CYBERNETICS, Vol.3, No.02, November 2019, pp. 90~99
P-ISSN 2579-9835
E-ISSN 2580-1465

Peringkasan Ekstraktif Teks Bahasa Indonesia dengan Pendekatan *Unsupervised* Menggunakan Metode Clustering

Dewi Pramudi Ismi[1], Fahri Ardianto[2]
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Jalan Ringroad Selatan, Tamanan, Banguntapan, Bantul [1][2]
*e-mail: dewi.ismi@tif.uad.ac.id[1]

Abstrak

Perkembangan teknologi informasi yang pesat membuat volume informasi yang dapat diakses oleh manusia melalui internet menjadi tidak terbendung. Hal ini menyebabkan pembaca harus dapat memilah informasi yang penting maupun merangkum informasi yang sangat masif. Pembaca perlu merangkum/meringkas banyaknya informasi menjadi informasi utama yang layak untuk ditelaah. Proses merangkum/meringkas informasi yang banyak dari berbagai sumber merupakan permasalahan yang tidak mudah. Peringkasan teks dapat dilakukan secara otomatis oleh komputer menggunakan teknologi kecerdasan buatan. Penelitian ini mengusulkan metode peringkasan teks berbasis clustering dengan K-means clustering. Metode yang diusulkan telah diuji dengan Rouge score menggunakan dataset benchmark. Berdasarkan hasil pengujian, sistem peringkasan teks yang dibangun cukup baik, yaitu memiliki nilai F_1 score dari Rouge-1 = 49,37%, Rouge-2 = 38,18% dan Rouge-L = 46,87%. Hasil yang diperoleh adalah bahwa metode ini lebih baik dari 3 metode unsupervised yang telah digunakan sebelumnya yaitu SumBasic, LSA dan LexRank.

Kata kunci: *peringkasan teks, k-means clustering, Rouge, metode unsupervised, kecerdasan buatan*

Abstract

The rapid development of information technology has made the volume of information accessible by humans via internet unstoppable. This causes the reader to be able to sort out important information or summarize massive information. Readers need to summarize/summarize the amount of information into the main information that deserves to be studied. The process of summarizing/summarizing a lot of information from various sources is a difficult problem. Text summarization can be done automatically by a computer using artificial intelligence technology. This study proposes a clustering-based text summarization method with K-means clustering. The proposed method has been tested with a Rouge score using a benchmark dataset. Based on the test results, the proposed text summarization system is good, which has an F1 score of Rouge-1 = 49.37%, Rouge-2 = 38.18% and Rouge-L = 46.87%. The results obtained shows that this method is better than the three unsupervised methods that have been used previously, namely SumBasic, LSA and LexRank.

Keywords*: text summarization, k-means clustering, Rouge, unsupervised method, artificial intelligence*

1. Pendahuluan

Received Agust 14, 2020; Revised September 07, 2020; Accepted September 17, 2020

Perkembangan teknologi informasi yang pesat membuat volume informasi yang dapat diakses oleh manusia melalui internet menjadi tidak terbendung. Hal ini menyebabkan pembaca harus dapat memilah informasi yang penting maupun merangkum informasi yang sangat masif. Pembaca perlu merangkum/meringkas banyaknya informasi menjadi informasi utama yang layak untuk ditelaah. Proses merangkum/meringkas informasi yang banyak dari berbagai sumber merupakan permasalahan yang tidak mudah. Pembaca membutuhkan waktu yang lama untuk dapat menyusuri setiap informasi dan mengambil informasi utama. Peringkasan teks dapat dilakukan secara otomatis oleh komputer menggunakan teknologi kecerdasan buatan[1][2].

