

# Implementasi Text Summarization Pada Ulasan Aplikasi Mobile JKN Menggunakan TF-IDF Dan Cosine Similarity

Valencia Ivena Lim<sup>1,\*</sup>, Fitria<sup>2</sup>, M. Hafid<sup>3</sup>

<sup>1,2,3</sup>Sistem Informasi, STMIK PPKIA Tarakanita Rahmawati Tarakan

## ABSTRACT

*Mobile JKN is an application developed by Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan that was developed for easier access to national health services. As of November 2024, this application has been downloaded 50 million times on the Google Play Store, and about 720 thousand reviews have been given by users. The reviews provided by users who have downloaded the Mobile JKN app are very useful and important for potential users and developers. However, the huge volume of reviews is a challenge in reading them one by one and can cause information overload. Based on the occurring problems, the author will apply text summarization to summarize the reviews of the JKN Mobile application by implementing the Term Frequency-Inverse Document Frequency (TF-IDF) and Cosine Similarity methods. The author added the Maximum Marginal Relevance (MMR) method because the TF-IDF and Cosine Similarity methods cannot produce a summary. Summarization is done by taking the most relevant reviews from among a collection of other reviews. This research resulted with the average Accuracy value of 29.6%, Precision 55.6%, and Recall 39.8%, with the highest value of Accuracy 38.4%, Precision 79.2%, and Recall 42.7%.*

**Keywords:** *Cosine Similarity, Text Summarization, TF-IDF, MMR*

## ABSTRAK

Aplikasi Mobile JKN merupakan aplikasi yang dikembangkan oleh Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan yang dikembangkan untuk kemudahan akses layanan kesehatan nasional. Per November 2024, aplikasi ini telah diunduh sekitar 50 juta kali di Google Play Store dan sekitar 720 ribu ulasan telah diberikan oleh para pengguna. Ulasan-ulasan yang diberikan oleh pengguna yang telah mengunduh aplikasi Mobile JKN sangat bermanfaat dan penting bagi calon pengguna dan pengembang. Akan tetapi, volume ulasan yang sangat besar menjadi tantangan dalam membacanya satu per satu dan dapat menimbulkan *information overload*. Berdasarkan permasalahan yang terjadi, maka penulis akan menerapkan *text summarization* untuk meringkas ulasan-ulasan aplikasi Mobile JKN dengan mengimplementasikan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity*. Penulis menambahkan metode *Maximum Marginal Relevance* (MMR) karena metode TF-IDF dan *Cosine Similarity* tidak dapat menghasilkan ringkasan. Peringkasan dilakukan dengan mengambil ulasan-ulasan yang paling relevan dari antara kumpulan ulasan lainnya. Penelitian ini menghasilkan nilai rata-rata *Accuracy* 29,6%, *Precision* 55,6%, dan *Recall* 39,8% dengan nilai tertinggi *Accuracy* 38,4%, *Precision* 79,2%, dan *Recall* 42,7%.

**Kata Kunci:** *Cosine Similarity, Text Summarization, TF-IDF, MMR*

---

Corresponding Author E-mail: 2150038@student.ppkia.ac.id \*

Received November 2024; revised Januari 2025; accepted Januari 2025; published Januari 2025

## 1. Pendahuluan

Google Play Store merupakan tempat untuk mengunduh aplikasi yang paling populer di kalangan pengguna Android. Seiring dengan meningkatnya jumlah pengguna Android, volume ulasan aplikasi di Google Play Store juga semakin meningkat. Besarnya volume ulasan yang tersedia dapat menjadi tantangan bagi calon pengguna dan pihak pengembang dalam mencari tahu kualitas dan performa aplikasi tersebut.

Sebelum mengunduh suatu aplikasi, sebagian calon pengguna akan membaca ulasan untuk mengetahui kualitas aplikasi tersebut. Namun, banyaknya jumlah ulasan yang tersedia dapat menimbulkan *information overload* yang diidentifikasi sebagai penyebab kelelahan pengguna yang dapat membebani kapasitas pengolahan informasi dan memunculkan duplikasi informasi yang tidak perlu [1].

Bagi pengembang aplikasi, ulasan dari pengguna sangat berguna dalam memperbaiki performa aplikasi dan merencanakan pembaruan di masa depan. Namun, dengan banyaknya ulasan yang diterima, ulasan yang tidak terstruktur, berisi informasi berulang, atau bahkan kurang relevan menjadi tantangan besar bagi pengembang dalam menyaring informasi yang relevan dan yang diprioritaskan untuk ditindaklanjuti.

Salah satu solusi untuk mengatasi masalah-masalah di atas adalah dengan menerapkan *text summarization*, yakni proses untuk meringkas teks tanpa mengurangi makna penting dari isi keseluruhan teks tersebut [2]. *Text summarization* merupakan terapan dari *text mining*, yaitu proses untuk menggali informasi dengan cara pengguna berinteraksi dengan dokumen-dokumen menggunakan alat analisis [3].

