ISBN : 979-587-705-4
Vol. 3 No. 1

Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode *Latent Semantic Analysis* (LSA) dan Teknik *Steinberger & Jezek*

Jerry Satiamy Saputra
Universitas Sriwijaya
Indonesia
jerrysatiamy@gmail.com

Muhammad Fachrurrozi
Universitas Sriwijaya
Indonesia
mfachrz@unsri.ac.id

Yunita
Universitas Sriwijaya
Indonesia
yunita.v1t4@gmail.com

Dokumen berita merupakan dokumen yang memuat berbagai macam informasi. Semakin banyak informasi yang terdapat pada suatu dokumen membuat dokumen tersebut semakin panjang. Membaca keseluruhan dokumen tersebut memakan banyak waktu. Ringkasan dokumen diperlukan untuk memudahkan memahami informasi yang berukuran besar dengan cepat. Peringkasan dokumen secara otomatis merupakan solusi untuk membantu mendapatkan intisari dari dokumen. Pada penelitian ini dilakukan penerapan metode Latent Semantic Analysis dan teknik Steinberger&Jezek yang digunakan untuk peringkasan teks otomatis. Jumlah data uji yang digunakan sebanyak 10 teks berita yang diambil dari data uji penelitian sebelumnya. Hasil penelitian yang telah dilakukan menghasilkan rata-rata recall 0.7027, precision 0.6973, dan f-measure 0.6974.

Keywords— Latent Semantic Analysis, Peringkasan teks otomatis, Steinberger&Jezek

I. PENDAHULUAN

Informasi merupakan hal yang sangat dibutuhkan oleh setiap orang. Semakin banyak informasi yang tersedia dalam suatu dokumen maka semakin panjang pula sebuah dokumen tersebut [2] . Untuk mendapatkan inti dari informasi tersebut dapat dilakukan dengan membaca isi dokumen secara keseluruhan, membutuhkan waktu yang cukup lama jika dibandingkan dengan membaca isi ringkasan dari suatu teks. Masalah tersebut dapat diselesaikan dengan cara melakukan peringkasan pada dokumen untuk mendapatkan dokumen yang lebih ringkas dari

Peringkasan secara manual juga tidak menyelesaikan masalah jika dokumen yang diringkas cukup panjang. Peringkasan teks otomatis atau yang dikenal sebagai Automatic Text Summarization merupakan solusi yang dapat digunakan dalam hal peringkasan teks dokumen dan tetap mempertahankan kualitas dari hasil ringkasan tersebut [8].

Ringkasan merupakan bentuk singkat dari sebuah teks yang dibuat dari satu dokumen atau lebih, dan mengambil informasi-informasi penting yang terdapat pada dokumen asli. Terdapat 2 teknik kriteria dalam peringkasan teks yaitu teknik ekstraksi dan teknik abstraksi. Teknik ekstraksi adalah teknik penyusunan kalimat dengan mengambil kalimat-kalimat penting yang terdapat pada dokumen asli dan menggabungkannya menjadi dokumen yang lebih pendek. Sedangkan, teknik abstraksi adalah teknik penyusunan kalimat dengan cara mengambil kalimat-kalimat penting pada dokumen asli lalu membuatnya dalam bentuk kalimat lain untuk dijadikan ringkasan [1].

II. PENELITIAN SEBELUMNYA

Peringkasan teks menggunakan Fuzzy Logic Scoring yang dilakukan oleh Patil dan Kukarni. Pada penelitiannya menggunakan 8 fitur ekstrasi yaitu; Tittle word, Sentence Length, Sentence Position, Numerical data, Thematic words, Sentence to sentence similarity, Term weight, dan Proper Nouns. Fuzzy Logic Scoring digunakan untuk pemberian bobot pada tiap kalimat dan kalimat dengan bobot tertinggi akan diambil dan disusun menjadi ringkasan. Hasil akhir yang didapat, metode Fuzzy Logic Scoring dapat meningkatkan kualitas peringkasan jadi lebih baik [3].

