

# Penerapan Algoritma Text Mining dan Lexrank dalam Meringkas Teks Secara Otomatis

Ahmad Fauzi

Fakultas Ilmu Komputer & Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: fauziac788@gmail.com

**Abstrak**—Pertumbuhan media serta berita online menjadikan penulis membuat penelitian di bidang peringkasan teks secara otomatis. Berita-berita yang menawarkan konsep yang cepat dan ringkas, akan tetapi pada kenyataannya berita digital tidak terorganisir dan menjadi begitu panjang menyebabkan pembaca sulit mencari inti point dari berita tersebut. Peringkasan dokumen menjadi cara yang efektif guna mendapatkan informasi dari dokumen tanpa membaca keseluruhan dokumen. Namun peringkasan dokumen untuk bahasa Indonesia masih tergolong sedikit dibandingkan dengan bahasa yang lain. Penelitian ini mengembangkan peringkasan dokumen secara otomatis menggunakan metode berbasis graf yaitu dengan Algoritma Lexrank yang dapat dibuktikan dengan penelitian yang sudah di uji menggunakan data berita bahasa Indonesia yang didapat dari liputan6.com. Jumlah kalimat yang diekstraksi adalah 25%-50% dari total kalimat dalam dokumen. Hasil dari ringkasan Lexrank memperlihatkan urutan dari bobot yang tertinggi yaitu =  $D2 = 1.433$ ,  $D10 = 1.289$ ,  $D3 = 1.253$ , .....  $D8 = 0.673$ . Nilai yang terbesar dari paragraf akan disusun kata sesuai urutan sehingga mendapatkan hasil ringkasan berita tersebut.

**Kata Kunci:** Peringkasan; Teks Otomatis; Lexrank; Berita

**Abstract**—The growth of media and online news has allowed writers to automate research in the field of text summarization. News that offers a quick and concise concept, but in reality digital news is not organized and it takes so long to find the essence of the news. Document summarization is an effective way to get information from a document without reading the entire document. However, document summaries for Indonesian are still relatively small compared to other languages. This study develops document summarization automatically using a graph-based method, namely the Lexrank Algorithm which can be proven by research that has been tested using Indonesian news data obtained from liputan6.com. The number of sentences extracted is 25%-50% of the total sentences in the document. The results of the Lexrank summary in order of the highest weight order are =  $D2 = 1.433$ ,  $D10 = 1.289$ ,  $D3 = 1.253$ , .....  $D8 = 0.673$ . The largest value from the summary will be arranged according to the order of words so as to get the summary of the news.

**Kata Kunci:** Summarization; Automatic Text; Lexrank; News

## 1. PENDAHULUAN

Algoritma *string matching* (pencocokan string) adalah algoritma untuk melakukan pencarian semua kemunculan string pendek (pattern) yang muncul dalam teks. *Pattern* yaitu string dengan panjang  $m$  karakter ( $m < n$ ). Teks (*text*) yaitu *long string* yang panjangnya  $n$  karakter. Ringkasan adalah sebuah penyajian peristiwa atau kejadian yang panjang di sajikan secara singkat. Teks merupakan suatu tatanan dari kata-kata yang digunakan untuk memberikan informasi, menjelaskan makna dan sebagainya. Peringkasan teks juga proses menghasilkan dokumen dengan besar tidak lebih dari 50% ukuran dokumen sumber. Peringkasan teks otomatis atau biasa disebut *automated text summarization* (ATS) merupakan salah satu upaya yang dilakukan untuk memberikan kemudahan kepada para pembaca agar lebih mudah mencari informasi secara cepat. [1]. Penyelesaian masalah penelitian dengan aplikasi peringkasan teks otomatis ini menggunakan pencocokan string (string matching) yaitu algoritma untuk melakukan pencarian yang akan menampilkan string pendek (pattern) dan string Panjang (teks). Dalam pencarian informasi berdasarkan *keyword*/inputan sangat berguna untuk pencarian terarah dan berguna untuk mengetahui informasi yang berhubungan dengan *keyword* yang dicari. Dengan bagaimana cara menerapkan menggunakan *Algoritma Text Mining* dan *Lexrank*, diharapkan agar proses peringkasan dapat memberi hasil ringkasan yang lebih baik.

