

# PERINGKAS TEKS OTOMATIS BERITA ONLINE MENGUNAKAN METODE *CROSS LATENT SEMANTIC ANALYSIS & COSINE SIMILARITY*

<sup>1</sup>Muhammad Afif Ubaidillah, <sup>2</sup>Ida Bagus Gede Dwidasmara, <sup>3</sup>Agus Muliantara

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana, Jalan Raya Kampus Udayana, Badung, Bali, Indonesia

<sup>1</sup>afifubaidillah.unud1@gmail.com, <sup>2</sup>dwidasmara@unud.ac.id, <sup>3</sup>muliantara@gmail.com

## **Abstrak**

Ringkasan merupakan suatu cara yang efektif untuk menyajikan suatu karangan yang panjang dalam bentuk yang singkat. Walaupun bentuknya ringkas, namun ringkasan itu tetap memertahankan pikiran pengarang dan pendekatannya yang asli. Namun dalam membuat ringkasan kita harus membaca berita atau artikel terlebih dahulu, sedangkan ringkasan dibuat dengan tujuan untuk meminimalkan waktu pembaca dan memberikan teks yang isinya langsung mengarah pada tujuan utama atau ide pokoknya. Pada penelitian ini memaparkan peringkasan teks otomatis berita online dari sebuah website menggunakan CLSA (*Cross Latent Semantic Analysis*) dan *Cosine Similarity*. Penelitian ini dilakukan untuk menguji seberapa baik hasil dan akurasi ringkasan yang dilakukan oleh CLSA dan *cosine similarity*. Penelitian ini menggunakan data sekunder dari berita dari media online yaitu web *balipost.com* dengan wilayah khusus Denpasar. Proses pengambilan data dilakukan dengan cara *crawling*. Data berita yang digunakan ialah sebanyak 161 berita, berita hasil ringkasan sistem nantinya akan dibandingkan dengan hasil ringkasan manual untuk mendapatkan akurasi. Dari hasil pengujian yang dilakukan oleh sistem didapatkan nilai rata – rata akurasi *F-Measure* sebesar 58%, rata – rata *Precision* 62% dan rata – rata *Recall* 57%. Hasil dari penelitian peringkasan teks otomatis dari berita online dengan menggunakan metode CLSA dan *cosine similarity* memberikan hasil dan akurasi ringkasan yang cukup.

**Keywords** : ringkasan, peringkasan teks otomatis, *crawling*, CLSA, *cosine similarity*

## **1. Pendahuluan**

Dunia informasi yang semakin tidak mengenal batas ruang dan waktu membuat semua orang bisa mengakses informasi kapan saja dan di mana saja. Salah satu informasi yang sangat banyak dicari adalah berita. Dalam prosesnya pengguna mengakses berita bisa melalui internet biasa dengan menggunakan bantuan mesin pencari seperti Google, Bing, dan berbagai mesin pencari lainnya.

Informasi dalam berita tentu menjadi kebutuhan pokok masyarakat saat ini karena dengan berita masyarakat dapat mengetahui segala peristiwa yang terjadi saat ini. Namun, banyaknya informasi dalam berita yang tersimpan dalam bentuk artikel, dokumen dan sebagainya bisa mencapai puluhan halaman bahkan ratusan. Untuk memahami satu artikel atau dokumen dengan banyak halaman membutuhkan waktu yang lama, hal ini tentu kurang efisien. Membaca sebuah berita dengan waktu yang singkat dan terburu-buru dapat mengakibatkan informasi yang di dapat kurang relevan atau tidak sesuai dengan yang di inginkan. Selain itu bagi orang yang membaca dokumen tebal, tidak cukup hanya dengan membaca lama tapi juga perlu membuat ringkasan kecil agar bisa di mengerti isi dari suatu dokumen yang di baca. Agar pembaca dapat menghemat waktu untuk mendapatkan tujuan utama atau ide pokok dari sebuah berita, maka perlu dibuat peringkasan teks.

*Summarization* (ringkasan) merupakan suatu cara yang efektif untuk menyajikan suatu karangan yang panjang dalam bentuk yang singkat. Walaupun bentuknya ringkas, namun ringkasan itu tetap memertahankan pikiran pengarang dan pendekatannya yang asli. Namun dalam membuat ringkasan kita harus membaca berita atau artikel terlebih dahulu, sedangkan ringkasan dibuat dengan tujuan untuk meminimalkan waktu pembaca dan memberikan teks yang isinya langsung mengarah pada tujuan utama atau ide pokoknya.

