Integrasi Peringkas Dokumen Otomatis Dengan Penggabungan Metode Fitur dan Metode *Latent Semantic Analysis* (LSA) Sebagai *Feature Reduction*

Junta Zeniarja [1], Abu Salam [2], Ardytha Luthfiarta [3], L Budi Handoko [4], Muhammad Jamhari [5]

[1,2,3,4] *Tekink Informatika, Univ. Dian Nuswantoro*
*Email: junta@dsn.dinus.ac.id, abu.salam@dsn.dinus.ac.id*1,
ardytha.luthfiarta@dsn.dinus.ac.id, handoko@dsn.dinus.ac.id
[5] *Magister Teknik Informatika, Univ. Dian Nuswantoro*
Email: vary.i.cat@gmail.com

ABSTRAK

Proses clustering dokumen memudahkan pengguna menemukan dokumen yang diinginkan. Dalam prosesnya dokumen yang akan dicluster direpresentasikan menggunakan Vector Space Model (VSM). Masalah klasik dalam VSM adalah matrik term-dokumen yang sangat jarang (banyak mengandung angka 0 dalam term-dokumen matrik) dan juga berdimensi tinggi, sehingga dapat mengurangi kinerja clustering dokumen. Oleh karena itu diperlukan suatu metode untuk bisa mengurangi dimensi term-dokumen dan menghilangkan term yang bernilai 0 tersebut sehingga dapat meningkatkan kinerja proses clustering. Dalam penelitian ini diusulkan model peringkas dokumen otomatis dengan penggabungan metode fitur dan latent semantic analysis (LSA) sebagai feature reduction pada proses clustering dokumen.

Tujuan dari penelitian ini adalah untuk meningkatkan akurasi dari clustering dokumen dengan pengkombinasian metode pada peringkas dokumen otomatis yang diintegrasikan sebagai feature reduction. Beberapa tahapan clustering dalam penelitian ini, yaitu preprocessing, peringkas dokumen otomatis dengan metode fitur ,LSA dan Kombinasi, pembobotan kata, feature selection, feature transformation dan algoritma clustering. Hasil penelitian menunjukkan tingkat akurasi menggunakan peringkas dokumen otomatis yang diintegrasikan sebagai feature reduction dengan menggabungkan metode fitur dan metode LSA mencapai 93,33 % yang diperoleh pada tingkat peringkas dokumen otomatis LSA Summary + Feature Summary 50% + Feature Selection 20% + LSA dibandingkan dengan feature selection 20 % tanpa menggunakan peringkas dokumen otomatis yang hanya mencapai tingkat akurasi 89,33 %.

Kata kunci : Text mining, Clustering Dokumen, Latent Semantic Analysis, Metode Fitur.

1. Latar Belakang

Clustering merupakan metode untuk mengorganisir secara otomatis koleksi data yang jumlahnya besar dengan partisi data set, sehingga objek dalam cluster yang sama lebih mirip satu sama lain daripada objek dalam cluster yang lain. Pengelompokan dokumen terkait untuk mengatur pengumpulan data teks besar. Dalam bidang Information Retrieval (IR), clustering dokumen digunakan secara otomatis untuk mengelompokkan dokumen yang memiliki topik yang sama [1]. Ringkasan dokumen dapat diartikan sebagai proses dari pembuatan intisari informasi terpenting dari sumber untuk menghasilkan versi yang lebih ringkas.

Volume dokumen yang sangat besar menyebabkan permasalahan optimalisasi hasil proses clustering dokumen, yaitu besarnya ruang vektor atau besarnya dimensi pada matrik term-dokumen dalam model ruang vektor atau vector space model (VSM). Dengan adanya masalah itu, dapat menurunkan kinerja dari pengelompokan dokumen. Sejumlah penelitian dilakukan untuk mengatasi masalah tersebut yaitu melalui Singular Value Decomposition (SVD) dengan menggunakan Latent Semantic Indexing (LSI). SVD dapat menjadikan matrik yang berdimensi lebih kecil dengan mengurai matrik term-dokumen. Akan tetapi, LSI melalui SVD dalam melakukan proses perhitungan memerlukan waktu yang relatif lebih lama [2].

