

# Clustering Peringkasan Teks Otomatis Dokumen Berita menggunakan Metode K-Means

## *Clustering Automatic Text Summarization of News Documents using the K-Means Method*

Ferdian Noprianto<sup>\*1</sup>, Surya Agustian<sup>2</sup>, Muhammad Irsyad<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau  
e-mail: <sup>\*1</sup>ferdiannoprianto@gmail.com

**Abstrak** – Perkembangan teknologi informasi yang pesat menyebabkan peningkatan secara drastis jumlah dokumen teks, artikel dan berita yang tersedia secara digital. Peningkatan jumlah dokumen tersebut menimbulkan kesulitan dalam memilih dan menyaring informasi yang relevan dengan efisien. Peringkasan teks otomatis menjadi solusi penting dalam memberikan informasi secara ringkas dan cepat. Penelitian ini mengusulkan peringkasan teks otomatis dengan teknik clustering menggunakan metode K-Means. Word2Vec digunakan untuk kalkulasi word embeddings dari kalimat-kalimat yang terdapat dalam setiap dokumen. Hasil ringkasan diperoleh dengan mengambil 1 kalimat yang memiliki jarak terdekat antara sentence embeddings-nya dengan centroid, dari masing-masing klaster. Evaluasi hasil ringkasan pada data testing dibandingkan dengan ringkasan manual buatan manusia menggunakan ROUGE, dengan F1-score sebagai metrik pengukuran utama. Metode ini cukup baik untuk tingkat kompresi yang kecil, yaitu F1-score rata-rata 51.96% untuk tingkat kompresi 30%. Sedangkan untuk tingkat kompresi 50% F1-score yang dicapai masih rendah.

Kata kunci – Peringkasan Teks Otomatis, K-Means Clustering, ROUGE, Word2vec

**Abstract** - The rapid development of information technology has led to a drastic increase in the number of text documents, articles and news available digitally. These large amount of documents makes people get difficulties in efficiently selecting and filtering relevant information. Automatic text summarization is an important solution in providing information concisely and quickly. This study proposes automatic text summarization with clustering techniques using the K-Means method. Word2Vec is used to calculate word embeddings from the sentences in each document. The summary is obtained by taking 1 sentence from each cluster, that has the shortest distance between the sentence embeddings and the centroid. The produced summaris od the data testing are compared with man-made summaries by using ROUGE, with F1-score as main measurement metric. The results obtained by this method are quite good for small compression rates, which is 51.96% for F1-score of ROUGE-L. Meanwhile, for the larger compression rates this method achieved a low average of F1-score.

Keywords – Automated Summarization, K-Means Clustering, ROUGE, Word2vec

## I. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah memicu peningkatan jumlah dokumen yang signifikan di internet. Tidak terkecuali dalam pemberitaan media *online*, artikel berita bertambah dalam jumlah yang massif dan melimpah. Namun, isi artikel terkadang banyak yang tidak penting dan perlu diketahui, atau berguna bagi pembaca. Sehingga membaca keseluruhan isi artikel akan menyulitkan dan membuat waktu, dan informasi yang diperoleh tidak semua relevan bagi pengguna. Oleh karena itu, peringkasan teks otomatis menjadi salah

satu solusi yang penting untuk memproses dan menyajikan informasi yang penting secara ringkas.

Peringkasan teks otomatis (*automatical text summarization*) adalah suatu teknik peringkasan teks dokumen secara otomatis menggunakan bantuan aplikasi komputer, untuk mengekstrak intisari informasi yang penting dari dalam dokumen sumber yang ada [1]. Peringkasan teks otomatis merupakan sebuah kecerdasan buatan (AI) di mana komputer dapat meringkas teks dalam kalimat atau dokumen menjadi bentuk yang lebih singkat untuk memudahkan orang menarik kesimpulan dari artikel panjang tanpa harus membaca teks keseluruhan dokumen [2].

Beberapa metode yang pernah dipublikasikan antara lain adalah menggunakan graf dan algoritma semut [1], *TextRank* [3], *LexRank* [4], beberapa yang disebutkan di dalam [5], seperti MMR, *clustering*, *machine learning* dan *deep learning*. Literatur *review* pada [5] menyatakan bahwa untuk penelitian peringkasan otomatis dokumen bahasa Indonesia masih cukup jarang, sehingga relatif sulit untuk dapat membandingkan hasil-hasil penelitian, disebabkan karena *dataset* yang digunakan berbeda. Di samping itu, pengukuran evaluasi metode yang dipakai juga masih belum standar.