Untuk memecahkan masalah tersebut diperlukan suatu perangkat atau aplikasi yang dapat meringkas teks secara otomatis. *Automated Text Summarization* (ATS) merupakan aplikasi berbasis

komputer yang berguna untuk menghasilkan ringkasan dari sebuah artikel atau berita dari sebuah website tanpa menghilangkan pikiran utama dari sebuah artikel atau berita tersebut. Tujuan utamanya adalah untuk menunjukkan beberapa kalimat utama yang digabung menjadi satu ringkasan, dengan harapan dapat mengurangi waktu untuk memahami isi dari sebuah artikel berita. Di Indonesia penelitian yang membahas tentang *Automated Text Summarization (ATS)* cukup banyak dengan berbagai macam metode yang berbeda-beda.

Dalam penelitian mengenai peringkasan teks otomatis yang dilakukan oleh (Gamaria M. dan Gunawan G., 2018) yang berjudul "Peringkasan dokumen berita Bahasa Indonesia menggunakan metode *Cross Latent Semantic Analysis (CLSA)*". Dalam penelitiannya Gamaria M. dan Gunawan G. membandingkan metode LSA dan CLSA dalam membuat aplikasi peringkasan dokumen berita Berbahasa Indonesia dan didapatkan kesimpulan bahwa peringkasan dokumen berita menggunakan CLSA didapat akurasi sebesar 72% dengan *compression rate* 30%.

Pada penelitian lainnya (Fernando W. dan Ednawati R. 2016) yang berjudul "Implementasi *Cross Method Latent Semantic Analysis* Untuk Meringkas Dokumen Berita Berbahasa Indonesia". Dalam penelitiannya Fernando W. dan Ednawati R. mengimplementasikan metode CMLSA dengan tujuan untuk mendapatkan akurasi melalui pengujian *Precision, Recall, dan F-Measure* dan didapatkan kesimpulan bahwa hasil dari pengujian *Precision* sebesar 72,25%, *Recall* sebesar 66,7%, dan *F-Measure* sebesar 69,6% dan dipatkan akurasi ringkasan dengan menggunakan metode CMLSA sebesar 69,6%.

Berhubungan dengan kajian terkait, penulis tertarik untuk membangun dan mengembangkan sebuah sistem peringkasan teks berita online dari sebuah website dengan berita berbahasa Indonesia menggunakan metode *Cross Latent Semantic Analysis (CLSA)* dan *Cosine Similarity*. Mengingat penelitian sebelumnya masih ada beberapa kekurangan, maka penulis akan melanjutkan dan mengembangkan penelitian sebelumnya dengan menambahkan kesamaan *similarity* antara judul berita dan dokumen yang dipilih oleh CLSA menggunakan metode *Cosine Similarity* untuk mendapatkan ringkasan yang lebih baik lagi dan untuk mengetahui berita tersebut sesuai atau tidak antara judul berita dengan isinya.

