, 0,139, 0,216 pada recall, precision, f-measure secara berurutan. Hasil penelitian menunjukkan bahwa penggunaan metode sentence extraction dan word sense disambiguation pada peringkasan teks multi-dokumen dapat meningkatkan kualitas hasil peringkasan teks.

Kata Kunci: *Part of speech (POS) tagging, Peringkasan multi-dokumen, Sentence distribution, Word sense disambiguation.*

ABSTRACT

Understanding the main meaning contained in some documents is certainly not easy and requires a long time. In response to the problem, research related to summarizing text documents has automatically been of particular concern in recent years. This study proposes a method of summarizing multi-document texts that can improve relevance between sentences using the method of sentence extraction and word sense disambiguation. The method of sentence extraction used is based on sentence distribution and part of speech (POS) tagging. Based on the text summarization test with the proposed method, the average value of ROUGE-1 is 0.712, 0.163, 0.247 on recall, precision, f-measure in sequence, while the results of testing multi-document text summarize without using word sense disambiguation get an average ROUGE-1 value of 0.685, 0.139, 0.216 on recall, precision, f-measure in sequence. The results of this study indicate that the use of the sentence extraction method and word sense disambiguation on summarizing multi-document text can improve the quality of text summarization.

Keywords: *Multi-documents summarization, Part of speech (POS) tagging, Sentence distribution, Word sense disambiguation.*

I. PENDAHULUAN

Melakukan peringkasan beberapa dokumen tanpa mengurangi kandungan informasi yang terkandung di dalamnya tentu tidak mudah dan membutuhkan waktu yang cukup lama, sehingga hal tersebut dibutuhkan solusi untuk menanganinya. Peringkasan teks dokumen secara otomatis merupakan proses mereduksi ukuran teks menjadi tidak lebih dari setengah teks aslinya. Peringkasan teks dokumen digunakan untuk menyampaikan informasi utama dari kumpulan dokumen secara umum [1]. Ringkasan teks yang dihasilkan dari satu dokumen disebut dengan peringkasan teks dokumen tunggal, sedangkan

ringkasan teks yang dihasilkan dari beberapa dokumen dengan topik yang sama disebut dengan peringkasan teks multi-dokumen [2]. Hasil peringkasan teks dokumen harus dapat mewakili nilai dokumen asli tanpa mengurangi konten maupun nilai informasi yang ada [3].

Sistem peringkasan teks dapat dikategorikan menjadi dua pendekatan, yaitu pendekatan secara ekstraktif dan abstraktif. Pendekatan ekstraktif adalah peringkasan teks dengan memilih kalimat-kalimat penting dari dokumen asli secara statistik. Sedangkan pendekatan secara abstraktif adalah proses peringkasan teks dengan memperhatikan konsep utama dari sebuah teks [3]. Peringkasan teks multi-dokumen dapat dilakukan

dengan memilih kalimat-kalimat representatif sebagai bahan penyusun ringkasan. Pembobotan dilakukan terhadap kalimat-kalimat guna mendapatkan kata-kata penting yang tersebar didalam dokumen untuk mendapatkan hasil ringkasan yang lebih informatif.

Penelitian sebelumnya telah melakukan peringkasan teks multi-dokumen berdasarkan *sentence distribution*. Peringkasan teks dilakukan dengan melakukan pembobotan terhadap setiap kalimat menggunakan *local sentence distribution* dan *global sentence distribution*. Kalimat dengan bobot paling tinggi akan diambil sebagai bahan penyusun ringkasan [4]. Peringkasan teks multi-dokumen yang dilakukan [4] hanya didasarkan pada pemilihan kalimat yang dianggap penting tanpa memperhatikan makna kata yang terdapat didalamnya. Kata-kata yang sama namun sebenarnya memiliki makna yang berbeda akan mempengaruhi hasil pengklasteran kalimat. Hal ini menyebabkan adanya kalimat-kalimat yang sebenarnya tidak relevan tetapi berada dalam satu kluster. Oleh sebab itu, dibutuhkan metode peringkasan multi-dokumen yang dapat memastikan kalimat-kalimat dalam satu kluster menjadi relevan.

Penelitian ini mengusulkan metode peringkasan multi-dokumen yang dapat meningkatkan relevansi antar kalimat dengan menggunakan metode *sentence extraction* dan *word sense disambiguation*. Penggunaan *word sense disambiguation* digunakan untuk mengatasi kata-kata yang serupa namun sebenarnya memiliki makna yang berbeda (polisemi). Pengecekan makna terhadap kata-kata polisemi, diharapkan dapat meminimalisir adanya kalimat yang sebenarnya tidak relevan namun berada dalam satu kluster.