Penelitian ini mengusulkan metode peringkasan dokumen untuk bahasa Indonesia dengan mengikuti pola penelitian sebagaimana pada [3] dan [4], dan menggunakan *dataset* yang sama sebagai *benchmark* evaluasi dan pengembangan metode. Pengukuran evaluasi hasil ringkasan menggunakan *ROUGE*, dengan *F1-score* sebagai metrik utama.

*TextRank* yang diterapkan di dalam [3] dan *LexRank* di dalam [4] memodifikasi cara pemilihan kalimat untuk menjadi ringkasan, dengan mengambil *ranking similarity score* dari yang tertinggi (*top-n*), terendah (*bottom-n*), atau yang di pertengahan (*middle-n*), di antara seluruh pasangan kalimat. Posisi mana yang menghasilkan *f1-score* pada *ROUGE-L* yang tertinggi lah yang akan dipilih sebagai model optimal.

Hasil ringkasan metode *LexRank* pada [4] untuk tingkat kompresi 50%, menghasilkan skor *f1-score* sebesar 67,05% untuk *ROUGE-L*. Sedangkan pengujian untuk tingkat kompresi 30%, rata-rata skor *f1-score* 54,76% untuk *ROUGE-L*. Sedangkan metode *TextRank* pada [3], memberikan hasil *f1-score* rata-rata 68,29% untuk *ROUGE-L* pada data *testing* dengan kompresi 50%. Hasil ini lebih tinggi daripada metode *LexRank*. Sementara itu, hasil ringkasan untuk tingkat kompresi 30% memberikan *f1-score* sebesar 43,59%, lebih rendah dibandingkan dengan *LexRank*.

Penelitian ini mengusulkan metode *clustering* untuk membangkitkan ringkasan dari dokumen. Sejumlah *n cluster* dibentuk berdasarkan level kompresi yang diinginkan. Kemudian dari setiap *cluster*, dihitung pusat *cluster*-nya (*centroid*), dan dihitung jarak antara setiap kalimat terhadap *centroid* tersebut. Kalimat dengan jarak terendah akan dipilih untuk menjadi ringkasan. Pengukuran jarak dilakukan dengan mengubah terlebih dahulu kalimat menjadi vektor kalimat, yang dihitung dari *pretrained word embeddings* Bahasa Indonesia dengan metode *word2vec* [5].

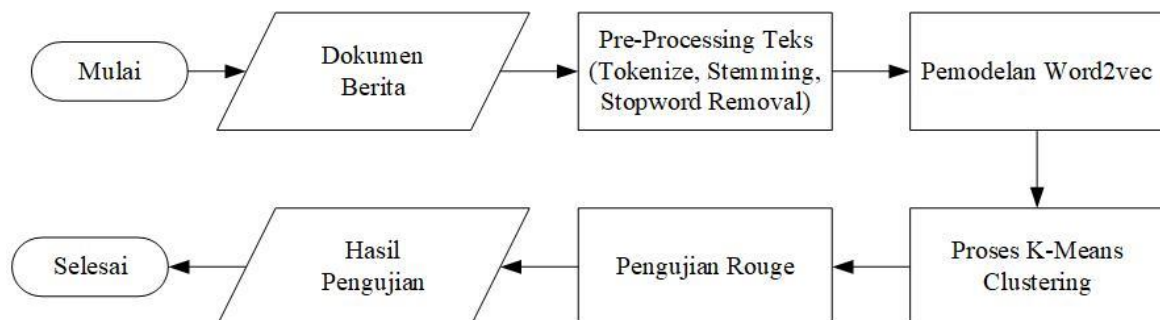
Bagian selanjutnya dari *paper* ini akan menjelaskan metode yang dikembangkan untuk peringkasan dokumen otomatis penelitian ini. Pada bab 3 akan dibahas mengenai hasil implementasi metode untuk pencarian model optimal, dan analisa hasil ringkasan pada data uji. Bagian terakhir ditutup dengan kesimpulan dan saran untuk penelitian selanjutnya.

## II. METODE

Penelitian ini dilakukan sebagaimana tahapan di dalam Gambar 1. *Dataset* yang dipakai diperoleh dari 300 artikel dokumen berita yang diambil dari *website-website* berita seperti *detik.com*, *kompas.com*, *tribunnews*, *kapanlagi.com*, *okezone.com*, dan sebagainya. Setiap kalimat di dalam dokumen diberikan skor oleh dua orang anotator. Kemudian diambil  $N$  kalimat dari total kalimat di dalam dokumen dengan skor tertinggi rata-rata dari dua penilai, sebagai ringkasan manual, sesuai dengan tingkat kompresi yang diinginkan. Hasil ringkasan manual ini disebut sebagai *gold standard*, yang digunakan untuk pengukuran performa metode. Dari 300 data, diambil sebanyak 150 data artikel sebagai data *training* atau data pengembangan sistem. Sedangkan 150 lainnya digunakan sebagai data uji.

