

## Peringkasan Teks Otomatis menggunakan Metode *Latent Semantic Analysis* pada Artikel Berita Ekonomi berbahasa Indonesia

Manat Hendry Fernando Sianturi<sup>1</sup>, Achmad Ridok<sup>2</sup>, Edy Santoso<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>manatsianturi@student.ub.ac.id, <sup>2</sup>acridokb@ub.ac.id, <sup>3</sup>edy144@ub.ac.id

### Abstrak

Teks artikel berita merupakan dokumen yang memuat berbagai macam informasi yang memiliki ragam topik salah satunya ekonomi. Teks artikel berita umumnya terdiri atas banyak paragraf sehingga dibutuhkan sebuah sistem untuk mengekstrak informasi untuk menyajikan ide-ide pokok atau informasi-informasi penting dari sebuah artikel berita kepada pembaca. Peringkasan dokumen secara otomatis merupakan solusi untuk membantu mendapatkan intisari dari dokumen. Pada penelitian ini memaparkan peringkasan artikel berita ekonomi berbahasa Indonesia menggunakan metode *Latent Semantic Analysis* yang menggunakan pendekatan aljabar linear *singular value decomposition* (SVD) dengan membentuk matriks representasi dari asosiasi term yang merupakan kata-kata pada dokumen yang berhubungan erat dari proses perhitungan TF-IDF. Pengujian penelitian ini memperoleh nilai rata-rata precision, recall, f-measure dan akurasi pada nilai compression rate sebesar 10% secara berurutan adalah 0.791667, 0.148054, 0.242747 dan 0.90152, pada nilai compression rate sebesar 30% secara berurutan adalah 0.475, 0.171828, 0.2264444 dan 0.787366 dan pada nilai compression rate sebesar 50% secara berurutan adalah 0.357857, 0.268677, 0.293111, 0.628009. Pada evaluasi pengujian yang dilakukan menghasilkan precision dan akurasi terbaik pada saat compression rate 10% dengan nilai 0.7416 dan akurasi 0.909853. Sedangkan untuk recall, dan f-measure terbaik dihasilkan pada saat compression rate 50% dengan nilai 0.25201 dan 0.284788.

**Kata kunci:** *peringkasan teks otomatis, latent semantic analysis, compression rate*

### Abstract

*The text of a news article is a document that contains various kinds of information on a variety of topics, one of them is the economy. The text of news articles generally consists of many paragraphs, so a system is needed to extract information to present the main ideas or important informations from a news article to readers. Automatic document summarization is a solution to help get the digest from the entire document. This research presents a summarization of Indonesian economic news articles using the Latent Semantic Analysis method, which uses a linear algebraic singular value decomposition (SVD) approach by forming a matrix representation of term associations, which are words in documents that are closely related in the TF-IDF calculation process. This research test obtained the average values of precision, recall, f-measure, and accuracy at a compression rate of 10% in order: 0.791667, 0.148054, 0.242747, and 0.90152; at a compression rate of 30% in order: 0.475, 0.171828, 0.2264444, and 0.787366; and at a compression rate of 50% in order: 0.357857, 0.268677, 0.293111, and 0.628009. In the evaluation of the tests carried out, it produces the best precision and accuracy when the compression rate is 10%, with a value of 0.7416 and an accuracy of 0.909853. As for recall, the best f-measure is produced when the compression rate is 50%, with a value of 0.252.*

**Keywords:** *automatic text summarization, latent semantic analysis, compression rate*

## 1. PENDAHULUAN

Informasi dapat beredar dengan cepat dan luas dari berbagai belahan dunia melalui internet. Kemudahan akses internet yang dapat

di akses di mana saja merupakan salah satu faktor utama yang menyebabkan informasi dapat dengan mudah diperoleh. Perkembangan internet dengan pesat juga berdampak terhadap bertambahnya jumlah informasi yang

mengakibatkan sangat sulit untuk mendapatkan informasi secara efisien (Desai & Shah, 2016). Berita merupakan sebuah informasi yang berguna untuk menyampaikan fakta kepada seluruh orang. Banyak situs yang menyediakan informasi berita yang terpercaya dan beragam topik, seperti detik.com, Kompas.com, kumparan.com dan masih banyak lagi situs lainnya. Masing-masing situs tersebut memiliki beraneka ragam topik berita antara lain olahraga, politik, kesehatan, dan ekonomi.