Berdasarkan permasalahan yang ada, dalam penelitian ini akan mengusulkan penggunaan fitur ringkasan ekstrak dengan penggabungan metode fitur dan latent semantic analysis (LSA) sebagai feature reduction pada proses clustering dokumen. Dengan melakukan proses peringkasan dokumen sebelum dilakukan proses clustering diharapkan mampu mengurangi besarnya matrik term-dokumen dengan tetap menjaga kualitas isi dari dokumen.

2. Dasar Teori

2.1 Clustering Dokumen

Algoritma yang umum digunakan pada proses clustering dokumen dan juga yang akan digunakan dalam penelitian adalah algoritma *K-means*, Dasar algoritma *K-means* dapat disusun menjadi 4 tahap sebagai berikut [3] :

1. Inisialisasi titik pusat Cluster
2. Masukan setiap dokumen ke cluster yang paling cocok berdasarkan ukuran kedekatan dengan centroid / titik tengah cluster.
3. Setelah semua dokumen masuk ke cluster. Hitung ulang centroid cluster berdasarkan dokumen yang berada di dalam cluster tersebut.
4. Jika centroid tidak berubah (dengan treshold tertentu) maka stop. Jika tidak, kembali ke langkah 2.

$$Sim(d_x,d_y) = \frac{\sum_{k=1}^{n} x_k \times y_k}{\sqrt{\sum_{k=1}^{n} x_k^2} \times \sqrt{\sum_{k=1}^{n} y_k^2}} \quad (1)$$

2.2 Tahap Preprocessing

Preprocessing adalah tahapan mengubah suatu dokumen ke dalam format yang sesuai agar dapat diproses oleh algoritma clustering [4]. Terdapat tiga tahapan dalam proses *Preprocessing* dalam penelitian ini, yaitu : Tokenization, Stopword, dan Stemming.

2.3 Peringkasan Teks Dokumen Otomatis *(Automatic Text Summarization)*

Peringkas dokumen teks otomatis adalah ringkasan dari sumber teks oleh mesin untuk menampilkan informasi paling penting dalam bentuk pendek dari teks aslinya dengan tetap menjaga intisari dari dokumen tersebut dan membantu pengguna dengan cepat memahami informasi dalam jumlah besar [4].

2.3.1 Metode Berbasis Fitur

7 fitur tahapan yang digunakan dalam penelitian ini adalah:

- Fitur Judul

$$Skor(S_i) = \frac{jumlah\ kata\ pada\ judul}{jumlah\ kata\ yang\ sama\ dengan\ judul} \quad (2)$$

- Panjang Kalimat

$$Skor(S_i) = \frac{Jumlah\ kata\ yang\ terdapat\ pada\ kalimat}{Jumlah\ kata\ yang\ terdapat\ pada\ kalimat\ terpanjang} \quad (3)$$

- Bobot Kata

$$Skor(S_i) = \frac{Jumlah\ TF-ISF\ dalam\ kalimat}{Maksimal\ jumlah\ TF-ISF} \quad (4)$$

TF-IDF = jumlah kata pada dokumen * idf

$$= \text{jumlah kata pada dokumen} * \log\left(\frac{df}{N}\right) \quad (5)$$

df = jumlah kalimat yang mengandung kata x
N = jumlah kalimat dalam pada dokumen

- Posisi Kalimat

Skor(S_i) = 1 untuk kalimat pertama dan kalimat terakhir. (6)
0 untuk kalimat lainnya.

- Kesamaan Antar Kalimat

$$sim_{cos}(d_i,d_j) = \frac{d_i \cdot d_j}{||d_i||.\ ||d_j||}$$

$$= \frac{\sum_{k=1}^{n} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} \times \sqrt{\sum_{k=1}^{n} w_{jk}^2}} \quad (7)$$

w_{ik} = Bobot kata pada dokumen
w_{jk} = Bobot kata pada query
sedangkan untuk menghitung skor dari fitur ini adalah [HYPERLINK \l "Sua081" 4] :

$$Skor(S_i) = \frac{\text{jumlah cosine similarity}}{jumlah\ maksimal\ similarity} \} \quad (8)$$

- Kata Tematik

$$Skor(S_i) = \frac{jumlah\ kata\ tematik\ dalam\ kalimat}{panjang\ kalimat\ (jumlah\ kata\ pada\ kalimat)} \quad (9)$$