Langkah pertama dari metode peringkasan teks otomatis yang diusulkan pada penelitian ini adalah *Text Preprocessing* (*tokenizing*, *stemming* dan *stopword removal*). Langkah kedua adalah menghitung vektor kalimat menggunakan *pretrained model Word2vec* Bahasa Indonesia. Langkah ketiga adalah proses klusterisasi teks berdasarkan vektor kalimat, kemudian diambil kalimat-kalimat yang paling dekat dengan pusat kluster (*centroid*). Setiap kluster dapat terdiri atas 1 kalimat atau lebih. Kalimat yang dipilih sebagai ringkasan adalah yang paling dekat jaraknya dengan *centroid*.

Setelah ringkasan diperoleh, hasilnya dibandingkan dengan *gold standard* menggunakan *ROUGE score*. Beberapa alternatif pengujian untuk mendapatkan model ringkasan yang optimal dilakukan. Model terbaik berdasarkan *F1-score* dari *ROUGE-L* yang tertinggi terhadap ringkasan data *training*, dipilih sebagai model final. Selanjutnya model final ini digunakan untuk meringkas artikel-artikel pada data *testing*.



Gambar 1. Tahapan Penelitian

### A. Pre-processing Data

#### 1). Tokenize

Pada tahap ini teks atau kalimat dipecah menjadi unit-unit yang disebut *gram* atau *token*. *Token* dapat berupa kata tunggal atau frase (*word n-gram*) dan berupa karakter/huruf (*character n-gram*), misalnya  $n=3,4,5$  dan seterusnya. Tujuan dari tokenisasi adalah untuk mempermudah pemrosesan dan analisis teks dengan memecahnya menjadi unit-unit yang lebih terdefinisi.

#### 2). Stemming

Dalam proses *stemming*, akhiran atau awalan dari kata-kata dihapus untuk mencapai bentuk dasar kata tersebut. Misalnya, kata-kata seperti "berlari", "berlarian", dan "berlariilah" akan distem menjadi bentuk dasar "lari". *Stemming* dapat membantu mengurangi variasi morfologi dalam teks seperti bentuk kata yang bervariasi (infleksi) atau kata-kata yang berasal dari akar yang sama.

### 3). *Stopword Removal*

*Stopword removal* merupakan proses dalam NLP yang melakukan penghapusan kata-kata yang umum atau terlalu sering muncul pada dokumen, dan tidak memiliki signifikansi untuk pemrosesan teks. *Stopword* biasanya berupa kata hubung, kata ganti, preposisi, partikel, seperti "dan", "atau", "juga", "dari", "pada", "walau", "pun", "yang", jarang memberikan kontribusi signifikan terhadap pemahaman dan analisis konten teks.

Tujuan utama dari *stopword removal* adalah untuk menghilangkan *noise* atau pengaruh yang tidak relevan dalam teks dan memfokuskan pada kata-kata kunci atau kata-kata yang lebih informatif [6]. Dengan menghapus *stopword*, ukuran korpus teks dapat dikurangi, dan proses pemrosesan dan analisis teks dapat dipercepat.

### B. *Word Embeddings Word2vec*

Pemodelan *Word2Vec* [7] adalah salah satu metode yang populer dalam pemrosesan bahasa alami (NLP) untuk merepresentasikan kata-kata dalam bentuk vektor numerik berdasarkan konteks kata-kata tersebut dalam teks [8]. Tujuan utama dari *Word2Vec* adalah untuk menangkap makna semantis kata-kata dan hubungan antara kata-kata dalam ruang vektor yang kontinu.

Fitur kalimat diekstraksi dengan model *Word2Vec* bahasa Indonesia yang sudah dilatih sebelumnya<sup>1</sup> (*pretrained*). *Word embeddings* yang dihasilkan dari metode *Word2Vec*, mengubah kata-kata menjadi bentuk vektor angka, sehingga dapat dianalisis dan dikomputasi untuk berbagai kebutuhan, seperti untuk klasifikasi sentimen [8 - 11]. Vektor kata yang dihasilkan oleh *word2vec* dapat menggambarkan hubungan semantik antar kata, misalnya kata "raja" dan "kerajaan", dibandingkan dengan vektor "presiden" dan "negara". Maka jarak antara vektor kata "raja" dan "kerajaan" akan dekat, dibandingkan antara jarak antara kata "raja" dan "negara". Demikian juga untuk kata "presiden" dengan kata "negara" dibandingkan dengan kata "kerajaan".