Metode yang digunakan dalam penelitian ini antara lain metode TF-IDF, *Cosine Similarity*, dan MMR. Metode TF-IDF digunakan untuk memberikan bobot pada hubungan suatu kata dengan dokumen, dengan mengabaikan kata-kata yang dianggap tidak penting [4]. Metode *Cosine Similarity* digunakan untuk mengukur sudut kosinus antara dua objek untuk membandingkan dua dokumen pada skala yang telah dinormalisasi, dilakukan dengan menghitung hasil perkalian titik antara kedua objek tersebut [5]. Metode *Maximum Marginal Relevance* (MMR) digunakan untuk meringkas dokumen tunggal ataupun multi dokumen dengan menghitung kesamaan antara kalimat dengan kalimat dan antara kalimat dengan *query* [6].

Awalnya penulis hanya mengajukan metode TF-IDF dan *Cosine Similarity*, tetapi karena kedua metode tersebut tidak dapat menghasilkan ringkasan maka penulis menambahkan metode MMR. Penulis memilih untuk menggunakan metode MMR karena metode ini umum digunakan dalam peringkasan teks, dapat mengurangi redundansi dalam peringkasan kalimat, dan

dapat menyeimbangkan relevansi dengan keberagaman [7][8].

Objek penelitian yang digunakan adalah ulasan aplikasi Mobile JKN. Aplikasi yang dikembangkan oleh BPJS Kesehatan ini dirancang untuk mem-permudah proses pendaftaran, perubahan data kepesertaan, dan pengaksesan informasi peserta keluarga, serta memberi fasilitas pemantauan biaya iuran, kemudahan akses ke layanan Fasilitas Kesehatan Tingkat Pertama (FKTP) dan Fasilitas Rujukan Tingkat Lanjutan) (FKTL), serta sarana untuk me-nyampaikan saran atau keluhan [9].

Penelitian ini diharapkan dapat membantu calon pengguna dan pihak pengembang dalam memahami kualitas dan performa aplikasi dengan lebih efisien. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan teknologi peringkasan teks otomatis, khususnya dalam bidang ulasan aplikasi.

## 2. Tinjauan Pustaka

Penelitian ini didasarkan pada penelitian yang dilakukan oleh [10] dengan topik peringkasan teks otomatis multi dokumen pada artikel berita gempa. Penelitian sebelumnya mengusulkan metode TF-IDF dan kompresi sebesar 75% terhadap gabungan dari ringkasan artikel pertama dan artikel kedua, menghasilkan sebuah ringkasan yang berisikan 6 kalimat. Penelitian sebelumnya memiliki akurasi hasil uji responden sebesar 54,45% dan uji kemiripan dokumen sebesar 78,023%.

Penulis menggunakan topik yang sama dengan penelitian sebelumnya, tetapi dengan objek dan metode penelitian yang berbeda. Objek yang digunakan dalam penelitian ini adalah ulasan aplikasi Mobile JKN, sedangkan metode yang diusulkan oleh penulis adalah TF-IDF dan *Cosine Similarity*. Penulis juga menambahkan metode MMR karena metode TF-IDF dan *Cosine Similarity* tidak dapat menghasilkan ringkasan sebab kedua metode tersebut hanya dapat digunakan untuk melakukan pembobotan dan pencarian.

## 3. Metode

### 3.1. Text Mining

*Text mining* adalah metode untuk menambang data tidak terstruktur dalam bentuk teks yang berasal dari sumber seperti dokumen, lampiran, atau kalimat. Metode ini bertujuan untuk menemukan kata-kata penting dalam dokumen sehingga dapat dilakukan analisis terkait dokumen tersebut. *Text mining* biasanya digunakan untuk mengambil kata-kata penting yang kemudian diklasifikasikan dan dianalisis hubungannya antar dokumen [11].

### 3.2. Web Scraping

Pengumpulan data dilakukan dengan menggunakan *web scraping*, yakni metode untuk

mengambil informasi dari situs web secara otomatis dengan mengekstraksi data yang tersedia [12]. Data yang dikumpulkan merupakan ulasan aplikasi Mobile JKN dalam bahasa Indonesia yang diambil dari Google Play Store. Jumlah data yang akan digunakan dalam penelitian ini adalah sebanyak 10 ulasan, diambil berdasarkan relevansi dan diurutkan berdasarkan tanggal terbaru.