Berdasarkan penelitian terdahulu yang telah dilakukan oleh Musfiroh Nurjannah, Hamdani, dan Inda Fitri Astuti pada jurnal Informatika Mulawarman, vol. 8 No. 3 September 2013 dalam jurnalnya yang berjudul “Penerapan Algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF) Untuk *Text Mining*” menyimpulkan bahwa “dari sistem ini melakukan penerapan algoritma *term frequency-inverse document frequency* untuk *text mining* sehingga membantu pengguna mendapatkan dokumen terkait yang sesuai dengan query yang telah diinputkan. Metode menggunakan *Algoritma term frequency-inverse document frequency* merupakan salah satu metode yang tepat untuk digunakan dalam pencarian kata di tiap dokumen. Sistem ini dapat melakukan pencarian sesuai dengan *query* secara bersamaan untuk banyak file. Metode pembobotan dokumen TF-IDF tidak selalu memberikan hasil performansi yang baik pada koleksi pengujian. Dalam algoritma TF-IDF, frekuensi kemunculan sebuah kata (*term*) dalam dokumen tidak mempengaruhi hasil perhitungan bobot dokumen oleh sistem (sifat *monotonicity* TF-IDF)[2].

Dalam sistem ini dapat melakukan pencarian sesuai dengan query secara bersamaan untuk banyak file. Hasil ringkasan berhasil mendapatkan kalimat yang relevan dengan isi dari dokumen dan berhasil mengurangi redundansi dari kalimat-kalimat yang sudah dijadikan ringkasan. Dengan demikian, pada penelitian ini di tujukan untuk mempermudah dan mempercepat proses pencarian yang dilakukan oleh pembaca dalam mencari sebuah teks dengan penerapan teks mining dan *lexrank*.

## 2. METODOLOGI PENELITIAN

### 2.1 Peringkasan Teks Otomatis (*Automatic Text Summarization*)

Ringkasan merupakan pengambilan bagian penting yang menggambarkan keseluruhan isi pokok dari dokumen asal. Hovy mendefinisikan ringkasan adalah teks yang dihasilkan dari sebuah teks atau banyaknya teks yang mengandung informasi dari teks orisinal dan tidak lebih dari setengah teks aslinya[4]. Peringkasan teks otomatis (*automatic text summarization*) merupakan suatu pembuatan bentuk teks yang lebih singkat dengan memanfaatkan aplikasi yang dijalankan atau yang dioperasikan menggunakan komputer.

### 2.2 Text Mining

*Text Mining* merupakan penerapan konsep data mining untuk mencari pola dalam teks, Proses penganalisaan teks digunakan untuk menemukan informasi yang bermanfaat untuk tujuan tertentu. Even Yahir mendefinisikan *text mining* merupakan penambangan data yang berupa teks dimana sumber data itu biasanya didapatkan dari dokumen dan tujuannya untuk mencari kata yang dapat mewakili isi dari dokumen sehingga dapat melakukan analisa suatu dokumen[5]. Penambahan teks juga memiliki tujuan dan menggunakan proses yang sama dengan penambahan data berupa salah satu teknik yang dapat digunakan untuk melakukan klasifikasi, dimana *text mining* merupakan variasi dari data mining yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar. Dan juga *text mining* dapat didefinisikan sebagai penerapan konsep dari teknik data mining untuk mencari pola dalam teks yang bertujuan untuk mencari informasi yang bermanfaat dengan tujuan tertentu.

### 2.3 Text Preprocessing

*Text Preprocessing* merupakan tahapan dari proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Sebuah teks yang harus dipisahkan, hal ini dapat dilakukan dalam beberapa tingkatan yang berbeda. Suatu dokumen dapat dipecahkan menjadi bab, sub bab, paragraf, kalimat dan pada akhirnya menjadi potongan kata. Selain itu pada tahapan ini keberadaan digit angka, huruf kapital atau karakter-karakter yang lainnya dihilangkan dan dirubah[1]. Berikut ini proses penjelasan tahapan *text preprocessing* :

1. *Case Folding*
2. *Tokenizing*
3. *Stemming*

### 2.4 LexRank

LexRank merupakan algoritma untuk peringkasan dokumen tunggal atau multi dokumen, yaitu satu ringkasan berasal dari satu dokumen atau lebih dari satu dokumen. LexRank menggunakan pendekatan berbasis centroid dalam proses peringkasan. LexRank digunakan sebagai bagian dari sistem yang lebih besar summarization yang menggabungkan skor LexRank (probabilitas stasioner) dengan fitur-fitur lain seperti posisi kalimat dan panjang menggunakan kombinasi linear dengan baik bobot yang ditentukan pengguna atau secara otomatis disetel. LexRank menerapkan langkah pengolahan pasca heuristik yang membangun ringkasan dengan menambahkan kalimat dalam urutan peringkat [5].