Untuk membentuk vektor kalimat (*sentence vector*), maka dilakukan penjumlahan elemen-wise antara vektor kata-kata penyusun kalimat, sebagaimana pada [12]. Dalam penelitian ini, vektor kalimat dinormalisasi dengan panjang kalimat (jumlah kata di dalam kalimat), sebagaimana persamaan (1).

$$StcVector = \frac{1}{N} \sum_i^N wordVector_i \dots(1)$$

Dimensi *Word2Vec* merupakan jumlah dimensi dalam ruang vektor di mana kata-kata direpresentasikan. Setiap kata dalam model *Word2Vec* direpresentasikan sebagai vektor numerik dalam ruang multidimensional, di mana setiap dimensi mewakili fitur atau atribut tertentu dari kata tersebut. Dalam penelitian ini menggunakan dimensi *Word2Vec* sebesar 100 dan 200. Tiap nilai dalam vektor tersebut menggambarkan tingkat kehadiran atau makna dari atribut tertentu dalam kata tersebut, seperti makna semantik, hubungan antarkata, atau pola linguistik.

### C. *Proses Clustering*

*Clustering* adalah bidang *data mining* yang melakukan pengelompokan data ke dalam dua atau lebih kelompok berdasarkan kemiripan ciri-ciri (fitur) yang dimiliki sampel di dalam *dataset*. Salah satu metode *clustering* yang populer adalah *K-means*. Dalam tugas-tugas NLP, pengelompokan data teks banyak dilakukan untuk berbagai tujuan. Antara lain seperti pada

[13], yang mengelompokkan data buku berdasarkan beberapa fitur. Sedangkan pada [14] metode clustering *K-means* digunakan untuk mengelompokkan obat-obatan berdasarkan beberapa ciri yang ditentukan.

Proses *K-Means Clustering* dilakukan dengan langkah-langkah sebagai berikut:

1. Menentukan jumlah *cluster* ( $K$ ): Nilai  $K$  harus ditentukan sebelumnya berdasarkan pengetahuan domain atau melalui analisis dan evaluasi yang iteratif.
2. Pemilihan pusat *cluster* awal: Langkah selanjutnya adalah memilih secara acak  $K$  titik awal sebagai pusat *cluster*. Pusat *cluster* ini akan mewakili lokasi kelompok yang akan dibentuk.
3. Mengelompokkan data: Setiap data kemudian akan ditempatkan ke dalam *cluster* terdekat berdasarkan jarak *Euclidean* antara data dan pusat *cluster* (*centroid*), sebagaimana persamaan (2).

$$\text{Jarak}(\text{VecVector}, \text{centroid}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \dots (2)$$

Dengan  $x_i$  adalah elemen vektor kalimat, dan  $y_i$  adalah elemen vektor *centroid*, dan  $n$  adalah dimensi dari vektor kalimat (dimensi *word embeddings*)

4. Menghitung pusat *cluster* baru: Setelah semua data ditempatkan dalam kelompok, pusat *cluster* baru dihitung sebagai rata-rata dari semua data yang ada di dalam kelompok. Pusat *cluster* baru ini akan menjadi representasi baru untuk *cluster* tersebut.
5. Iterasi: Langkah-langkah 3 dan 4 akan diulang secara iteratif hingga tidak ada perubahan anggota *cluster* lagi atau perubahan *cluster* sangat minimal. Hal ini menandakan algoritma *K-Means Clustering* sudah konvergen.

Selain *K-means*, metode *clustering* lainnya adalah *K-medoids*, yang cara kerjanya mirip dengan *K-means*. Pada *K-means*, pusat *cluster* ditentukan dari rata-rata fitur anggota *cluster*-nya, sedangkan *K-medoids* mengambil salah satu anggota sebagai pusat *cluster*-nya. Baru kemudian anggota *cluster* diukur jaraknya terhadap pusat *cluster* tersebut. Penelitian [15] membandingkan performa *clustering* menggunakan *K-means* dan *K-medoids* untuk data obat.

Penelitian ini membentuk *cluster* berdasarkan fitur dari *sentence vector* teks. Jumlah *cluster*  $K$  ditentukan dari tingkat kompresi yang dipilih, berdasarkan persentase jumlah kalimat ringkasan terhadap jumlah kalimat di dalam dokumen yang diringkaskan. Misalnya suatu dokumen tersusun atas 50 kalimat, maka tingkat kompresi 50% akan memilih 25 kalimat dari dokumen. Sehingga jumlah *cluster* yang dibutuhkan adalah 25 *cluster*. Proses pembentukan *cluster* menggunakan *libray pyhton sklearn* [16].