### 3.3. Text Preprocessing

*Text preprocessing* adalah langkah awal dalam pemrosesan teks yang bertujuan untuk menjadikan teks lebih terstruktur dan mempersiapkannya sebagai data yang siap diolah [13]

Berikut adalah tahapan-tahapan yang ada di dalam *text preprocessing* menurut [14]:

- a. *Tokenizing*, proses penguraian teks yang semula berupa kalimat menjadi kata.
- b. *Filtering*, tahap mengambil kata penting dari hasil proses *tokenizing*. Tahap ini dapat menggunakan algoritma *stop list* maupun *word list*.
- c. *Stemming*, tahap untuk mencari kata dasar dari hasil *filtering*.
- d. *Tagging*, tahap untuk mencari *root* (bentuk awal) dari tiap kata lampau dari hasil proses *stemming*. Biasanya tahap ini digunakan dalam pemrosesan teks berbahasa Inggris.
- e. *Analyzing*, tahap penentuan seberapa jauh keterhubungan antar suatu kata terhadap suatu dokumen atau kalimat dengan menghitung nilai/bobot ke-terhubungan. Biasanya menggunakan metode TF-IDF.

### 3.4. TF-IDF

TF-IDF merupakan metode statistik yang digunakan untuk menilai pentingnya sebuah kata kunci dalam sebuah dokumen, khususnya dalam konteks kumpulan dokumen yang lebih besar. TF-IDF menghasilkan bobot atau skor yang menunjukkan seberapa penting kata kunci tersebut dalam dokumen tertentu [15].

Perhitungan bobot masing-masing dokumen terhadap kata kunci dapat dilihat pada persamaan 1.

$$W_{d,t} = TF_{d,t} * IDF \quad (1)$$

Dimana  $W$  adalah bobot dokumen ke- $n$ ,  $d$  adalah dokumen,  $t$  adalah kata kunci, TF adalah singkatan dari *Term Frequency*, dan IDF adalah singkatan dari *Inverse Document Frequency*.

Perolehan nilai TF dapat dilihat pada persamaan 2.

$$TF_{d,t} = \frac{\text{Kemunculan KK dalam dokumen}}{\text{Total kata dalam dokumen}} \quad (2)$$

Dimana KK adalah singkatan dari kata kunci.

Perolehan nilai IDF dapat dilihat pada persamaan 3.

$$IDF = 1 + \log_{10} \left( 1 + \log_{10} \left( \frac{D}{df} \right) \right) \quad (3)$$

Dimana  $D$  adalah total dokumen dan  $df$  adalah jumlah dokumen yang mengandung kata kunci.

Selain itu, jika hasil pembagian  $D/df$  menghasilkan 0, maka nilai IDF langsung ditetapkan sebagai 0 untuk menghindari nilai tak terdefinisi.

### 3.5. Cosine Similarity

*Cosine Similarity* digunakan untuk mengukur tingkat kesamaan antara dua objek yang dinyatakan dalam vektor dengan kata kunci dari sebuah dokumen sebagai ukuran. Tingkat kesamaan yang dihasilkan oleh metode ini berada dalam rentang antara 0 hingga 1, dimana nilai 0 menunjukkan kedua objek tersebut berbeda, dan nilai 1 menunjukkan kedua objek tersebut identik [16].

Persamaan yang digunakan untuk menghitung tingkat kesamaan dapat dilihat pada persamaan 4.

$$\text{CosSim}(d_j, d_q) = \frac{\sum_{i=1}^t (w_{ij} * w_{iq})}{\sqrt{(\sum_{i=1}^t w_{ij}^2 * w_{ij}) * (\sum_{i=1}^t w_{iq}^2 * w_{iq})}} \quad (4)$$

Dimana  $\text{CosSim}(d_j, d_q)$  adalah tingkat kesamaan dokumen dengan *query* tertentu,  $w_{ij}$  adalah bobot dari *term* ke- $i$  dalam vektor untuk dokumen ke- $j$ ,  $w_{iq}$  adalah bobot dari *term* ke- $i$  dalam vektor untuk *query* ke- $q$ , dan  $t$  adalah jumlah *term* yang unik dalam *dataset*.

### 3.6. MMR

MMR adalah metode ekstraksi ringkasan yang digunakan untuk meringkas satu dokumen atau beberapa dokumen. Metode MMR bekerja dengan menghitung kesamaan antar bagian teks dan menghasilkan ringkasan yang bebas dari redundansi serta hanya mengambil informasi yang relevan [17].

Persamaan untuk menghitung nilai MMR dapat dilihat pada persamaan 5.

$$\text{MMR} = \text{argmax} [\lambda * \text{Sim1}(S_i, Q) - (1 - \lambda) * \text{maxSim2}(S_i, S')] \quad (5)$$

Dimana MMR adalah nilai MMR,  $\lambda$  adalah parameter yang mempengaruhi tingkat relevansi,  $S_i$  adalah vektor bobot kata yang menjadi kandidat,  $S'$  adalah vektor bobot kata lainnya selain kandidat,  $Q$  adalah vektor bobot kata dari *query*,  $\text{Sim1}(S_i, Q)$  adalah nilai *similarity* antar kalimat ke- $i$  dengan *query*, dan  $\text{Sim2}(S_i, S')$  adalah nilai *similarity* antar kalimat ke- $i$  dengan kalimat hasil ekstraksi.

