

# **Penerapan Algoritma Maximum Marginal Relevance Dalam Peringkasan Teks Secara Otomatis**

**Ade Kurniawan\*, Mohd. Irsan Humaidy**

Fakultas Ilmu Komputer & Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: <sup>1</sup>kurniawancyber7@gmail.com, <sup>2</sup>irsanhumaidy@gmail.com  
Email Penulis Korespondensi: kurniawancyber7@gmail.com

**Abstrak**—Mesin pencari merupakan perangkat penelusur informasi dari dokumen-dokumen yang tersedia. Namun mesin pencari akan memberikan hasil pencarian yang sangat banyak, sehingga untuk mencari sebuah dokumen kita harus membuka satu persatu. Peringkasan teks memungkinkan kita untuk mendapatkan kunci informasi dari sebuah dokumen secara cepat tanpa mengharuskan kita untuk membaca isi dokumen secara manual. Salah satu metode yg digunakan adalah Maximum Marginal Relevance (MMR). MMR tergolong kepada kategori peringkasan ekstraktif (memilih kalimat yang ada pada dokumen sebagai kalimat pokok dari isi dokumen). MMR dilakukan dengan melakukan pembobotan untuk setiap kalimat yang ada pada dokumen. Diharapkan dengan diterapkannya Metode Maximum Marginal Relevance pada peringkasan teks secara otomatis ini dapat memperkecil tingkat redundansi data serta dapat membantu pembaca dalam memahami makna dari hasil ringkasan dan dapat memberikan informasi baik

**Kata Kunci:** Peringkasan ;Teks Otomatis; Metode Maximum Marginal Relevance

**Abstract**—The search engine is a device for searching information from available documents. But the search engine will give you a lot of search results, so to find a document we have to open a meeting. Text summarization allows us to get key information from a document quickly without requiring us to read the contents of the document manually. One of the methods used is Maximum Marginal Relevance (MMR). MMR belongs to the extractive summary category (choosing the sentence in the document as the main sentence of the document content). MMR is done by weighting each sentence in the document. It is hoped that with the application of the Maximum Marginal Relevance Method this automatic summary of text can reduce the level of data redundancy and can help readers understand the meaning of the summary results and provide good information

**Kata Kunci:** Summarization; Automatic Text; Maximum Marginal Relevance Method

## **1. PENDAHULUAN**

Kemajuan teknologi adalah sesuatu yang tidak bisa kita hindari dalam kehidupan ini, pencarian sebuah informasi pada sebuah dokumen sangatlah penting dan sudah menjadi aktivitas sehari-hari, seperti mencari sebuah jurnal untuk tugas atau mencari sebuah referensi. Ringkasan dibutuhkan untuk mendapatkan isi artikel secara ringkas. Ringkasan merupakan suatu bentuk karangan panjang yang dihadirkan dalam jumlah singkat, tujuannya untuk memberitahu pada pembaca inti dari suatu pikiran utama

Dengan banyaknya sebuah informasi dalam bentuk dokumen teks yang akan dibaca, maka akan sulit untuk menemukan teks yang ingin dicari. Oleh karena itu muncul ide dari peneliti untuk mempermudah pembaca mengetahui isi teks tanpa harus membaca teks secara keseluruhan. Peringkasan teks otomatis merupakan pembuatan bentuk yang lebih singkat dari suatu teks dengan memanfaatkan aplikasi yang dijalankan dan dioperasikan pada komputer[1]. Untuk mencari suatu text yang sesuai dengan pilihan, biasanya dengan memasukkan suatu kata kunci (query) agar menemukan teks yang sesuai (relevan). Tetapi ada beberapa baris-baris kalimat yang terdapat pada suatu dokumen tidak mengandung query yang dimasukkan. Agar menemukan teks yang sesuai, maka dibutuhkan suatu ringkasan.

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). Aplikasi peringkasan teks ini menggunakan tipe pendekatan ekstraktif, yaitu metode maximum marginal relevance. Maximum marginal relevance merupakan salah satu metode ekstraksi ringkasan (extractive summary) yang digunakan untuk meringkas dokumen tunggal atau multi dokumen. MMR meringkas dokumen dengan menghitung kesamaan (similarity) antara kalimat dengan kalimat dan antara kalimat dengan query . Maximum marginal relevance ini sendiri mampu dalam mengurangi redundansi pada kalimat yang akan diringkas [2].