#### D. Evaluasi *ROUGE score*

Untuk mengukur peforma sistem maka digunakanlah pengujian *ROUGE*, Pengujian *ROUGE* (*Recall-Oriented Understudy for Gisting Evaluation*) adalah metode evaluasi yang umum digunakan dalam penilaian dan perbandingan sistem peringkasan teks atau sistem pemerinkkatan otomatis. *ROUGE* digunakan untuk mengukur sejauh mana sebuah ringkasan teks yang dihasilkan oleh sistem peringkasan mencerminkan informasi yang terdapat dalam dokumen sumber.

##### 1) *ROUGE-N Recall* dan *Precision*

*ROUGE-N* merupakan metrik mengukur kesamaan *n-gram* antara ringkasan teks yang dihasilkan dan dokumen *gold standard*. Nilai  $N$  menunjukkan jumlah kata yang dibandingkan. Misalnya, *ROUGE-1* akan mengukur kesamaan *unigram* (*word 1-gram*),

ROUGE-2 akan mengukur kesamaan *bigram* (*word 2-gram*), dan seterusnya. Metrik yang diukur adalah *Recall* dan *Precision*, dihitung melalui persamaan (3-6)

$$R1 \text{ Recall} = \frac{\text{jumlah unigram kata yang sama}}{\text{Total kata di ringkasan manual}} \dots (3)$$

$$R1 \text{ Precision} = \frac{\text{jumlah unigram kata yang sama}}{\text{Total kata di ringkasan sistem}} \dots (4)$$

$$R2 \text{ Recall} = \frac{\text{jumlah bigram kata yang sama}}{\text{Total kata di ringkasan manual}} \dots (5)$$

$$R2 \text{ Precision} = \frac{\text{jumlah bigram kata yang sama}}{\text{Total kata di ringkasan sistem}} \dots (6)$$

## 2) ROUGE-L

Metrik ini mengukur kesamaan dalam hal urutan kata (*longest common subsequence*) antara ringkasan dan dokumen *gold standard*. ROUGE-L memberikan perhatian pada urutan kata yang penting dalam menjaga kohesi dan koherensi ringkasan. *Recall* dan *Precision* untuk ROUGE-L dihitung dengan persamaan (7-8).

$$RL \text{ Recall} = \frac{\text{LCS(system,gold standard)}}{\text{Total kata di ringkasan manual}} \dots (7)$$

$$RL \text{ Precision} = \frac{\text{LCS(sistem,gold standard)}}{\text{Total kata di ringkasan sistem}} \dots (8)$$

## 3) F-measure

*F-Measure* adalah suatu metrik pengukuran yang diturunkan dari *E-Measure*, yang menggabungkan *recall* dan *precision* dengan bobot tertentu, dengan jumlah bobot keduanya adalah 1. Bila ingin berfokus pada *recall*, maka bobot *recall* dibuat lebih besar dari 0.5, dan sebaliknya. Ketika kita ingin mengukur performa sistem dengan *recall* dan *precision* yang seimbang, keseimbangan yang baik, disebut sebagai *F1-measure*, dan lebih populer dengan *F1-score*. *F1-score* (pengukuran seimbang antara *precision* dan *recall*) lebih banyak dipakai untuk evaluasi di berbagai tugas-tugas penelitian NLP. Untuk ROUGE-1, ROUGE-2, dan ROUGE-L, *F1-score* dihitung dengan memasukkan nilai *precision* dan *recall* masing-masing pada persamaan (9).

$$f1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots (9)$$

## III. HASIL DAN PEMBAHASAN

### A. Set-Up Penelitian

Skema eksperimen yang dilakukan mengikuti cara yang digunakan dalam [3] dan [4]. Dataset yang terdiri dari 300 artikel dari berbagai sumber media online, dibagi menjadi dua bagian, yaitu 150 artikel digunakan sebagai data untuk pengembangan sistem (*training*), dan 150 lainnya digunakan sebagai data uji. Tingkat kompresi (CR=*compression rate*) adalah ukuran ringkasan yang terbentuk berbanding ukuran artikel asal, yaitu ditetapkan sebesar 30% dan 50%. Model *word embeddings word2vec* yang diperiksa adalah berdimensi 100 dan 200 untuk mencari model optimalnya.

Untuk *scoring*, digunakan nilai rata-rata untuk *Precision*, *Recall*, dan *F1-score* dari semua artikel yang sudah diringkaskan. Berdasarkan nilai rata-rata yang terbaik dari tingkat kompresi, dipilih model final yang akan diterapkan untuk membentuk ringkasan pada data uji.

### B. Model K-Means Clustering untuk CR 30% pada Data Training

Tabel 1 dan 2 berikut ini adalah hasil ringkasan yang dihasilkan menggunakan tingkat kompresi 30% dengan dimensi *word2vec* 100 dan 200 pada data *training*. Hasil ini mencakup

nilai *precision*, *recall*, untuk skor *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*.