### 3.7. Confusion Matrix

*Confusion matrix* adalah metode yang digunakan untuk menghitung akurasi dalam suatu konsep klasifikasi. Evaluasi dengan *confusion matrix* menghasilkan beberapa matriks seperti nilai *Accuracy*, *Precision*, dan *Recall*.

*Accuracy* merupakan persentase ketepatan data yang telah diklasifikasikan dengan benar. *Precision* mengacu pada proporsi kasus yang diprediksi positif dan benar positif dalam data. *Recall* menggambarkan proporsi kasus positif yang berhasil diprediksi dengan tepat [18].

Persamaan yang digunakan untuk menghitung *Accuracy*, *Precision*, dan *Recall* dapat dilihat pada persamaan 6, 7, dan 8.

$$Accuracy = \frac{\text{Jumlah Kata yang Sama}}{\text{Banyak Kata pada Ringkasan S \& M}} \quad (6)$$

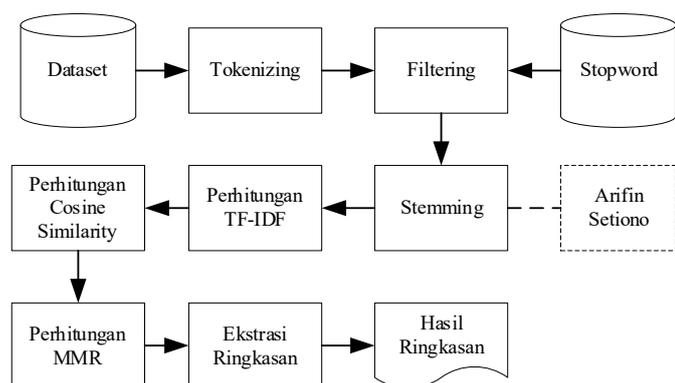
$$Precision = \frac{\text{Jumlah Kata yang Sama}}{\text{Banyak Kata pada Ringkasan S}} \quad (7)$$

$$Recall = \frac{\text{Jumlah Kata yang Sama}}{\text{Banyak Kata pada Ringkasan M}} \quad (8)$$

Dimana *Ringkasan S* adalah ringkasan sistem dan *Ringkasan M* adalah ringkasan manual.

### 4. Hasil dan Pembahasan

Alur proses sistem *text summarization* ini dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Sistem

Berdasarkan Gambar 1, proses *text summarization* dimulai dengan pe-ngumpulan *dataset*, lalu data-data tersebut akan melalui proses *tokenizing*, *filtering* menggunakan *stopword*, dan *stemming* menggunakan algoritma Arifin Setiono. Setelah itu, dilakukan pembobotan dengan menggunakan metode TF-IDF dan *Cosine Similarity*. Kemudian melakukan perhitungan metode MMR sebelum mengekstrasi ringkasan dan menghasilkan ringkasan yang relevan.