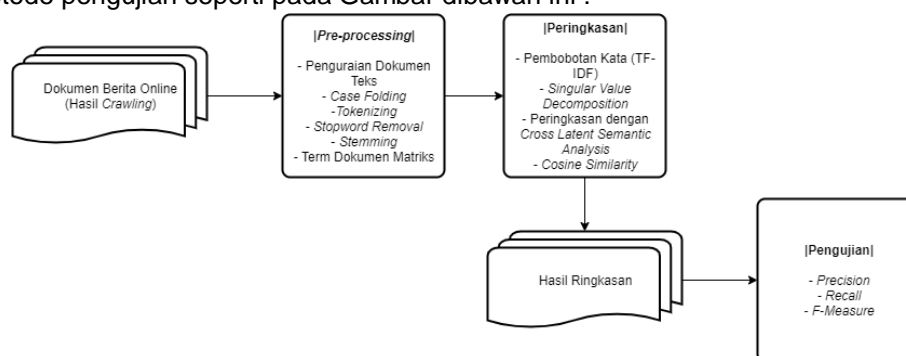
## 2. Metodologi Penelitian

### 2.2.1 Data

Data yang digunakan dalam penelitian ini ialah data sekunder. Untuk mengumpulkan data secara simultan dan untuk mengefisiensi waktu digunakan teknik *crawling* atau sering disebut *web crawler*. *Web crawler* ini nantinya akan bekerja untuk mengumpulkan semua data atau informasi secara otomatis pada web tertentu, setelah mengambil data dari halaman web dan menyimpannya ke dalam suatu media penyimpanan dalam bentuk .txt. Data yang digunakan yaitu data berita online dari situs <http://www.balipost.com/> khusus daerah Denpasar. Data berita tersebut akan digunakan sebagai masukan atau input yang akan di ringkas secara otomatis menggunakan bantuan mesin (peringkasan teks otomatis). Data ringkasan merupakan data dari hasil ringkasan yang dihasilkan oleh sistem dengan menggunakan Metode *Cross Latent Semantic Analysis* dengan *Cosine Similarity*. Hasil dari data ringkasan adalah kalimat - kalimat yang dipilih oleh sistem dan jumlahnya sudah dihitung oleh sistem. Data berita yang akan dikumpulkan sebanyak 161 data berita untuk diuji (*data testing*).

### 2.2.2 Arsitektur Sistem

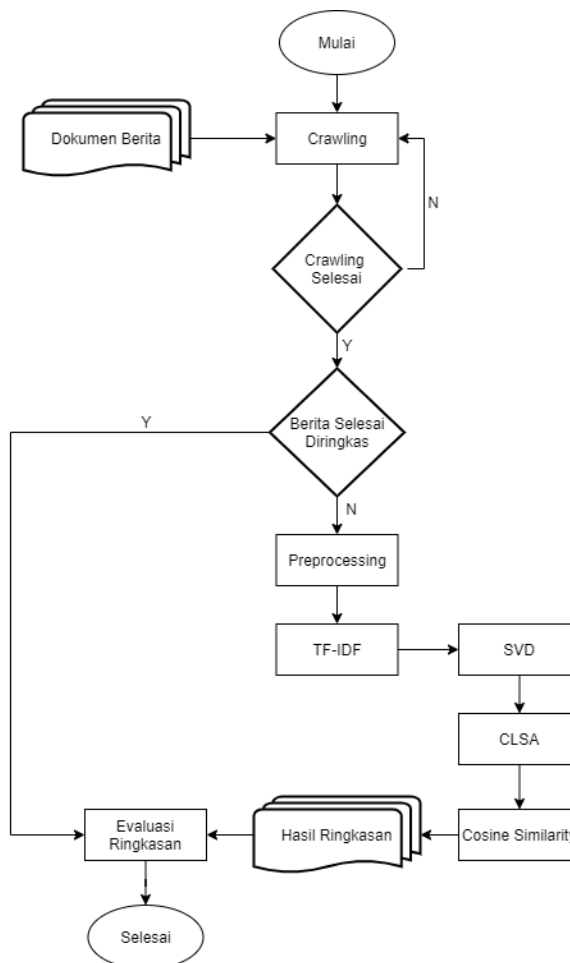
Arsitektur peringkasan dokumen Bahasa Indonesia yang digunakan untuk menggambarkan sistem kerja pada penelitian ini terdiri dari tiga tahapan yaitu, *preprocessing*, peringkasan dan metode pengujian seperti pada Gambar dibawah ini :



Gambar 2.1 Arsitektur Sistem Peringkasan Teks Otomatis.

### 2.2.3 Proses Penelitian

Pada bagian ini akan digambarkan alur secara umum dari penelitian yang akan dilakukan penulis, yaitu dimulai dari pengumpulan data berita dari media online, proses dari metode yang digunakan hingga menghasilkan sebuah ringkasan. Berikut adalah flowchart proses penelitian :



Gambar 2.2 Flowchart Proses Penelitian

## 3. Hasil dan Pembahasan

### 3.3.1 Implementasi Sistem

Sistem diimplementasikan menggunakan bahasa pemrograman Python karena struktur program yang sederhana dan mampu menghemat waktu dalam pembuatan. Implementasi dilakukan dengan Python versi 3.7.1 menggunakan *framework* flask agar sistem berbasis website. Dari sisi *hardware* dan *software* digunakan beberapa perangkat sebagai alat bantu dalam penelitian ini diantaranya sebagai berikut :

#### a. Software

1. JetBrains PyCharm Community Edition 2018.3.4 x64
2. Google Chrome
3. MongoDB 4.2.3
4. Robo 3T 1.3.1
5. Balsmiq.cloud

#### b. Hardware

1. Processor Intel® Core™ i5 M540
2. Memory DDR3 4096 Megabytes
3. Harddisk 256 Gigabytes

### 3.3.2 Implementasi Crawling

Proses *Crawling* ini melibatkan bantuan library *crawling* pada python untuk mengambil semua data teks berupa judul dan isi dari berita pada web terkait dan kemudian akan diproses hingga menghasilkan sebuah file berkestensi *.txt*. Proses *crawling* pada penelitian ini menggunakan web

<http://www.balipost.com/> dengan wilayah khusus Denpasar. Cara kerja *crawling* ini ialah mengambil setiap tag html, tag yang akan diambil ialah title (judul berita) dan content (isi berita), namun sebelum *crawling* melakukan prosesnya diperlukan pengaturan (*setting*) secara manual untuk mengambil setiap tag dari website *balipost.com*.