### 2.5 Metode Berbasis Graf

Metode berbasis graf tergolong memodelkan teks ke dalam bentuk graf dengan menjadikan unit-unit teks sebagai vertex dan menambahkan edges pada graf berdasarkan hubungan antar unit teks yang dijadikan vertex, kemudian menentukan tingkat pentingnya setiap vertex berdasarkan struktur graf keseluruhan. Graf tekstual adalah graf yang dibangun dari teks dengan tujuan untuk melakukan perankingan halaman-halaman web, penerapan perankingan graf tekstual adalah untuk melakukan perankingan terhadap unit-unit teks. Dari hasil perankingan dapat dipilih unit-unit teks paling penting yang akan menjadi penyusun ringkasan ekstraktif. Pada perankingan graf tekstual, teks direpresentasikan menjadi sebuah graf. Vertex atau node pada graf tekstual adalah unit teks yang akan diranking, yaitu dapat berupa kata-kata, kalimat-kalimat, atau paragraf-paragraf dalam teks. Edge atau link dalam graf menunjukkan keterhubungan antar vertex atau node. Keterhubungan tersebut dapat berupa similarity antar kalimat ataupun hubungan leksikal atau gramatikal antar kata atau frasa

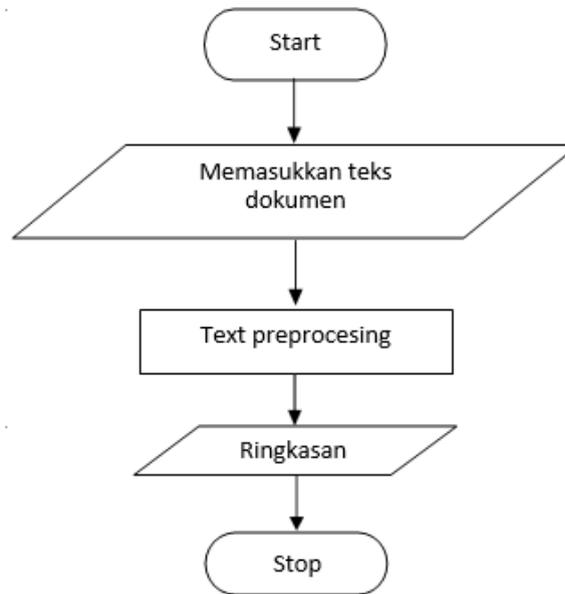
## 3. HASIL DAN PEMBAHASAN

Ringkasan yang dibutuhkan untuk mendapatkan isi berita secara ringkas. Konsep sederhana dalam mengambil bagian penting dari keseluruhan isi berita kemudian menyajikannya kembali dalam bentuk yang lebih singkat. Peringkasan teks otomatis yang akan dibuat merupakan sistem yang secara otomatis dapat membaca teks single dokumen dan akan menghasilkan sebuah ringkasan. Metode yang digunakan dalam peringkasan menggunakan pendekatan ekstraksi yaitu algoritma Lexrank. Proses secara umum dalam pembuatan ringkasan otomatis pada skripsi ini, yaitu text preprocessing, meliputi pemecahan kalimat, case folding, filtering, tokenizing kata dan stemming. Ketika teks akan diringkas, proses yang dilakukan adalah:

1. User memasukkan teks dokumen yang akan diringkas dan memasukkan kalimat query.

2. Kemudian sistem melakukan pemrosesan teks (text preprocessing), yaitu pemecahan kalimat, case folding, filtering, tokenizing kata dan stemming..

Adapun gambaran proses peringkasan otomatis secara umum dapat dilihat pada Gambar 1. berikut



**Gambar 1.** Proses Peringkasan Otomatis

Untuk mengidentifikasi maka memerlukan data dokumen, penulis mengambil data tentang berita yang ingin diringkas. Data di ambil secara real- time dari website liputan6.com. Pengguna (reader) dapat membaca berita kriminal secara ringkas dengan menyertakan poin dan unsur penting dalam sebuah berita. Pengguna juga dapat menentukan berapa jumlah baris ringkasan yang akan ditampilkan dalam aplikasi. Untuk memudahkan dalam melakukan analisa, maka penulis telah menentukan judul berita “Menteri Teten: Kalau UMKM Terganggu, Kita Khawatir Pengangguran Bertambah Banyak” adapun isi berita dari website liputan6.com sebagai berikut :

Menteri Koperasi dan UKM, Teten Masduki mengatakan sektor UMKM harus dijaga agar tidak makin terpuruk di tengah pandemi Covid-19. Alasannya sektor ini menyerap 97 persen tenaga kerja sehingga jika UMKM terganggu, angka pengangguran akan bertambah. "Karena kalau UMKM ini terganggu maka kita khawatir angka pengangguran (bertambah) banyak," kata Teten dalam Penandatanganan Nota Kesepahaman Kemenkop UKM dan BKPM, Jakarta, Kamis (17/9). Masa pandemi ini jua menjadi ujian bagi para pelaku UMKM. Sebab, para pekerja formal yang kehilangan pekerjaan kemungkinan besar banting stir menjadi pelaku usaha mikro. Akibatnya, persaingan antar pelaku usaha menjadi tinggi. "Bahkan orang yang di PHK dari sektor formal juga akibatnya harus diserap juga oleh UMKM, sehingga tingkat persaingan UMKM tinggi," tutur Teten. Maka, mantan Kepala Staf Kepresidenan ini tengah berupaya untuk menjaga pelaku UMKM agar tidak tumbang. UMKM harus tetap bertahan meskipun dalam kondisi ekonomi sulit. "Untuk itu kita harus berusaha agar UMKM kita ini bisa bertahan dalam kondisi yang cukup sulit," kata dia. Hal ini dilakukan agar tidak ada lagi penambahan jumlah angka pengangguran. "Karena faktor yang utama itu agar tidak menambah jumlah (pengangguran)," kata Teten mengakhiri.