*Dataset* yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Daftar Ulasan

No	Ulasan
1	tolong dibenarkan lagi aplikasinya. setiap fitur pasti banyak errornya. Menerima kode otp saja harus menunggu lama banget. Aplikasi ini sangat membantu jika semua fitur aplikasi bisa digunakan dengan nyaman tetapi malah banyak errornya.
2	Mendaftar dengan 2 akun berbeda, kacau sekali untuk buka menu harus verifikasi tetapi saat ingin verifikasi yang dicantumkan nomor berbeda. Bagaimana bisa verifikasi kalau yang muncul nomor orang lain? Sudah coba buat akun baru dengan ktp lain sama saja mengawur.
3	Berminggu-minggu faskes tingkat lanjutnya tidak mau muncul, pemeriksaan ibu saya jadi terhambat terus, sudah direfresh berkali-kali tetap begitu. Sekiranya aplikasi sudah tidak berfungsi mending ditiadakan saja daripada hanya menambah penyimpanan di hp. Jika jawaban ulasan saya adalah akan dilakukan perbaikan sistem, lebih baik tidak usah dijawab! Cukup tindakan saja yang perlu ditunjukkan.
4	Parah berputar-putar mau log in saja, karena lupa password harus verifikasi sms, pertama ada warning jaringan internet lemah padahal pakai wifi full akses, kedua ada peringatan nomor kartu salah padahal benar, ketiga peringatan belum terdaftar. 3x salah tunggu time selesai. Akhirnya coba mendaftar sudah isi semua kolom sewaktu di oke ada peringatan anda sudah terdaftar.... Gubraaak!!, wah jawabannya disuruh update padahal baru install versi terbaru tuh...
5	Data sudah tersimpan di profil tetapi masuk menu kartu untuk cetak kartu online tidak bisa padahal semua data sudah terdaftar di jkn. Tertulis malah belum terdaftar. Aplikasi sampah dan menyulitkan rakyat kecil.
6	Admin tolong perbaiki fitur perubahan data yang ganti nomor hp, di bagian pengisian persyaratan perubahan data tidak bisa dipencet bagaimana mau mengisinya, Harusnya ini tugas kalian yang cek secara berkala fitur-fitur yang ada di aplikasi supaya masyarakat lebih mudah mengakses!!! Secepatnya diperbaiki!!! Kalau bisa persyaratannya jangan foto pegang KTP, kayak mau pinjol saja!! antisipasi untuk kalian para pengelola, ada oknum yang menyalahgunakan!!!!
7	OTP harus menggunakan SMS sudah sangat tidak zaman lagi, mata duitan minta pulsa. Padahal banyak pilihan lain seperti email atau WA bahkan bisa google autentikasi. Aplikasi yang sangat mata duitan mempersulit bukan membantu pelayanan masyarakat
8	Apa-apaan sih aplikasinya, setiap mengambil nomor antrian, katanya sudah ada nomor antrian, sementara nomornya pun tidak muncul, sudah 2 hari dicoba, jaringan hp lancar dan bagus, dicoba juga dengan wifi sama saja hasilnya
9	Setiap mau daftar setelah mengisi semuanya dan waktu mengisi verifikasi lewat emailnya... waktu mau mengecek kode dan sudah dapat kodenya waktu buka aplikasinya mesti harus kembali ke awal lagi...

No	Ulasan
	Daftar lagi kirim verifikasi lagi ganti lagi... Dan seterusnya...
10	Ini bagaimana sih padahal sudah dikirim kode Verifikasi via SMS dan email dan sudah saya masukkan sesuai prosedur tetapi kok keterangan akun belum terverifikasi, Kalau memang belum terverifikasi, kode yang dikirim waktu pendaftaran untuk apa? Mana Pandawa JKN juga tidak ada menu buat verifikasi <i>virtual account</i> , kalau kayak begini mau dipakai untuk bersalin jadi ribet dan bikin pusing pengguna, anggaran saja miliaran tetapi pelayanan tidak memuaskan

Tahapan pertama yang dilakukan setelah mengumpulkan data ulasan adalah *text preprocessing*. *Text preprocessing* terdiri atas tahap *tokenizing*, *filtering*, *stemming*, dan *analyzing*.

*Tokenizing* adalah proses pemotongan kalimat menjadi kata-kata individu (*token*). Contoh tahap *tokenizing* dapat dilihat pada Tabel 2.

**Tabel 2.** Tokenizing

Kalimat	Tokenizing
tolong dibenarkan lagi aplikasinya. setiap fitur pasti banyak errornya. Menerima kode otp saja harus menunggu lama banget. Aplikasi ini sangat membantu jika semua fitur aplikasi bisa digunakan dengan nyaman tetapi malah banyak errornya.	tolong dibenarkan lagi dibenarkan lagi aplikasinya setiap fitur fitur pasti banyak errornya menerima kode otp saja harus menunggu lama banget aplikasi ini sangat membantu jika semua fitur aplikasi bisa digunakan dengan nyaman tetapi malah banyak errornya

Langkah selanjutnya adalah *filtering*, yakni proses pengambilan kata-kata penting. Contoh tahap *filtering* dilihat pada Tabel 3.

**Tabel 3.** Filtering

Tokenizing	Filtering
tolong dibenarkan lagi aplikasinya setiap fitur pasti banyak errornya menerima kode otp saja harus menunggu lama banget aplikasi ini sangat membantu jika semua fitur aplikasi bisa digunakan dengan nyaman tetapi malah banyak errornya	tolong dibenarkan aplikasinya fitur errornya menerima kode otp menunggu aplikasi membantu fitur aplikasi nyaman errornya

Langkah selanjutnya adalah *stemming*, yakni proses menghilangkan imbuhan. Contoh tahap *stemming* dapat dilihat pada Tabel 4.

**Tabel 4.** Stemming

Filtering	Stemming
tolong dibenarkan aplikasinya fitur errornya menerima kode otp menunggu aplikasi membantu fitur aplikasi	tolong benar aplikasi fitur error terima kode otp tunggu aplikasi bantu fitur aplikasi

Filtering	Stemming
nyaman errornya	nyaman error

Setelah selesai melakukan tahapan *text preprocessing* pada semua dokumen, *term* atau kata yang dihasilkan dari tahap *stemming* akan digunakan dalam tahap *analyzing* yang meliputi perhitungan metode TF-IDF dan *Cosine Similarity*.