Ringkasan adalah sebuah teks singkat yang dihasilkan oleh satu atau lebih teks yang panjang. Ringkasan mengandung sebagian besar informasi dari teks asli [1]. Menurut [3], ada dua jenis teknik peringkasan teks dilihat dari bagaimana ringkasan dihasilkan. Kedua jenis ringkasan tersebut ialah ringkasan ekstraktif dan ringkasan abstraktif. Peringkasan ekstraktif memilih beberapa kalimat dari dokumen asli untuk merepresentasikan dokumen secara keseluruhan tanpa mengubah struktur kalimat-kalimat tersebut. Sedangkan peringkasan abstraktif menyusun ulang kalimat-kalimat menjadi ringkasan berdasarkan kata-kata inti yang terdapat pada dokumen asli. Peringkasan secara abstraktif lebih susah dilakukan daripada peringkasan secara ekstraktif.

Penelitian tentang sistem peringkasan teks otomatis khususnya yang berbahasa Indonesia sudah beberapa kali dilakukan oleh para peneliti. Peringkasan teks Bahasa Indonesia dapat dilakukan dengan menggunakan algoritma Modified Discrete Differential Evolution [4]. Namun penelitian tersebut hanya menguji peringkasan dengan akurasi, dan belum menggunakan nilai *rouge* yang merupakan standar pengujian untuk peringkasan teks. Peringkasan teks Bahasa Indonesia juga dapat dilakukan secara unsupervised dengan metode *Latent Semantic Analysis* (LSA) dan pengujiannya dikaitkan dengan hasil clustering dokumen Bahasa Indonesia[5]. Meskipun penelitian tersebut menunjukkan bahwa peringkasan teks yang dilakukan dapat meningkatkan F-Measure dari hasil clustering dokumen, akan tetapi sebenarnya hasil peringkasan dokumen dengan LSA pada penelitian tersebut belum dievaluasi. Penelitian lainnya menggunakan *compression rate* untuk menentukan panjangnya ringkasan dalam proses peringkasan teks Bahasa Indonesia[6]. Pada penelitian tersebut, nilai *precision* dan *recall* dari hasil peringkasan masih rendah.

Penelitian ini bertujuan menggunakan pendekatan yang berbeda dari penelitian-penelitian sebelumnya. Penelitian ini menggunakan pendekatan *unsupervised* dalam melakukan peringkasan teks Bahasa Indonesia. Pendekatan *unsupervised* dipilih karena lebih mudah serta tidak memerlukan data latih dalam membangun model peringkasan teks. Penelitian ini akan menggunakan metode clustering untuk menghasilkan kalimat-kalimat inti yang akan mewakili keseluruhan isi dokumen dalam ringkasan. Hasil ringkasan yang diperoleh dalam penelitian ini akan diuji menggunakan nilai Rouge yang merupakan metode pengujian peringkasan teks yang standar. Nilai Rouge diperoleh dengan membandingkan hasil ringkasan yang dihasilkan oleh sistem dengan hasil ringkasan standar yang dibuat oleh manusia.

2. Landasan Teori

2.1. K-means Clustering

Teknik *Clustering* merepresentasikan tupel-tupel data sebagai suatu objek. *Clustering* membagi objek-objek menjadi beberapa grup atau *cluster*, sehingga objek-objek dalam satu *cluster* memiliki kemiripan satu sama lain. Istilah kemiripan dalam hal ini mengacu kepada seberapa dekat jarak antar objek yang terdapat pada suatu ruang [7]. Menurut [8] langkah-langkah klasterisasi dengan metode *K-Means Clustering* adalah sebagai berikut:

a. Pilih jumlah cluster K. Inisialisasi *k* pusat cluster (*centroid*). Hal ini dilakukan dengan cara memberi nilai pusat-pusat cluster dengan angka-angka random.

b. Tempatkan setiap data ke cluster terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data ke cluster tertentu

ditentukan jarak antara data dengan pusat cluster. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat cluster. Jarak paling antara satu data dengan satu cluster tertentu akan menentukan suatu data masuk dalam cluster mana. Untuk menghitung jarak semua data ke setiap titik pusat cluster dapat menggunakan teori jarak euclidean yang dirumuskan dengan persamaan (1) berikut:

$$D(i,j) = \sqrt{(X_{1i} - X_{1j})^2 + (X_{2i} - X_{2j})^2 + \cdots + (X_{ki} - X_{kj})^2} \quad (1)$$

dimana:

D(i,j) = Jarak data ke *i* ke pusat cluster *j*

X_{ki} = Data *i* pada atribut data ke *k*

X_{kj} = Titik pusat ke *j* pada atribut ke *k*

c. Hitung kembali pusat cluster dengan keanggotaan cluster yang sekarang. Pusat cluster adalah rata-rata dari semua data dalam setiap clusternya.

d. Ulangi langkah (2) dan (3) dengan memakai pusat cluster yang baru. Jika pusat cluster sudah tidak berubah lagi, maka proses pengclusteran selesai, atau dinamakan konvergen.