Adapun proses penerapan algoritma Text Mining dan Lexrank dalam meringkas teks secara otomatis sebagai berikut ini:

### 1. Pemecahan Kalimat

Pada aktivitas yang dilakukan dalam proses ini adalah memecah string dokumen utuh menjadi kalimat-kalimat dengan menghilangkan delimiter atau tanda baca yang menyusunnya seperti titik “.”, tanda tanya “?”, dan tanda seru “!”.

**Tabel 1.** Pemecahan Kalimat

No	Kalimat
1	Menteri Koperasi dan UKM Teten Masduki mengatakan sektor UMKM harus dijaga agar tidak makin terpuruk di tengah pandemi
2	Covid-19 sektor ini menyerap angka 97 persen tenaga kerja sehingga jika UMKM terganggu angka pengangguran akan bertambah
3	Karena kalau UMKM ini terganggu maka kita khawatir angka pengangguran bertambah banyak kata Teten dalam Penandatanganan Nota Kesepahaman Kemenkop UKM dan BKPM
4	Masa pandemi ini juga menjadi ujian bagi para pelaku UMKM
5	Sebab para pekerja formal yang kehilangan pekerjaan kemungkinan besar banting stir menjadi pelaku usaha mikro
6	Akibatnya persaingan antar pelaku usaha menjadi tinggi
7	Bahkan orang yang di PHK dari sektor formal juga akibatnya harus diserap juga oleh UMKM sehingga tingkat persaingan UMKM tinggi tutur Teten

8	Maka mantan Kepala Staf Kepresidenan ini tengah berupaya untuk menjaga pelaku UMKM agar tidak tumbang
9	UMKM harus tetap bertahan meskipun dalam kondisi ekonomi sulit
10	Untuk itu kita harus berusaha agar UMKM kita ini bisa bertahan dalam kondisi yang cukup sulit kata dia
11	Hal ini dilakukan agar tidak ada lagi penambahan jumlah angka pengangguran
12	Karena faktor yang utama itu agar tidak menambah jumlah pengangguran kata Teten mengakhiri

## 2. Case Folding

Dokumen yang telah dipotong menjadi beberapa kalimat kemudian di tahap case folding ini yang selanjutnya dilakukan yaitu mengubah teks menjadi huruf kecil, menghilangkan angka dan tanda baca maupun simbo-simbol karena dianggap sebagai delimiter, sistem hanya menerima karakter huruf saja.

**Tabel 2.** Hasil Case Folding kata

No	Kalimat
1	menteri koperasi dan ukm teten masduki mengatakan sektor umkm harus dijaga agar tidak makin terpuruk di tengah pandemi
2	covid sektor ini menyerap angka persen tenaga kerja sehingga jika umkmterganggu angka pengangguran akan bertambah
3	karena kalau umkm ini terganggu maka kita khawatir angka pengangguran bertambah banyak kata teten dalam penandatanganan nota kesepahaman kemenkop ukm dan bkpm
4	masa pandemi ini juga menjadi ujian bagi para pelaku umkm
5	Sebab para pekerja formal yang kehilangan pekerjaan kemungkinan besar bantingstir menjadi pelaku usaha mikro
6	akibatnya persaingan antar pelaku usaha menjadi tinggi
7	bahkan orang yang di phk dari sektor formal juga akibatnya harus diserap juga oleh umkm sehingga tingkat persaingan umkm tinggi tutur teten
8	maka mantan kepala staf kepresidenan ini tengah berupaya untuk menjaga pelaku umkm agar tidak tumbang
9	umkm harus tetap bertahan meskipun dalam kondisi ekonomi sulit
10	Untuk itu kita harus berusaha agar umkm kita ini bisa bertahan dalam kondisiyang cukup sulit kata dia
11	hal ini dilakukan agar tidak ada lagi penambahan jumlah angka pengangguran
12	karena faktor yang utama itu agar tidak menambah jumlah penganggurankatateten mengakhiri

## 3. Filtering

Pada tahapan ini akan dilakukan adalah mengambil kata-kata penting dari hasil case folding kalimat dan membuang kata-kata yang dianggap kurang penting. Algoritma yang dipakai adalah stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting).