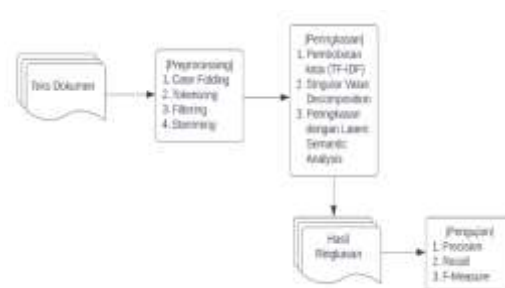
Banyaknya informasi yang tersedia pada data teks dokumen yang salah satunya adalah artikel berita membuat proses pengolahan data dari dokumen artikel berita menjadi lebih sulit dilakukan serta tidak efisien. Sehingga diperlukan sebuah peringkasan teks otomatis yang dapat membantu pengurangan isi dokumen yang lebih ringkas dan singkat dengan mengambil inti dokumen serta membuang term tanpa mengurangi makna dari dokumen tersebut (Luthfiarta, et al., 2013). Sehingga dihasilkan sebuah informasi yang bermanfaat dengan jelas dan ringkas, dimana hal ini dapat menghemat waktu serta tenaga yang ada.

Peringkasan teks otomatis (*automatic text summarization*) adalah pembuatan ringkasan dari sebuah teks secara otomatis dengan memanfaatkan komputer. Peringkasan teks otomatis juga merupakan proses mengurangi dokumen teks dengan komputer untuk menciptakan sebuah ringkasan yang mempertahankan poin penting dari dokumen asli (Mustaqhfi, Abidin, & Kusumawati, 2011).

Peringkasan teks dilakukan agar dapat membantu pembaca memperoleh inti dari sebuah informasi dari sebuah berita dengan waktu yang cepat. Beberapa metode peringkasan teks dilakukan. Pada penelitian ini akan dicoba pengembangan peringkasan teks dengan menggunakan metode latent semantic analysis (LSA). Metode TF-IDF dari penelitian yang dilakukan oleh (Evan, Pranowo, & Purnomo, 2014) sebagai metode dalam proses pembobotan kata berdasar frekuensi kemunculan kata sehingga kata dengan nilai TF-IDF tinggi, merupakan kata itu memiliki hubungan erat dengan dokumen yang terdapat kemunculan kata tersebut. Kemudian dilakukan langkah selanjutnya yang pernah dilakukan oleh (Jamhari, Noersasongko, & Subagyo, 2014) yaitu metode latent semantic analysis merupakan metode yang menerapkan ekstraksi dari makna tersembunyi atau dikenal dengan makna semantik yang ada pada sebuah kalimat.

Pada penelitian ini akan dibangun sebuah sistem untuk melakukan peringkasan teks agar dapat membantu pembaca memperoleh inti dari sebuah informasi dari sebuah berita ekonomi dengan waktu yang cepat dengan menggunakan metode latent semantic analysis (LSA).

## 2. METODE PENELITIAN



Gambar 1. Alur Peringkasan Teks

Gambar 1 merupakan alur peringkasan teks yang dilakukan penulis, yaitu dimulai dari pengumpulan data teks dokumen dari internet diantaranya berita ekonomi dari situs berita detik.com, dilakukannya *preprocessing* dari pengurain teks dokumen, *case folding*, *tokenizing*, *stopword (filtering)*, *stemming* dan menghasilkan *term* dokumen. Setelahnya dilakukan peringkasan dengan metode pembobotan kata (TF-IDF), menghitung *singular value decomposition* implementasi *cross latent semantic analysis* untuk menghasilkan hasil ringkasan yang kemudian diuji dengan teknik *Performance Measure*.

### 2.1 Preprocessing Text

*Preprocessing text* merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Suatu teks tidak dapat diproses langsung oleh algoritma pencarian, oleh karena itu dibutuhkan *preprocessing text* untuk mengubah teks menjadi data numerik. Proses ini terdiri dari beberapa tahap pembersihan dokumen seperti berikut:

#### 1. Case Folding

Merupakan tahapan dimana akan dilakukan pemisahan *string* atau struktur di setiap kalimat dengan pembatas simbol titik menjadi komponen-komponen terpisah. Setiap dokumen yang telah dipecah akan dimasukkan ke dalam *list* kalimat. Keluaran dari hasil segmentasi berupa kumpulan kalimat yang akan digunakan pada proses berikutnya.

## 2. Tokenizing

*Tokenizing* adalah proses memecah dokumen menjadi kumpulan kata. *Tokenizing* dapat dilakukan dengan menghilangkan tanda baca dan memisahkannya per-spasi. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

## 3. Stopword Removal atau Filtering

*Stopwords removal* merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak.