Penelitian mengenai peringkasan teks telah dilakukan oleh Geetha J K dan Deepamala menerapkan metode Latent Semantic Analysis untuk peringkasan teks pada artikel berbahasa Kanada. Metode ini terdiri dari beberapa tahap yaitu mengubah dokumen menjadi matriks, pemberian bobot, menambahkan SVD ke [illegible] Pemilihan kalimat dua metode yang berbeda yaitu Steinberger&Jezek dan Cross method [4].

Penelitian mengenai peringkasan teks otomatis yang dilakukan oleh Alami, Meknassi, Ouatik, dan Ennahnahi membandingkan tiga jenis stemming untuk peringkasan teks bahasa arab yaitu Khoja's stemmer, Larkey's stemmer, dan Alkhalil's stemmer. Proses pemilihan kalimat yang akan dijadikan bagian dari ringkasan menggunakan Cosine Similarity dan metode MMR. Pengujian dilakukan dengan membandingkan nilai recall, precision, dan f-measure hasil peringkasan

ISBN : 979-587-705-4
Vol. 3 No. 1

dengan menggunakan tiga jenis stemmer yang berbeda. Hasil dari penelitian yang didapat adalah Khoja's stemmer merupakan stemmer terbaik dengan nilai recall, precision, dan f-measure tertinggi yaitu 0.35, 0.57, dan 0.44 untuk proses peringkasan teks Berbahasa Arab [9].

Penelitian mengenai peringkasan teks yang dilakukan oleh Xiong dan Luo dengan membandingkan metode Latent Semantic Analysis dan metode Maximal Marginal Relevance didapatkan hasil bahwa metode Latent Semantic Analysis lebih baik [5].

Penelitian lainnya yang dilakukan oleh Niladri Chatterjee dengan menggunakan metode Genetic Algorithm. Pada penelitian ini dilakukan pengoptimalisasian nilai Fitness Function yang dipengaruhi oleh tiga buah faktor yaitu Cohesion Factor, Topic Relation Factor, dan Readability Factor. Terdapat beberapa process pada metode yang dipakai yaitu Selection, Crossover, dan Mutation. Pengujian dilakukan terhadap beberapa dokumen dan didapatkan nilai Precision terbesar yaitu 83% dan nilai Max Fitness terbesar 71% [10].

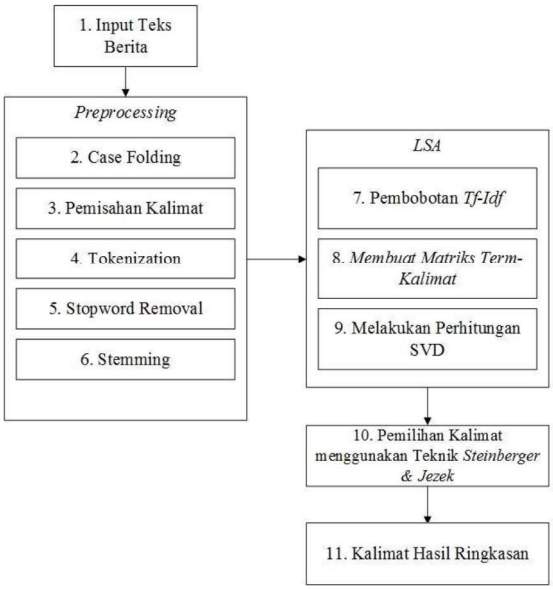
Penelitian lainnya yang dilakukan oleh Luthfiarta, Zeniarja, dan Salam menerapkan metode Latent Semantic Analysis pada proses Clustering dokumen dan menghasilkan tingkat akurasi yang lebih baik [6].

Penelitian ini dilakukan untuk mengetahui tingkat akurasi pada peringkasan dokumen berita berbahasa Indoneisa menggunakan metode Latent Semantic Analysis dan teknik Steinberger&Jezek pada pemilihan kalimat.