**Tabel 3.** Hasil Filtering kata

No	Kalimat
1	sektor umkm harus dijaga agar tidak makin terpuruk di tengah pandemicovid
2	umkm terganggu angka pengangguran akan bertambah
3	kita khawatir angka pengangguran bertambah banyak
4	masa pandemi ini juga menjadi ujian bagi para pelaku umkm
5	para pekerja formal yang kehilangan pekerjaan kemungkinan besarbanting stir menjadi pelaku usaha mikro
6	akibatnya persaingan antar pelaku usaha menjadi tinggi
7	orang yang di phk harus diserap juga oleh umkm sehingga tingkat persaingan umkm tinggi
8	mantan kepala staf kepresidenan berupaya untuk menjaga pelaku umkm agar tidak tumbang
9	umkm harus tetap bertahan meskipun dalam kondisi ekonomi sulit
10	Untuk itu kita harus berusaha agar umkm bisa bertahan dalam kondisiyang cukup sulit
11	hal ini dilakukan agar tidak ada lagi penambahan jumlah angka pengangguran
12	faktor yang utama itu agar tidak menambah jumlah pengangguran katateten mengakhiri

## 4. Activity Diagram Tokenizing

Tahap tokenizing adalah tahap pemotongan string menjadi potongan kata kemudian disusun menjadi baris. Pemotongan string kalimat-kalimat hasil filtering berdasarkan delimiter yang menyusunnya yaitu karakter spasi (" ").

**Tabel 4.** Hasil Tokenizing kata

No	Kata	No	Kata	No	Kata	No	Kata
1	angka	5	usaha	9	pekerja	13	pelaku
2	pengangguran	6	ekonomi	10	kondisi	14	khawatir
3	banyak	7	akan	11	umkm	15	ujian
4	sulit	8	tidak	12	phk	16	tinggi

### 5. Activity Diagram Stemming

Aktifitas pada tahap stemming ini menentukan kata dasar atau akar kata dari token kata. Dalam bahasa Indonesia, afiks (imbuhan) terdiri dari sufiks (akhiran), infiks (sisipan), dan prefiks (awalan). Karena proses penambahan infiks dalam bahasa Indonesia jarang terjadi maka proses stemming yang dibangun hanya menangani kata yang mengalami penambahan prefiks dan sufiks. Aturan- aturan proses stemming telah dijelaskan pada sub bab 2 dalam proses stemming pada bahasa Indonesia. Yang menjadi perbedaan dengan algoritma- algoritma stemming lain adalah proses stemming ini menggunakan bantuan kamus-kamus kecil dengan tujuan membedakan suatu kata yang mengandung imbuhan baik prefiks maupun sufiks dengan suatu kata dasar yang salah satu suku katanya merupakan bagian dari imbuhan, terutama dengan kata dasar yang mempunyai suku kata lebih dari dua.

**Tabel 5.** Hasil Stemming kata

No	Kata	No	Kata	No	Kata
1	angka	5	usaha	9	pelaku
2	penggangguran	6	harus	10	kondisi
3	agar	7	sulit	11	Jumlah
4	bertambah	8	tinggi	12	pandemi

### 6. Representasi Graf

Pada proses representasi graf, perhitungan similarity dilakukan dengan menggunakan persamaan Idf-modified-cosine menghitung bobot dari setiap term yang terdapat di setiap pasang kalimat dengan pembobotan tf-idf dengan menggunakan persamaan. Hasil dari pembobotan tf-idf untuk term yang terdapat di setiap kalimat kemudian dinormalisasi. Untuk setiap term akan dihitung nilai IDF dengan menggunakan persamaan, lalu dilakukan perhitungan bobot dengan menggunakan persamaan dan perhitungan panjang vektor menggunakan persamaan, setelah itu akan dilakukan perhitungan cosine similarity perkalimat dengan menggunakan persamaan.

**Tabel 6.** Perhitungan Nilai IDF

Term	Term Frequency												IDF		
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	DF	d/df	Log(n/df)
angka	0	2	1	0	0	0	0	0	0	0	1	0	3	4	0.602
penggangguran	0	0	1	0	0	0	0	0	0	0	1	1	3	4	0.602
agar	1	0	0	0	0	0	0	1	0	1	0	1	4	3	0.477
bertambah	0	1	1	0	0	0	0	0	0	0	0	0	2	6	0.778
usaha	0	0	0	0	1	1	0	0	0	0	0	0	2	6	0.778
harus	1	0	0	0	0	0	1	0	1	1	0	0	4	3	0.477
sulit	0	0	0	0	0	0	0	0	1	1	0	0	2	6	0.778
tinggi	0	0	0	0	0	1	1	0	0	0	0	0	2	6	0.778
pelaku	0	0	0	1	1	1	0	1	0	0	0	0	4	3	0.477
kondisi	0	0	0	0	0	0	0	0	1	1	0	0	2	6	0.778
jumlah	0	0	0	0	0	0	0	0	0	0	1	1	2	6	0.778
pandemi	1	0	0	1	0	0	0	0	0	0	0	0	2	6	0.778