## 4. Stemming

Merupakan sebuah tahapan mengubah data hasil *filtering* menjadi kata dasar dengan menghilangkan imbuhan yang ada pada setiap kata. Pada proses ini digunakan *library python* Sastrawi, dengan menghilangkan imbuhan pada kata dan menjadikan kata dasar sehingga mengurangi variasi kata dengan kata yang juga memiliki kata dasar sama.

## 2.2 TF-IDF

Metode *Term Frequency Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. TF-IDF ini adalah sebuah dokumen atau dalam sekelompok kata. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen. (Intan, 2006)

Pada algoritma TF-IDF digunakan rumus untuk menghitung bobot (W) masing-masing dokumen terhadap kata kunci dengan rumus (Al-Talib, 2013) pada Persamaan 1 berikut:

$$TF - IDF(ti, dj) = tf(ti, dj) \log N / ni \quad (1)$$

Keterangan:

TF-IDF = pembobotan kata

tf(ti, dj) = banyaknya kata yang dicari

N = total dokumen dalam *dataset*

ni = total dokumen

## 2.3 Latent Semantic Analysis

*Latent Semantic Analysis* (LSA) merupakan sebuah metode yang memanfaatkan model statistik matematis untuk menganalisa struktur

semantik suatu teks. LSA dapat digunakan untuk memberikan nilai pada teks dengan mengkonversikan teks menjadi matriks-matriks yang diberi nilai pada masing-masing *term* untuk dicari kesamaan dengan *term* referensi. Metode LSA menerima masukan berupa file teks yang selanjutnya direpresentasikan sebagai matriks. *Latent semantic Analysis* (LSA) diusulkan dan dipatenkan pada tahun 1998 oleh Scott Deerwester, Thomas Launder, dan Susan Dumais.

LSA sendiri digunakan untuk mengidentifikasi semantik kalimat-kalimat dengan peringkat tertinggi dan yang beda dari yang lain. LSA menggunakan metode aljabar linear *singular value decomposition* (SVD).

Pada perhitungan *Singular value decomposition* (SVD) diperlukan *term* matriks yang telah dibentuk sebelumnya. Tujuan dilakukan SVD yaitu untuk menemukan pola atau hubungan baru antar kata. Hasil dari perhitungan SVD menghasilkan 3 komponen matriks yaitu dua matriks *orthogonal* dan satu matrik diagonal. SVD dari pemfaktoran sebuah matriks A yang berdimensi t x d adalah seperti pada Persamaan 2 berikut.

$$A_{t \times d} = U_{t \times n} \Sigma V_{d \times n}^T \quad (2)$$

Keterangan :

A = Input Matriks

U = Matriks berdimensi t x n

V = Matriks berdimensi d x n

$\Sigma$  = Matriks diagonal berdimensi k x k

t = jumlah baris matriks

d = jumlah kolom matriks

Berikut langkah-langkah untuk mencari nilai U dan V dalam perhitungan SVD:

1. Hitung  $A^T \times A$ . Pada tahap awal dilakukan perkalian antara matriks A yang ditranspose dengan matriks.
2. Cara determinan hasil perkalian matriks  $A^T \times A$ . Pada langkah kedua dilakukan determinan  $A^T \times A$  untuk mencari nilai eigen berikut merupakan persamaan rumus determinan seperti pada Persamaan 3 dan 4 berikut:

$$\det[A^T \times A] = 0 \quad (3)$$

$$|A^T \times A - \lambda I| = 0 \quad (4)$$

Setelah dilakukan proses determinan pada matriks  $A^T \times A$  maka akan dihasilkan nilai eigen sesuai dengan banyaknya kolom. Jika

matriks mempunyai tiga kolom maka nilai eigen terdapat  $\lambda_1, \lambda_2, \lambda_3$ . Setelah ditemukan nilai eigen atau *eigen value*, nilai ini dapat digunakan untuk mencari nilai *singular value*. Nilai *singular value* diperoleh dari akar dari tiap nilai eigen. Hasil nilai *singular* digunakan untuk membentuk matriks  $S$  dengan ketentuan  $S_1 > S_2 > S_3 \dots > S_n$ .

