

(Jurnal Edukasi dan Penelitian Informatika)

ISSN(e): 2548-9364 / ISSN(p): 2460-0741

Vol. 7 No. 2 Agustus 2021

# Peringkasan Teks Otomatis pada Modul Pembelajaran Berbahasa Indonesia Menggunakan Metode *Cross Latent Semantic Analysis* (*CLSA*)

Yunita Maulidia Sari<sup>#1</sup>, Nenden Siti Fatonah<sup>#2</sup>

\*Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana Jl. Meruya Selatan No. 1, Kembangan, Jakarta 11650

> ¹yunyunitamsr@gmail.com ²nendenfatonah@gmail.com

Abstrak — Perkembangan teknologi yang pesat membuat kita lebih mudah dalam menemukan informasi-informasi yang dibutuhkan. Permasalahan muncul ketika informasi tersebut sangat banyak. Semakin banyak informasi dalam sebuah modul maka akan semakin panjang isi teks dalam modul tersebut. Hal tersebut akan memakan waktu yang cukup lama untuk memahami inti informasi dari modul tersebut. Salah satu solusi untuk mendapatkan inti informasi dari keseluruhan modul dengan cepat dan menghemat waktu adalah dengan membaca ringkasannya. Cara cepat untuk mendapatkan ringkasan sebuah dokumen adalah dengan cara peringkasan teks otomatis. Peringkasan teks otomatis (Automatic Text Summarization) merupakan teks yang dihasilkan dari satu atau lebih dokumen, yang mana hasil teks tersebut memberikan informasi penting dari sumber dokumen asli, serta secara otomatis hasil teks tersebut tidak lebih panjang dari setengah sumber dokumen aslinya. Penelitian ini bertujuan untuk menghasilkan peringkasan teks otomatis pada modul pembelajaran berbahasa Indonesia dan mengetahui hasil akurasi peringkasan teks otomatis yang menerapkan metode Cross Latent Semantic Analysis (CLSA). Jumlah data yang digunakan pada penelitian ini sebanyak 10 file modul pembelajaran yang berasal dari modul para dosen Universitas Mercu Buana, dengan format .docx sebanyak 5 file dan format .pdf sebanyak 5 file. Penelitian ini menerapkan metode Term Frequency-Inverse Document Frequency (TF-IDF) untuk pembobotan kata dan metode Cross Latent Semantic Analysis (CLSA) untuk peringkasan Pengujian akurasi pada peringkasan modul pembelajaran dilakukan dengan cara membandingkan hasil ringkasan manual oleh manusia dan hasil ringkasan sistem. Yang mana pengujian ini menghasilkan rata-rata nilai fmeasure, precision, dan recall tertinggi pada compression rate 20% dengan nilai berturut-turut 0.3853, 0.432, dan 0.3715.

Kata kunci.... Cross Latent Semantic Analysis, Modul Pembelajaran, Peringkasan Teks Otomatis, TF-IDF.

## I. PENDAHULUAN

Seperti yang kita ketahui perkembangan kemajuan teknologi yang pesat membuat manusia menjadi lebih

mudah dalam menemukan informasi-informasi yang dibutuhkan. Baik itu informasi yang didapatkan dari membaca di internet maupun membaca buku, modul, artikel, majalah, koran, dan lain-lain. Salah satu sumber informasi dapat diperoleh dalam sebuah modul pembelajaran yang biasanya digunakan sebagai media belajar oleh mahasiswa atau mahasiswi. Modul ialah bahan ajar yang dibuat secara sistematis dengan menggunakan bahasa yang mudah dipahami oleh para siswa, yang sesuai usia dan tingkat pengetahuan siswa sehingga para siswa bisa belajar dengan bantuan dari pendidik [1]. Modul ialah bahan ajar yang ditujukan kepada mahasiswa atau mahasiswi dalam mencapai tujuan belajarnya, yang dibuat secara sistematis serta sesuai kebutuhan sehingga mahasiswa atau mahasiswi dapat belajar secara mandiri.