Pada *crawling* data yang diambil adalah data berita terbaru, pada saat pengambilan data kemungkinan besar akan terjadi berita tidak terambil semua, minimal data berita yang terambil sesuai dengan jumlah berita yang ada di *pagination* dan berita tidak akan terurut karena dalam hal ini *crawling* yang digunakan memiliki sifat *asynchronous*.

### 3.3.3 Implementasi Preprocessing

Pada tahap ini teks yang sebelumnya sudah dicrawling dalam bentuk .txt kemudian akan dilakukan tahap *preprocessing* yang dipecah menjadi 4 bagian yaitu, *Case Folding*, *Stemming*, *Stopword Removal*, dan *Tokenizing*.

Proses pertama yaitu *Case Folding* yang mana proses ini untuk menyeragamkan teks dengan cara mengubah semua huruf teks menjadi huruf kecil. Contoh *Case Folding* seperti dibawah ini :

**Teks Asli** : Uji coba Tempat Pembuangan Sampah Sementara di Kreneng.

**Hasil** : uji coba tempat pembuangan sampah sementara di kreneng

Setelah melewati proses *Case Folding*, langkah selanjutnya adalah melakukan proses *Stemming* yaitu mengubah kata menjadi kata dasar. Tujuan *stemming* sendiri adalah mereduksi kata dari hasil token sebelumnya untuk memperoleh *index* kemunculan kata yang baik pada setiap dokumen kalimat. Contoh *Stemming* seperti dibawah ini :

**Teks Asli** : uji coba tempat pembuangan sampah sementara di kreneng

**Hasil** : uji, coba, tempat, buang, sampah, sementara, kreneng

Proses selanjutnya ialah *Stopword Removal* dimana proses ini untuk menghapus kata-kata yang dianggap tidak penting, hal ini berfungsi agar dapat memaksimalkan informasi yang penting pada teks berita. Contoh *Stopword Removal* seperti dibawah ini :

**Teks Asli** : uji coba tempat pembuangan sampah sementara di kreneng

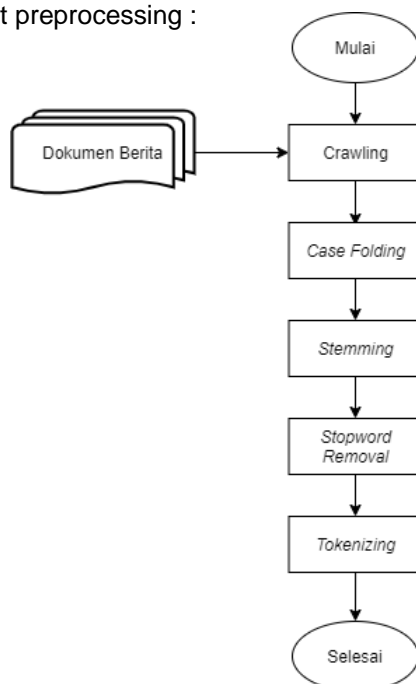
**Hasil** : uji, coba, tempat, pembuangan, sampah, sementara, kreneng

Langkah terakhir dari *preprocessing* adalah melakukan *Tokenizing* yaitu proses ini dilakukan untuk memisah *String* dari sebuah kalimat menjadi kata.

**Teks Asli** : uji coba tempat pembuangan sampah sementara di kreneng

**Hasil** : uji, coba, tempat, pembuangan, sampah, sementara, di, kreneng

Dibawah ini merupakan flowchart preprocessing :



Gambar 3.1 Flowchart Preprocessing

### 3.3.4 Implementasi TF-IDF

Setelah tahap *preprocessing* selesai maka proses selanjutnya ialah proses TF-IDF dimana ini merupakan proses awal pembentukan matriks SVD. Proses TF-IDF diawali dengan mengambil semua *term* yang telah disimpan sebelumnya pada file txt. Kemudian dari setiap file akan diambil kata yang dianggap penting kemudian ditambahkan pada list *term*. Setelah itu dari semua dokumen akan dihitung frekuensi kemunculan *term* penting pada list *term*. Nilai tersebut akan digunakan untuk menghitung IDF dan terakhir akan dilakukan perhitungan  $\logarithmic\ TF * IDF$ . Perulangan tersebut akan dilakukan sebanyak jumlah *term* penting \* jumlah kalimat pada setiap berita. Untuk rumus TF-IDF seperti dibawah ini :

$$W = tf * IDF \quad (1)$$

Dimana :