Berdasarkan penelitian terdahulu yang telah dilakukan oleh Eko Budi Setiawan, dan Aji Teja Hartanto, dalam jurnalnya yang berjudul “Implementasi Metode Maximum Marginal Relevance (MMR) dan Algoritma Steiner Tree untuk Menentukan Storyline Dokumen Berita “menyimpulkan bahwa “ sistem ini melakukan penerapan algoritma maximum margina relevance ini dapat membantu pengguna mendapatkan dokumen yang terkait sesuai dengan query yang telah diinputkan. Metode menggunakan maximum marginal relevance merupakan salah satu metode yang tepat untuk digunakan dalam pencarian kata ditiap dokumen[3].

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 metode maximum marginal relevance.

## 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*

### 2.5 Maximum Marginal Relevance

*Maximum Marginal Relevance* (MMR) merupakan salah satu metode ekstraksi ringkasan (*extractive summary*) yang biasa digunakan untuk meringkas suatu dokumen tunggal maupun multi dokumen[6]. MMR adalah salah satu teknik peringkasan yang ditujukan untuk mengambil sebuah informasi yang *relevan* dan tidak mengandung *redundansi*(perulangan). Cara kerja MMR meringkas sebuah dokumen dengan menghitung kesamaan bagian teks dengan tujuan mendapatkan skor kalimat berdasarkan kesamaan (*similarity*) dengan *query* yang diberikan yang dapat mengurangi *redundansi* pada hasil ringkasan yang di dapat. Metode Maximum Marginal Relevance (MMR) sering digunakan untuk peringkasan teks karena metode MMR sederhana dan juga efisien [2]. Jika kesamaan antara satu kalimat dengan kalimat yang lain tinggi, maka terdapat kemungkinan terjadi *redundansi*. Metode MMR dapat mengurangi *redundansi* dengan rumus pada Persamaan 7.

$$MMR = \arg[\lambda * Sim1(S_i, Q) - (1 - \lambda) * \max Sim2(S_i, S')] \quad (1)$$

$\lambda$  = parameter yang memengaruhi tingkat relevansi

$S_i$  = vektor bobot kata yang menjadi kandidat

$S'$  = vektor bobot kata lainnya selain kandidat

$Q$  = vektor bobot kata dari *query*

$Sim1(S_i, Q)$  = nilai *similarity* antar kalimat ke-i dengan *query*

$Sim2(S_i, S')$  = nilai *similarity* antar kalimat ke-i dengan kalimat hasil ekstraksi

$Sim$  adalah nilai cosine similarity antara dua vektor.  $\lambda$  merupakan nilai koefisien yang mengatur relevansi kalimat dan mengurangi *redundansi*. Nilai parameter  $\lambda$  adalah 1 atau 0 atau antara ( $0 < \lambda < 1$ ). Pada saat parameter  $\lambda=1$  maka nilai MMR yang diperoleh cenderung relevan terhadap dokumen asli. Ketika  $\lambda = 0$  maka nilai MMR yang diperoleh akan cenderung relevan dengan kalimat yang diekstrak sebelumnya. Oleh karena itu, nilai  $\lambda$  perlu dioptimalkan agar mendapatkan ringkasan yang baik dengan nilai  $\lambda$  antara 0 sampai 1. Untuk peringkasan dengan dokumen yang kecil, seperti artikel akan menghasilkan hasil ringkasan yang baik jika nilai parameter  $\lambda = 0,7$

## 3. HASIL DAN PEMBAHASAN

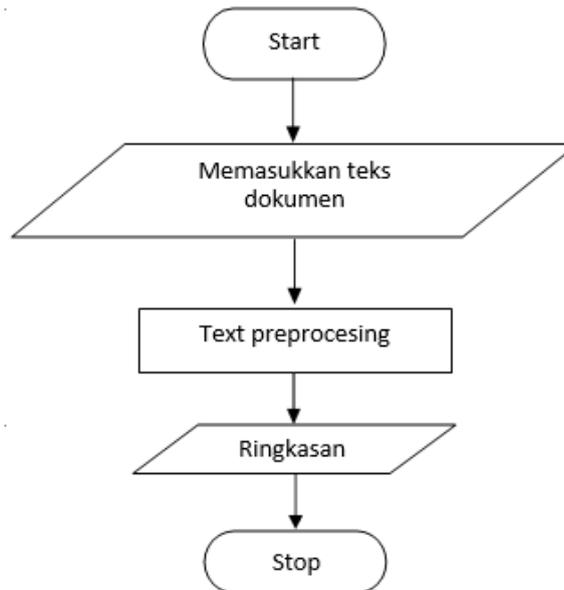
Membaca merupakan bagian dari kebutuhan manusia, baik membaca, surat kabar, majalah maupun buku. Perkembangan teknologi saat ini berdampak pada penggunaan internet untuk mempublikasi artikel di situs-situs internet. Demikian juga dengan artikel-artikel seperti berita banyak yang diunggah di situs-situs surat kabar online. Ringkasan dibutuhkan untuk mendapatkan isi artikel secara ringkas. Konsep sederhana ringkasan yaitu mengambil bagian penting dari keseluruhan isi