Semakin banyak dan mudahnya informasi yang diperoleh tidak sejalan dengan minat baca masyarakat. Salah satu kegiatan yang wajib dilakukan guna memperkaya pengetahuan dan informasi ialah dengan cara membaca. Permasalahan muncul ketika informasi yang terdapat dalam suatu modul tersebut sangat banyak. Semakin banyak informasi dalam sebuah modul maka akan semakin panjang isi teks dalam modul tersebut. Hal tersebut tentu saja akan memakan waktu yang cukup lama untuk memahami inti dari informasi dalam modul tersebut. Salah satu solusi untuk mendapatkan inti informasi dari keseluruhan modul dengan cepat dan menghemat waktu adalah dengan membaca bagian ringkasan dari dokumen tersebut.

Cara cepat dalam memperoleh ringkasan dari dokumen adalah dengan cara peringkasan teks otomatis. Peringkasan teks otomatis (*Automatic Text Summarization*) merupakan teks yang dihasilkan dari satu atau lebih dokumen, yang mana hasil teks tersebut memberikan informasi penting dari sumber dokumen asli, serta secara otomatis hasil teks tersebut tidak lebih panjang dari setengah sumber dokumen aslinya.[2] Peringkasan teks otomatis adalah proses membuat teks yang Panjang menjadi berkurang dengan

bantuan program dari komputer sehingga dihasilkan sebuah ringkasan yang berisikan kalimat-kalimat penting yang mana hasil peringkasan tersebut tidak melebihi dari setengah dokumen teks asli nya.[3],[4] Sehingga dengan menggunakan peringkasan teks otomatis mendapatkan ide pokok maupun informasi penting dari sebuah modul pembelajaran dengan jelas, tepat, dan ringkas, tanpa merubah makna dari isi modul tersebut. Peringkasan teks otomatis juga dapat menghemat waktu dalam mendapatkan inti informasi dari sebuah modul tersebut. Peringkasan teks sendiri pada umumnya terdiri dari dua teknik yaitu, peringkasan teks ekstraksi dan abstraksi. Teknik ekstraksi adalah teknik penyusunan kalimat dengan memilih 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[5].

Penelitian terdahulu banyak yang membahas peringkasan teks otomatis (Automatic Text Summarization) dengan menggunakan berbagai metode, diantaranya penelitian yang dilakukan dilakukan oleh (Gamaria dan Gunawan, 2018) mereka melakukan penelitian pada artikel berita dengan menerapkan metode Cross Latent Semantic Analysis (CLSA). Pada penelitian ini membandingkan metode LSA dan CLSA dengan menggunakan data yang berasal dari 240 artikel www.kompas.com. Setelah melalui proses preprocessing, pembobotan kata (TF-IDF), perhitungan metode CLSA dan LSA, hasil pengujian menggunakan metode CLSA lebih baik dibandingkan dengan metode LSA. Hasil pengujian peringkasan dokumen berita menggunakan CLSA mendapatkan akurasi sebesar 72% dengan compression rate 30%.[6] Penelitian lainnya dilakukan oleh (Dika dkk, 2017), dimana pada penelitian ini dilakukan peringkasan teks otomatis pada 30 artikel kesehatan dengan menggunakan metode Lexical Chain dan Word Sense Disambiguation. Penelitian ini menghasilkan nilai akurasi dengan menggunakan metode Lexical Chain dengan Word Sense Disambiguation memiliki rata-rata precision sebesar 66% dan recall sebesar 34% sedangkan untuk metode Lexical Chain tanpa Word Sense Disambiguation memiliki rata-rata precision sebesar 57% dan recall sebesar 29%. Kesimpulannya dengan menggunakan metode Lexical Chain dan Word Sense Disambiguation mendapatkan akurasi yang lebih baik daripada hanya menggunakan metode Lexical Chain tanpa Word Sense Disambiguation.[7]

Berdasarkan pada penelitian sebelumnya tentang peringkasan teks otomatis (*Automatic Text Summarization*) [6],[7] yang menunjukkan bahwa dengan menggunakan metode *Cross Latent Semantic Analysis* (CLSA) dapat memberikan hasil akurasi yang lebih baik dibanding metode lainnya. Maka pada penelitian ini akan dilakukan peringkasan teks otomatis berbahasa Indonesia dengan menerapkan teknik ekstraksi dan metode *Cross Latent Semantic Analysis* (CLSA) terhadap modul pembelajaran

yang memiliki jumlah kalimat yang lebih banyak dibanding penelitian sebelumnya. Penelitian ini bertujuan untuk mengetahui hasil akurasi ringkasan teks otomatis dengan menerapkan metode *Cross Latent Semantic Analysis* dan menghasilkan peringkasan teks otomatis pada modul pembelajaran berbahasa Indonesia dengan cepat dan tepat, sehingga mahasiswa yang membaca modul tersebut dapat mengetahui inti informasi dari mata kuliah yang akan atau sedang diambil. Diharapkan ketika mahasiswa yang hendak mengambil mata kuliah, mereka sudah mendapatkan gambaran tentang mata kuliah yang akan mereka ambil dengan membaca ringkasan modul tersebut.