II. STUDI LITERATUR

Penelitian terkait peringkasan teks dokumen banyak dilakukan sebelumnya dengan metode yang berbeda. Penelitian yang dilakukan oleh [5] mengajukan metode peringkasan teks dokumen menggunakan *Integer Linear Programming* (ILP) pada teks dokumen tunggal. Metode pembobotan yang digunakan mengkombinasikan antara cakupan dan posisi dari suatu kalimat untuk menentukan penting atau tidaknya suatu konsep. Penelitian lain melakukan peringkasan teks dokumen tunggal secara statistik pada artikel berita Cina. Penelitian yang dilakukan oleh [6] dimulai dengan mengumpulkan kosakata baru pada setiap artikel berita dan diverifikasi dengan menggunakan layanan terjemahan online. Kemudian, segmentasi kata secara statistik dilakukan dengan menghitung nilai frekuensi relatif dari kata-kata yang sama (n-grams). Seluruh nilai n-grams kemudian dijumlahkan sebagai nilai bobot dari suatu kalimat. Kalimat yang dianggap penting adalah kalimat dengan nilai bobot paling tinggi. penelitian yang dilakukan oleh [5][6] hanya terbatas pada dokumen tunggal. Proses peringkasan teks juga hanya didasarkan pada nilai statistik dari suatu kalimat

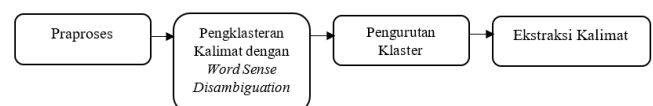
tanpa memperhatikan kandungan makna pada setiap kalimatnya.

Selain peringkasan teks dokumen tunggal, peringkasan teks multi-dokumen juga banyak menjadi perhatian dalam beberapa tahun terakhir. Penelitian oleh [7] melakukan peringkasan teks multi-dokumen pada berita Turki. Proses peringkasan dilakukan dengan terlebih dahulu mengelompokkan kalimat berdasarkan kemiripan topik. Pengelompokan kalimat berdasarkan topik didapat dengan menghitung nilai kemiripan cosine antar kalimat. Proses peringkasan kemudian dilakukan menggunakan *Latent Semantic Analysis* (LSA). Penelitian terkait peringkasan teks multi-dokumen lainnya dilakukan oleh [8]. Penelitian tersebut melakukan peringkasan teks multi-dokumen Bahasa Indonesia berdasarkan *Latent Dirichlet Allocation* (LDA) dan *significance sentence*. Tujuan dari penelitian ini adalah menemukan nilai-nilai optimal dari setiap parameter yang digunakan pada *Latent Dirichlet Allocation* (LDA).

Pada umumnya, penelitian terkait peringkasan teks dokumen hanya memperhatikan frekuensi dan persebaran kata dalam suatu dokumen. Penelitian ini mengusulkan sebuah metode peringkasan teks multi-dokumen berdasarkan metode *sentence extraction* dan *word sense disambiguation*. Proses pengecekan makna suatu kata dalam kalimat dilakukan dengan menggunakan *WordNet*, sedangkan proses *word sense disambiguation* menggunakan algoritma Lesk. Penelitian sebelumnya yang dilakukan oleh [9] menunjukkan bahwa *WordNet* dapat digunakan dalam membedakan makna dari kata polisemi dalam suatu kalimat, sedangkan penelitian [10] menunjukkan bahwa algoritma Lesk pada *WordNet* dapat digunakan dalam proses *word sense disambiguation*.

III. METODE PENELITIAN

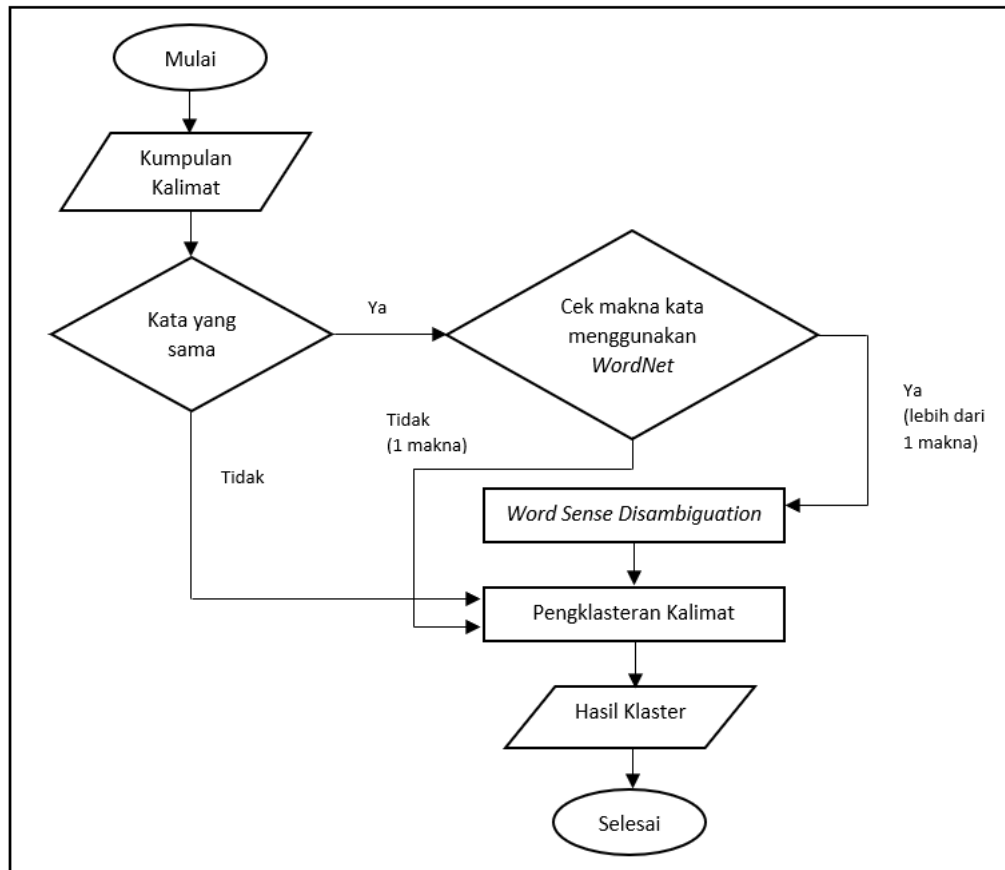
Pada penelitian ini, proses peringkasan teks multi-dokumen didasarkan pada empat tahapan utama, yaitu 1) Praproses, 2) Pengklasteran kalimat dengan *word sense disambiguation*, 3) Pengurutan kluster, dan 4) Ekstraksi kalimat. Secara garis besar, tahapan dalam proses peringkasan teks multi-dokumen dapat dilihat pada Gambar 1.