dari artikel 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 maximum marginal relevance (MMR), seperti pada penelitian yang dilakukan oleh Jaime Carbonell dan Jade Goldstein

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

Algoritma maximum marginal relevance digunakan untuk meranking kalimat-kalimat sebagai tanggapan terhadap query yang diberikan oleh user. Perhitungan mmr dilakukan dengan iterasi dengan mengkombinasikan dua matrik cosine similarity yaitu relevansi antara query terhadap keseluruhan kalimat dan similarity antara kalimat dengan kalimat. Pengguna yang menginginkan informasi ruang sampel disekitar query, maka harus menetapkan  $h$  pada nilai yang lebih rendah. Sedangkan bagi pengguna yang ingin fokus untuk memperkuat dokumen-dokumen yang agar lebih relevan, maka harus menetapkan  $h$  pada nilai yang lebih dekat dengan  $h$ . Kalimat dengan nilai MMR tertinggi dari setiap perhitungan iterasi akan di ambil, yang kemudian dipilih untuk dijadikan sebagai ringkasan. Iterasi akan berhenti ketika hasil mmr maksimum sama dengan 0

Untuk mengidentifikasi suatu dokumen maka memerlukan data dokumen, yang ingin diringkas Untuk memudahkan dalam melakukan analisa, maka penulis telah menentukan judul **“Pengaruh Dari Pendidikan Terhadap Kualitas Anak Di Pedalaman”**

Berikut ini adalah contoh teks yang diinputkan dalam proses *text preprocessing* “Sudah bukan hal yang sifatnya rahasia lagi bahwa anak-anak yang kebetulan tinggal di kawasan pedalaman amat sulit untuk memperoleh pendidikan yang layak sebagaimana yang didapatkan oleh anak-anak secara umum, terlebih anak-anak yang belajar di kawasan perkotaan. Selain sulit memperoleh pendidikan yang layak, mereka pun juga seringkali sulit mendapatkan air yang bersih. Bahkan pendidikan yang mereka dapatkan seringkali hanya sebatas kelayakan dari pendidikan Indonesia. Mereka juga banyak yang ketinggalan jaman dan cenderung tidak mengikuti perkembangan jaman. Selain itu, mereka bahkan juga seringkali tidak bisa mengenal alat komunikasi semisal Hp yang menjadi salah satu trend jaman modern”

### 1. Pemecahan Kalimat

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

**Tabel 1.** Hasil Pemecahan Kalimat

No	Kalimat
1	Sudah bukan hal yang sifatnya rahasia lagi bahwa anak-anak yang kebetulan tinggal di kawasan pedalaman amat sulit untuk memperoleh pendidikan yang layak sebagaimana yang didapatkan oleh anak-anak secara umum, terlebih anak-anak yang belajar di kawasan perkotaan

No	Kalimat
2	Selain sulit memperoleh pendidikan yang layak, mereka pun juga seringkali sulit mendapatkan air yang bersih.
3	Bahkan pendidikan yang mereka dapatkan seringkali hanya sebatas kelayakan dari pendidikan Indonesia.
4	Mereka juga banyak yang ketinggalan jaman dan cenderung tidak mengikuti perkembangan jaman.
5	Selain itu, mereka bahkan juga seringkali tidak bisa mengenal alat komunikasi semisal Hp yang menjadi salah satu trend jaman modern