III. METODOLOGI

Pada penelitian ini terdiri dari preprocessing dan processing. Preprocessing adalah proses awal pengolahan data sebelum proses utama dilakukan. Tahap-tahap yang dilakukan pada proses preprocessing adalah casefolding, sentence segmentation, tokenization, stopword removal, dan stemming [1].

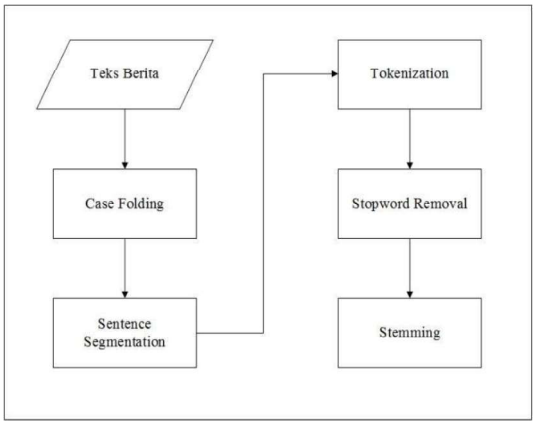
Proses selanjutnya setelah preprocessing adalah implementasi metode *Latent Semantic Analysis* (LSA), pembobotan *tf-idf,* pembuatan matriks *term-sentence,* melakukan perhitungan SVD terhadap matriks inputan hasil pembobotan, pemilihan kalimat menggunakan teknik *Steinberger&Jezek*, dan hasil terakhir adalah didapatkannya kalimat hasil ringkasan sistem. Alur proses sistem secara keseluruhan dapat dilihat pada Gambar 2.



Gambar 2. Alur Sistem

A. Preprocessing

Proses preprocessing terdiri dari casefolding, yaitu proses merubah karakter inputan menjadi huruf kecil. Proses pemisahan kalimat dari paragraf dokumen inputan, pada proses ini tanda baca seperti titik (.), tanda tanya (?), dan tanda seru (!) akan menjadi penanda berakhirnya suatu kalimat.



Gambar 1. Tahapan preprocessing

Proses selanjutnya yaitu tokenization, yaitu proses pemecahan kalimat menjadi per-kata yang dilakukan berdasarkan spasi sebagai pemisah tiap kata. Proses selanjutnya yaitu penghapusan kata-kata yang tidak memiliki makna atau kata yang kurang berarti dan sering muncul dalam kumpulan kata. Setelah itu dilakukan proses stemming, yaitu proses mengubah suatu kata menjadi kata dasarnya (root word) menggunakan algoritma tertentu. Proses stemming merupakan proses terakhir pada preprocessing. Tahapan pada preprocessing dapat dilihat pada Gambar1.

Stemming dalam penelitian ini menggunakan InaNLP Library yang merupakan library natural language processing yang dikembangkan oleh Tim Lab Grafika dan Intelegensia buatan dari Sekolah Teknik

ISBN : 979-587-705-4
Vol. 3 No. 1

Elektro dan Informatika Institut Teknologi Bandung [7]. Pada InaNLP Library terdapat fungsi-fungsi sebagai berikut :

1. Pemisah kalimat
2. Tokenisasi
3. Normalisasi Kata
4. Stopword
5. Stemmer
6. POS Tagger
7. NE Tagger
8. Phrase Chunker
9. Parser (Early & CYK)
10. Semantic Analyzer

Pada penelitian ini, InaNLP Library akan digunakan pada fase preprocessing. Fungsi pada InaNLP Library yang dipakai adalah pemisah kalimat, tokenisasi, normalisasi kata, stopword, dan stemmer. InaNLP juga digunakan sebagai sumber kamus kata dasar Bahasa Indonesia dan kamus stopword list.