Gambar 1. Tahapan dalam Peringkasan Teks Multi-Dokumen

A. Praproses

Tahap praproses atau *preprocessing* merupakan



Gambar 2. Pengecekan Term Antara 2 Kalimat

tahapan awal yang dilakukan pada proses peringkasan teks multi-dokumen. Proses *preprocessing* terdiri dari empat tahapan utama yaitu *tokenizing*, *Part of speech (POS) tagging*, *stopwords removal*, dan *stemming*. *Tokenizing* merupakan proses mengubah kalimat-kalimat menjadi *token* atau kata. *Part of speech (POS) tagging* dilakukan untuk memberikan label jenis kata pada setiap kata pada suatu kalimat. Pemberian label jenis kata pada setiap kata dilakukan secara otomatis dengan menggunakan library NLTK pada Python. *Stopwords removal* adalah proses penghapusan kata-kata yang dianggap tidak penting seperti kata depan dan kata penghubung dengan menggunakan kamus stoplist pada NLTK. Tahap terakhir dari *preprocessing* adalah *stemming*. *Stemming* dilakukan untuk mengubah setiap kata menjadi kata dasar. Library yang digunakan dalam proses *stemming* pada penelitian ini adalah *SnowballStemmer*. Dokumen yang telah dilakukan proses *preprocessing* siap digunakan pada proses peringkasan teks multi-dokumen selanjutnya.

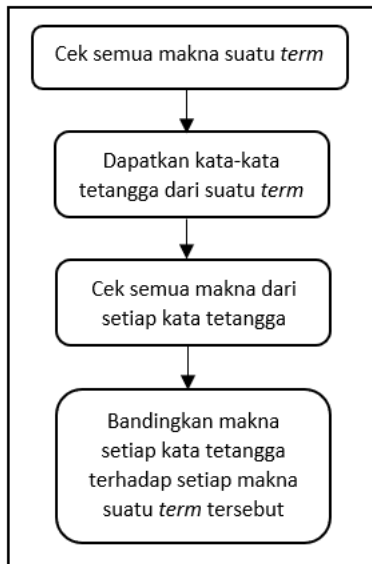
B. Pengklasteran Kalimat dengan Word Sense Disambiguation

Pengklasteran kalimat atau *sentence clustering* merupakan proses pengelompokan kalimat menjadi beberapa kluster. Kalimat dengan nilai similaritas yang

tinggi akan memiliki peluang untuk berada dalam satu kluster. Pengelompokan kalimat didasarkan pada nilai *Histogram Ratio (HR)* yang dimiliki oleh setiap kalimat. Nilai *Histogram Ratio (HR)* inilah yang akan menentukan pada kluster berapa suatu kalimat akan dimasukkan. Secara garis besar, proses pengelompokan kalimat menjadi beberapa kluster dibagi menjadi dua tahapan, yaitu *word sense disambiguation* dan Pembobotan kalimat menggunakan *Histogram Ratio (HR)*.

1) *Word Sense Disambiguation*: *Word sense disambiguation* adalah proses pengecekan makna antar beberapa kata yang serupa. *Word sense disambiguation* digunakan untuk menangani kata-kata dengan tulisan yang sama namun sebenarnya memiliki makna yang berbeda (polisemi). Proses pengecekan makna antar kata dilakukan menggunakan *WordNet*. Pada beberapa waktu terakhir, *WordNet* menjadi populer digunakan sebagai pendekatan dalam membedakan makna dari kata-kata polisemi. *WordNet* sendiri merupakan sebuah database leksikal yang dikembangkan di Universitas Princeton untuk bahasa Inggris [9]. *WordNet* mengatur kata benda, kata kerja, kata sifat dan kata keterangan ke dalam kelompok sinonim dan menggambarkan hubungan antara kelompok-kelompok sinonim tersebut membentuk jaringan semantik antar kata. Proses *word*