- W = bobot dokumen ke-d terhadap kata ke-t
- tf = banyaknya kata yang akan dicari pada sebuah berita
- IDF = Inversed Document Frequency

### 3.3.5 Implementasi SVD (*Singular Value Decomposition*)

Setelah matriks TF-IDF selesai proses selanjutnya ialah proses *Singular Value Decomposition* dimana ini bertujuan untuk mereduksi matriks agar lebih sedikit. Setelah nilai *term* didapat dari proses TF-IDF selanjutnya ialah mereduksi matriks dengan cara SVD, matriks yang digunakan pada SVD adalah matriks  $A^T A$ , sebelum itu matriks yang sudah terbuat, matriks  $m \times n$  akan ditranspose menjadi  $n \times m$ . kemudian akan dilanjutkan pencarian nilai  $AA^T$ ,  $A^T A$  dan pencarian nilai *eigenvector* matriks  $V$  dan *eigenvalue*  $A^T A$ . setelah itu proses akan dilanjutkan dengan mencari nilai *singular* matriks  $S$  dan kemudian diakarkan. Untuk rumus SVD seperti dibawah ini :

$$A = USV^T \quad (3)$$

Dimana :

- A = matriks dokumen
- U = matriks *orthogonal*  $m \times m$  (*Left Singular Vector*)
- S = matriks diagonal
- V = matriks *orthogonal*  $n \times n$  (*Right Singular Vektor*)

### 3.3.6 Implementasi CLSA (*Cross Latent Semantic Analysis*)

Hasil matriks pada SVD kemudian akan dilanjutkan ke proses *Cross Latent Semantic Analysis* dimana pada proses ini mencari rata – rata pada matriks  $V^T$  kemudian akan dilakukan seleksi pada matriks  $V^T$  dan terakhir dihitung nilai *length* pada matriks  $V^T$ . Pada proses *Cross Latent Semantic Analysis*, setelah matriks  $V^T$  didapat maka akan dilakukan seleksi pada setiap matriks  $V^T$ , kemudian dilanjutkan dengan menghitung nilai *length* dengan rumus persamaan (4) yang sudah diimplementasikan ke kode diatas untuk memperoleh skor dari tiap-tiap kalimat. Baris - baris pada matriks atau kalimat yang mempunyai nilai *length* yang tinggi akan di jadikan sebagai ringkasan. Setelah itu akan dilakukan juga pencarian rata-rata dari matriks  $V^T$  dan terakhir ialah menentukan hasil ringkasan berdasarkan skor tertinggi dari dokumen kalimat. Untuk CLSA seperti dibawah ini :

1. Membentuk matriks  $A_{mn}$
2. Menemukan *eigenvector* (matriks  $V$ ) dan *eigenvalue* dari matriks  $A^T A$ .
3. Mencari nilai *singular* (matriks  $S$ ), dengan cara mengurutkan nilai yang paling tertinggi dan diakarkan.
4. Melakukan *Transpose* pada *eigevector* untuk membentuk matriks  $V^T$ .
5. Menghitung nilai rata-rata dari matriks  $V^T$  seperti pada Tabel 2.1.
6. Melakukan seleksi pada setiap nilai matriks  $V^T$ , apabila nilai tersebut lebih kecil dari nilai rata-rata pada setiap dokumen kalimat, maka nilai pada matriks  $V^T$  diubah menjadi 0 dan membentuk matriks  $V$  yang baru pada Tabel 2.2.
7. Menghitung nilai *length* pada setiap matriks  $V^T$  dengan menggunakan Rumus dibawah ini untuk memperoleh skor dari tiap-tiap dokumen kalimat.

$$length = \sqrt{\sum_{j=1}^n V_{ij}^2 * S_{jj}^2} \quad (4)$$

Keterangan :

- i : baris matriks
- j : kolom matrik

Baris - baris pada matriks atau kalimat kalimat yang mempunyai nilai *length* yang tinggi akan di jadikan sebagai ringkasan.