- Data Numerik

$$\text{Skor}(S_i) = \frac{\text{jumlah data numerik}}{\text{panjang kalimat (jumlah kata pada kalimat)}} \quad (10)$$

2.3.2 Metode Berbasis LSA (Latent Semantic Analysis)

LSA *(Latent Semantic Analysis)* adalah metode statistik aljabar yang mengekstrak struktur semantik yang tersembunyi dari kata dan kalimat [5], untuk mencari interelasi diantara kalimat dan kata, digunakan metode aljabar Singular Value Decomposition (SVD). Disamping mempunyai kapasitas relasi model diantara kata dan kalimat, SVD ini mempunyai kapasitas reduksi noise yang membantu untuk meningkatkan akurasi [6].

2.4 Document Representation Vector Space Model

VSM mengubah koleksi dokumen kedalam matrik *term-document* [7]. Pada gambar 1, dimana *d* adalah dokumen dan *w* adalah bobot atau nilai untuk setiap term.

$$A_{mxn} = \begin{array}{c} d1 \quad d2 \quad dn \\ \downarrow \quad \downarrow \quad \downarrow \\ \begin{bmatrix} \omega_{11} & \omega_{11} & \cdots \omega_{11} \\ \omega_{11} & \omega_{11} & \cdots \omega_{11} \\ \vdots & & \vdots \\ \omega_{m1} & \omega_{m2} & \cdots \omega_{mn} \end{bmatrix} \begin{array}{l} \leftarrow t1 \\ \leftarrow t2 \\ \\ \leftarrow tm \end{array} \end{array}$$

Gambar 1: Matrik Term-dokumen

2.5 TFIDF

TF adalah banyaknya kemunculan suatu *term* dalam suatu dokumen, IDF adalah perhitungan logaritma antara pembagian jumlah total dokumen dengan cacah dokumen yang mengandung suatu *term*, dan TFIDF adalah perkalian antara TF dengan *IDF* [3]. Dalam penelitian ini digunakan TFIDF sebagai metode *term weighting*.

$$IDF = log \frac{D}{DF} \quad (11)$$

$$TFIDF(t) = TF * log \frac{D}{DF} \quad (12)$$

2.6 Similiarity Measure

Pada Vector Space Model Dokumen direpresentasikan dalam bentuk $d = \{w_1, w_2, w_{3,\ldots}, w_n\}$ dimana *d* adalah dokumen dan *w* adalah nilai bobot setiap term dalam dokumen [3]. Dalam penelitian ini untuk menghitung persamaan antar dokumen akan mengukur jarak antar 2 dokumen d_i dan d_j, dengan menggunakan rumus *cosines similiarity*.

$$similiarity\ (d_i, d_j) = cosines\ \theta = \frac{\vec{di}.\vec{dj}}{||di||.||dj||} \quad (13)$$

2.7 Teknik Dimension Reduction

2.7.1 Feature Selection

Disebutkan bahwa hasil dari clustering teks mempunyai ketergantungan dengan kesamaan dokumen sehingga, kontribusi dari sebuah term dapat diartikan sebagai kontribusi terhadap kesamaan dokumen [3].

$$TC(t) = \sum_{i,j \cap 1 \neq j} f(t, di).f(t, dj) \quad (14)$$

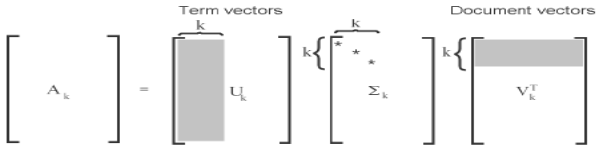
$f(t,d)$ merupakan bobot *tf*idf* dari term *t* di dokumen *d*.

2.7.2 Singular Value Decomposition

Latent Semantic Indexing (LSI) melalui metode Singular Value Decomposition (SVD) mengurai matrik term-document menjadi 3 matrik *U*, *S* dan *V* yang memiliki dimensi lebih kecil [3].

$$A = USV^T \quad (15)$$

U merupakan matrik term yang berdimensi *m* x *k*, *S* adalah matrik diagonal yang berisi eigen value berdimensi kxk dan V^T adalah matrik dokumen yang memiliki dimensi *k* x *n*.



mxn *mxk* *kxk* *kxn*

Gambar 2: Dekomposisi truncated SVD

Truncated SVD menggunakan pendekatan rank-k untuk mengurangi SVD [7], Dalam penelitian ini menggunakan peringkat-k pembulatan nilai akar dari jumlah 150 dokumen yang diproses, yaitu pembulatan dari $\sqrt{150}$ = 12.