sense disambiguation diawali dengan membandingkan setiap term antar kalimat. Pengecekan makna akan dilakukan pada setiap *term* serupa pada kedua kalimat menggunakan *WordNet*. *Term* akan dianggap sama jika keduanya memiliki *synset* yang sama. Tetapi jika *synset* yang dihasilkan berbeda, maka keduanya dianggap *term* yang berbeda. Proses pengecekan *term* antar dua kalimat secara garis besar ditunjukkan pada Gambar 2. Algoritma yang digunakan pada proses *word sense disambiguation* untuk menentukan makna antar *term* adalah algoritma Lesk pada *WordNet*. Algoritma Lesk dilakukan dengan cara mengambil kata-kata tetangga dari *term* target. Setiap kata tetangga akan dilakukan pengecekan makna. Makna dari setiap kata tetangga kemudian akan dihitung kemiripannya dengan setiap *synset* dari *term* target. *Synset* dengan nilai kemiripan tertinggi yang akan dipilih sebagai makna dari suatu *term*. Secara garis besar, proses algoritma Lesk dapat dilihat pada Gambar 3.



Gambar 3. Tahapan Algoritma Lesk

2) Perhitungan Similaritas: Pembobotan dengan menggunakan *Similarity-based Histogram Clustering* (SHC) dilakukan terhadap setiap kalimat pada dokumen. Nilai bobot inilah yang akan digunakan untuk menentukan pada kluster mana suatu kalimat akan bergabung. Nilai kemiripan antar dua kalimat dapat dihitung dengan menjumlahkan kata yang sama antara dua kalimat ($|S_i \cap S_j|$) dibagi dengan jumlah kata pada kedua kalimat ($|S_i| + |S_j|$). Rumus yang digunakan dalam menghitung kemiripan antar dua kalimat S_i dan S_j dapat dilihat pada (1).

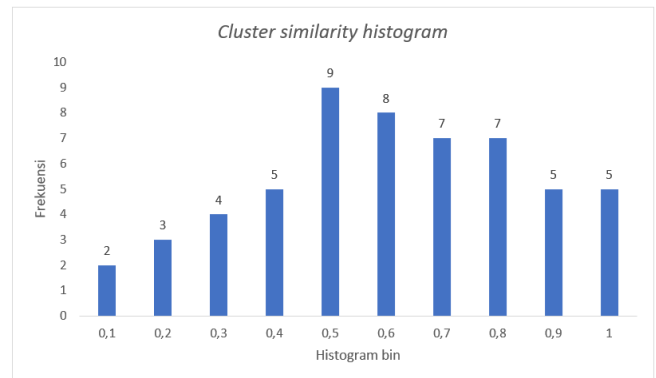
$$sim(S_i, S_j) = \frac{(2 * |S_i \cap S_j|)}{|S_i| + |S_j|} \quad (1)$$

Nilai similaritas antar kalimat pada suatu kluster disimpan dalam sebuah list $Sim=[Sim_1, Sim_2, Sim_3, \dots, Sim_n]$

dengan n merupakan jumlah pasangan kalimat pada suatu kluster.

3) *Similarity histogram* berdasarkan pengklusteran kalimat: Tahap *clustering* pada penelitian ini menggunakan metode penambahan secara dinamis pada tiap kalimat untuk membentuk kluster. Setiap kalimat pada dokumen akan diproses sebanyak satu kali dan ditambahkan pada kluster yang sesuai. Nilai perhitungan untuk menentukan kluster mana suatu kalimat akan diinputkan maka digunakan nilai *similarity histogram*.

Konsep utama dari *Similarity histogram-based clustering* adalah untuk menjaga setiap kluster se-koheren mungkin. Tingkat koheren dari suatu kluster dapat dimonitor melalui *cluster similarity histogram*. Histogram yang dimiliki oleh tiap kluster terdiri dari beberapa bin yang menunjukkan nilai similaritas antar kalimat. Tinggi dari tiap bin merepresentasikan jumlah pasang kalimat yang berada pada interval tersebut. Contoh *cluster similarity histogram* ditunjukkan pada Gambar 4.



Gambar 4. Cluster Similarity Histogram

Nilai similarity histogram suatu cluster dapat dinotasikan dengan $H = \{ h_i, i=1, \dots, n_b \}$,

$$h_i = \text{count}(sim_k) \quad sim_{ii} \leq sim_k \leq sim_{ui}$$

dimana jumlah bin pada histogram dilambangkan dengan n_b , jumlah kalimat pada bin i dilambangkan dengan h_i , nilai similaritas terendah pada bin i dilambangkan dengan sim_{ii} , dan nilai similaritas tertinggi pada bin i dilambangkan dengan sim_{ui} .

Secara garis besar, perhitungan *histogram ratio* tiap kluster ditunjukkan pada (2).

$$HR = \frac{\sum_{i=1}^{n_b} h_i}{\sum_{i=1}^{n_b} h_j} \quad (2)$$

Kalimat akan masuk ke dalam suatu kluster apabila kalimat tersebut memiliki karakteristik yang sesuai. Kluster baru akan dibentuk apabila suatu kalimat tidak sesuai dengan karakteristik kluster manapun.