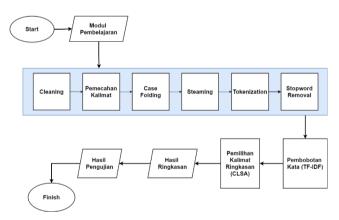
### II. METODE PENELITIAN

# A. Data Penelitian

Dataset yang dipakai penulis pada penelitian ini diperoleh secara manual. Pada penelitian ini menggunakan sampel data sebanyak 10 file modul pembelajaran yang berasal dari modul para dosen Universitas Mercu Buana yang tidak dikunci atau *lock*, dengan format .docx sebanyak 5 file dan format .pdf sebanyak 5 file.

## B. Tahapan Penelitian

Pada penelitian ini terdapat beberapa tahapan penelitian diantaranya; *start*, modul pembelajaran (.docx & .pdf), *preprocessing*, pembobotan kata (TF-IDF), peringkasan teks (CLSA), hasil peringkasan modul pembelajaran, hasil pengujian atau evaluasi, dan *finish*. Seperti pada contoh Gambar 1.



Gambar. 1 Tahapan penelitian

- 1) Modul Pelajaran .docx / .pdf: Tahapan pertama melakukan peringkasan teks otomatis adalah dengan menginputkan data atau dokumen yang berupa modul pembelajaran dengan format .docx dan .pdf.
- 2) Preprocessing: tahap awal pada pemrosesan peringkasan berita[8]. Tahapan dalam preprocessing dilakukan dengan tahapan cleaning, pemecahan kalimat, case folding, stemming, tokenization, dan stopword removal.
- a) Cleaning: Pada tahap ini kumpulan kalimat yang telah dipecah, selanjutnya dilakukan proses cleaning yang bertujuan untuk membersihkan noise yang ada dalam

teks. Beberapa contoh *noise* yang akan dibersihkan yaitu; angka, tanda buka kurung, dan lain-lain.[9]

- b) Pemecahan Kalimat: Pada tahapan pemecahan kalimat yang berarti memecah dokumen yang telah diinputkan menjadi per kalimat. Pemisah (delimiter) yang digunakan yaitu (., ?). Hasil dari pemecahan kalimat selanjutnya dapat digunakan untuk tahapan selanjutnya.[10]
- c) Case Folding: Setelah melalui tahap cleaning, selanjutnya yaitu tahapan case folding. Tahap ini berfungsi untuk mengganti huruf kapital (uppercase) menjadi huruf kecil (lowercase). Sehingga teks menjadi sama rata dan tidak ditemukan lagi huruf kapital (uppercase) dalam data.[7]
- d) Stemming: Tahap ini bertujuan untuk mengubah kata berimbuhan menjadi kata dasar. Contohnya seperti pada kata "membeli" diganti menjadi "beli", kata "berlibur" diganti menjadi "libur". Dalam penelitian ini algoritma stemming yang digunakan yaitu algoritma dari Nazief dan Adriani.[11] Algoritma Nazief-Adriani ialah algoritma stemming yang mengubah kata yang terdapat pada dokumen menjadi kata akarnya (root word) dengan memakai aturan tertentu. Misalnya pada kata bersama, kebersamaan, menyamai, akan diubah menjadi kata "sama".
- e) Tokenization: Tahap ini berfungsi untuk memotong atau memisahkan atau memecah kalimat menjadi per kata berdasarkan spasi sebagai pemotong/ pemisah/ pemecah kata tersebut.[9]
- f) Stopword Removal: Tahap terakhir pada preprocessing adalah stopword removal. Dalam tahap ini kata-kata yang dianggap tidak penting akan dilakukan penghapusan, yang bertujuan untuk mengurangi jumlah kata yang akan diproses. Contoh kata yang akan dihilangkan seperti; "dan", "atau", "dia", "ia", "adalah", "dari", dan lain-lain.[12]
- 3) Pembobotan Kata Term frequency-Inverse document frequency (TF-IDF): merupakan ekstraksi kalimat dengan cara memberikan nilai atau bobot pada kata[13]. TF-IDF ialah salah satu metode perhitungan bobot kata dengan cara mengekstraksi ciri suatu teks.[14] Proses perhitungan TF-IDF dilakukan agar mendapatkan bobot kata yang terdapat pada suatu dokumen. Semakin sering kata tersebut muncul pada sebuah dokumen maka nilainya semakin besar. Bobot kata akan memperhitungkan kebalikan frekuensi dokumen yang terdapat sebuah kata (Inverse Document Frequency).[12]