**Tabel 1.** Pengujian Data *Training* dengan CR 30% dan dimensi *Word2vec* 100

No	Artikel ID	Rouge-1			Rouge-2			Rouge-L		
		R	P	F1	R	P	F1	R	P	F1
1	Doc-001	42.18	81.65	55.62	32.61	74.47	45.36	42.18	81.65	55.62
2	Doc-002	50.00	86.21	63.29	40.00	80.00	53.33	50.00	86.21	63.29
3	Doc-003	43.06	79.49	55.86	33.68	68.42	45.14	42.36	78.21	54.95
...	...	...	...	...	...	...	...	...	...	...
148	Doc-148	27.50	55.00	36.67	11.64	27.42	16.35	25.00	50.00	33.33
149	Doc-149	62.50	84.91	72.00	51.85	82.35	63.64	62.50	84.91	72.00
150	Doc-150	50.00	78.01	60.94	37.44	68.55	48.43	49.33	76.96	60.12
<b>Rata-rata</b>		<b>42.29</b>	<b>70.90</b>	<b>52.63</b>	<b>30.42</b>	<b>57.86</b>	<b>39.61</b>	<b>41.70</b>	<b>69.91</b>	<b>51.90</b>

**Tabel 2.** Pengujian Data *Training* dengan CR 30% dan dimensi *Word2vec* 200

No	Artikel ID	Rouge-1			Rouge-2			Rouge-L		
		R	P	F1	R	P	F1	R	P	F1
1	Doc-001	41.71	86.27	56.23	32.92	77.37	46.19	41.23	85.29	55.59
2	Doc-002	32.00	55.17	40.51	20.00	42.86	27.27	32.00	55.17	40.51
3	Doc-003	48.61	80.46	60.61	37.31	69.90	48.65	47.92	79.31	59.74
...	...	...	...	...	...	...	...	...	...	...
148	Doc-148	35.83	69.35	47.25	21.92	50.00	30.48	35.00	67.74	46.15
149	Doc-149	48.61	66.04	56.00	33.33	52.94	40.91	45.83	62.26	52.80
150	Doc-150	52.68	81.35	63.95	41.41	74.60	53.26	52.35	80.83	63.54
<b>Rata-rata</b>		<b>42.46</b>	<b>71.80</b>	<b>52.98</b>	<b>30.84</b>	<b>59.41</b>	<b>40.28</b>	<b>41.89</b>	<b>70.85</b>	<b>52.28</b>

Dari data *training* didapatkan hasil rata-rata untuk *F1-score* pada Model *K-Means Clustering* untuk CR 30% dengan dimensi *word2vec* 100 adalah 51.94%. Sedangkan bila menggunakan dimensi *word2vec* 200 mendapatkan *F1-score* 52.28%. Maka untuk tingkat kompresi 30%, model yang dipakai adalah model dengan dimensi *word2vec* 200.

### C. Model *K-Means Clustering* untuk CR 50% pada Data *Training*

Hasil ringkasan untuk data *training* dengan tingkat kompresi 50% dilaporkan pada Tabel 3 dan 4, dengan dimensi *word2vec* 100 dan 200.

**Tabel 3.** Pengujian Data *Training* dengan CR 50% dan dimensi *Word2vec* 100

No	Artikel ID	Rouge-1			Rouge-2			Rouge-L		
		R	P	F1	R	P	F1	R	P	F1
1	Doc-001	62.09	77.51	68.95	54.97	70.24	61.67	61.61	76.92	68.42
2	Doc-002	64.00	64.00	64.00	50.00	54.55	52.17	64.00	64.00	64.00
3	Doc-003	70.14	78.29	73.99	65.80	70.17	67.91	69.44	77.52	73.26
...	...	...	...	...	...	...	...	...	...	...
148	Doc-148	42.50	54.84	47.89	21.23	28.97	24.51	39.17	50.54	44.13
149	Doc-149	66.67	47.06	55.17	52.78	43.51	47.70	65.28	46.08	54.02
150	Doc-150	65.77	72.86	69.14	52.42	62.96	57.21	64.77	71.75	68.08
<b>Rata-rata</b>		<b>58.31</b>	<b>65.61</b>	<b>61.36</b>	<b>45.80</b>	<b>54.04</b>	<b>49.25</b>	<b>57.64</b>	<b>64.84</b>	<b>60.65</b>

**Tabel 4.** Pengujian Data *Training* dengan CR 50% dan dimensi *Word2vec* 200

No	Artikel ID	Rouge-1			Rouge-2			Rouge-L		
		R	P	F1	R	P	F1	R	P	F1
1	Doc-001	63.51	74.86	68.72	56.21	68.05	61.56	63.51	74.86	68.72
2	Doc-002	64.00	64.00	64.00	50.00	54.55	52.17	64.00	64.00	64.00