Perhitungan IDF dilakukan dengan menggunakan rumus

$$IDF = \log \left( \frac{N}{dfi} \right)$$

N : Jumlah kalimat dalam berita = 12

dfi : Jumlah kalimat yang berisi kata “Tembak” = 3

$$IDF = \log \left( \frac{12}{3} \right)$$

$$IDF = \log(4)$$

$$IDF = 0.602$$

**Tabel 7.** Nilai Bobot

Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
T1 angka	0	1.204	0.602	0	0	0	0	0	0	0	0.602	0
T2 penggangguran	0	0	0.602	0	0	0	0	0	0	0	0.602	0.602
T3 agar	0.477	0	0	0	0	0	0	0.477	0	0.477	0	0.477
T4 bertambah	0	0.778	0.778	0	0	0	0	0	0	0	0	0
T5 usaha	0	0	0	0	0.778	0.778	0	0	0	0	0	0
T6 harus	0.477	0	0	0	0	0	0.477	0	0.477	0.477	0	0
T7 sulit	0	0	0	0	0	0	0	0	0.778	0.778	0	0
T8 tinggi	0	0	0	0	0	0.778	0.778	0	0	0	0	0
T9 pelaku	0	0	0	0.477	0.477	0.477	0	0.477	0	0	0	0

T10 kondisi	0	0	0	0	0	0	0	0	0.778	0.778	0	0
T11 jumlah	0	0	0	0	0	0	0	0	0	0	0.778	0.778
T12 pandemi	0.778	0	0	0.778	0	0	0	0	0	0	0	0

Perhitungan IDF dilakukan dengan menggunakan rumus  $IDF = \log(N)$  Setelah itu, dilakukan perhitungan nilai IDF sesuai dengan frekuensi kata yang muncul dalam satu kalimat. Nilai IDF akan dikalikan dengan jumlah berapa kali kata yang muncul dalam suatu kalimat (sentence). Term yang akan dihitung adalah T1 “Angka” pada kalimat 1 (D2):  
 Perhitungan bobot dilakukan dengan menggunakan rumus  $Ws.t = TFd.t * IDFt$   
 Diketahui: TF : Banyaknya term i pada sebuah kalimat = 2  
 (pada S1, term “Tembak” muncul sebanyak 3 kali) IDF : Inversed Document Frequency = 0.602 Maka,  
 $Ws.t = TFd.t * IDFt$   
 $= 2 * 0.602$   
 $Ws.t = 1.204$

**Tabel 8.** Nilai Panjang Vektor Setiap Kalimat

IDF <sup>2</sup>											
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
0	1.449	0.362	0	0	0	0	0	0	0	0.362	0
0	0	0.362	0	0	0	0	0	0	0	0.362	0.362
0.227	0	0	0	0	0	0	0.227	0	0.227	0	0.227
0	0.605	0.605	0	0	0	0	0	0	0	0	0
0	0	0	0	0.605	0.605	0	0	0	0	0	0
0.227	0	0	0	0	0	0.227	0	0.227	0.227	0	0
0	0	0	0	0	0	0	0	0.605	0.605	0	0
0	0	0	0	0	0.605	0.605	0	0	0	0	0
0	0	0	0.227	0.227	0.227	0	0.227	0	0	0	0
0	0	0	0	0	0	0	0	0.605	0.605	0	0
0	0	0	0	0	0	0	0	0	0	0.605	0.605
0.605	0	0	0.605	0	0	0	0	0	0	0	0
<b>1.059</b>	<b>2.054</b>	<b>1.329</b>	<b>0.832</b>	<b>0.832</b>	<b>1.437</b>	<b>0.832</b>	<b>0.454</b>	<b>1.437</b>	<b>1.664</b>	<b>1.329</b>	<b>1.194</b>
<b>1.029</b>	<b>1.433</b>	<b>1.253</b>	<b>0.912</b>	<b>0.912</b>	<b>1.198</b>	<b>0.912</b>	<b>0.673</b>	<b>1.198</b>	<b>1.289</b>	<b>1.253</b>	<b>1.092</b>