Proses TF-IDF dapat dituliskan seperti pada persamaan 1 dan persamaan 2:

$$W = TF * IDF$$
 (1)

Keterangan:

W: bobot dokumen ke-d terhadap kata ke-t (nilai bobot dari setiap kata pada sebuah dokumen).

TF: jumlah kata yang dihitung pada kalimat (D)

IDF : Inverse Document Frequency

Untuk nilai IDF dapat dicari dengan Rumus 2:

$$IDF = Log\left(\frac{N}{DF}\right)$$
(2)

Keterangan:

IDF: banyaknya dokumen yang mengandung kata (Document Frequency) dengan jumlah kalimat (D)

N : total kalimat dalam dokumen

DF: banyaknya kalimat yang mengandung kata.

- 4) Pemilihan Kalimat Ringkasan (CLSA): Setelah mendapatkan hasil pembobotan dari setiap kata dengan menggunakan algoritma TF-IDF. Maka tahap selanjutnya adalah pemilihan kalimat Ringkasan dengan menggunakan algoritma Cross Latent Semantic Analysis (CLSA). CLSA ialah pengembangan dari metode Latent Semantic Analysis (LSA). Steinberger dan Jezek merupakan pencetus pertama kali metode ini. Berikut ini adalah proses step by step CLSA dalam melakukan peringkasan pada dokumen:[6]
- a) Matriks Amn yang pertama kali dibuat, dengan menerapkan Singular Value Decomposition (SVD) yang dirumuskan seperti pada Rumus 3:

$$A_{mn} = U_{mn} . S_{nn} . V^{T}_{nn}$$

$$(3)$$

Keterangan:

 $A_{mn}$ : matriks awal

 $U_{mm}$ : matriks ortogonal U

 $S_{mn}$ : matriks diagonal S

 $V^{T}_{nn}$ : transpose matriks orthogonal V

- b) Eigenvector (matriks V) dan eigenvalue ditemukan dari matriks A<sup>T</sup>A.
- Nilai singular (matriks S) dicari, dengan cara mengurutkan nilai yang tertinggi lalu selanjutnya diakarkan.
- d) Lakukan Transpose pada eigenvector yang bertujuan untuk membuat matriks V<sup>T</sup>.
- e) Matriks V<sup>T</sup> akan dihitung rata-rata nilainya.
- f) Melakukan seleksi pada setiap nilai matriks V<sup>T</sup>.
- g) Melakukan perhitungan pada nilai length di setiap matriks  $V^T$ . Perhitungan tersebut dapat menggunakan Rumus 4 sebagai berikut:

$$Length = \sqrt{\sum_{j=1}^{n} V_{ij}^{2} * S_{jj}^{2}}$$

$$\tag{4}$$

Hasil ringkasan ditentukan menurut skor tertinggi dari setiap kalimat yang telah dilakukan perhitungan.

5) Evaluasi Hasil Ringkasan: Berdasarkan penelitian sebelumnya, penulis melakukan penelitian ini dengan menggunakan metode evaluasi atau pengujian intrinsik menggunakan perhitungan f-measure.[15] Nilai f-measure ialah hasil dari precision dan recall. Recall adalah

sebuah metode yang mampu mengambil nilai tertinggi yang sebagian besar kalimatnya sesuai (benar), *precision* ialah jumlah kalimat yang berhasil diambil oleh sistem.[6]

# III. HASIL DAN PEMBAHASAN

Pengujian terhadap modul pembelajaran sebagai contoh data untuk mendapatkan nilai akurasi hasil dari peringkasan teks otomatis yang telah dibuat. Pengujian diterapkan pada 10 modul pembelajaran dengan format .docx dan .pdf. Tingkat akurasi diperoleh dengan cara membandingkan antara hasil ringkasan manual yang

dibuat oleh manusia dengan hasil ringkasan oleh sistem. Pengujian tingkat akurasi pada penelitian ini diukur dengan cara perhitungan *f-measure*, *precision*, dan *recall*.