Dari 132 *term* yang didapatkan dari tahap *stemming*, penulis hanya akan menampilkan 10 *term* dalam dokumentasi hasil perhitungan untuk mendapatkan nilai TF dan *df*. Contoh hasil perhitungan untuk mendapatkan nilai TF dan *df* dapat dilihat pada Tabel 5.

Tabel 5. TF & df

Term	TF										df
	D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10	
tolong	1	0	0	0	0	1	0	0	0	0	2
benar	1	0	0	0	0	0	0	0	0	0	1
aplikasi	3	0	1	0	1	1	1	1	1	0	7
fitur	2	0	0	0	0	2	0	0	0	0	2
error	2	0	0	0	0	0	0	0	0	0	1
terima	1	0	0	0	0	0	0	0	0	0	1
kode	1	0	0	0	0	0	0	0	2	2	3
otp	1	0	0	0	0	0	1	0	0	0	2
tunggu	1	0	0	1	0	0	0	0	0	0	2
bantu	1	0	0	0	0	0	1	0	0	0	2

Setelah mendapatkan nilai TF dan *df* dari semua *term*, maka langkah selanjutnya adalah menghitung nilai IDF. Berikut adalah contoh perhitungan IDF terhadap *term* “tolong”:

$$IDF = 1 + \log_{10} \left( 1 + \log_{10} \left( \frac{10}{2} \right) \right) = 1,23$$

Setelah menghitung nilai IDF, langkah selanjutnya adalah menghitung bobot dokumen ke-n. Berikut adalah contoh perhitungan bobot (*w*) dokumen 1 terhadap *term* “tolong”:

$$w = 1 * 1,23 = 1,23$$

Setelah mendapatkan nilai bobot, maka langkah selanjutnya adalah menghitung tingkat similaritas dokumen terhadap kata kunci. Di sini, penulis memilih kata “error”, “layan”, dan “verifikasi” sebagai kata kunci. Berikut adalah contoh perhitungan tingkat similaritas dokumen 1 terhadap kata kunci:

$$\text{Bobot } (w) \text{ D1} = 2,602 + 0 + 0 = 2,602$$

Setelah menyelesaikan perhitungan metode TF-IDF, tahap selanjutnya adalah melakukan perhitungan

metode *Cosine Similarity*, dilakukan dengan membandingkan dokumen ke dokumen.

Awal perhitungan metode *Cosine Similarity* mirip seperti perhitungan metode TF-IDF. Perbedaannya adalah metode *Cosine Similarity* hanya menghitung dua dokumen saja, berbeda dengan metode TF-IDF yang menghitung semua dokumen. Contoh perhitungan bobot antara D1 dengan D2 dapat dilihat pada Tabel 6.

Tabel 6. Bobot

D	Term	TF		df	D/df	IDF	W	
		D1	D2				D1	D2
D1	tolong	1	0	1	2	1,11	1,114	0
D1	benar	1	0	1	2	1,11	1,114	0
D1	aplikasi	3	0	1	2	1,11	3,343	0
D1	fitur	2	0	1	2	1,11	2,229	0
D1	error	2	0	1	2	1,11	2,229	0
D1	terima	1	0	1	2	1,11	1,114	0
D1	kode	1	0	1	2	1,11	1,114	0
D1	otp	1	0	1	2	1,11	1,114	0
D1	tunggu	1	0	1	2	1,11	1,114	0
D1	bantu	1	0	1	2	1,11	1,114	0
D1	nyaman	1	0	1	2	1,11	1,114	0
D2	daftar	0	1	1	2	1,11	0	1,114
D2	2	0	1	1	2	1,11	0	1,114
D2	akun	0	2	1	2	1,11	0	2,229
D2	beda	0	2	1	2	1,11	0	2,229
D2	kacau	0	1	1	2	1,11	0	1,114
D2	buka	0	1	1	2	1,11	0	1,114
D2	menu	0	1	1	2	1,11	0	1,114
D2	verifikasi	0	3	1	2	1,11	0	3,343
D2	cantun	0	1	1	2	1,11	0	1,114
D2	nomor	0	2	1	2	1,11	0	2,229
D2	muncul	0	1	1	2	1,11	0	1,114
D2	orang	0	1	1	2	1,11	0	1,114
D2	coba	0	1	1	2	1,11	0	1,114
D2	ktp	0	1	1	2	1,11	0	1,114
D2	awur	0	1	1	2	1,11	0	1,114
<b>Total</b>							<b>16,71</b>	<b>22,28</b>

Selanjutnya adalah menghitung bobot dokumen dengan persamaan seperti yang dapat dilihat pada Tabel 7.