Setelah mendapatkan nilai bobot langkah selanjutnya adalah menghitung nilai panjang vector setiap kalimat. Perhitungan dilakukan dengan memangkatkan nilai IDF. Setelah itu, nilai term dalam satu kalimat dijumlahkan lalu diakarkan. Panjang vektor kalimat yang akan dihitung adalah D1 Perhitungan Panjang Vektor Setiap Kalimat menggunakan rumus:

$$|Di| = \sqrt{|Di| = \sqrt{(IDFt1)^2 + (IDF2)^2 + \dots + (IDFtn)^2}}$$

$$|D1| = \sqrt{0 + 0 + (0.477)^2 + 0 + 0 + (0.477)^2 + 0 + 0 + 0 + 0 + (0.778)^2}$$

$$|D1| = \sqrt{0.227 + 0.227 + 0.605}$$

$$|D1| = \sqrt{1.059}$$

$$|D1| = 1.029$$

**Tabel 9.** Perhitungan Cosine Similarity

Term	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
T1	0	0.675	0.619	0	0	0	0	0	0	0	0.619	0
T2	0	0	0.619	0	0	0	0	0	0	0	0.619	0.711
T3	0.671	0	0	0	0	0	0	1.028	0	0.536	0	0.633
T4	0	0.615	0.704	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0.967	0.736	0	0	0	0	0	0
T6	0.671	0	0	0	0	0	0.758	0	0.577	0.536	0	0
T7	0	0	0	0	0	0	0	0	0.736	0.684	0	0
T8	0	0	0	0	0	0.736	0.967	0	0	0	0	0
T9	0	0	0	0.758	0.758	0.577	0	1.028	0	0	0	0
T10	0	0	0	0	0	0	0	0	0.736	0.684	0	0
T11	0	0	0	0	0	0	0	0	0	0	0.704	0.807
T12	0.857	0	0	0.967	0	0	0	0	0	0	0	0

Setelah menghitung nilai bobot dan nilai panjang vektor, maka selanjutnya dihitung nilai cosine similarity. Perhitungan dilakukan dengan membagi nilai IDF dengan akar dari nilai IDF dikalikan nilai panjang vektor. Perhitungan cosine similarity. Perhitungan cosine similarity menggunakan T3 pada D1.

$$Idf - modified - cosine(x,y) = \frac{A.B}{|A| \cdot |B|} = \frac{IDF}{\sqrt{IDF \times Nilai\ Bobot}}$$

$$Idf - modified - cosine(x,y) = \frac{0.477}{\sqrt{0.477} \times 1.029}$$

$$\text{Idf - modified - cosine (x,y)} = \frac{0.477}{\frac{0.690 \times 1.029}{0.477}}$$

$$\text{Idf - modified - cosine (x,y)} = \frac{0.477}{0.710}$$

$$\text{Idf - modified - cosine (x,y)} = 0.671$$

Proses pemeringkatan tersebut merupakan proses untuk menentukan bobot akhir setiap *vertex* yang ada di dalam suatu *graf*. Setelah bobot akhir tiap *vertex* didapatkan, bobot akhir tiap *vertex* diurutkan mulai dari yang terbesar sampai terkecil. Berikut adalah hasil akhir proses pemeringkatan terhadap salah satu berita bias dilihat pada tabel 3.10.

**Tabel 10.** Nilai Akhir Bobot

Kalimat	Bobot Akhir
D2	1.433
D10	1.289
D3	1.253
D11	1.253
D6	1.198
D9	1.198
D12	1.092
D1	1.029
D4	0.912
D7	0.912
D5	0.912
D8	0.673

## 7. Ekstraksi Kalimat

Proses ekstraksi kalimat dilakukan dengan mengambil 25% kalimat dari total kalimat yang ada pada dokumen. Kalimat yang diekstraksi adalah kalimat yang memiliki peringkat paling tinggi. Setelah didapat bobot akhir lalu masuk dengan proses ekstraksi kalimat. Proses ekstraksi kalimat merupakan proses memilih kalimat dari sebuah dokumen yang akan dijadikan ringkasan. Penelitian ini menggunakan tingkat kompresi sebesar 25% sampai 50% dan kalimat yang dipilih adalah kalimat dengan peringkat paling tinggi. Misalkan suatu dokumen memiliki 20 kalimat, maka kalimat yang akan diekstraksi sebanyak 5 hingga 10 kalimat yang peringkatnya paling tinggi. Jika hasil perhitungan 25% dari total kalimat adalah bilangan desimal maka hasil perhitungan tersebut dibulatkan ke atas. Misalkan 25% kalimat yang diekstraksi adalah 5.5 maka banyak kalimat yang dijadikan ringkasan adalah 6 kalimat. Kalimat dengan nilai bobot tertinggi akan keluar menjadi hasil ringkasan berita tersebut sesuai dengan urutan paragrafnya.