2.2. Peringkasan Teks

Peringkasan teks adalah proses menghasilkan ringkasan dari suatu teks atau artikel yang panjang. Hasil peringkasan teks adalah ringkasan. Ringkasan adalah teks yang lebih pendek dari teks asal namun mengandung informasi-informasi inti/penting dari teks asal. Peringkasan teks memiliki dua jenis yaitu peringkasan teks abstraktif dan peringkasan teks ekstraktif. Peringkasan ekstraktif dilakukan dengan mengekstraksi kalimat-kalimat atau frase-frase yang sering muncul dalam dokumen teks asal dan menggabungkan kalimat atau frase tersebut menjadi ringkasan. Pada peringkasan teks ekstraktif, kalimat-kalimat diambil tanpa mengubah kalimat-kalimat tersebut (sesuai teks asal). Peringkasan abstraktif menggunakan pendekatan yang berbeda dengan peringkasan teks ekstraktif. Pada peringkasan abstraktif, ide utama yang terdapat pada dokumen teks asal digali, kemudian ringkasan disusun dengan mengkonstruksi kembali kalimat-kalimat yang mengandung ide-ide utama. Pada peringkasan abstraktif, kalimat-kalimat penyusun ringkasan tidak sama dengan kalimat-kalimat pada teks awal [9].

Hasil ringkasan teks dievaluasi dengan menggunakan nilai Rouge (*Recall-Oriented Understudy for Gisting Evaluation*) [10]. Nilai Rouge diperoleh dengan membandingkan hasil ringkasan teks oleh sistem dengan hasil ringkasan teks yang dibuat oleh manusia. Pengukuran Rouge tersebut menghitung jumlah unit yang *overlap* seperti n-gram, urutan kata, dan pasangan kata antara ringkasan yang dihasilkan komputer untuk dievaluasi dan ringkasan ideal yang dibuat oleh manusia.

3. Metode Penelitian

Pada penelitian ini secara umum akan dilakukan 3 tahapan. Tahap pertama yaitu text preprocessing meliputi *text tokenizing*, *stemming* dan *stopword removal*. Tahap kedua yaitu perhitungan bobot teks dengan algoritma TF-IDF. Tahap ketiga yaitu klasterisasi teks berdasarkan bobot TF-IDF dengan jumlah cluster yang diinginkan, kemudian kalimat dengan jarak terdekat dengan pusat cluster (centroid) diambil sehingga diperoleh kalimat-kalimat inti setiap cluster.

3.1. Text Pre-processing

Tahapan ini diperlukan agar dokumen hasil pengumpulan data yang akan diproses berada dalam bentuk yang tepat dan dapat diproses pada tahapan selanjutnya [5]. Dalam bidang *text mining* dan *natural language processing,* teks yang akan diproses perlu diubah ke dalam bentuk numerik sehingga dapat dilakukan operasi-operasi matematis untuk mencapai tujuan yang diinginkan. Dalam penelitian ini akan lakukan 3 tahap *text pre-processing*, yaitu *text tokenizing, stopword removal* dan *stemming.*

3.1.1. Text Tokenization

Sebelum diproses dengan menggunakan sebuah metode *text mining*, dokumen teks yang akan diolah harus dipecah terlebih dahulu untuk mendapatkan frasa dengan inti yang berbobot. Ini dapat terjadi pada beberapa level yang berbeda. Dokumen dapat dipecah kedalam bentuk bab, bagian, paragraf, kalimat, kata-kata, dan bahkan suku kata atau fonem. Pendekatan yang paling sering ditemukan dalam sistem *text mining* melibatkan pemenggalan teks menjadi perkalimat ataupun perkata, proses yang seperti inilah yang disebut *text tokenization* [11].