B. Pembuatan matriks term-sentence

Setelah langkah preprocessing dilakukan didapatkan kumpulan kata-kata dasar. Pembuatan matriks *term-sentence* yang dimana baris merupakan banyak kata hasil stemming, dan kolom merupakan banyak jumlah kalimat pada dokumen inputan. Tiap cell berisikan bobot tiap kata. Proses pembobotan tiap kata dilakukan dengan menggunakan TFIDF. *Term Frequency* (TF) adalah jumlah frekuensi kemunculan suatu *term* pada suatu kalimat, sedangkan *Inverse Document Frequency* (IDF) adalah perhitungan logaritma pembagian dari total jumlah kalimat dengan frekuensi kalimat yang memuat suatu *term* [4].

$$Tf - Idf = \left(\frac{f(i,j)}{WordsPerSentence[j]}\right) * log\left(\frac{TotalNumOfSentences}{SentencesPerWord[i]}\right) \quad (1)$$

Keterangan :

f(i,j) = frekuensi kata i pada kalimat j.
Tf = *f(i,j)*/ jumlah banyak kata dalam kalimat[j].
Idf = *log(*jumlah banyaknya kalimat/frekuensi kalimat yang mengandung kata[i]*)*.

Contoh :

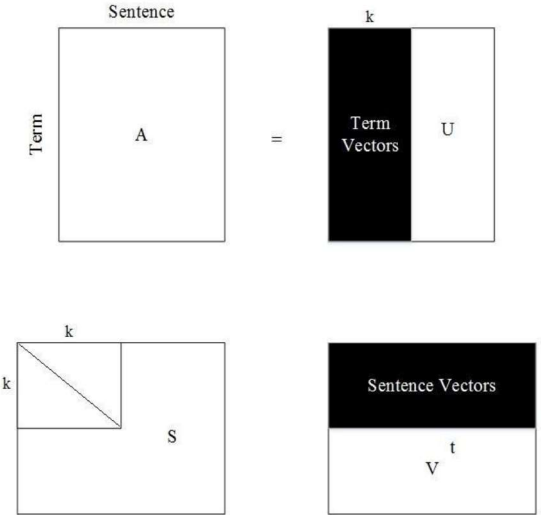
Kal1 : minta tambah armada atur perintah transportasi online berlaku
Kal2 : pekan atur perintah berlaku
Kal3 : lihat permen april berlaku tinggal minggu
Kal4 : rudi operator ojek online gojek sanggup minta tambah armada bogor
Kal5 : iya gojek sanggup

Tabel 1. Matriks *term-sentence*

	Kal1	Kal2	Kal3	Kal4	Kal5
minta	1	0	0	1	0
tambah	1	0	0	1	0
armada	1	0	0	1	0
atur	1	1	0	0	0
perintah	1	1	0	0	0
transportasi	1	0	0	0	0
online	1	0	0	1	0
berlaku	1	1	1	0	0
pekan	0	1	0	0	0
lihat	0	0	1	0	0
permen	0	0	1	0	0
april	0	0	1	0	0
tinggal	0	0	1	0	0
minggu	0	0	1	0	0
rudi	0	0	0	1	0
operator	0	0	0	1	0
ojek	0	0	0	1	0
gojek	0	0	0	1	1
sanggup	0	0	0	1	1
bogor	0	0	0	1	0
iya	0	0	0	0	1

C. Penerapan SVD pada Matriks

Singular Value Decomposition (SVD) merupakan salah satu teknik aljabar linear yang digunakan untuk menguraikan (dekomposisi) suatu matriks menjadi tiga buah matriks baru, yaitu matriks orthogonal U, matriks diagonal S, dan Transpose matriks orthogonal V.



Gambar 3. Proses SVD

$$A_{m\times n} = U_{m\times n} . S_{n\times n} . V^T_{n\times n} \quad (2)$$

Keterangan :

$A_{m\times n}$ = matriks A dengan nilai m ≥ n
$U_{m\times n}$ = matriks ortogonal berukuran m × n
$S_{n\times n}$ = matriks diagonal berukuran n × n, dengan elemen matriks positif atau nol
$V^T_{n\times n}$ = matriks ortogonal berukuran n × n, yang merupakan transpose matriks V.