C. Pengurutan Klaster

Setelah seluruh kalimat bergabung dalam klaster, langkah selanjutnya adalah melakukan pengurutan terhadap nilai klaster. Semakin besar suatu klaster memiliki jumlah kata dengan nilai bobot yang tinggi, semakin besar kemungkinan klaster tersebut berada pada urutan teratas. Perhitungan nilai bobot pada masing-masing klaster ditunjukkan pada (3).

$$weight(c_j) = \sum \log(1 + (count(w))) \quad (3)$$

$Count(w)$ merupakan jumlah kata w pada dokumen input dan nilai $count(w)$ melebihi nilai $threshold$. Seluruh klaster yang telah terbentuk kemudian diurutkan dari nilai bobot yang paling tinggi hingga klaster dengan nilai bobot terendah. Semakin tinggi nilai bobot suatu klaster menunjukkan semakin penting nilai informasi dari klaster tersebut.

D. Ekstraksi Kalimat

Tahap selanjutnya adalah memilih kalimat utama yang dianggap paling penting dari setiap klaster yang disebut dengan ekstraksi kalimat atau *sentence extraction*. Metode *sentence extraction* yang digunakan pada penelitian ini didasarkan pada dua hal, yaitu *part of speech (POS) tagging* dan *sentence distribution*.

1) *POS Tagging*: *POS Tagging* adalah proses pemberian label pada setiap kata. Label yang diberikan didasarkan pada empat komponen utama yaitu a) noun, b) *adjective, verb, participle*, c) adverbs, dan d) tag lainnya [11]. Pemberian POS Tag pada setiap kata dilakukan secara otomatis menggunakan *library NLTK*.

2) *Sentence Distribution*: Distribusi kalimat dalam dokumen dapat dijadikan dasar dalam menentukan penting atau tidaknya suatu kalimat dalam sebuah klaster. Metode *sentence distribution* yang digunakan pada penelitian ini didasarkan pada dua hal, yakni *local sentence distribution* dan *global sentence distribution*.

Local sentence distribution digunakan untuk menentukan penting atau tidaknya suatu kalimat dalam klaster. Kalimat yang terdiri dari kata-kata yang banyak tersebar dalam suatu klaster akan dianggap lebih penting dan mendapat posisi yang lebih tinggi dalam suatu klaster [12]. Sedangkan *global sentence distribution* digunakan untuk menentukan penting atau tidaknya suatu kalimat dalam kumpulan klaster. Suatu kalimat yang memiliki sebaran *term* yang tinggi dalam klaster namun memiliki sebaran yang rendah pada klaster lainnya dianggap lebih penting dan memiliki nilai *global sentence distribution* yang lebih tinggi [12].

Terdapat beberapa tahapan yang perlu dilakukan pada proses *local sentence distribution*, yaitu:

Step 1: Peluang persebaran term

Persebaran p_j dari *term j* didapat dengan membagi jumlah *term* yang berbeda pada kalimat s_i pada klaster k $|s_{ik}|$ dengan total $|s_{ik}|$ pada klaster k . Persebaran *term* pada setiap kalimat pada sebuah klaster secara garis besar ditunjukkan pada (4).

$$p_j = \frac{|s_{ik}|}{\sum |s_{ik}|} \quad (4)$$

Step 2: Persebaran *term* pada kalimat pada suatu klaster

Persebaran *term j* pada kalimat pada klaster k dapat dihitung dengan menggunakan (5).

$$x_{jk}^2 = \sum_{j=1}^{|c_k|} \frac{(n_{ji} - n_{jk}p_j)^2}{n_{jk}p_j}, \quad (5)$$

dimana n_{ji} merupakan frekuensi *term j* pada kalimat i , sedangkan n_{jk} merupakan frekuensi *term j* pada klaster k , dan $|c_k|$ merupakan jumlah term pada kalimat.

Step 3: Tingkat persebaran term pada klaster
Tingkat persebaran *term j* pada klaster k dapat diperoleh dengan menggunakan (6) dan (7).

$$U_{jk} = \frac{1}{1 + x_{jk}^2} \quad (6)$$

$$St_{jk} = \log_2(1 + \frac{p_{jk}}{p_k}), \quad (7)$$

dimana p_{jk} adalah jumlah kalimat yang mengandung *term j* pada klaster k , sedangkan p_k adalah jumlah kalimat pada klaster k .

Step 4: Nilai bobot *term j* pada klaster k

Nilai bobot *term j* pada klaster k (W_{jk}) dapat diperoleh menggunakan (8).

$$W_{jk} = \log_2(1 + U_{jk} * St_{jk}) \quad (8)$$

Step 5: *Local sentence distribution*

Nilai *local sentence distribution* dari suatu kalimat ($W_{ls}(S_{ik})$) didapat dengan menjumlahkan seluruh bobot *term* W_{jk} dari kalimat tersebut dibagi dengan jumlah *term* yang membentuk kalimat s_i $|s_{ik}|$. Secara umum, perhitungan *local sentence distribution* dapat dilihat pada (9).

$$W_{ls}(S_{ik}) = \frac{1}{|s_{ik}|} \sum W_{jk} \quad (9)$$

Sedangkan tahapan yang perlu dilakukan untuk mendapat nilai *global sentence distribution* antara lain:

Step 1: Peluang persebaran *term* pada set klaster

Persebaran p_{jc} dari *term j* pada klaster k didapat dengan membagi jumlah *term* yang berbeda pada klaster k $|C_k|$ dengan total *term* yang berbeda pada seluruh klaster

|C|. Persebaran *term* dalam kluster dapat dihitung dengan menggunakan (10).

$$p_{jc} = \frac{|C_k|}{|C|} \quad (10)$$

Step 2: Persebaran term pada set kluster

Persebaran *term j* pada seluruh kluster dapat dihitung dengan menggunakan (11).

$$x_j^2 = \sum_{k=1}^{|C|} \frac{(n_{jk} - n_j p_{jc})^2}{n_j p_{jc}}, \quad (11)$$

dimana n_{jk} merupakan frekuensi *term j* pada kluster k , sedangkan n_j merupakan frekuensi *term j* pada seluruh kluster, dan $|C|$ merupakan jumlah term pada kluster.

Step 3: Tingkat persebaran *term* pada set kluster

Tingkat persebaran *term j* pada set kluster dapat diperoleh dengan menggunakan (12) dan (13).

$$U_j = \frac{1}{1 + x_j^2} \quad (12)$$

$$St_j = \log_2 \left(1 + \frac{p}{p_j} \right), \quad (13)$$

Dimana p_j adalah jumlah kluster yang mengandung *term j*, sedangkan p adalah jumlah seluruh kluster.

Step 4: Nilai bobot *term j* pada seluruh kluster

Nilai bobot *term j* pada seluruh kluster (W_{jc}) dapat diperoleh menggunakan (14).

$$W_{jc} = \log_2(1 + U_j * St_j) \quad (14)$$

Step 5: *Global sentence distribution*

Nilai *global sentence distribution* dari suatu kalimat ($W_{gs}(S_{ik})$) didapat dengan menjumlahkan seluruh bobot term $W_{t_{g,j}}$ dari kalimat tersebut dibagi dengan jumlah term yang membentuk kalimat s_i $|S_{ik}|$. Secara umum, perhitungan *global sentence distribution* dapat dilihat pada (15).

$$W_{gs}(S_{ik}) = \frac{1}{|S_{ik}|} \sum W_{jk} \quad (15)$$

Nilai bobot tiap kalimat pada suatu kluster didapat dengan mengalikan nilai *local sentence distribution* dan *global sentence distribution* pada kalimat tersebut. Kalimat dengan nilai bobot tertinggi akan dianggap sebagai kalimat penting dan dapat mewakili seluruh kalimat pada kluster tersebut. Perhitungan bobot tiap kalimat berdasarkan *local sentence distribution* dan *global sentence distribution* dapat diperoleh menggunakan (16).

$$weight(S_{ik}) = W_{ls}(S_{ik}) * W_{gs}(S_{ik}) \quad (16)$$

IV. HASIL DAN ANALISIS

Dataset yang digunakan pada penelitian ini adalah data berupa berita yang didapat dari *Document Understanding Conference (DUC) 2004 task 2*. Data DUC terdiri dari 50 topik berita dengan masing-masing topik terdiri dari 10 dokumen berita. Evaluasi pengujian yang dilakukan adalah dengan membandingkan peringkasan teks multi-dokumen tanpa menggunakan *word sense disambiguation* dan peringkasan teks multi-dokumen dengan menggunakan *word sense disambiguation*. Hasil pengujian didapat dengan membandingkan nilai ROUGE-1 dan ROUGE-2. Semakin tinggi nilai ROUGE yang didapat, menunjukkan ringkasan tersebut memiliki kualitas yang lebih baik.

A. Perbandingan Hasil ROUGE-1 dan ROUGE-2

Hasil peringkasan teks multi-dokumen kemudian dihitung menggunakan nilai ROUGE. ROUGE sendiri merupakan sebuah metode untuk menentukan kualitas hasil ringkasan teks dibandingkan dengan hasil ringkasan manual oleh manusia [13]. Dari 50 topik berita yang telah dilakukan peringkasan, perbandingan rata-rata hasil ROUGE-1 dan ROUGE-2 pada peringkasan teks multi-dokumen tanpa menggunakan *word sense disambiguation* dengan peringkasan menggunakan *word sense disambiguation* dapat dilihat pada Tabel I.

TABEL I
Hasil Evaluasi dengan ROUGE

Metode Peringkasan	ROUGE-1			ROUGE-2		
	Recall	Precision	F-measure	Recall	Precision	F-measure
<i>Sentence Extraction</i>	0.685	0.139	0.216	0.323	0.096	0.140
Metode usulan	0.712	0.163	0.247	0.364	0.122	0.173

Dari hasil pengujian didapat bahwa peringkasan teks multi-dokumen dengan hanya menggunakan metode *sentence extraction* memiliki nilai ROUGE-1 sebesar 0.68 dan ROUGE-2 sebesar 0.32. Sedangkan peringkasan teks multi-dokumen dengan menggunakan metode *sentence extraction* dan *word sense disambiguation* mendapatkan hasil ROUGE-1 sebesar 0.71 dan ROUGE-2 sebesar 0.36. Dari hasil pengujian didapatkan bahwa peringkasan teks multi-dokumen dengan memperhatikan *word sense disambiguation* dan *sentence extraction* memiliki nilai ROUGE-1 dan ROUGE-2 yang lebih baik mesti memiliki perbedaan nilai yang tidak terlalu tinggi. Hal ini bisa jadi dikarenakan karena jumlah kata polisemi yang terdapat pada set dokumen tidak terlalu banyak, sehingga hasil ROUGE-1 maupun ROUGE-2 yang didapat tidak memiliki perbedaan yang cukup signifikan dibandingkan dengan peringkasan teks multi-dokumen tanpa menggunakan *word sense disambiguation*.