2.8 Evaluation Measure

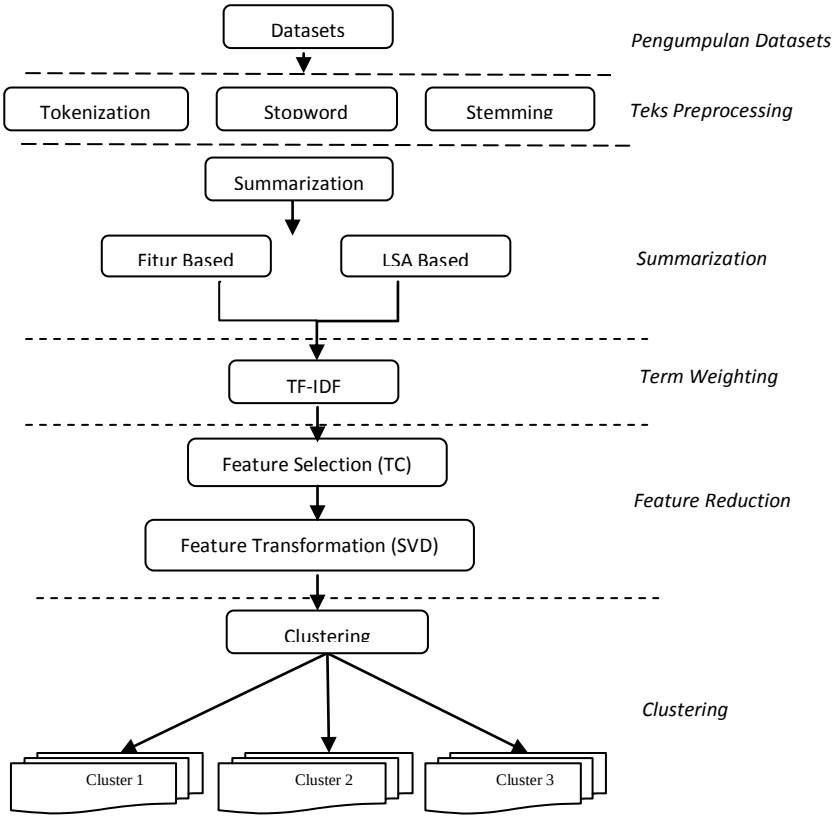
Recall dan precision kategori *i* dalam cluster *j* diperoleh dari persamaan berikut :

$$Recall\ (i,j) = \frac{nij}{ni} \qquad Precision\ (i,j) = \frac{nij}{nj} \tag{16}$$

Dinama n_{ij} merupakan jumlah dokumen kategori *i* dalam cluster *j*, n_i adalah jumlah dokumen dalam kategori *i* dan n_j merupakan jumlah dokumen dalam cluster *j* [3]. Kemudian untuk menghitung F-measure yang digunakan adalah persamaan berikut:

$$F(i,j) = \frac{2*(Precision*Recall)}{(Precision+Recall)} \qquad F = \sum_i \frac{ni}{n} max_{j=1,\ldots,k} F(i,j) \tag{17}$$

3. Model yang Diusulkan



Gambar 3: Model yang diusulkan.