# A. Hasil Proses TF-IDF

Keluaran yang dihasilkan dari tahap TF-IDF akan disimpan ke dalam *array*. Contoh hasil perhitungan nilai TF-IDF pada setiap *term* atau kata yang terdapat dalam setiap dokumen atau data yang dipergunakan dalam penelitian ini dapat dijumpai pada Tabel I. Hasil TF-IDF didapatkan dari rumus persamaan (1) dan (2).

TABEL I HASIL TF-IDF

Kata		Frekuensi Kata di Setiap Kalimat									DF	D/DF	IDF=	TF	W=TF*	
IIIII	D0	D1	D2	D3	D4		D50	D51	D52	D53	D54		<i>D,D</i> 1	log(D/DF)	11	IDF
a	0	0	0	0	0		0	0	0	0	1	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
aktifitas	1	0	0	0	0		0	0	0	0	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
asmstrong	0	0	0	0	0		0	0	0	1	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
ariyant	0	0	0	0	0		0	0	0	0	1	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
bahas	1	0	0	0	0		0	0	0	0	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
behavior	0	0	0	0	0		0	0	0	0	1	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
center	0	1	0	0	0		0	0	0	0	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
edition	0	0	0	0	0		0	0	0	1	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
fakultas	0	0	0	0	0		0	0	0	0	1	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74
field	0	1	0	0	0		0	0	0	0	0	1	55/1=55	Log(55)= 1.74	1	1 * 1.74 = 1.74

Pada Tabel I terlihat nilai bobot untuk setiap *term* atau kata nya. Setelah didapatkan bobot untuk setiap *term* atau kata dalam dokumen. Maka selanjutnya hasil TF-IDF seperti pada Tabel I akan digunakan untuk proses berikutnya yaitu proses pemilihan kalimat ringkasan dengan menerapkan metode *Cross Latent Semantic Analysis*.

## B. Hasil Proses CLSA

Setelah dilakukan pembobotan kata memakai matriks TF-IDF selesai, proses berikutnya adalah Singular Value Decomposition (SVD) dimana proses ini bertujuan untuk mereduksi matriks agar tidak terlalu panjang. Matriks yang digunakan pada SVD adalah matriks A<sup>T</sup>A, sebelum itu matriks yang sudah dibuat yaitu matriks m x n akan ditranspose menjadi n x m.[16] Kemudian akan dilanjutkan pencarian nilai AA<sup>T</sup>, A<sup>T</sup>A dan pencarian nilai eigenvector matriks V dan nilai eigenvalue A<sup>T</sup>A. Kemudian proses akan dilanjutkan dengan mencari nilai singular matriks S yang selanjutnya akan diakarkan. Hasil matriks pada SVD kemudian akan dilanjutkan ke proses Cross Latent Semantic Analysis dimana pada proses ini mencari rata rata pada matriks V<sup>T</sup> kemudian akan dilakukan seleksi pada matriks V<sup>T</sup> dan terakhir dihitung nilai *length* pada matriks V<sup>T</sup> untuk dijadikan sebagai hasil ringkasan.

TABEL III HASIL SCORE LENGTH

Kalimat	Length
0	2.378916469406922
1	2.4906230811582497
2	1.3506302642655612
3	1.985021059726594
4	1.985021059726594
50	1.2890134865930443
51	1.0957155064668238
52	0.5485170339757925
53	0.23890463877447993
54	0.22114392759375243

Berbeda dengan metode *Latent Semantic Analysis* (LSA), yang berdasarkan dari kecocokan antara kalimat dengan judul dokumen, pada metode *Cross Latent Semantic Analysis* (CLSA) hal yang dijadikan sebagai parameter dari hasil ringkasan adalah *length* dari kalimat yang terdapat pada dokumen uji atau dataset. Hasil dari kalimat yang memiliki nilai *length* tertinggi akan dipilih sebagai hasil ringkasan. Untuk mencari nilai *length* suatu kalimat dapat menggunakan rumus persamaan (4). Setelah didapatkan nilai *length* setiap kalimatnya. Proses selanjutnya akan dilakukan pencarian rata-rata dari matriks V<sup>T</sup> dan proses terakhir adalah mendapatkan hasil ringkasan menurut skor tertinggi kalimat yang diuji.