B. Perbandingan Hasil Peringkasan Teks Multi-dokumen

Hasil peringkasan teks multi-dokumen dengan menggunakan *word sense disambiguation* dan tanpa menggunakan *word sense disambiguation* dapat dilihat pada Tabel II.

V. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, peringkasan teks multi-dokumen dengan hanya menggunakan metode *sentence extraction* memiliki nilai ROUGE-1 sebesar 0.68 dan ROUGE-2 sebesar 0.32. Peringkasan teks multi-dokumen dengan menggunakan metode *sentence extraction* dan *word sense disambiguation* mendapatkan hasil ROUGE-1 sebesar 0.71 dan ROUGE-2 sebesar 0.36. Hasil pengujian menunjukkan bahwa peringkasan teks multi-dokumen dengan menggunakan metode *sentence extraction* dan *word sense disambiguation* dapat meningkatkan kualitas hasil peringkasan teks. Penelitian ini belum mampu menangani kata-kata yang berbeda namun memiliki makna sama (sinonim). Penelitian selanjutnya diharapkan dapat mengembangkan peringkasan teks multi-dokumen dengan memperhatikan *word sense disambiguation* dan sinonim kata.

DAFTAR PUSTAKA

- [1] D. Kim and J.-H. Lee, "Multi-document Summarization by Creating Synthetic Document Vector Based on Language Model," *International Conference on Soft Computing and Intelligent Systems and International Symposium on Advanced Intelligent Systems 978-1-5090-2678-4/16 \$31.00 © 2016 IEEE DOI 10.1109/SCIS&ISIS.2016.159 605 2016 Joint 8th International*, vol. 20, no. 4, pp. 513–532, 2016.
- [2] M. Ozsoy, I. Cicekli, and F. Alpaslan, "Text summarization of turkish texts using latent semantic analysis," *In Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, Stroudsburg, PA, USA. Association for Computational Linguistics*, no. August, pp. 869–876, 2010.
- [3] G. Yapius, A. Erwin, M. Galinium, and W. Muliady, "Automatic Multi-Document Summarization for Indonesian Documents Using Hybrid Abstractive-Extractive Summarization Technique," *International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia*, pp. 0–4, 2014.
- [4] K. Sarkar, "Sentence Clustering-Based Summarization of Multiple Text Documents," *TECHNIA – International Journal of Computing Science and Communication Technologies*, vol. 2, no. 1, pp. 974–3375, 2009.
- [5] H. Oliveira, R. Lima, R. D. Lins, F. Freitas, M. Riss, and S. J. Simske, "A Concept-Based Integer Linear Programming Approach for Single-Document Summarization," *Proceedings - 2016 5th Brazilian Conference on Intelligent Systems, BRACIS 2016*, pp. 403–408, 2017.
- [6] J. Wang and J. Yang, "Statistical Single-Document Summarization for Chinese News Articles," *International Conference on Advanced Information Networking and Applications Workshops*, pp. 183–188, 2012.
- [7] F. Demirci, E. Karabudak, and B. Ilgen, "Multi-document summarization for Turkish news," *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017.
- [8] W. Agus, K. Retno, and S. Bayu, "Multi Document Summarization for the Indonesian Language Based on Latent Dirichlet allocation and Significance sentence," *International Conference on Information and Communications Technology (ICOIACT)*, pp. 520–524, 2018.
- [9] U. R. Dhungana, S. Shakya, and K. Baral, "Word Sense Disambiguation Using WSD Specific Wordnet of Polysemy Words," *International Journal on Natural Language Computing*, vol. 3, no. 4, pp. 29–41, 2013.
- [10] K. Samhith, S. A. Tilak, and G. Panda, "Word Sense Disambiguation using WordNet Lexical Categories," *International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)-2016*, pp. 1664–1666, 2016.
- [11] O. P. K. Umam, FW Putro, G Qorik, "Coverage, Diversity, and Coherence Optimization for Multi-Document Summarization," *Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)*, vol. 1, pp. 1–10, 2015.
- [12] A. Z. Arifin, M. Z. Abdullah, and A. W. Rosyadi, "Sentence Extraction Based on Sentence Distribution and Part of Speech Tagging for Multi-document Summarization," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 2, pp. 843–851, 2018.
- [13] K. Ganesan, "Evaluation of Summarization Tasks [Online]." Tersedia : <https://github.com/RxNLP/ROUGE-2.0> .