## 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	sudah bukan hal yang sifatnya rahasia lagi bahwa anak anak yang kebetulan tinggal di kawasan pedalaman amat sulit untuk memperoleh pendidikan yang layak sebagaimana yang didapatkan oleh anak-anak secara umum terlebih anak anak yang belajar di kawasan perkotaan
2	selain sulit memperoleh pendidikan yang layak mereka pun juga seringkali sulit mendapatkan air yang bersih
3	bahkan pendidikan yang mereka dapatkan seringkali hanya sebatas kelayakan dari pendidikan Indonesia
4	mereka juga banyak yang ketinggalan jaman dan cenderung tidak mengikuti perkembangan jaman
5	selain itu mereka bahkan juga seringkali tidak bisa mengenal alat komunikasi semisal hp yang menjadi salah satu trend jaman modern

## 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). *Stoplist (stopword)* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag- of-words*. Dalam tahap pembuangan kata-kata yang tidak penting adalah kata hasil parsing dicek dengan kamus (kumpulan kata) *stopword*. Jika kata parsing ada yang sama dengan *stopword* maka kata akan dibuang atau dihapus. *Stopword* yang digunakan dalam penelitian ini dapat dilihat pada halaman lampiran

**Tabel 3.** Hasil *Filtering* kata

No	Kalimat
1	anak tinggal kawasan pedalaman sulit memperoleh pendidikan layak kawasan perkotaan
2	sulit memperoleh pendidikan layak mereka seringkali sulit mendapatkan air bersih
3	pendidikan mereka dapatkan sebatas kelayakan pendidikan Indonesia
4	mereka banyak ketinggalan jaman
5	mereka seringkali mengenal alat komunikasi hp trend jaman modern

## 4. 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
1	Air	14	Layak
2	Alat	15	Memperoleh
3	Anak	16	Medapatkan
4	Banyak	17	Mengenal
5	Bersih	18	Mereka
6	Dapatkan	19	Modern
7	Hp	20	Pendidikan
8	Indonesia	21	Perkotaan
9	Jaman	22	Sebatas
10	Kawasan	23	Seringkali
11	Kelayakan	24	Sulit
12	Ketinggalan	25	Tinggal
13	Komunikasi	26	Trend

## 5. Stemming

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
1	Air	13	Komunikasi
2	Alat	14	Kota
3	Anak	15	Layak
4	Banyak	16	Mereka
5	Batas	17	Modern
6	Bersih	18	Pendidikan
7	Dapat	19	Peroleh
8	Hp	20	Seringkali
9	Indonesia	21	Sulit
10	Jaman	22	Tinggal
11	Kawasan	23	Trend
12	Kenal		

#### 6. Pembobotan TF-IDF

Pembobotan secara otomatis biasanya berdasarkan jumlah kemunculan suatu kata dalam sebuah dokumen (*term frequency/tf*) dan jumlah kemunculannya dalam koleksi dokumen (*inverse document frequency/idf*). Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. Aktifitas pembobotan tf-idf menghasilkan bobot tiap kata. Pembobotan tf-idf dilakukan untuk pembobotan tahap selanjutnya, yaitu digunakan untuk menghitung bobot *relevance query* dan bobot *similarity* kalimat. Alur *activity* pada bobot tf-idf untuk *relevance query* adalah sama seperti *activity* pada bobot tf-idf untuk *similarity* kalimat. Perhitungan bobot tf-idf untuk *relevance query* dan *similarity* kalimat dapat dilihat pada Lampiran 3. Berikut ini *activity* diagram pembobotan tf-idf secara umum.

**Tabel 6.** Perhitungan Nilai IDF

Term	Term Frekuensi							IDF
	D1	D2	D3	D4	D5	DF	d/tf	Log(n/df)
Air	0	1	0	0	0	1	5	0.698
Alat	0	0	0	0	1	1	5	0.698
Anak	6	0	0	0	0	1	5	0.698
Banyak	0	0	0	1	0	1	5	0.698
Batas	0	0	1	0	0	1	5	0.698
Bersih	0	1	0	0	0	1	5	0.698
Dapat	0	1	1	0	0	2	2.5	0.397
Hp	0	0	0	0	1	1	5	0.698
Indonesia	0	0	1	0	0	1	5	0.698
Jaman	0	0	0	1	0	1	5	0.698
Kawasan	1	0	0	0	0	1	5	0.698
Kenal	0	0	0	0	1	1	5	0.698
Komunikasi	0	0	0	0	1	1	5	0.698
Kota	1	0	0	0	0	1	5	0.698
Layak	1	1	1	0	0	3	1.666	0.221
Mereka	0	1	1	1	1	4	1.25	0.096
Modern	0	0	0	0	1	1	5	0.698
Pendidikan	1	1	2	0	0	3	1.666	0.221
Peroleh	1	1	0	0	0	2	2.5	0.397
Seringkali	0	1	1	0	1	3	1.666	0.221
Sulit	1	2	0	0	0	2	2.5	0.397
Tinggal	1	0	0	1	0	2	2.5	0.397
Trend	0	0	0	0	1	1	5	0.698