TABEL IIIII HASIL LENGTH TERTINGGI

Length	Kalimat
2.4906230811582497	1
2.4533655041777607	6
2.378916469406922	0
2.210745915374316	7
2.029591520928515	10
0.5273028879978168	13
0.24114957607657286	31
0.23890463877447993	53
0.22114392759375243	54
0.11030317113124717	37

Dari Tabel III, dapat diketahui *length* kalimat yang paling panjang yaitu terdapat pada kalimat ke-1. Sedangkan *length* kalimat yang memiliki hasil *length* paling pendek adalah kalimat ke-37. Dari hasil perhitungan length pada Tabel III, dapat diambil kalimat hasil ringkasannya.

# C. Hasil Pengujian Peringkasan Teks Otomatis

Pengujian dilakukan dengan cara membandingkan teks yang dihasilkan dari ringkasan teks otomatis yang menerapkan metode Cross Latent Semantic Analysis dengan teks vang dihasilkan secara manual. Untuk mendapatkan hasil pengujian ringkasan teks otomatis penelitian ini memakai metode precision, recall, dan fmeasure yang dihitung menggunakan compression rate. Nilai persentase untuk compression rate yaitu 10%, 20%, 30%, 40%, dan 50% pada setiap dokumen yang digunakan. Dipilihnya nilai persentase tersebut dikarenakan dengan nilai compression rate 10% sampai dengan 50% telah dapat dikatakan cukup untuk menghasilkan sebuah ringkasan yang cukup baik.[5] Pengujian ini bertujuan untuk mendapatkan nilai akurasi atau performa dari hasil peringkasan teks otomatis yang menerapkan metode Cross Latent Semantic Analysis (CLSA) yang didapatkan dari hasil nilai precision, recall, dan f-measure. Selanjutnya akan dilakukan analisis dan diambil kesimpulan mengenai tingkat akurasi dari ringkasan menggunakan metode Cross Latent Semantic Analysis untuk meringkas modul pembelajaran. Hasil rata-rata pengujian peringkasan teks otomatis dengan memakai metode f-measure, precision, dan recall pada compression rate 10%, 20%, 30%, 40%, dan 50% terdapat pada Gambar. 3.



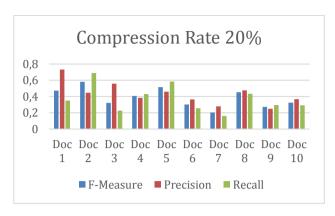
Gambar. 2 Rata-rata pengujian CR 10%, 20%, 30%, 40%, dan 50%

Dengan melakukan pengjian pada semua dokumen yang berjumlah 10 dokumen, maka didapat hasil rata-rata untuk metode *f-measure*, *precision*, dan *recall* dengan *compression rate* 10%, 20%, 30%, 40%, dan 50%. Nilai rata-rata hasil *f-measure*, *precision*, dan *recall* tertinggi terdapat pada *compression rate* 20% dengan nilai berturutturut 0.3853, 0.432, dan 0.3715 atau 38.53%, 43.2%, dan 37.15%. Sedangkan untuk nilai rata-rata hasil *f-measure*, *precision*, dan *recall* terendah terdapat pada *compression rate* 30% dengan nilai berturut-turut 0.3524, 0.5318, dan 0.2766 atau 35.24%, 53.18%, dan 27.66%.

TABEL IVV HASIL PENGUJIAN DENGAN CR 20%

Compression Rate 20%							
Dokumen	F-Measure	Precision	Recall				
Doc 1	0.474	0.731	0.35				
Doc 2	0.58	0.448	0.688				
Doc 3	0.322	0.558	0.226				
Doc 4	0.407	0.385	0.431				
Doc 5	0.514	0.46	0.583				
Doc 6	0.301	0.365	0.256				
Doc 7	0.205	0.28	0.161				
Doc 8	0.453	0.475	0.432				
Doc 9	0.272	0.25	0.297				
Doc 10	0.325	0.368	0.291				
Rata-rata	0.3853	0.432	0.3715				
Rata- rata %	38.53	43.2	37.15				

Dari hasil pengujian pada *compression rate* 20% menggunakan metode *f-measure*, *precision*, dan *recall* didapatkan rata-rata secara berurutan yaitu 0.3853, 0.432, dan 0.3715. Sedangkan untuk nilai *f-measure*, *precision*, dan *recall* tertinggi terdapat pada Doc 2 yang secara berurutan yaitu 0.58, 0.448, dan 0.688.