Pada penelitian ini untuk mempermudah perhitungan matriks digunakan Java Matrix (JAMA) Library. JAMA merupakan library perhitungan matriks pada java yang dikembangkan oleh MathWorks dan National Institute of Standards and Technology (NIST). Tujuan penggunaan library JAMA sendiri adalah untuk mempermudah perhitungan matriks pada

ISBN : 979-587-705-4
Vol. 3 No. 1

penelitian yang akan dilakukan [8]. Pada JAMA Library terdapat fungsi-fungsi dekomposisi matriks sebagai berikut :

1. Cholesky Decomposition
2. LU Decomposition
3. QR Decomposition
4. Eigenvalue Decomposition
5. Singular Value Decomposition

Pada penelitian ini akan digunakan fungsi Singular Value Decomposition pada library JAMA untuk perhitungan SVD pada metode LSA.

D. Sentence Selection

Teknik pemilihan kalimat yang digunakan pada perangkat lunak ini adalah teknik Steinberger&Jezek. Teknik peringkasan ini membutuhkan Matriks V dan Matriks singular S yang telah dihasilkan dari proses SVD.

$$length = \sqrt{\sum_{j=1}^{n} V_{ij} * S_{jj}} \quad (3)$$

Matriks V merupakan matriks yang dimana tiap barisnya mendefinisikan kalimat dan kolom mendefinisakan konsep (hubungan), setiap kolom pada matriks tersebut dihitung nilai rata-rata nya, lalu nilai pada setiap elemen masing-masing kolom pada matriks yang kurang dari nilai rata-rata digantikan dengan nol. Tujuan dari langkah ini adalah untuk menghapus kalimat yang mengganggu, yaitu kalimat yang tidak memiliki hubungan [4].

	Con1	Con2	Con3	Con4	Length
Sen1	0.0169	-0.0111	0.0126	-0.1321	0.4250
Sen2	0.0199	-0.0133	0.0200	-0.8936	0.1012
Sen3	0.0783	-0.0034	0.0078	-0.3976	0.5455
Sen4	0.0190	-0.0095	0.0099	-0.0293	0.1006
Average	0.0335	-0.0093	0.0126	-0.3631	

Tabel 2. Teknik Steinberger&Jezek

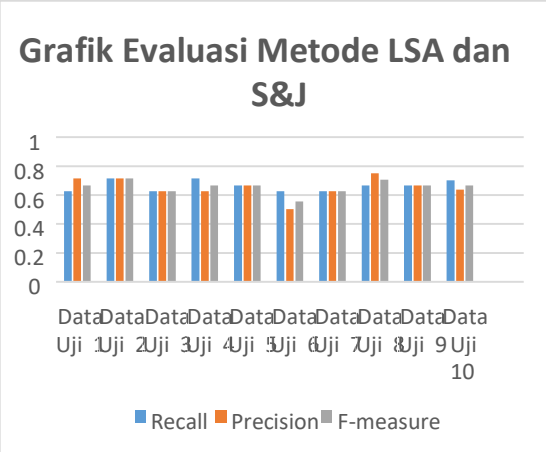
IV. HASIL & PEMBAHASAN

Kami melakukan pengujian terhadap sampel data untuk menilai tingkat akurasi dengan perhitungan secara manual. Pengujian dilakukan dengan menggunkan 10 teks berita. Tingkat akurasi dilihat berdasarkan perbandingan antara hasil ringkasan manual dengan hasil ringkasan sistem. Pengujian ini merupakan hasil evaluasi yang dilakukan terhadap hasil dari ringkasan perangkat lunak dengan metode Latent Semantic Analysis dan teknik Steinberger&Jezek pada pemilihan kalimat. Pengujian tingkat akurasi diukur dengan perhitungan recall, precision, dan f-measure. Pengujian terhadap data uji dilakukan dengan memasukkan sampel data kedalam sistem dan memprosesnya untuk mendapatkan sebuah ringkasan.