3. Hitung setiap eigen vektor  $V$  dengan mengevaluasi Persamaan 5.

$$|A^T \times A - \lambda I| \times X_i \quad (5)$$

dimana nilai *eigen value* dihitung pada tahap sebelumnya. Hasil perhitungan menghasilkan vektor  $V_1, V_2, V_3, \dots, V_n$

4. Bentuk matriks  $V$  dengan tiap eigen vektor  $V_1, V_2, V_3, \dots, V_n$ . Menjadi kolom pada matriks  $V$ .

5. Bentuk matriks  $U$  dengan rumus seperti pada Persamaan 6.

$$U = AVS^{-1} \quad (6)$$

setelah melakukan proses SVD maka langkah selanjutnya yaitu dilakukan reduksi matriks hasil perhitungan SVD. Tujuannya untuk menghilangkan *noise* atau kata-kata yang dianggap tidak penting tanpa menghilangkan korelasi antar kata dengan kalimat pada dimensi pada matriks.

6. Untuk proses tahap pemilihan kalimat yang nantinya akan dijadikan ringkasan dengan berdasarkan kalimat yang ada pada matriks  $V^T$ . Salah satu metode penyusunan ringkasan atau ekstraksi ringkasan dengan menggunakan pendekatan *method latent semantic analysis*. Pada metode ini pada proses pemilihan kalimat atau ekstraksi ringkasannya dengan melakukan perhitungan nilai rata-rata dan panjang (*length*) yang diperoleh dari matriks  $V^T$  serta matriks  $\sum$  seperti pada Persamaan 7 berikut.

$$length = \sqrt{\sum_{j=1}^n V_{ij}^2 \times S_{jj}^2} \quad (7)$$

Keterangan :

i = baris matriks

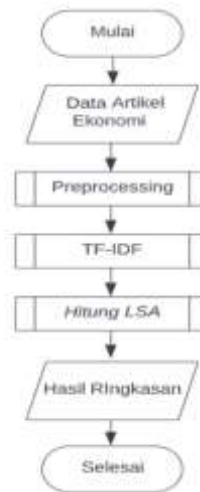
j = kolom matriks

### 3. IMPLEMENTASI

Tahapan implementasi pada penelitian ini dilakukan dengan dua tahapan algoritme yakni diantaranya *preprocessing* yang digunakan untuk mereduksi data teks menjadi *term-term* yang signifikan dalam rangka memangkas waktu

komputasi yang kemudian dihitung dengan menggunakan TF-IDF.

Tahapan pada implementasi sistem adalah ditunjukkan pada Gambar 2 sebagai berikut



Gambar 2. Alur Proses Penelitian

Setelah diperoleh nilai TF-IDF maka proses perhitungan dengan LSA (*latent semantic analysis*) akan di proses dengan membuat ringkasan dengan pendekatan identifikasi kesamaan makna dengan peringkat tertinggi suatu kalimat-kalimat dan yang berbeda dari yang lain dengan menganalisa antara kumpulan dokumen dan *term* yang memiliki keterkaitan dengan dokumen dan *term* satu sama lain yang mana LSA ini menggunakan SVD (*singular value decomposition*) dengan pendekatan ke fitur semantik. SVD digunakan untuk mereduksi *noise* untuk membantu meningkatkan akurasi. Hasil peringkasan dokumen disusun berdasar jumlah persentase *compression rate* yang mengambil sejumlah CR dari keseluruhan jumlah dokumen dimana sebelumnya dilakukan perhitungan dengan proses ekstraksi ringkasan menggunakan *cross method* yang mengurutkan ringkasan berdasar perhitungan nilai panjang setiap kalimat.

### 4. HASIL DAN PENGUJIAN

Pengujian pada *compression rate* ini dilakukan dengan dilakukan uji coba sebanyak 3 kali dari besar persentasenya yaitu dengan nilai 10%, 30% dan 50% untuk setiap data teks dokumen yang akan digunakan.

Dipakainya nilai variasi persentase diatas karena dengan nilai *compression rate* dengan besar 10-50% sudah dapat dikatakan cukup untuk menghasilkan sebuah ringkasan yang cukup baik (Alguliev & Aliguliyev, 2009).

Namun sebaiknya sebuah ringkasan tidak lebih dari 50% dari seluruh isi dokumen teks.

Dalam pengujian *compression rate* data artikel terlebih dahulu di uji agar mendapatkan nilai dari True positive (tp) merupakan kalimat yang ada dalam ringkasan manual dan muncul dalam ringkasan sistem. False positive (fp) kalimat yang ada dalam ringkasan manual tapi tidak muncul di sistem. False negative (fn) merupakan kalimat yang ada pada ringkasan manual tapi tidak muncul pada ringkasan sistem. Dapat dilihat pada Tabel 1 Kelas Pengujian.