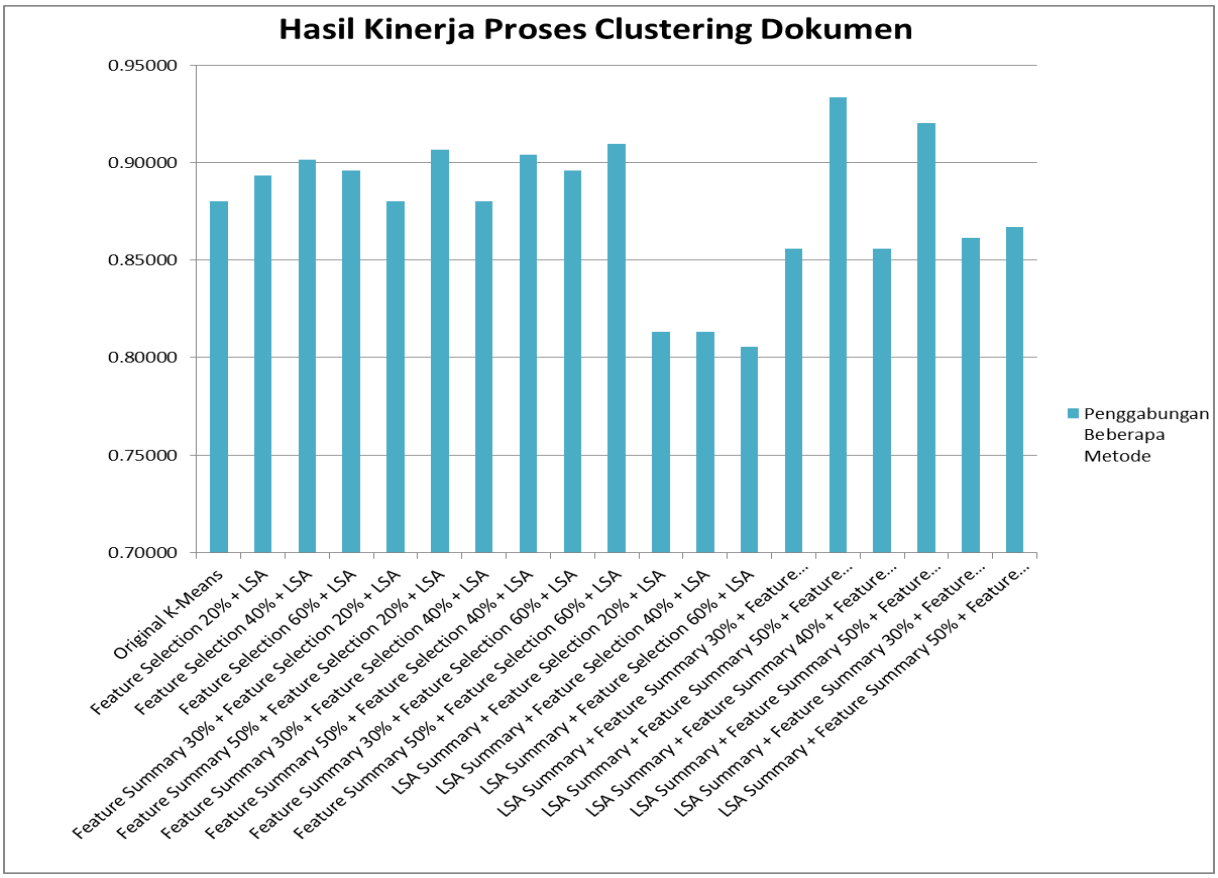
Dalam penelitian ini, mengusulkan beberapa tahapan model yang diterapkan, dimana seperti yang tergambar dalam Gambar 3 yaitu :

1. Tahap *Pengumpulan Datasets* yang berupa dokumen teks dalam bahasa Indonesia.
2. Tahap *Teks Preprocessing* yang terdiri dari *tokenization*, *stopword removal* dan *stemming*.
3. Tahap *Summarization* yaitu proses peringkas dokumen otomatis dengan penggabungan metode fitur dan Latent Semantic Analysis (LSA).
4. Tahap *Term-Weighting* yang akan menghasilkan matrik term-dokumen dengan dimensi m x n (Matrik $A_{m\,x\,n}$) dimana *m* adalah jumlah term dan *n* adalah jumlah dokumen.
5. Tahap *Feature Reduction* yang terdiri dari proses Feature Selection (TC) dimana hasil dari clustering teks mempunyai ketergantungan dengan kesamaan dokumen, sehingga kontribusi dari sebuah term dapat diartikan sebagai kontribusi terhadap kesamaan dokumen dan Feature Transformation (SVD), dimana SVD akan mengurai matrik term-dokumen menjadi 3 matrik yang berdimensi lebih kecil ($A_{c\,x\,n} = U * S * V^T$)
6. Tahap *Clustering*, menggunakan matrik V^T yang digunakan dalam proses pengelompokan dokumen dengan algoritma k-means.

4. Hasil dan Pembahasan

4.1. Akurasi

Dari hasil penelitian yang dilakukan dapat dibuktikan bahwa integrasi peringkas dokumen otomatis dengan menggabungkan metode fitur dan Latent Semantic Analysis (LSA) dapat meningkatkan akurasi hasil clustering pada dokumen teks berbahasa Indonesia.