Gambar. 3 Pengujian dengan CR 20%

Dari hasil pengujian menggunakan metode *f-measure*, precision, dan recall pada algoritma Cross Latent Semantic Analysis menghasilkan nilai akurasi yang cukup rendah dibandingkan dengan penelitian sebelumnya dengan menggunakan algoritma yang sama Cross Latent Semantic Analysis. Selain dikarenakan perbedaan banyaknya jumlah kata atau kalimat dalam dokumen yang diproses, hasil pemilihan ringkasan antara sistem dan pakar juga dapat memengaruhi hasil akhir dari akurasi peringkasan teks otomatis. Sebagai contoh penelitian (Prabowo et al., 2017) yang menerapkan metode Normalized Google Distance Dimana dan *K-Means*.[17] penelitian tersebut menggunakan 10 data dokumen KIKBI. Yang menghasilkan rata-rata nilai akurasi precision, recall, dan relative utility secara berurutan pada pakar pertama ialah 0,20, 0,47, 0,48 dan pada pakar kedua ialah 0,27, 0,43, 0,45. Jika dibandingkan dengan hasil penelitian yang dilakukan oleh penulis dengan menerapkan metode Cross Latent Semantic Analysis. Maka hasil ringkasan yang memakai metode Cross Latent Semantic Analysis menghasilkan nilai rata-rata akurasi yang lebih baik dibandingkan dengan memakai metode Normalized Google Distance dan K-Means.

# IV. KESIMPULAN

Hasil pengujian peringkasan teks otomatis menggunakan algoritma Cross Latent Semantic Analysis, menghasilkan rata-rata nilai akurasi f-measure, precision, dan recall pada compression rate 20% secara berturut-turut yaitu 0.3853, 0.432, dan 0.3715. Sedangkan pengujian untuk nilai f-measure, precision, dan recall tertinggi terdapat pada Doc 2 yang secara berurutan yaitu 0.58, 0.448, dan 0.688. Pengujian ini menghasilkan nilai akurasi yang cukup rendah dibandingkan dengan penelitian sebelumnya dengan menggunakan algoritma yang sama Cross Latent Semantic Analysis. Selain dikarenakan perbedaan banyaknya jumlah kata atau kalimat dalam dokumen yang diproses, hasil pemilihan ringkasan antara sistem dan secara manual oleh manusia juga dapat memengaruhi hasil akhir dari akurasi peringkasan teks otomatis.

Dari penelitian yang dilakukan, terdapat saran untuk penelitian yang akan datang. Adapun sarannya yaitu dengan lebih memperhatikan delimiter pada saat melakukan pemecahan kalimat. Karena ada beberapa kata yang menjadi ambigu ketika delimiter yang digunakan tanda titik (.). Contohnya seperti dalam sebuah kalimat terdapat kata yang disertai gelar atau singkatan misalnya Dr. Sari, Sari, M.Kom, M.M. yang seharusnya untuk kata tersebut tidak dipecah berdasarkan hanya tanda titik (.). 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 Cross Latent Semantic Analysis. Selanjutnya akan dilakukan pengecekan kemiripan antara judul dengan kalimat yang sudah diringkas. Dengan menambahkan algoritma Cosine Similarity diharapkan dapat menghasilkan akurasi yang lebih baik lagi.