8. Menentukan hasil ringkasan berdasarkan skor tertinggi dari dokumen kalimat.

### 3.3.7 Implementasi Cosine Similarity

Setelah proses CLSA selesai, selanjutnya ialah proses *Cosine Similarity* dimana proses ini digunakan untuk mencocokkan antara judul dengan dokumen kalimat berita. Pada proses *Cosine Similarity* ini, berita yang sudah berhasil diringkas CLSA kemudian akan dilakukan pengecekan kemiripan antara judul dengan berita yang sudah diringkas. Data berita ringkasan yang digunakan ialah data yang sudah dijadikan per file.txt. dan judul yang sudah didapat saat *crawling*, yang nantinya setiap file judul akan dibandingkan kemiripan dengan dokumen berita (ringkasan berita). 161 File berita dan 161 judul berita sudah dikelompokkan dengan folder yang berbeda. Untuk rumus *cosine similarity* seperti dibawah ini :

$$\text{Cosinus} \rightarrow \text{sim}(d_j, q) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_i W_{ij} \cdot W_q}{\sqrt{\sum_i W_{ij}^2} \cdot \sqrt{\sum_i W_q^2}}$$

Dimana :

$W_{ij}$  : bobot kata  $i$  pada dokumen  $j$

$W_q$  : bobot *query*

Berikut merupakan contoh perhitungan *cosine similarity* yang digunakan pada penelitian ini :

- **Kalimat (S1)** : tempat pembuangan sampah sementara (tps) underground di kreneng, denpasar, sudah diuji coba.
- **Judul (S2)** : tps underground kreneng diuji coba

**Tabel 3.1 Perhitungan Cosine Similarity**

Kata	TF		DF	IDF	W = TF * IDF	
	S1	S2			S1	S2
tempat	1	0	1	0.301	0.301	0
pembuangan	1	0	1	0.301	0.301	0
sampah	1	0	1	0.301	0.301	0
sementara	1	0	1	0.301	0.301	0
tps	1	1	2	0	0	0
underground	1	1	2	0	0	0
di	1	0	1	0.301	0.301	0
kreneng	1	1	2	0	0	0
denpasar	1	0	1	0.301	0.301	0
sudah	1	0	1	0.301	0.301	0
diuji	1	1	2	0	0	0
coba	1	1	2	0	0	0
					sum(S1)	sum(S2)
<b>Nilai bobot tiap kalimat</b>					2.107	0

### 3.3.8 Evaluasi Hasil Ringkasan

Hasil ringkasan pada penelitian ini menggunakan metode evaluasi instrik yaitu metode *Precision*, *Recall* dan *F-Measure* untuk memperoleh hasil akurasi antara ringkasan manual dengan sistem. Setelah hasil ringkasan CLSA dihasilkan, diperlukan juga ringkasan manual untuk menguji dan mencari hasil akurasi ringkasan tersebut. terdapat 161 data berita ringkasan sistem dan 161 data berita ringkasan manual (ringkasan manual) yang akan diuji. Berikut adalah rumus *Precision*, *Recall*, dan *F-Measure*:

$$\text{Precision} = \frac{\text{Kalimat Ringkasan Sistem} \cap \text{Ringkasan Manual (TP)}}{\sum \text{Kalimat Ringkasan Sistem (TP + FP)}} \quad (1)$$

$$\text{Recall} = \frac{\text{Kalimat Ringkasan Sistem} \cap \text{Ringkasan Manual (TP)}}{\sum \text{Kalimat Ringkasan Manual (TP + FN)}} \quad (2)$$

$$F - Measure = \frac{2 * Precision * Recall}{Recall + Precision} \quad (3)$$

Dari Rumus 1, Rumus 2, dan Rumus 3 diketahui bahwa, ada tiga komponen yang penting yaitu *True Positive* (TP) adalah jumlah dokumen kalimat yang dipilih oleh peringkas manual, *False Positive* (FP) merupakan jumlah dokumen kalimat yang dipilih oleh sistem benar tetapi menurut peringkas manual salah dan *False Negative* (FN) adalah jumlah dokumen kalimat yang benar menurut peringkas manual tetapi salah menurut sistem. Pada tahap pengujian, data berita ringkasan sistem dan data ringkasan manual sudah dikelompokkan pada masing-masing folder. Setelah itu file akan dibaca satu persatu oleh sistem dan dicocokkan antara ringkasan sistem dan ringkasan manual. Sesuai dengan rumus pada persamaan (1), (2), dan (3), rumus sudah diimplemetasikan lewat pengkodean seperti diatas untuk menguji ringkasan secara otomatis. Berikut merupakan contoh perhitungan hasil akurasi ringkasan yang digunakan pada penelitian ini :