Berikut adalah Pengujian Hasil Ringkasan teks secara otomatis menggunakan metode Maximum Marginal Relevance

**Tabel 10.** Hasil Ringkasan

Kalimat	Hasil Ringkasan
<i>Covid-19 sektor ini menyerap angka 97 persen tenaga kerja sehingga jika UMKM terganggu angka pengangguran akan bertambah</i>	Covid-19 menyerap 97% persen tenaga kerja sehingga UMKM terganggu.
Karena kalau UMKM ini terganggu maka kita khawatir angka pengangguran bertambah banyak kata Teten dalam Penandatanganan Nota Kesepahaman Kemenkop UKM dan BKPM, Jakarta, Kamis(17/9).	Kalau UMKM ini terganggu maka kita khawatir penganggur tambah banyak," kata Teten dalam Penandatanganan Nota Kesepahaman Kemenkop UKM dan BKPM, Jakarta, Kamis (17/9).
Akibatnya persaingan antar pelaku usaha menjadi tinggi UMKM harus tetap bertahan meskipun dalam kondisi ekonomi sulit	Akibatnya persaingan antar pelaku usaha menjadi tinggi UMKM harus tetap bertahan meskipun ekonomi sulit
Karena faktor yang utama itu agar tidak menambah jumlah pengangguran kata Teten mengakhiri Menteri Koperasi dan UKM Teten Masduki mengatakan sektor UMKM harus dijaga agar tidak makin terpuruk di tengah pandemic	Karena faktor yang utama itu agar tidak menambah jumlah pengangguran kata Teten mengakhiri Menteri Koperasi dan UKM Teten Masduki mengatakan sektor UMKM dijaga agar tidak makin terpuruk di tengah pandemi
Masa pandemi ini juga menjadi ujian bagi para pelaku UMKM	Masa pandemi ujian bagi UMKM
Sebab para pekerja formal yang kehilangan pekerjaan kemungkinan besar banting stir menjadi pelaku usaha mikro Maka mantan Kepala Staf Kepresidenan ini tengah berupaya untuk menjaga pelaku UMKM agar tidak tumbang	Para pekerja kehilangan pekerjaan mungkin besar banting stir. Maka mantan Kepala Staf Kepresidenan ini tengah berupaya untuk menjaga UMKM agar tidak tumbang.

#### 4. KESIMPULAN

Dalam melakukan peringkasan teks otomatis menggunakan metode algoritma text mining dan lexicrank dapat menghasilkan ringkasan secara otomatis tanpa menghilangkan makna dalam teks. Penerapan algoritma *text mining* yang diterapkan untuk meringkas teks secara otomatis masih terbilang lemah dikarenakan masih memiliki nilai yang sama. Adanya nilai bobot dari suatu *term* dapat memudahkan paragraf mana yang akan diawali untuk diringkas diringkas Hasil ringkasan dari sistem peringkasan teks otomatis pada penelitian ini adalah kalimat inti yang mirip dengan query dan berdasarkan urutan bobot, jadi untuk pengembangan penelitian berikutnya diharapkan hasil ringkasan memiliki urutan berdasarkan sistematika yang baik. Dalam membangun aplikasi peringkasan teks otomatis ini penulis menyadari bahwa terdapat kelemahan, oleh sebab itu dalam merancang peringkasan teks ini dapat dikombinasikan dengan algoritma lain agar dapat menghasilkan lebih baik.

#### REFERENCES

- [1] K. G. Subhawa, I. Atastina, A. Nursikuwagus, F. T. Informatika, and U. Telkom, "Algoritma De ( Differential Evolution )," pp. 0–4, 2013.
- [2] M. Nurjannah and I. Fitri Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman," J. Inform. Mulawarman, vol. 8, no. 3, pp. 110– 113, 2013.
- [3] Badudu dan Sutan Mohammad Zain, Efektifitas Bahasa Indonesia. Jakarta: Balai Pustaka, 2010.
- [4] D. Mahdiana, "Pengadaan Barang Dengan Metodologi Berorientasi Obyek: Studi Kasus PT. Liga Indonesia," J. Telemat., vol. 3, no. 2, pp. 36–43, 2016.
- [5] Mulyadi, Sistem Informasi Akuntansi. Jakarta: Salemba Empat, 2016.
- [6] M. M. Prof. Dr. H. Sahyar, M.S., ALGORITMA DAN PEMOGRAMathar, I.MANAJEMEN INFORMASI DAN KESEHATAN.MAN. 2016., 2018.
- [7] D. R. Radev and G. Erkan, "LexRank : Graph-based Centrality as Saliencie in Text Summarization," J. Artif. Intell. Res., vol. 22, no. 1, pp. 457–479, 2004..
- [8] S. Xie and Y. Liu, "Using corpus and knowledge-based similarity measure in Maximum Marginal Relevance for meeting summarization," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2, no. 2, pp. 4985–4988, 2008, doi: 10.1109/ICASSP.2008.4518777.
- [9] F. David, Grossman and phir, Information Retrieval: Algorithm and Heuristics. Kluwer Academic Publisher, 1998.
- [10] H. W. A. Kesuma, "Penerapan Metode TF-IDF dan Cosine Similarity dalam Aplikasi Kitab Undang-Undang Hukum Dagang." 2016.