3	Doc-003	68.75	75.00	71.74	60.10	62.37	61.21	68.06	74.24	71.01
...	...	...	...	...	...	...	...	...	...	...
148	Doc-148	48.33	64.44	55.24	30.82	43.69	36.14	45.83	61.11	52.38
149	Doc-149	70.83	57.30	63.35	62.96	56.20	59.39	70.83	57.30	63.35
150	Doc-150	66.11	76.06	70.74	53.52	67.31	59.63	65.77	75.68	70.38
<b>Rata-rata</b>		<b>57.42</b>	<b>65.15</b>	<b>60.68</b>	<b>44.76</b>	<b>53.45</b>	<b>48.38</b>	<b>56.75</b>	<b>64.40</b>	<b>59.97</b>

Dari data-data di atas, dipilih model optimal untuk kompresi 50% menggunakan dimensi *word2vec* 100, karena hasil *F1-score* untuk *ROUGE-L*-nya lebih baik, yaitu 60.65%.

#### D. Penerapan Model Optimal untuk Data Uji

Model-model optimal yang terpilih dari tingkat kompresi 30% dan 50%, selanjutnya diterapkan untuk memproduksi ringkasan secara otomatis untuk data uji. Hasil implementasi tersebut diukur performanya dengan *ROUGE-score* di antara metode-metode penelitian sebelumnya, dan dilaporkan pada Tabel 5 berikut.

**Tabel 5.** Pengujian Model Optimal K-means di antara metode lainnya

No.	CR	Metod	Rouge-1(%)			Rouge-2(%)			Rouge-L(%)		
			R	P	F1	R	P	F1	R	P	F1
1		TextRank [3]	43,28	48,02	45,00	30,24	34,05	31,62	41,91	46,52	43,59
2	30%	LexRank [4]	<b>53,35</b>	<b>60,12</b>	<b>55,82</b>	<b>44,33</b>	48,09	<b>45,51</b>	<b>52,35</b>	58,96	<b>54,76</b>
3		K-means	42,03	<b>70,79</b>	52,27	30,68	<b>58,68</b>	39,87	41,57	<b>70,02</b>	51,70
4		TextRank [3]	<b>72,32</b>	66,15	<b>68,76</b>	<b>65,08</b>	57,29	<b>60,60</b>	<b>71,81</b>	65,72	<b>68,29</b>
5	50%	LexRank [4]	67,03	<b>69,09</b>	67,53	59,74	<b>59,58</b>	59,14	66,53	<b>68,62</b>	67,05
6		K-means	58,37	64,28	60,89	45,89	53,16	48,97	57,83	63,66	60,31

Dari hasil pada Tabel 5 di atas, terlihat bahwa untuk tingkat kompresi yang rendah (CR=30%), metode *K-means* dapat memproduksi ringkasan yang kompetitif, yaitu dari segi *F1-score* untuk ketiga *ROUGE*, dapat melampaui metode *TextRank* [3], namun masih lebih rendah dari hasil yang dicapai metode *LexRank* [4]. Namun dari segi *Precision*, metode *K-means* dapat mengungguli metode lainnya.

Sedangkan untuk tingkat kompresi yang tinggi (CR=50%), metode *K-means* tidak dapat menyaingi kedua metode terdahulu, dan cenderung memiliki gap *F1-score* yang cukup jauh dari keduanya. Hal ini disebabkan karena metode *clustering* membentuk *cluster* terlebih dahulu. Apabila tingkat kompresi 50%, artinya ada setengah dari jumlah kalimat dalam dokumen yang harus menjadi anggota *cluster* terpisah, sehingga besar kemungkinan akan banyak *cluster* hanya terdiri atas 1 anggota saja. Dan pemilihan kalimat yang mewakili tiap *cluster* untuk menjadi ringkasan menjadi tidak efektif.

#### IV. KESIMPULAN

Dari pengujian data *training* yang telah dilakukan, dengan tingkat kompresi 30%, terlihat bahwa hasil ringkasan lebih baik saat menggunakan dimensi *word2vec* 200, dengan rata-rata *F1-score* pada *ROUGE-L* sebesar 52,28% berbanding 51,90% untuk *word2vec* berdimensi 100.

Sementara itu, untuk CR=50%, hasil ringkasan lebih baik saat menggunakan dimensi *word2vec* 100 dengan rata-rata *F1-score* untuk *ROUGE-L* sebesar 60,65%, sedangkan nilai rata-rata *F1-score* untuk dimensi *word2vec* 200 adalah 59.97%.