**Tabel 3.2 Evaluasi Hasil Peringkasan**

Dok	JK	KRM	JKRM	KRS	JKRS	KR	JKR
1	12	2,3,6,8	4	1,2,3	3	2,3	2
2	12	1,2,7	3	1,2,3	3	1,2	2
3	10	1,3,7	3	1,2,7	3	1,7	2
4	7	1,2	2	1,2	3	1,2	2
5	11	1,2,4,11	4	1,2,6	3	1,2	2
6	9	1,2,8	3	1,2,4	3	1,2	2

Sedangkan untuk hasil pengujian dari *Precision* (P), *Recall* (R) dan *F-Measure* (F-M) peneliaitan ini adalah :

**Tabel 3.3 Hasil Pengujian Precision**

P	Fi	FK	Xi	FiXi
0,23 – 0,37	12	12	0,30	3,6
0,40 – 0,46	11	23	0,43	4,73
0,50 – 0,58	42	65	0,54	22,68
0,60 – 0,69	46	111	0,64	29,67
0,70 – 0,78	39	150	0,74	28,86
0,80 – 1,00	11	161	0,90	9,9
	<b>161</b>			99,44
<b>Rata - rata</b>				<b>0,62</b>

**Tabel 3.4 Hasil Pengujian Recall**

R	Fi	FK	Xi	FiXi
0,50 – 0,53	65	65	0,515	33,475
0,54 – 0,57	45	110	0,555	24,975
0,58 – 0,61	26	136	0,595	15,47
0,62 – 0,66	16	152	0,64	10,24
0,70 – 0,72	5	157	0,71	3,55
0,75 – 0,80	5	161	0,775	3,1
	161			90,81
<b>Rata - rata</b>				<b>0,57</b>

**Tabel 3.4 Hasil Pengujian *F-Measure***

<b>F-M</b>	<b>Fi</b>	<b>FK</b>	<b>Xi</b>	<b>FiXi</b>
0,33 – 0,37	3	3	0,354	1
0,40 – 0,47	14	17	0,435	6
0,50 – 0,59	70	87	0,546	38
0,60 – 0,69	60	147	0,648	39
0,70 – 0,76	12	159	0,731	9
0,80 – 0,85	2	161	0,829	2
	161			95
<b>Rata – rata</b>				<b>0,58</b>

Dari penjabaran tabel diatas didapatkan hasil akhir pengujian peringkasan teks otomatis menggunakan *cross latent semantic analysis* dan *cosine similarity* ialah, akurasi nilai rata – rata *F-Measure* 0.58 (58%), *Precision* 0.62 (62%) dan *Recall* 0.57 (57%). Sesuai dengan standarisasi ringkasan yang dijadikan pedoman, hasil ringkasan yang dihasilkan oleh sistem sudah menghasilkan ringkasan yang cukup.

#### **4. Kesimpulan**

Dari penelitian tentang peringkasan teks otomatis menggunakan metode *Cross Latent Semantic Analysis* dan *Cosine Similarity* didapatkan kesimpulan sebagai berikut:

1. Hasil akurasi ringkasan berita menggunakan metode *cross latent semantic analysis* dan *cosine similarity* didapat akurasi nilai rata – rata *F-Measure* ialah 0.58 (58%), rata – rata *Precision* 0.62 (62%) dan rata – rata *Recall* 0.57 (57%) dari jumlah data 161 berita dari web <http://www.balipost.com/> khusus daerah Denpasar.
2. Dari nilai *Precision*, *Recall* dan *F-Measure* yang telah di dapatkan, dapat diketahui bahwa implementasi metode *cross latent semantic analysis* dan *cosine similarity* pada peringkasan teks otomatis untuk meringkas berita online berbahasa Indonesia dapat menghasilkan ringkasan yang cukup karena dari rata – rata *F-Measure* hanya sekitar 58% yang didapat dari jumlah kalimat hasil ringkasan sistem yang sesuai dengan kalimat hasil ringkasan manual.

Untuk kebutuhan pengembangan penelitian ini terdapat beberapa saran yang nantinya dapat dilanjutkan oleh peneliti berikutnya, berikut adalah beberapa saran yang dapat diberikan :

1. Untuk pengembangan disarankan nantinya sistem dibuat agar bisa melakukan klasifikasi berita berdasarkan daerah atau kasus.
2. Untuk penelitian selanjutnya dapat melakukan peringkasan teks dengan algoritma yang sama dengan menggunakan teknik abstraksi.
3. Disarankan untuk mengembangkan penelitian menggunakan bahasa lain selain Bahasa Indonesia.