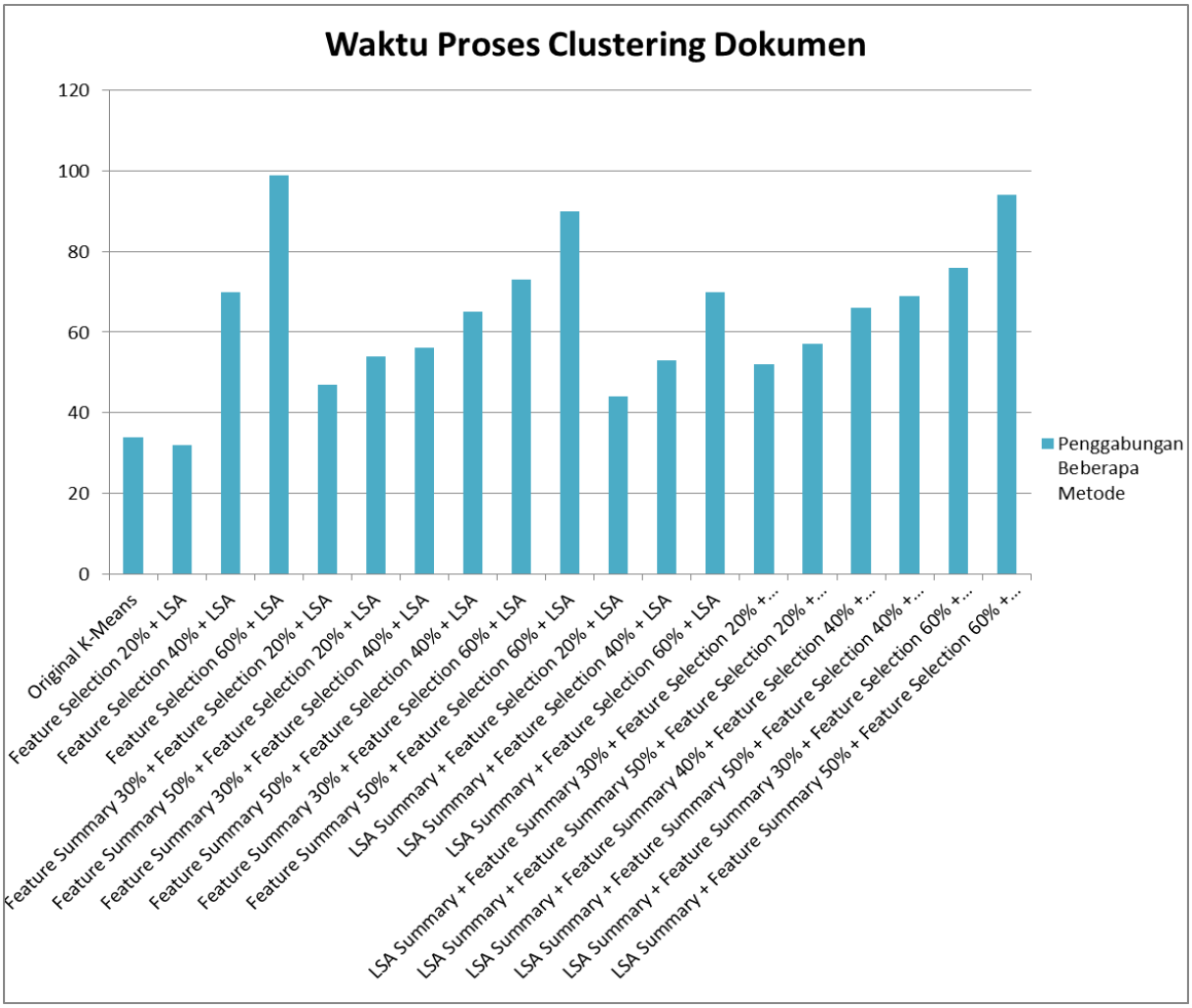


Gambar 4: Hasil kinerja proses clustering dokumen

Tingkat akurasi menggunakan peringkas dokumen otomatis yang diintegrasikan sebagai feature reduction dengan menggabungkan metode fitur dan metode LSA pada percobaan di atas mencapai 93,33 % yang diperoleh pada tingkat peringkas dokumen otomatis LSA Summary + Feature Summary 50% + Feature Selection 20% + LSA dibandingkan dengan feature selection 20 % tanpa menggunakan peringkas dokumen otomatis yang hanya mencapai tingkat akurasi 89,33 %. Dari gambar 4 juga dapat dilihat penurunan tingkat akurasi untuk % feature selection yang lain, akan tetapi pada proporsi 60 % feature selection integrasi peringkas dokumen otomatis dengan metode LSA mengalami penurunan tingkat akurasi.