$$Recall = \frac{kalimat\ ringkasan\ sistem \cap ringkasan\ manual}{\sum kalimat\ ringkasan\ manual} \quad (3)$$

$$Precision = \frac{kalimat\ ringkasan\ sistem \cap ringkasan\ manual}{\sum kalimat\ ringkasan\ sistem} \quad (4)$$

$$f - measure = \frac{2*precision*recall}{recall+precision} \quad (5)$$



Gambar 4. Grafik Evaluasi Metode LSA dan Steinberger&Jezek

Pada Gambar 4 menunjukkan diagram grafik batang hasil pengujian terhadap 10 data uji yang telah dilakukan. Data uji diambil dari data uji pada penelitian mengenai peringkasan teks sebelumnya. Hasil akhir yang didapat nilai rata-rata recall sebesar 0.6626, precision 0.7157, dan rata-rata fmeasure sebesar 0.6554.

V. KESIMPULAN

Pada penelitian ini kami telah menerapkan metode Latent Semantic Analysis dan teknik Steinberger&Jezek pada peringkasan dokumen teks berita Berbahasa Indonesia dengan hasil ringkasan sebesar 50% dari teks dokumen asli. Nilai recall tertinggi yang didapat sebesar 0.71, nilai precision tertinggi sebesar 0.75, dan nilai f-measure tertinggi sebesar 0.71.

Untuk pengembangan lebih lanjut, disarankan untuk tidak hanya mempertimbangkan bobot dari setiap kalimat tetapi juga memperhitungkan relasi antara kalimat-kalimat tersebut. Selain menghasilkan peringkasan teks ekstraksi, penelitian selanjutnya dapat dikembangkan dengan metode lain agar bisa menghasilkan peringkasan teks abstraksi.

REFERENSI

[1] Babar, S. A., & Patil, P. D. (2015). Improving Performance of Text Summarization. Procedia Computer Science, 46(Icict 2014), 354–363. https://doi.org/10.1016/j.procs.2015.02.031.

[2] Hariharan, S., & Srinivasan, R. (2008). Investigations in single document summarization by extraction method. Proceedings of the 2008 International Conference on Computing, Communication and Networking, ICCCN 2008, (Icccn), 0–4. https://doi.org/10.1109/ICCCNET.2008.4787677.

[3] Patil, M., & Kulkarni, N. (2014). Text Summarization Using Fuzzy

Logic. Paragraph, 1(3), 42–45. Retrieved from http://www.ijirae.com/images/downloads/may-specialissue/MYCS10082.May2014.10.pdf.

[4] Geetha J K, & Deepamala N. (2015). Kannada text summarization using Latent Semantic Analysis. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1508–1512). IEEE. https://doi.org/10.1109/ICACCI.2015.7275826

[5] Xiong, S., & Luo, Y. (2015). A new approach for multi-document summarization based on latent semantic analysis. In Proceedings - 2014 7th International Symposium on Computational Intelligence and Design, ISCID 2014 (Vol. 1, pp. 177–180). https://doi.org/10.1109/ISCID.2014.27

[6] Luthfiarta, A., Zeniarja, J., & Salam, A. (2013). Algoritma Latent Semantic Analysis (LSA) Pada Peringkas Dokumen Otomatis Untuk Proses Clustering Dokumen. Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013), 2013(November), 13–18.

[7] Wijanto, M. C. (2015). Sistem Pendeteksi Pengirim Tweet dengan Metode Klasifikasi Naive Bayes, 1, 172–182.

[8] Dokun, O., & Celebi, E. (2015). Single-Document Summarization Using Latent Semantic Analysis, 1(2).

[9] Alami, N., Meknassi, M., Ouatik, S. A., & Ennahnahi, N. (2016). Impact of stemming on Arabic text summarization, 338–343.

[10] Niladri Chatterjee, A. M. and S. G. (2012). Single Document Extractive Text Summarization Using Genetic Algorithms. In Third International Conference on Emerging Applications of Information Technology (EAIT) Single 2012 (pp. 225–254). https://doi.org/10.1785/0120040116