**Tabel 7.** Nilai Bobot

Term	D1	D2	D3	D4	D5
Air	0	0.698	0	0	0
Alat	0	0	0	0	0.698
Anak	4.188	0	0	0	0
Banyak	0	0	0	0.698	0
Batas	0	0	0.698	0	0
Bersih	0	0.698	0	0	0
Dapat	0	0.397	0.397	0	0
Hp	0	0	0	0	0.698
Indonesia	0	0	0.698	0	0
Jaman	0	0	0	0.698	0
Kawasan	0.698	0	0	0	0
Kenal	0	0	0	0	0.698
Komunikasi	0	0	0	0	0.698
Kota	0.698	0	0	0	0
Layak	0.221	0.221	0.221	0	0
Mereka	0	0.221	0.221	0.221	0.221
Modern	0	0	0	0	0.698
pendidikan	0.221	0.221	0.442	0	0
Peroleh	0.397	0.397	0	0	0
Seringkali	0	0.221	0.221	0	0.221
Sulit	0.397	0.794	0	0	0
Tinggal	0.397	0	0	0.397	0
Trend	0	0	0	0	0.698

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

D1	D2	D3	D4	D5
0	0.487	0	0	0
0	0	0	0	0.487
17,539	0	0	0	0
0	0	0	0.487	0
0	0	0.487	0	0
0	0.487	0	0	0
0	0.157	0.157	0	0
0	0	0	0	0.487
0	0	0.487	0	0
0	0	0	0.487	0
0.487	0	0	0	0
0	0	0	0	0.487
0	0	0	0	0.487
0.487	0	0	0	0
0.048	0.048	0.048	0	0
0	0.048	0.048	0.048	0.048
0	0	0	0	0.487
0.048	0.048	0.048	0	0
0.157	0.157	0	0	0
0	0.048	0.048	0	0.048
0.157	0.63	0	0	0
0.157	0	0	0.157	0
0	0	0	0	0.487
<b>17540.54</b>	<b>2.11</b>	<b>1.323</b>	<b>1.179</b>	<b>3.018</b>
<b>132.44</b>	<b>1.452</b>	<b>1.15</b>	<b>1.085</b>	<b>1.737</b>

**Tabel 9.** Perhitungan *Cosine Similarity*

D1	D2	D3	D4	D5
0	0.575	0	0	0
0	0	0	0	0.481
1.000	0	0	0	0
0	0	0	0.771	0
0	0	0.727	0	0
0	0.575	0	0	0

D1	D2	D3	D4	D5
0	0.434	0.936	0	0
0	0	0	0	0.481
0	0	0.727	0	0
0	0	0	0.771	0
0.006	0	0	0	0
0	0	0	0	0.481
0	0	0	0	0.481
0.006	0	0	0	0
0.003	0.324	0.409	0	0
0	0.324	0.409	0.414	0.27
0	0	0	0	0.481
0.003	0.324	0.592	0	0
0.004	0.434	0	0	0
0	0.324	0.409	0	0.27
0.004	0.614	0	0	0
0.004	0	0	0.581	0
0	0	0	0	0.481

## 7. Proses Pemeringkatan

### a. Perhitungan Nilai IDF

Pada tabel.5 hasil *stemming* kata yang akan di ambil sebagai kata *term* (t) dan menjadi keyword untuk berita yang diuji. Kalimat pada tajuk berita akan dicek jika mengandung *term* tersebut. Setelah diketahui berapa kali kata yang muncul dalam setiap kalimat dalam berita (*term frequency*), maka langkah selanjutnya adalah menghitung nilai IDF setiap *term* yang muncul dalam kalimat. Perhitungan dilakukan dengan menggunakan persamaan 2.1. Hasil dari perhitungan IDF tertera pada tabel 6. Perhitungan IDF menggunakan T1 “Angka” pada D2 Perhitungan IDF dilakukan dengan menggunakan rumus  $IDF = \log (N)$