4.2. Waktu

Waktu rata-rata yang diambil diukur mulai dari proses preprocessing sampai dengan hasil clustering diperoleh.



Gambar 5: Waktu proses clustering dokumen

Pada % feature selection yang semakin kecil feature reduction yang diintegrasikan dengan peringkas dokumen otomatis membutuhkan tambahan waktu komputasi tersendiri, dari percobaan yang dilakukan untuk 20% feature selection terdapat peningkatan waktu komputasi dari percobaan clustering tanpa peringkas dokumen otomatis, menggunakan peringkas dokumen otomatis dengan proporsi 30%, 40% dan 60%. Akan tetapi pada proporsi feature selection yang semakin besar, % peringkas dokumen otomatis dapat menurunkan waktu komputasi yang ada, pada percobaan 40% dan 60% feature selection dapat dilihat pada Gambar 5 bahwa integrasi peringkas dokumen otomatis sebagai feature reduction dapat mengurangi rata-rata waktu komputasi yang dibutuhkan.

5. Kesimpulan

Berdasarkan percobaan-percobaan yang telah dilakukan dapat disimpulkan bahwa Peringkas Dokumen Otomatis dengan Penggabungan Metode Fitur dan Latent Semantic Analysis (LSA) pada Proses Clustering Dokumen Teks Berbahasa Indonesia berpengaruh pada penggabungan metode Fitur dan LSA yang dapat dilihat melalui tingkat akurasi yang berbeda-beda pada masing-masing percobaan. Hasil penelitian menunjukkan bahwa integrasi peringkas dokumen otomatis dengan penggabungan metode Fitur dan LSA dapat meningkatkan kinerja clustering dokumen sampai dengan 93,33 %, mengalami peningkatan dari tingkat akurasi 90,67 % untuk proses feature reduction standar tanpa menggunakan peringkas dokumen otomatis dan 88 % tingkat akurasi clustering standar. Hasil pengujian menunjukkan bahwa kombinasi tahapan teks preprocessing yang terbaik yaitu peringkat dokumen otomatis dengan menggabungkan LSA Summary + Feature Summary 50% + Feature Selection 20% + LSA dimana pada percobaan tersebut menghasilkan tingkat akurasi yang tertinggi, yaitu sebesar 93.33%.

DAFTAR PUSTAKA

[1] L. Muflikhah and B. Baharudin, "Document Clustering Using Concept Space and Cosine Similarity Measurement," *2009 Int. Conf. Comput. Technol. Dev.*, pp. 58–62, 2009.

[2] C. Supriyanto and A. Affandy, "Kombinasi Teknik Chi Square Dan Singular Value Decomposition Untuk Reduksi Fitur Pada Pengelompokan Dokumen," *Semantik*, vol. 2011, Semantik, 2011.

[3] A. Salam, C. Supriyanto, and A. Fahmi, "Integrasi Peringkas Dokumen Otomatis Sebagai Feature Reduction Pada Clustering Dokumen," vol. 2012, Semantik, pp. 145–150, 2012.

[4] L. Suanmali, N. Salim, and M. Binwahlan, "Automatic text summarization using feature based fuzzy extraction," *J. Teknol. Mklm.*, vol. 2, Desember, 2008.

[5] M. Ozsoy, I. Cicekli, and F. Alpaslan, "Text summarization of turkish texts using latent semantic analysis," *Proc. 23rd Int. ...*, 2010.

[6] M. Ozsoy, F. Alpaslan, and I. Cicekli, "Text summarization using latent semantic analysis," *J. Inf. Sci.*, February, pp. 1–81, 2011.

[7] R. Peter and G. Shivapratap, "Evaluation of SVD and NMF methods for latent semantic analysis," *... J. Recent ...*, vol. 1, no. 3, pp. 308–310, 2009.