3.1.2. Stopword Removal

Stopwords adalah kata umum (c*ommon words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. *Stopword removal* sendiri berarti menghilangkan kata-kata seperti “dan”, “atau”, “tapi”, “akan” atau kata-kata lain yang sangat umum yang ketika hanya ada kata tersebut maka tidak mengandung arti atau makna apa pun [12].

3.1.3. *Stemming*

Tahapan ini bertugas mengidentifikasi bentuk akar kata dengan menghapus akhiran, seperti pada kata "memulihkan" yang berasal sebagai "pulih" [12]. Tujuan dari tahapan ini adalah agar nilai frekuensi dari tiap-tiap istilah dengan akar kata yang sama, dapat diakumulasikan menjadi satu buah kelompok frekuensi akar kata yang sama.

3.2. Pembobotan TF-IDF

Pembobotan suatu kata dapat diperoleh berdasarkan jumlah kemunculan kata tersebut dalam sebuah dokumen. Rasio kemunculan kata tersebut dinamakan *term frequency* (*tf*) sedangkan jumlah kemunculan *term* dalam koleksi dokumen disebut *inverse document frequency* (*idf*). Bobot suatu istilah semakin besar jika istilah tersebut sering muncul dalam suatu dokumen dan semakin kecil jika istilah tersebut muncul dalam banyak dokumen [13][14] . Berikut ini disajikan persamaan untuk mendapatkan nilai pembobotan menggunakan TF-IDF:

$$IDF = 1 + log\frac{D}{df} \quad (2)$$

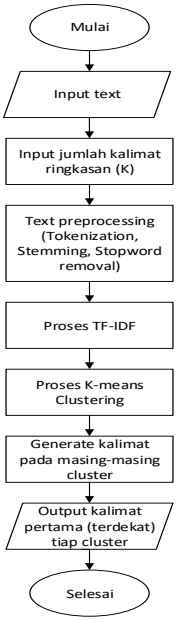
$$TFIDF = TF * (1 + log\frac{D}{df}) \quad (3)$$

3.3. Klasterisasi Kalimat

Pengklasteran kalimat dilakukan dengan algoritma K-Means Clustering setelah didapatkan nilai TFIDF setiap kata pada setiap dokumen. Jumlah klaster yang dihasilkan menyesuaikan dengan jumlah kalimat dalam ringkasan yang diinginkan. Setiap klaster terdiri dari beberapa kalimat dan masing-masing klaster akan diwakili oleh satu kalimat. Satu kalimat ini adalah kalimat terdekat dengan pusat klaster. Satu kalimat dari setiap klaster ini yang kemudian menyusun ringkasan teks secara keseluruhan.

4. Hasil dan Pembahasan

Aplikasi yang telah dibangun dalam penelitian ini diberi nama Ringkas.net. Ringkas.net memiliki alur kerja (workflow) seperti pada Gambar 1 berikut ini.



Gambar 1. Alur kerja aplikasi peringkasan teks dengan clustering

4.1. Text Preprocessing

Tahap *text preprocessing* dilakukan pada inputan berupa teks utuh yang akan diringkas. Teks yang diambil berupa teks yang memiliki sejumlah kalimat dan diusahakan lebih banyak dari jumlah *cluster* yang diinputkan. Tahap *text preprocessing* terdiri dari *text tokenizing, stemming dan stopword removal.* Contoh teks utuh dapat dilihat seperti pada Gambar 2.

Teks	Campak adalah infeksi virus yang ditandai dengan munculnya ruam di seluruh tubuh dan sangat menular. Campak bisa sangat mengganggu dan mengarah pada komplikasi yang lebih serius. Gejala campak mulai muncul sekitar satu hingga dua minggu setelah virus masuk ke dalam tubuh. Bercak atau ruam berwarna merah kecokelatan akan muncul di kulit setelah beberapa hari kemudian. Urutan kemunculan bercak ini dari belakang telinga, sekitar kepala, kemudian ke leher.