Tabel 1. Kelas Pengujian

| Artikel | Kelas Pengujian |    |    |    |     |    |    |    |     |    |    |    |
|---------|-----------------|----|----|----|-----|----|----|----|-----|----|----|----|
|         | 10%             |    |    |    | 30% |    |    |    | 50% |    |    |    |
|         | TP              | FP | FN | TN | TP  | FP | FN | TN | TP  | FP | FN | TN |
| 1       | 2               | 1  | 0  | 8  | 2   | 1  | 1  | 8  | 3   | 0  | 3  | 7  |
| 2       | 1               | 0  | 1  | 11 | 1   | 0  | 3  | 9  | 1   | 1  | 4  | 7  |
| 3       | 2               | 1  | 0  | 15 | 2   | 1  | 2  | 13 | 2   | 2  | 3  | 11 |
| 4       | 3               | 0  | 1  | 16 | 4   | 1  | 2  | 13 | 4   | 2  | 4  | 10 |
| 5       | 1               | 0  | 1  | 6  | 1   | 0  | 2  | 5  | 1   | 1  | 3  | 4  |
| 6       | 1               | 1  | 0  | 12 | 1   | 2  | 1  | 9  | 2   | 1  | 3  | 8  |
| 7       | 1               | 1  | 0  | 11 | 1   | 2  | 1  | 8  | 2   | 1  | 3  | 7  |
| 8       | 2               | 0  | 1  | 11 | 2   | 1  | 2  | 9  | 3   | 0  | 4  | 7  |
| 9       | 1               | 0  | 1  | 7  | 1   | 0  | 2  | 6  | 1   | 1  | 3  | 4  |
| 10      | 1               | 0  | 1  | 5  | 1   | 0  | 2  | 4  | 1   | 0  | 4  | 2  |

Maka dari data tersebut akan dilakukan perhitungan *precision* dan *recall* sebagai berikut dengan acuan:

a. *Precision*

*Precision* adalah berapa banyak dokumen yang berhasil diambil oleh sistem. Rumus perhitungan *precision* seperti pada Persamaan 8.

$$\frac{tp}{tp+fp} \quad (8)$$

b. *Recall*

*Recall* ialah kemampuan untuk mengambil peringkat teratas yang sebagian besar relevan (benar). Rumus perhitungan *recall* seperti pada Persamaan 9.

$$\frac{tp}{tp+fn} \quad (9)$$

c. *F-Measure*

*F-measure* digunakan untuk mengukur kualitas *recall* dan *precision*. Rumus perhitungan *f-measure* seperti pada Persamaan 10. .

$$\frac{2 \times (Precision \times recall)}{precision + recall} \quad (10)$$

d. Akurasi

Akurasi merupakan salah satu perhitungan evaluasi yang mengkombinasikan antara kalimat yang terpilih oleh manusia, kalimat yang tidak terpilih oleh manusia, dan kalimat hasil pengambil sistem. Rumus perhitungan akurasi seperti pada persamaan 11.

$$\frac{tp+tn}{tp+fn+fp+tn} \quad (11)$$

Dimana :

*tp* = jumlah kalimat yang berhasil diekstrak sistem sesuai dengan kalimat yang diringkas chatgpt

*fp* = jumlah kalimat yang diekstrak sistem tetapi tidak terdapat dalam kalimat yang diringkas chatgpt

*fn* = jumlah kalimat yang diringkas chatgpt tetapi tidak terdapat dalam kalimat yang diekstrak sistem.

*tn* = jumlah kalimat yang tidak diringkas chatgpt dan tidak diekstrak sistem.

#### 4.1 Pengujian Compression rate 10%

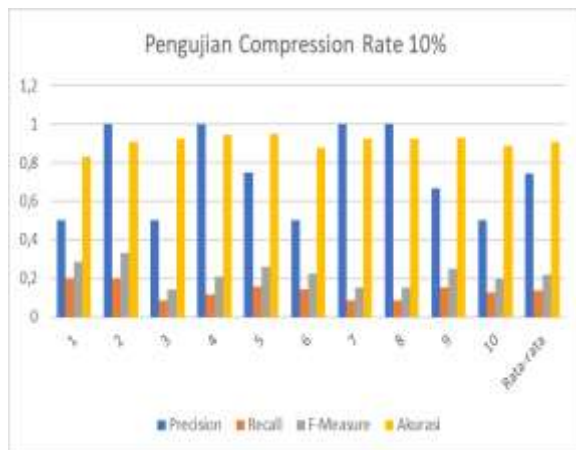
Tabel 2. Pengujian *Compression rate* 10%