## References

- [1] Ade R., M. Zidny N., & Auliya B. (2019). Penerapan *Cosine Similarity* dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional*, Vol.2, No.1
- [2] Andika, Ari. 2015. Perancangan Aplikasi Pengukuran Similaritas pada Dokumen dengan Metode Semantic. *Majalah Ilmiah Informasi dan Teknologi Ilmiah (INTI)* V(3): 13-19.
- [3] Asian, J. (2007). *Effective Techniques for Indonesian Teks Retrieval*. Melbourne: RMIT University.
- [4] Badry, R. M., Eldin, A. S., & Elzanfally, D. S. (2013). Teks Summarization within the Latent Semantic Analysis Framework: Comparative Study. *International Journal of Computer Applications*, 81(11), 40-45.
- [5] [cloudflare.com](https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/), "What Is a Web Crawler How and How Web Spider Work". [diakses 27/01/20]. <https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/>
- [6] Firdaus, A. dkk. 2014. Aplikasi Pendeteksi Kemiripan pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani dan Metode Cosine Similarity. *Jurnal Teknologi Informasi* 10(1): 96-109.
- [7] Geetha, J. K., & Deepamala, N. (2015). Kannada teks summarization using Latent Semantic Analysis. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1508-1512). Pune: IEEE.
- [8] 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
- [9] Gunawan, F. E., Juandi, A. V., & Soewito, B. (2015). An automatic teks summarization using teks features and singular value decomposition for popular articles in Indonesia language. *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)* (pp. 27-32). Surabaya: IEEE. doi:10.1109/ISITIA.2015.7219948
- [10] Makbule Gulcin Ozsoy, Ilyas Cicekli, and Ferda Nur Alp, "Text Summarization of Turkish Texts Using Latent Semantic Analysis," *Proceedings of the 23<sup>rd</sup> international conference on computational linguistics*, pp. 869 - 876, 2010.
- [11] Mandar G., Gunawan G. (2018). Peringkasan dokumen berita Bahasa Indonesia menggunakan metode *Cross Latent Semantic Analysis*. *Jurnal Ilmiah Teknologi Sistem Informasi* 3 (2) 94-104, 96-100
- [12] Mustaqhfiri, M., Abidin, Z., & Kusumawati, R. (2011). Peringkasan teks otomatis berita berbahasa Indonesia menggunakan metode Maximum Marginal Relevance. *MATICS*, 4(4), 134-147.
- [13] Najibullah, A., & Mingyan, W. (2015). Otomatisasi peringkasan dokumen sebagai pendukung sistem manajemen surat. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 1(1), 1-6.
- [14] Nurina S. W. G., Indriati, & Ratih K. D. (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*. Vol. 2, No. 9, September 2018, hlm. 2821-2828
- [15] Ozsoy, M. G., Cicekli, I., & Alpaslan, F. N. (2010). Teks summarization of Turkish tekss using latent semantic analysis. *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 869-876). Beijing: ACM.
- [16] Pressman, R. S., 2012. Pendekatan Praktisi Rekayasa Perangkat Lunak. *Yogyakarta: Andi Offset*.
- [17] Setiawan P., "Pengertian Ringkasan, Manfaat, Ciri, Cara dan Langkah" <https://www.gurupendidikan.co.id/pengertian-ringkasan/>. 07/10/2019.
- [18] Steinberger, J., & Ježek, K. (2004). Using Latent Semantic Analysis in Teks Summarization and Summary Evaluation. *Proc. ISIM '04*, (pp. 93–100).
- [19] Torres-Moreno, J.-M. (2014). *Automatic teks summarization* (Vol. 5). Hoboken: Wiley-ISTE.
- [20] Winata F., Rainarli E. (2016). Implementasi Cross Method Latent Semantic Analysis Untuk Meringkas Dokumen Berita Berbahasa Indonesia. *Techno.COM*, Vol. 15, No. 4. 266-273
- [21] Zeniarja, J., Salam, A., Luthfiarta, A., Handoko, L. B., & Jamhari, M. (2013). Integrasi peringkasan dokumen otomatis dengan penggabungan metode fitur dan metode Latent Semantic Analysis (LSA) sebagai Feature Reduction. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2013 (SEMANTIK 2013)* (pp. 191-197). Semarang: Universitas Dian Nuswantoro