#### REFERENSI

- [1] Y. N. Kholisho, "Pengembangan Modul Pembelajaran Multimedia untuk Meningkatkan Minat dan Pemahaman Konsep Mahasiswa Prodi Pendidikan Informatika," *EDUMATIC J. Pendidik. Inform.*, vol. 1, no. 1, p. 17, 2017, doi: 10.29408/edumatic.v1i1.733.
- [2] I. P. G. Hendra Saputra, "Peringkasan Teks Otomatis Untuk Dokumen Bahasa Bali Berbasis Metode Ektraktif," *J. Ilmu Komput.*, vol. 10, no. 1, pp. 33–38, 2018.
- [3] M. A. Ubaidillah, I. B. G. Dwidasmara, and A. Muliantara, "Peringkas Teks Otomatis Berita Online Menggunakan Metode Cross Latent SemanticAnalysis & Cosine Similarity," *JELIKU* (*Jurnal Elektron. Ilmu Komput. Udayana*), vol. 9, no. 1, p. 105, 2020, doi: 10.24843/jlk.2020.v09.i01.p11.
- [4] S. Twinandilla, S. Adhy, B. Surarso, and R. Kusumaningrum, "Multi-Document Summarization Using K-Means and Latent Dirichlet Allocation (LDA) - Significance Sentences," *Procedia Comput. Sci.*, vol. 135, pp. 663–670, 2018, doi: 10.1016/j.procs.2018.08.220.
- [5] J. Saputra, M. Fachrurrozi, and Yunita, "Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode Latent Semantic Analysis (LSA) dan Teknik Steinberger & Jezek," *Comput. Sci. ICT*, vol. 3, no. 1, pp. 215–219, 2017.
- [6] G. Mandar and G. Gunawan, "Peringkasan dokumen berita Bahasa Indonesia menggunakan metode Cross Latent Semantic Analysis," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 2, p. 94, 2017, doi: 10.26594/register.v3i2.1161.
- [7] D. M. Fazar and N. I. Widiastuti, "Lexical Chain dan Word Sense Disambiguation Untuk Peringkasan Artikel Berbahasa Indonesia," *Techno. Com*, vol. 16, no. 2, pp. 195–207, 2017, doi: 10.33633/tc.v16i2.1413.
- [8] I. No and J. Pinto, "Available Online at www.ijarcs.info TEXT SUMMARIZER FOR URL AND . DOCX FILES," vol. 11, no. 4, pp. 18–22, 2020.
- [9] M. D. Diana, B. D. Satya, and F. Rahutomo, "Penerapan Algoritma Score-Based pada Peringkasan Teks Cerpen Otomatis," *Semin. Inform. Apl. POLINEMA* 2020, 2020.
- [10] D. Andriani and M. T. Furqon, "Peringkasan Teks Otomatis Pada Artikel Berita Hiburan Berbahasa Indonesia Menggunakan Metode BM25" vol. 3, no. 3, pp. 2603–2610, 2019.
- BM25," vol. 3, no. 3, pp. 2603–2610, 2019.

  [11] H. Noprisson and V. Ayumi, "Implementasi Algoritma Nazief-Adriani Pada Fitur Tebak Kata di Web Edukasi Bahasa Indonesia," 

  JSAI (Journal Sci. Appl. Informatics), vol. 1, no. 1, pp. 18–23, 2018, doi: DOI:10.36085/jsai.v1i1.10.
- [12] H. Jayadianti, R. Damayanti, and ..., "Latent Semantic Analysis (Lsa) Dan Automatic Text Summarization (Ats) Dalam Optimasi Pencarian Artikel Covid 19," Semin. Nas. ..., vol. 2020, no. Semnasif, pp. 52–59, 2020, [Online]. Available: http://jurnal.upnyk.ac.id/index.php/semnasif/article/view/4085.

- [13] K. E. Dewi and E. Widiastuti, Nelly Indriani Rainarli, "Evaluasi Sentence Extraction pada Peringkasan Dokumen Otomatis," *Pros. Semin. Nas. Inform. dan Apl.*, no. September, p. A-8 sd. A-12, 2017.
- [14] R. D. Himawan and Eliyani, "Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa," J. Edukasi dan Penelit. Inform., vol. 7, no. 1, pp. 58–63, 2021, [Online]. Available: https://jurnal.untan.ac.id/.
- [15] C. Shah and A. Jivani, "A Hybrid Approach of Text Summarization Using Latent Semantic Analysis and Deep Learning," 2018 Int. Conf. Adv. Comput. Commun. Informatics, pp. 2039–2044, 2018.
- [16] M. M. Tf-idf, "Pendekatan dengan Naturan Language Analysis . Beberapa Teknik Pendekatan statistika," pp. 134–142, 2019.
- [17] D. A. Prabowo, M. A. Fauzi, and Y. A. Sari, "Peringkasan Teks Ekstraktif Kepustakaan Ilmu Komputer Bahasa Indonesia Menggunakan Metode Normalized Google Distance dan K-means," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 1, no. 12, pp. 1697–1707, 2017.