| Dokumen   | Comparison Rate |          |           |         |
|-----------|-----------------|----------|-----------|---------|
|           | 10%             |          |           |         |
| Dokumen   | Precision       | Recall   | F-Measure | Akurasi |
| 1         | 0.5             | 0.2      | 0.28571   | 0.83333 |
| 2         | 1               | 0.2      | 0.33333   | 0.90909 |
| 3         | 0.5             | 0.08333  | 0.14285   | 0.92307 |
| 4         | 1               | 0.11764  | 0.21052   | 0.94444 |
| 5         | 0.75            | 0.15789  | 0.26087   | 0.95    |
| 6         | 0.5             | 0.14285  | 0.22222   | 0.875   |
| 7         | 1               | 0.08333  | 0.15384   | 0.92307 |
| 8         | 1               | 0.08333  | 0.15384   | 0.92307 |
| 9         | 0.66667         | 0.15384  | 0.25      | 0.92857 |
| 10        | 0.5             | 0.125    | 0.2       | 0.88889 |
| Rata-rata | 0.791667        | 0.148054 | 0.242747  | 0.90152 |

Berdasarkan Tabel 2. dapat dilihat bahwa nilai rata-rata *precision*, *recall*, *fmeasure* dan akurasi pada nilai *compression rate* sebesar 10% secara berurutan adalah 0.791667, 0.148054,



0.242747 dan 0.90152. Sedangkan nilai *precision*, *recall*, *f-measure* dan akurasi tertinggi pada *compression rate* sebesar 10% secara berurutan adalah 1, 0.2, 0.33333 dan 0.90909 yakni pada dokumen 2 seperti yang terdapat pada Gambar 3. Persentase *compression rate* menentukan banyaknya kalimat yang akan dipilih sebagai hasil ringkasan.



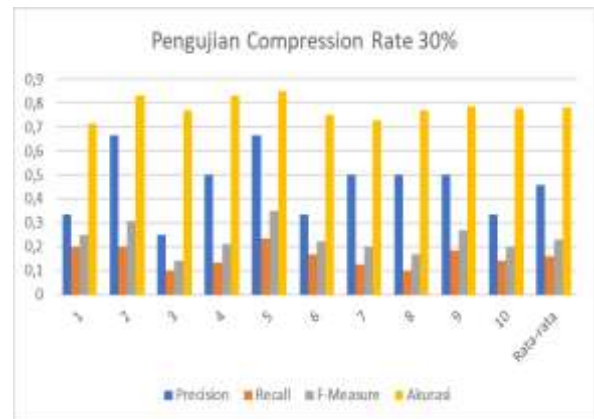
Gambar 3. Grafik Compression rate 10%

#### 4.2 Pengujian Compression rate 30%

Tabel 3. Pengujian *Compression rate* 30%

| Dokumen   | Comparison Rate |          |           |          |
|-----------|-----------------|----------|-----------|----------|
|           | 30%             |          |           |          |
|           | Precision       | Recall   | F-Measure | Akurasi  |
| 1         | 0.33333         | 0.2      | 0.25      | 0.71428  |
| 2         | 0.66667         | 0.2      | 0.30769   | 0.83333  |
| 3         | 0.25            | 0.1      | 0.14285   | 0.76923  |
| 4         | 0.5             | 0.13333  | 0.21052   | 0.83333  |
| 5         | 0.66667         | 0.23529  | 0.34782   | 0.85     |
| 6         | 0.33333         | 0.16667  | 0.22222   | 0.75     |
| 7         | 0.5             | 0.125    | 0.2       | 0.72727  |
| 8         | 0.5             | 0.1      | 0.16667   | 0.76923  |
| 9         | 0.5             | 0.18181  | 0.26667   | 0.78571  |
| 10        | 0.33333         | 0.14285  | 0.2       | 0.77778  |
| Rata-rata | 0.475           | 0.171828 | 0.226444  | 0.787366 |

Berdasarkan Tabel 3 dari hasil pengujian dapat dilihat bahwa nilai rata-rata *precision*, *recall*, *f-measure* dan akurasi pada nilai *compression rate* sebesar 30% secara berurutan adalah 0.475, 0.171828, 0.226444 dan 0.787366. Sedangkan nilai *precision*, *recall*, *f-measure* dan akurasi tertinggi pada *compression rate* sebesar 30% secara berurutan adalah 0.66667, 0.23529, 0.34782 dan 0.85 yakni pada dokumen 5 seperti yang terdapat pada Gambar 4.