Diketahui:

N : Jumlah kalimat dalam dokumen = 5

dfi : Jumlah kalimat yang berisi kata “air” = 1

$$IDF = \log (N)$$

$$IDF = \log ( 5 )$$

$$IDF = \log (4)$$

$$IDF = 0.698$$

### b. Perhitungan Bobot

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 “air” muncul sebanyak 1 kali)

IDF : Inversed Document Frequency = 0.698

Maka,

$$Ws.t = TFd.t * IDFt$$

$$= 1 * 0.698$$

$$Ws.t = 0.698$$

### c. Perhitungan Nilai Panjang Vektor Setiap Kalimat

Setelah mendapatkan nilai bobot langkah selanjutnya adalah menghitung nilai panjang vector setiap kalimat dengan menggunakan persamaan 2.3. 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:

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

$$|D1| = \sqrt{0^2 + 0 + (4.118)^2 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + (0.698)^2 + 0 + 0 + (0.698)^2 + (0.221)^2 + 0 + 0 + (0.221)^2 + (0.397)^2 + 0 + (0.397)^2 + (0.397)^2 + 0}$$

$$|D1| = \sqrt{17541}$$

$$|D1| = 132.4$$

### d. Perhitungan Cosine Similarity

Setelah menghitung nilai bobot dan nilai panjang vektor, maka selanjutnya dihitung nilai *cosine similarity* dengan menggunakan persamaan 2.3. Perhitungan dilakukan dengan membagi nilai IDF dengan akar dari nilai IDF dikalikan nilai panjang vektor. Perhitungan cosine similarity menggunakan rumus:

Perhitungan cosine similarity menggunakan T3 pada D1.

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

$$\text{Idf - modified - cosine } (x, y) = \frac{17539}{\sqrt{0.477} \times 132.40}$$

$$\text{Idf - modified - cosine } (x, y) = \frac{17539}{132.43 \times 132.40}$$

$$\text{Idf - modified - cosine } (x, y) = \frac{17539}{17533}$$

$$\text{Idf - modified - cosine } (x, y) = 1.000$$

**Tabel 10.** Nilai Akhir Bobot

Kalimat	Bobot Akhir
D1	132.4
D2	1.452
D3	1.15
D4	1.085
D5	1.737

#### 4. KESIMPULAN

Aplikasi peringkasan teks otomatis menggunakan metode algoritma maximum marginal relevance dapat menghasilkan ringkasan secara otomatis tanpa menghilangkan makna dalam teks. Dapat menerapkan Algoritma maximum marginal relevance dalam melakukan peringkasan menggunakan aplikasi peringkasan teks secara otomatis. 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] M. Mustaqhfiri, Z. Abidin, and R. Kusumawati, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," *Matics*, 2012, doi: 10.18860/mat.v0i0.1578.
- [2] N. F. Saraswati, Indriati, and R. S. Perdana, "Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 5494–5502, 2018, doi: 10.1016/s1010-6030(01)00380-x.
- [3] E. B. Setiawan and A. T. Hartanto, "Implementasi Metode Maximum Marginal Relevance (MMR) dan Algoritma Steiner Tree untuk Menentukan Storyline Dokumen Berita," *Jurnal ULTIMATICS*, vol. 8, no. 1, 2017, doi: 10.31937/ti.v8i1.499.
- [4] E. Hovy and C.-Y. Lin, "Automated text summarization and the SUMMARIST system," p. 197, 1996, doi: 10.3115/1119089.1119121.
- [5] M. Inderjeet, "Summarization Evaluation: An Overview," *Pflege Z.*, vol. 62, no. 6, pp. 337–341, 2009.
- [6] F. T. Waruwu and R. Mandala, "Perbandingan Algoritma Knuth Morris Pratt dan Boyer Moore Dalam Pencocokan String Pada Aplikasi Kamus Bahasa Nias," *J. Ilm. INFOTEK*, vol. 1, no. 1, pp. 36–43, 2016.
- [7] S. Andini, "Klasifikasi Dokumen Teks Menggunakan Algoritma Naive Bayes dengan Bahasa Pemrograman Java," *Teknol. Inf. Pendidik.*, vol. 6, no. 2, pp. 140–147, 2013, [Online]. Available: <http://www.jurnal-tip.net/jurnal-resource/file/13-Vol6No2Sep2013-Silfia Andini.pdf>.
- [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.