Gambar 2. Contoh teks bahasa indonesia yang akan diringkas

4.1.1. Text Tokenization

Pada tahapan ini dilakukan untuk memecah teks utuh menjadi per kalimat untuk kemudian disimpan kedalam DataFrame. Contoh tokenisasi teks dapat dilihat pada Gambar 3.

Dokumen 1	Campak adalah infeksi virus yang ditandai dengan munculnya ruam di seluruh tubuh dan sangat menular.
Dokumen 2	Campak bisa sangat mengganggu dan mengarah pada komplikasi yang lebih serius.
Dokumen 3	Gejala campak mulai muncul sekitar satu hingga dua minggu setelah virus masuk ke dalam tubuh.
Dokumen 4	Bercak atau ruam berwarna merah kecokelatan akan muncul di kulit setelah beberapa hari kemudian.
Dokumen 5	Urutan kemunculan bercak ini dari belakang telinga, sekitar kepala, kemudian ke leher.

Gambar 3. Contoh dokumen teks yang telah ditokenisasi (per kalimat)

Pada contoh dokumen teks pada Gambar 3, teks Bahasa Indonesia yang akan diringkas dipisahkan per kalimat. Dokumen 1 merupakan kalimat pertama, dokumen 2 merupakan kalimat kedua, dst.

4.1.2. Stemming

Tahapan ini membuang semua imbuhan yang terdapat pada suatu kata. Tahapan ini menerapkan library Sastrawi yang mengimplementasikan algortima Nazief Adriani. Contoh teks yang sudah dibuang imbuhannya dapat dilihat pada gambar 4 contoh *stemming* teks.

D1	D2	D3	D4	D5
campak	campak	gejala	bercak	urut
adalah	bisa	campak	atau	muncul
infeksi	sangat	mulai	ruam	bercak
virus	ganggu	muncul	warna	ini
yang	dan	sekitar	merah	dari
tanda	arah	satu	cokelat	belakang

Gambar 4. Contoh hasil *stemming* teks pada dokumen

4.1.3. Stopword Removal

Tahap terakhir pada *text preprocessing* adalah tahap penghilangan kata henti atau kata-kata lain yang minim makna seperti “dan”, “atau”, “tapi”, “akan” atau kata-kata sejenis. Contoh teks yang sudah dihilangkan kata hentinya dapat dilihat pada gambar 5 contoh penghilangan kata henti.

D1	D2	D3	D4	D5
campak	campak	gejala	bercak	urut
infeksi	sangat	campak	ruam	muncul
virus	ganggu	mulai	warna	bercak
tanda	arah	muncul	merah	belakang
muncul	komplikasi	satu	cokelat	telinga
ruam	lebih	hingga	muncul	kepala

Gambar 5. Contoh penghilangan kata (*stopwords*)

4.2. Pembobotan Kata menggunakan TF-IDF

Pembobotan dilakukan untuk menemukan bobot (W) dari setiap kata yang terdapat pada suatu dokumen (D). TF-IDF sendiri merupakan persamaan yang mengalikan frekuensi suatu *term* atau katapada suatu dokumen (TFi) dengan hasil log pembagian antara jumlah dokumen keseluruhan (D) dibagi jumlah dokumen yang mengandung *term*/kata tersebut (IDFi). Contoh teks yang sudah dihitung TF-IDF nya pada Tabel 1 Pembobotan Kata.