TABEL II
Perbandingan Hasil Peringkasan

Metode <i>Sentence Extraction</i>	Metode Usulan
<p>The deaths and the ostentatious display of the heads by killers prompted neighboring Russian regions to close off roads into Chechnya, and caused an outpouring of outrage and lament from Russian and foreign officials.</p> <p>The chief executive of Granger Telecom, Ray Verth, said Tuesday that Chechen authorities had mounted a rescue attempt Monday night that went tragically wrong.</p> <p>The men, who were working to restore phone services in the breakaway Russian republic, had been abducted by gunmen in October.</p> <p>French President Jacques Chirac spoke to President Boris Yeltsin by telephone and thanked him for Russia's part in freeing Cochetel, the Kremlin press service said.</p> <p>Chechen President Aslan Maskhadov said Wednesday that one alleged kidnapper had been arrested and authorities gleaned enough information to launch a rescue operation for the hostages, three Britons and a New Zealander.</p> <p>The videotape, shown Thursday by Chechnya's vice president, showed hostage Peter Kennedy claiming that the four men had been sent to Chechnya to listen in on phone conversations in the breakaway republic.</p> <p>But during the rescue attempt the kidnappers panicked and decapitated the captives, Maskhadov said, according to Russian media reports.</p> <p>Since Chechnya's two-year war of independence against Russia ended in 1996, hundreds of people have been kidnapped in the breakaway republic and neighboring regions by armed gangs who are mostly motivated by ransom.</p> <p>Chechen acting First Vice Premier Turpal Atgeriyev, who is heading the investigation into the killings, said authorities had several suspects in the abduction, but have failed to make any progress, according to the ITAR-Tass news agency.</p> <p>Chechnya's top prosecutor, who was investigating the killings and was apparently kidnapped Thursday, was also released Saturday, Interfax said.</p> <p>No ransom was paid for Cochetel's release, Mikhailov said.</p> <p>Chechen President Aslan Maskhadov will allow British government agents to come to Chechnya to investigate the killing, Atgeriyev said, according to the Interfax news agency.</p> <p>A French United Nations official kidnapped in southern Russia more than 10 months ago has been freed and was flown to Moscow Saturday, news reports reported.</p> <p>A truck blew up when it drove over a land mine in breakaway Chechnya on Friday, killing two people and injuring two others, a news agency said.</p> <p>Mansur Tagirov vanished Thursday evening while returning to Chechnya's capital city Grozny from a small village nearby, news reports said.</p> <p>Mansur Tagirov vanished Thursday evening while returning to Chechnya's capital city Grozny from a small village nearby, news reports said.</p> <p>Spokesmen at the British and New Zealand embassies in Moscow said they were checking on reports the four men had been killed, but could not comment.</p> <p>The government has said it was trying to rescue the men at the time of their deaths, which have outraged Britain and Russia.</p>	<p>A concrete criminal group was identified, an organizer of the abductions was arrested and an approximate location of the hostages was determined, Maskhadov said in a statement.</p> <p>In footage of the operation broadcast by Russia's NTV television station, Cochetel was shown lying face-down, handcuffed, behind a white van.</p> <p>Vincent Cochetel, 37, who headed the regional office of the U.N. High Commissioner for Refugees in North Ossetia, was seized by three masked gunmen in the region's capital of Vladikavkaz on Jan. 29.</p> <p>The videotape, shown Thursday by Chechnya's vice president, showed hostage Peter Kennedy claiming that the four men had been sent to Chechnya to listen in on phone conversations in the breakaway republic.</p> <p>French President Jacques Chirac spoke to President Boris Yeltsin by telephone and thanked him for Russia's part in freeing Cochetel, the Kremlin press service said.</p> <p>Chechen authorities found the decapitated heads of four kidnapped foreigners Tuesday along a highway near a remote village after a two-month search in the breakaway region in southern Russia.</p> <p>No ransom was paid for Cochetel's release, Mikhailov said.</p> <p>The chief executive of Granger Telecom, Ray Verth, said Tuesday that Chechen authorities had mounted a rescue attempt Monday night that went tragically wrong.</p> <p>He planned to discuss the killings with Russian Foreign Minister Igor Ivanov, who is in Brussels for talks at NATO.</p> <p>But during the rescue attempt the kidnappers panicked and decapitated the captives, Maskhadov said, according to Russian media reports.</p> <p>Meanwhile, Chechen Deputy Prime Minister Turpal Atgeriyev said authorities had arrested a man who confessed to kidnapping the four foreigners.</p> <p>He said the bodies had not been found.</p> <p>Chechen acting First Vice Premier Turpal Atgeriyev, who is heading the investigation into the killings, said authorities had several suspects in the abduction, but have failed to make any progress, according to the ITAR-Tass news agency.</p> <p>It was unclear where Cochetel was held, in the breakaway republic of Chechnya or in North Ossetia.</p> <p>Mansur Tagirov vanished Thursday evening while returning to Chechnya's capital city Grozny from a small village nearby, news reports said.</p> <p>A Russian soldier taken hostage during the war was released Friday after a year of efforts to free him, the ITAR-Tass news agency reported.</p> <p>It was unclear where Cochetel was held, in the breakaway republic of Chechnya or in North Ossetia.</p> <p>Tagirov was investigating the deaths Britons Peter Kennedy, Darren Hickey, and Rudolf Petschi, and Stanley Shaw of New Zealand.</p>