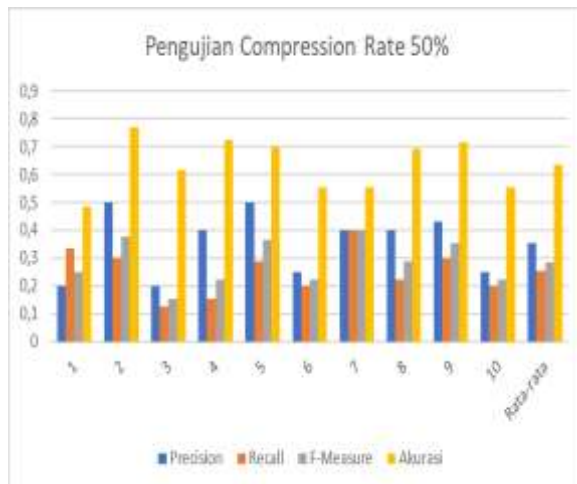
Gambar 4. Grafik Compression rate 30%

#### 4.3 Pengujian Compression rate 50%

Tabel 4. Pengujian *Compression rate* 50%

| Dokumen   | Comparison Rate |          |           |          |
|-----------|-----------------|----------|-----------|----------|
|           | 50%             |          |           |          |
|           | Precision       | Recall   | F-Measure | Akurasi  |
| 1         | 0.2             | 0.33333  | 0.25      | 0.48257  |
| 2         | 0.5             | 0.3      | 0.375     | 0.76923  |
| 3         | 0.2             | 0.125    | 0.15384   | 0.61538  |
| 4         | 0.4             | 0.15384  | 0.22222   | 0.72222  |
| 5         | 0.5             | 0.28571  | 0.36363   | 0.7      |
| 6         | 0.25            | 0.2      | 0.22222   | 0.55556  |
| 7         | 0.4             | 0.4      | 0.4       | 0.55556  |
| 8         | 0.4             | 0.22222  | 0.28571   | 0.6923   |
| 9         | 0.42857         | 0.3      | 0.35294   | 0.71428  |
| 10        | 0.25            | 0.2      | 0.22222   | 0.55556  |
| Rata-rata | 0.357857        | 0.268677 | 0.293111  | 0.628009 |

Berdasarkan Tabel 4. dari hasil pengujian dapat dilihat bahwa nilai rata-rata *precision*, *recall*, *f-measure* dan akurasi pada nilai *compression rate* sebesar 50% secara berurutan adalah 0.357857, 0.268677, 0.293111, 0.628009. Sedangkan nilai *precision*, *recall*, *f-measure* dan akurasi tertinggi pada *compression rate* sebesar 50% secara berurutan adalah 0.5, 0.3, 0.375 dan 0.76923 yakni pada dokumen 2 seperti pada Gambar 5.



Gambar 5. Grafik Compression rate 50%

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian Peringkasan teks otomatis menggunakan metode *latent semantic analysis* pada artikel berita ekonomi berbahasa Indonesia dapat diterapkan dengan cara *Latent semantic analysis* sebagai algoritma untuk mendapat kalimat-kalimat yang memiliki keterkaitan kata dengan pendekatan secara semantik dengan menggunakan SVD sebagai fitur penghilang redudansi atau *noise* pada kata tertentu. Serta penggunaan metode *Latent Semantic Analysis* sebagai pengekstraksi ringkasan yang akan dipilih dari artikel berita ekonomi dalam data teks dokumen.

Dari pengujian yang telah dilakukan dapat ditarik kesimpulan bahwa Metode *Latent Semantic Analysis* dapat digunakan untuk melakukan peringkasan teks. Selanjutnya, Pengujian dengan menggunakan *compression rate* berbeda dapat memengaruhi nilai *precision*, *recall*, *f-measure* dan akurasi.

Hasil akurasi pada penelitian ini diperoleh dengan cara pengujian akurasi *precision*, *recall*, *f-measure* dan akurasi terhadap seluruh teks dokumen yang digunakan. Dimana diperoleh ringkasan terbaik dengan nilai *compression rate* 10% yang didapati nilai *precision*, *recall*, *f-measure* serta akurasi secara berurutan adalah 0.7916, 0.14850, 0.2427 dan 0.9015

Untuk penelitian yang akan datang diharapkan bisa menerapkan algoritma *Cosine Similarity* atau metode lainnya yang sesuai kebutuhan. Hal tersebut bertujuan untuk mencocokkan antara judul dengan kalimat hasil ringkasan. Pada proses *Cosine Similarity*, dokumen yang sudah berhasil diringkas menggunakan metode *Latent Semantic Analysis*. Dengan menambahkan algoritma