Tabel 1. Contoh perhitungan pembobotan kata

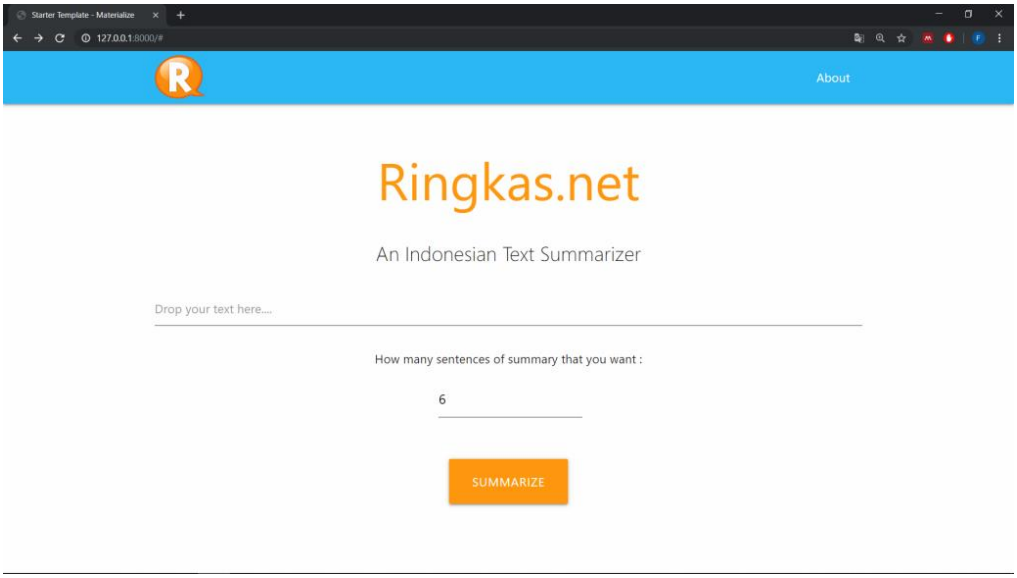
Term	***Term Frequency***					***df***	***D/df***	***idf + 1***	***Term Weight (W)*** W = tf x (idf + 1)				
	d1	**d2**	**d3**	**d4**	**d5**				**d1**	**d2**	**d3**	**d4**	**d5**
campak	1	1	1	0	0	3	1,7	1,22	1,22	1,22	1,22	0,00	0,00
infeksi	1	0	0	0	0	1	5	1,70	1,70	0,00	0,00	0,00	0,00
virus	1	0	1	0	0	2	2,5	1,40	1,40	0,00	1,40	0,00	0,00
tanda	1	0	0	0	0	1	5	1,70	1,70	0,00	0,00	0,00	0,00
muncul	1	0	1	1	1	4	1,3	1,10	1,10	0,00	1,10	1,10	1,10
ruam	1	0	0	1	0	2	2,5	1,40	1,40	0,00	0,00	1,40	0,00
seluruh	1	0	0	0	0	1	5	1,70	1,70	0,00	0,00	0,00	0,00
tubuh	1	0	1	0	0	2	2,5	1,40	1,40	0,00	1,40	0,00	0,00

4.3. Clustering dengan K-Means

K-Means Clustering telah berhasil digunakan untuk mengelompokkan kalimat-kalimat dalam bentuk bobot TF-IDF ke dalam k cluster, dimana k adalah jumlah cluster yang diinputkan oleh pengguna. Pada proses klasterisasi ini, terbentuk kalimat-kalimat yang berdekatan berada dalam cluster yang sama, sehingga dapat diambil satu kalimat untu merepresentasikan cluster tersebut. Kalimat yang memiliki jarak paling dekat dengan centroid dianggap sebagai kalimat yang paling merepresentasikan kalimat-kalimat lain yang tergabung dalam 1 cluster. Kalimat-kalimat yang representatif ini kemudian digabungkan menjadi satu ringkasan.

Implementasi Sistem

Implementasi penelitian ini adalah sebuah sistem yang berjalan pada platform web dengan menggunakan framework Flask sebagai framework yang mendukung bahasa pemrograman Python. Beberapa antarmuka sistem yang telah dibangun disajikan pada Gambar 6, Gambar 7, dan Gambar 8.