Metode *K-Means Clustering* dapat diterapkan pada tugas peringkasan teks otomatis, dengan kinerja yang cukup baik untuk tingkat kompresi yang rendah, yang menghasilkan sedikit kalimat ringkasan. Sedangkan bila angka tingkat kompresinya yang tinggi, artinya ringkasan masih cukup panjang, maka metode ini memiliki kinerja yang kurang baik. Hal ini terlihat dari perbandingan hasil evaluasi terhadap dua metode lainnya, yang menunjukkan bahwa



metode *K-means* cukup kompetitif untuk memproduksi ringkasan yang pendek.

### Daftar Pustaka

- [1] I. W. A. Setyadi, D. C. Khrisne, and I. M. A. Suyadnya, "Automatic Text Summarization Menggunakan Metode Graph dan Metode Ant Colony Optimization," vol. 17, no. 1, pp. 124–130, 2018.
- [2] K. Ivanedra and M. Mustikasari, "Implementasi Metode Recurrent Neural Network Pada Text the Implementation of Text Summarization With Abstractive," J. Teknol. Inf. dan Ilmu Komput., vol. 6, no. 4, 2019, doi: 10.25126/jtiik.201961067.
- [3] F. Husniah, S. Agustian, and I. Afrianty, "Peringkasan Teks Otomatis Artikel Berbahasa Indonesia Menggunakan Algoritma Textrank," Teknoka 7, 2022.
- [4] Halimah, Surya Agustian, and Siti Ramadhani, "Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank," J. CoSciTech (Computer Sci. Inf. Technol., vol. 3, no. 3, pp. 371–381, 2022, doi: 10.37859/coscitech.v3i3.4300.
- [5] Y. Yuliska and K. U. Syaliman, "Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia," IT J. Res. Dev., vol. 5, no. 1, pp. 19–31, 2020, doi: 10.25299/itjrd.2020.vol5(1).4688.
- [6] K. E. Dewi and N. I. Widiastuti, "The Design of Automatic Summarization of Indonesian Texts Using a Hybrid Approach," J. Teknol. Inf. dan Pendidik., vol. 15, no. 1, pp. 37–43, 2022, doi: 10.24036/jtip.v15i1.451.
- [7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. "Distributed Representations of Words and Phrases and their Compositionality". In Proceedings of NIPS, 2013.
- [8] H. Juwiantho et al., "Sentiment Analysis Twitter Bahasa Indonesia Berbasis WORD2VEC Menggunakan Deep Convolutional Neural Network," J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 1, pp. 181–188, 2020, doi: 10.25126/jtiik.202071758.
- [9] M. Rusli, "Ekstraksi Fitur Menggunakan Model Word2Vec Pada Sentiment Analysis Kolom Komentar Kuisisioner Evaluasi Dosen Oleh Mahasiswa," Klik - Kumpul. J. Ilmu Komput., vol. 7, no. 1, p. 35, 2020, doi: 10.20527/klik.v7i1.296.
- [10] S. Khomsah, "Sentiment Analysis On YouTube Comments Using Word2Vec and Random Forest," Telematika, vol. 18, no. 1, p. 61, 2021, doi: 10.31315/telematika.v18i1.4493.
- [11] D. I. Af'idah, Dairoh, S. F. Handayani, and R. W. Pratiwi, "Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen," J. Inform. Jurunal Pengemb. IT, vol. 6, no. 3, pp. 156–161, 2021.
- [12] A. Fadilah, "Penerapan Algoritma K-Nearest Neighbor untuk Mendeteksi Ujaran Kebencian dan Bahasa Kasar pada Twitter Bahasa Indonesia," Tugas Akhir, Universitas Islam Negeri Sultan Syarif Kasim Riau, 2021.
- [13] J. Nasir, "Penerapan Data Mining Clustering Dalam Mengelompokkan Buku Dengan Metode K-Means," Simetris J. Tek. Mesin, Elektro dan Ilmu Komput., vol. 11, no. 2, pp. 690–703, 2021, doi: 10.24176/simet.v11i2.5482.
- [14] G. Gustientiedina, M. H. Adiya, and Y. Desnelita, "Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan," J. Nas. Teknol. dan Sist. Inf., vol. 5, no. 1, pp. 17–24, 2019, doi: 10.25077/teknosi.v5i1.2019.17-24.
- [15] R. A. Farissa, R. Mayasari, and Y. Umaidah, "Perbandingan Algoritma K-Means dan K-Medoids Untuk Pengelompokkan Data Obat dengan Silhouette Coefficient di Puskesmas Karangsambung," J. Appl. Informatics Comput., vol. 5, no. 2, pp. 109–116, 2021, doi: 10.30871/jaic.v5i1.3237.
- [16] Pedregosa et al., Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825–2830, 2011.