*Cosine Similarity* diharapkan dapat menghasilkan akurasi yang lebih baik lagi,

## 6. DAFTAR PUSTAKA

- Desai, N., & Shah, P., 2016. Automatic Text Summarization Using Supervised Machine Learning Technique for Hindi Language. *International Journal of Research in Engineering and Technology*. Volume 5, pp.361–367.
- Luthfiarta, A., Zeniarja, J., Salam, A., 2014. Integrasi peringkasan dokumen otomatis dengan algoritma latent semantic analysis (LSA) pada peringkasan dokumen otomatis untuk proses clustering dokumen. Vol. 13, No. 1, pp.61-68.
- Mustaqhfiri, M., Abidin, Z., & Kusumawati, R. (2011). Peringkasan teks otomatis berita berbahasa Indonesia menggunakan metode Maximum Marginal Relevance. *MATICS*, 4(4), 134-147.
- Sari, Y. M., Fatonah, N. S. (2021). Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode Cross Latent Semantic Analysis (CLSA). *Jurnal Edukasi dan Penelitian Informatika*. Vo. 7, No. 2.
- Gotami, N. S., Indriati, I., & Dewi, R. K. (2018). Peringkasan Teks Otomatis Secara Ekstraktif Pada Artikel Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Latent semantic analysis. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), 2821-2828
- Dalal, V., & Malik, L. G. (2014). A Survey of Extractive and Abstractive Text Summarization Techniques, 1-2.
- Jamhari, M., Noersasongko, E., & Subagyo, H. (2014). Pengaruh Peringkasan Dokumen Otomatis Dengan Penggabungan Metode Fitur dan Latent semantic analysis (LSA) Pada Proses Clustering Dokumen Teks Berbahasa Indonesia. 1, 2355-5920
- Evan, F. H., Pranowo, & Purnomo, S. Y. (2014). Pembangunan Perangkat Lunak Peringkasan Dokumen dari Banyak Sumber Menggunakan Sentence Scoring dengan Metode TF-IDF. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 17-22.

- Babar, S. A., & Patil, P. D., 2015. Improving Performance of Text Summarization. *Procedia Computer Science*. pp.354–363.
- N. F. Saraswati, I. Indriati and R.S. Perdana, “Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2018.
- A. Ridok, (2014). “Peringkasan Dokumen Bahasa Indonesia Berbasis Non-Negative Matrix Factorization (NMF),” *Jurnal Teknologi Informasi dan Ilmu Komputer*.
- Babar, S. A., & Patil, P. D., 2015. Improving Performance of Text Summarization. *Procedia Computer Science*. pp.354–363.
- Nabila. S, dan Fitriansyah, A. 2020. Peringkasan Dokumen Tugas Akhir Secara Otomatis Menggunakan Metode TF-IDF Dan K-Means
- Widyasanti. K, Putra. D, dan Rusjyanthi. K 2018. Seleksi Fitur Bobot Kata Dengan Menggunakan Metode TF-IDF Untuk Ringkasan Bahasa Indonesia
- Kesuma, H. 2016. Penerapan Metode TF-IDF Dan Cosine Similarity Dalam Aplikasi Kitab Undang-Undang Hukum Dagang
- Melita. R, Amrizal. V, Suseno. H, dan Dirjam, T. 2018. Penerapan Metode Term Frequency inverse document frequency (tf-idf) dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam)
- Al-Talib, G. dan Hassan, H. 2013. A Study on Analysis of SMS Classification Using TF-IDF Weighting. *International Journal of Computer Networks and Communications Security* 1(5): 189-194.
- Zaman, B. dan Winarko, E. 2011. Analisis Fitur Kalimat untuk Peringkasan Teks Otomatis pada Bahasa Indonesia. *Indonesia Journal Computing and Cybernetics System (IJCCS)* 5(2): 60-68.
- Karyono, G. dan Utomo, F. 2012. Temu Balik Informasi pada Dokumen Teks Berbahasa Indonesia dengan Metode Vector Space Retrieval Model. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan* 2012. 23 Juni 2012: 283-289
- Fitri, Meisya. Perancangan Sistem Temu Balik Informasi dengan Metode Pembobotan Kombinasi TF-IDF untuk Pencarian Dokumen Berbahasa Indonesia. <http://jurnal.untan.ac.id/index.php/justin/article/view/1319>. 15 Agustus 2015 (13.42).
- Intan, R. dan Defeng, A. HARD: Subject-based Search Engine Menggunakan TF-IDF dan Jaccard's Coefficient. <http://jurnalindustri.petra.ac.id/index.php/ind/article/view/16502>. 23 Januari 2016 (22.55).