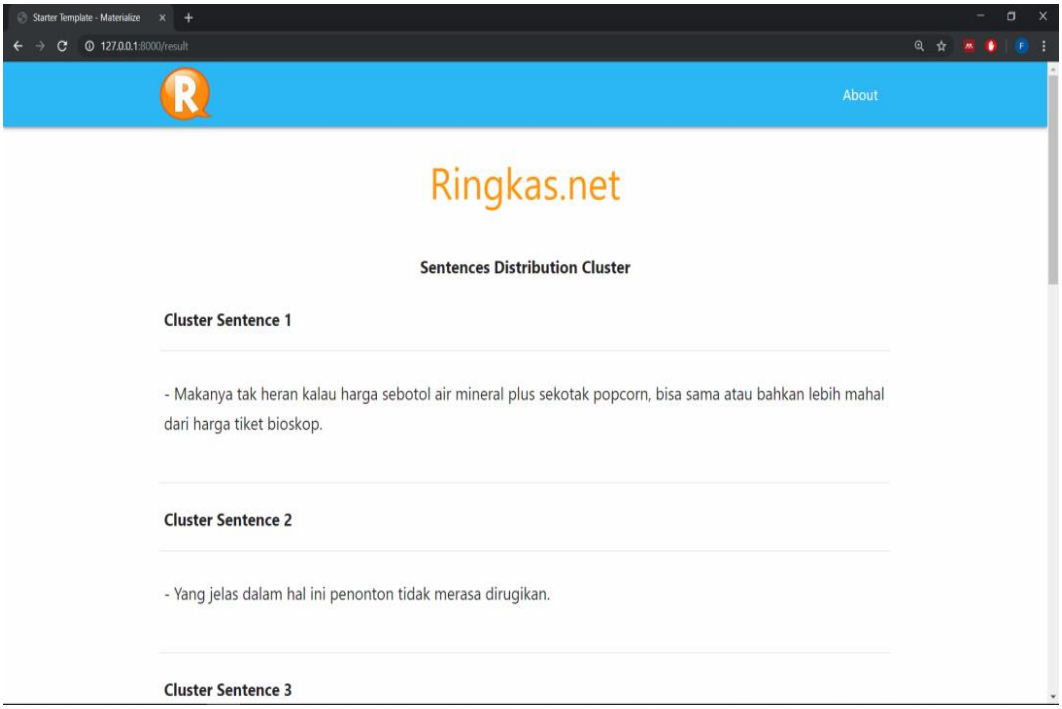


Gambar 6. Tampilan dashboard

Halaman dashboard berisi *form* untuk menginputkan teks yang ingin diringkas dan juga untuk menginputkan jumlah kalimat ringkasan yang ingin dihasilkan. Input jumlah ringkasan ini dijadikan acuan untuk jumlah cluster yang digunakan pada saat proses clustering dengan K-means. Setelah pengguna menginputkan teks dan jumlah ringkasan, selanjutnya pengguna menekan tombol bertuliskan *summarize* untuk meringkas teks yang sudah diinputkan sebelumnya. Secara otomatis teks yang ada akan diolah dan halaman akan diarahkan ke halaman hasil ringkasan yang ditampilkan pada Gambar 7.

Gambar 7. Halaman hasil ringkasan

Distribusi kalimat untuk setiap cluster juga ditampilkan pada aplikasi yaitu halaman distribusi cluster (Gambar 8.)



Gambar 8. Halaman distribusi kalimat per cluster

Pengujian dengan *Rouge*

Tahap pengujian dalam penelitian ini adalah mengukur seberapa akurat ringkasan yang dihasilkan oleh sistem ini apabila dibandingkan dengan ringkasan yang dihasilkan oleh manusia. Dalam pengujian ini digunnakan dataset *benchmark* untuk peringkasan teks Bahasa Indonesia yaitu dataset Indosum [2]. Dataset ini berisi artikel-artikel teks Bahasa Indonesia yang telah diringkas secara manual oleh manusia. Untuk mengetahui akurasi peringkasan, penelitian ini menggunakan metode pengujian *Rouge Test* yang merupakan metode yang akan menghasilkan nilai *precision, recall* dan *F1 score. Rouge* sendiri merupakan akronim dari *Recall-Oriented Understudy for Gisting Evaluation* atau dalam bahasa Indonesia berarti evaluasi intisari yang berorientasi *recall*. Nilai akurasi yang diambil dari pengujian ini hanyalah nilai F_1 *score* dari *Rouge-1, Rouge-2 dan Rouge-L* sesuai dengan pengujian yang dilakukan di Indosum [2]. Untuk hasil pengujian dapat dilihat pada Tabel 2 Hasil pengujian *Rouge Test.*

Tabel 2. Hasil pengujian *Rouge Test.*

	Dataset Pengujian ke-					**Rerata**	**x100 %**	**Std. Deviasi**
	1	2	3	4	5			
Rouge-1	0,489	0,494	0,492	0,501	0,493	0,494	49,37	0,44
Rouge-2	0,376	0,381	0,383	0,389	0,380	0,382	38,18	0,48
Rouge-L	0,463	0,469	0,468	0,476	0,468	0,469	46,87	0,46

5. Kesimpulan

Berdasarkan hasil pengujian yang telah diperoleh, maka kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Sistem dapat mempermudah *user* untuk meringkas artikel secara instan.
2. Sistem dapat meningkatkan efisiensi waktu pembaca dalam menemukan inti pembahasan dari suatu teks artikel.
3. Berdasarkan hasil pengujian, sistem peringkasan teks yang dibangun cukup baik, yaitu memiliki nilai F_1 *score* dari *Rouge-1* = 49,37%, *Rouge-2* = 38,18% *dan Rouge-L* = 46,87%.
4. Dibandingkan dengan hasil yang diperoleh pada Indosum [2], metode yang diterapkan pada penelitian ini memiliki akurasi yang lebih baik dari 3 metode lainnya dalam rumpun metode pendekatan *unsupervised* (SumBasic, LSA dan LexRank).

Daftar Pustaka

[1] H. P. Luhn, “The Automatic Creation of Literature Abstracts,” *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 2010.
[2] K. Kurniawan and S. Louvan, “I NDO S UM : A New Benchmark Dataset for Indonesian Text Summarization,” *2018 Int. Conf. Asian Lang. Process.*, pp. 215–220, 2018.
[3] E. Hovy and C.-Y. Lin, “Automated text summarization and the SUMMARIST system,” p. 197, 1996.
[4] N. W. H. Setyawan, “Peringkasan Teks Bahasa Indonesia Menggunakan Modified Discrete Differential Evolution Algorithm,” 2016.
[5] A. Luthfiarta, J. Zeniarja, and A. Salam, “Algoritma Latent Semantic Analysis (LSA) Pada Peringkas Dokumen Otomatis Untuk Proses Clustering Dokumen,” *Semin. Nas. Teknol. Inf. Komun. Terap. 2013 (SEMANTIK 2013)*, vol. 2013, no. November, pp. 13–18, 2013.
[6] A. Romadhony, Z. R. Fariska, N. Yusliani, and L. Abednego, “Text Summarization untuk Dokumen Berita Berbahasa Indonesia,” *J. Telkom Univ.*, pp. 408–414, 2017.
[7] J. Han, M. Kamber, and J. Pei, *Data Mining Concept and Techniques, 3rd ed.* 2012.
[8] B. Sentosa and A. Umam, *Data Mining dan Big Data Analytics*. Yogyakarta: Media Pustaka, 2018.
[9] V. Gupta, G.S Lehal, "A Survey of Text Summarization Extractive Techniques", *Journal of Emerging Technologies in Web Intelligence,* Vol. 2, No. 3. 2010.
[10] C.Y. Lin, "Rouge: A Package for Automatic Evaluation of Summaries", Proceedings of the ACL 2004 Workshop on Text Summarization Branches Out (WAS 2004), pages 74-81. Barcelona, Spain.
[11] R. Feldman and J. Sanger, *The Text Mining Handbook*. 2006.
[12] M. W. Berry, *Survey of Text Mining Clustering, Classification, and Retrieval Scanned by Velocity*, vol. 32, no. 10. 2004.
[13] D. A. Grossman and O. Frieder, *Information Retrieval: Algorithms and Heuristics*, vol. 461. 2004.
[14] M. Mustaqhfiri, Z. Abidin, and R. Kusumawati, “Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance,” *Matics*, 2012.