Tabel 7. Bobot Dokumen

Bobot Dokumen				
D1 <sup>2</sup>	D2 <sup>2</sup>	D1*D2	D1 <sup>2</sup> *D1	D2 <sup>2</sup> *D2
1,242	0	0	1,384	0
1,242	0	0	1,384	0
11,175	0	0	37,356	0
4,967	0	0	11,068	0
4,967	0	0	11,068	0
1,242	0	0	1,384	0
1,242	0	0	1,384	0
1,242	0	0	1,384	0
1,242	0	0	1,384	0
1,242	0	0	1,384	0
1,242	0	0	1,384	0
0	1,242	0	0	1,384
0	1,242	0	0	1,384

0	4,967	0	0	11,068
0	4,967	0	0	11,068
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	11,175	0	0	37,356
0	1,242	0	0	1,384
0	4,967	0	0	11,068
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	1,242	0	0	1,384
0	1,242	0	0	1,384
<b>31,041</b>	<b>39,732</b>	<b>0</b>	<b>70,561</b>	<b>85,779</b>

Langkah selanjutnya adalah meng-hitung nilai CosSim, yakni tingkat kesamaan dokumen dengan *query*. Berikut adalah hasil perhitungan CosSim antara D1 dengan D2:

$$\text{CosSim} = \frac{0}{\sqrt{(70,561) * (85,779)}} = 0$$

Setelah menghitung nilai CosSim pada setiap pasang dokumen, langkah selanjutnya adalah melakukan per-hitungan metode MMR. Berikut adalah contoh perhitungan metode MMR terhadap dokumen 1 pada iterasi ke-1:

$$\text{MMR} = (0,5 * 2,602) - ((1 - 0,5) * 0,188) = 1,207$$

Dimana 0,5 adalah nilai parameter  $\lambda$  yang dipilih oleh penulis; 2,602 diambil dari tingkat similaritas D1 terhadap kata kunci, dan 0,188 merupakan nilai maksimal CosSim pada D1. Berikut adalah contoh perhitungan metode MMR terhadap dokumen 1 pada iterasi ke-2:

$$\text{MMR} = (0,5 * 1,207) - ((1 - 0,5) * 0,188) = 0,509$$

Iterasi ke-2 hingga ke-10 tidak menggunakan tingkat similaritas dokumen terhadap kata kunci seperti pada iterasi ke-1, melainkan menggunakan nilai MMR dari iterasi sebelumnya. Hasil perhitungan metode MMR dari iterasi pertama hingga iterasi terakhir dapat dilihat pada Tabel 8.

**Tabel 8.** MMR

Iterasi	D1	D2	D3	D4	D5
1	1,207	1,601	-	0,524	-
2	0,509	<b>0,683</b>	-	0,213	-
3	<b>0,160</b>	-	-	0,058	-
			0,078		0,107

4	-	-	-	-	-
			0,083	<b>0,020</b>	0,115
5	-	-	-	-	-
			0,086		0,119
6	-	-	-	-	-
			<b>0,088</b>		0,120
7	-	-	-	-	-
					<b>0,121</b>
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-

**Tabel 8.** Lanjutan MMR

Iterasi	D6	D7	D8	D9	D10
1	-	0,561	-	0,948	<b>2,709</b>
	0,072		0,076		
2	-	0,227	-	0,277	-
	0,108		0,113		
3	-	0,059	-	-	-
	0,126		0,132	0,059	
4	-	-	-	-	-
	0,135	0,024	0,142	0,227	
5	-	-	-	-	-
	0,140	<b>0,066</b>	0,147	0,311	
6	-	-	-	-	-
	0,142		0,149	0,353	
7	-	-	-	-	-
	0,143		0,150	0,374	
8	-	-	-	-	-
	<b>0,144</b>		0,151	0,384	
9	-	-	-	-	-
			<b>0,151</b>	0,390	
10	-	-	-	-	-
				<b>0,392</b>	

Nilai-nilai di dalam tabel adalah nilai MMR yang telah didapatkan. Nilai yang tulisannya ditebalkan merupakan nilai maksimal MMR dalam iterasi tersebut. Dokumen dimana nilai maksimal MMR didapat tidak akan digunakan dalam perhitungan iterasi selanjutnya, diberi tanda “-” dalam tabel.

Setelah seluruh proses perhitungan selesai, didapati 3 dokumen yang terpilih sebagai ringkasan, yaitu D10, D2, dan D1. Ketiga dokumen tersebut dipilih karena memiliki nilai MMR tertinggi dan bernilai positif. Berikut ringkasan ulasan aplikasi Mobile JKN yang dihasilkan:

“Ini bagaimana sih padahal sudah dikirim kode Verifikasi via SMS dan email dan sudah saya masukkan sesuai prosedur tetapi kok keterangan akun belum terverifikasi, Kalau memang belum terverifikasi, kode yang dikirim waktu pendaftaran untuk apa? Mana Pandawa JKN juga tidak ada menu buat *verifikasi virtual account*, kalau kayak begini mau dipakai untuk bersalin jadi ribet dan bikin pusing pengguna, anggaran saja miliaran tetapi pelayanan tidak memuaskan. Mendaftar dengan 2 akun berbeda, kacau sekali untuk buka menu

harus verifikasi tetapi saat ingin verifikasi yang dicantumkan nomor berbeda. Bagaimana bisa verifikasi kalau yang muncul nomor orang lain? Sudah coba buat akun baru dengan ktp lain sama saja mengawur. tolong dibenarkan lagi aplikasinya. setiap fitur pasti *banyak error*nya. Menerima kode otp saja harus menunggu lama banget. Aplikasi ini sangat membantu jika semua fitur aplikasi bisa digunakan dengan nyaman tetapi malah banyak *error*nya.”

Tahap selanjutnya adalah meng-hitung *confusion matrix*. Untuk pengujian, penulis telah mengumpulkan 3 responden. Tabel *Confusion Matrix* untuk membandingkan hasil ringkasan sistem dengan hasil ringkasan manual oleh responden 1 dapat dilihat pada Tabel 9.

**Tabel 9.** Confusion Matrix (1)

Jumlah kata yang sama	27
Banyak kata pada ringkasan sistem (S)	48
Banyak kata pada ringkasan manual (M)	69
Banyak kata pada ringkasan S & M	90

Berikut adalah hasil perhitungan *confusion matrix* terhadap hasil ringkasan manual dari responden 1:

$$\text{Accuracy} = \frac{27}{90} \times 100\% = 30,0\%$$

$$\text{Precision} = \frac{27}{48} \times 100\% = 56,3\%$$

$$\text{Recall} = \frac{27}{69} \times 100\% = 39,1\%$$

Tabel *Confusion Matrix* untuk membandingkan hasil ringkasan sistem dengan hasil ringkasan manual oleh responden 2 dapat dilihat pada Tabel 10.

**Tabel 10.** Confusion Matrix (2)

Jumlah kata yang sama	38
Banyak kata pada ringkasan sistem (S)	48
Banyak kata pada ringkasan manual (M)	89
Banyak kata pada ringkasan S & M	99

Berikut adalah hasil perhitungan *confusion matrix* terhadap hasil ringkasan manual dari responden 2:

$$\text{Accuracy} = \frac{38}{99} \times 100\% = 38,4\%$$

$$\text{Precision} = \frac{38}{48} \times 100\% = 79,2\%$$

$$\text{Recall} = \frac{38}{89} \times 100\% = 42,7\%$$

Tabel *Confusion Matrix* untuk membandingkan hasil ringkasan sistem dengan hasil ringkasan manual oleh responden 3 dapat dilihat pada Tabel 11.

**Tabel 11.** Confusion Matrix (3)

Jumlah kata yang sama	15
Banyak kata pada ringkasan sistem (S)	48
Banyak kata pada ringkasan manual (M)	40
Banyak kata pada ringkasan S & M	73

Berikut adalah hasil perhitungan *confusion matrix* terhadap hasil ringkasan manual dari responden 3:

$$\text{Accuracy} = \frac{15}{73} \times 100\% = 20,5\%$$

$$\text{Precision} = \frac{15}{48} \times 100\% = 31,3\%$$

$$\text{Recall} = \frac{15}{40} \times 100\% = 37,5\%$$

Berdasarkan ketiga hasil perhitungan *confusion matrix*, didapati rata-rata *Accuracy* sebesar 29,6%, rata-rata *Precision* sebesar 55,6%, dan rata-rata *Recall* sebesar 39,8%, dengan nilai tertinggi *Accuracy* sebesar 38,4%, *Precision* sebesar 79,2%, dan *Recall* sebesar 42,7%.

## 5. Penutup

Berdasarkan analisa yang dilakukan oleh penulis, penulis menarik kesimpulan bahwa penelitian ini mampu meng-identifikasi ulasan-ulasan yang relevan, berdasarkan pada kata kunci yang telah ditentukan. Metode TF-IDF dan *Cosine Similarity* tidak dapat menghasilkan ringkasan sehingga penulis menambahkan metode MMR. Mengenai evaluasi hasil ringkasan, semakin banyak jumlah kata yang terdapat dalam hasil ringkasan manual, maka semakin tinggi pula hasil perhitungan *confusion matrix*.

## 6. Daftar Pustaka

- [1] R. Kumalasari, “Opini: Fenomena Penghindaran Informasi pada Infodemik Covid-19 Melalui Media Sosial”, *Jurnal Sistem Informasi, Teknologi Informasi, dan Edukasi Sistem Informasi*, vol. 2, no. 2, pp. 112-118, 2021. doi: 10.25126/justsi.v2i2.54
- [2] M. S. Utomo, J. S. Wibowo, and E. N. Wahyudi, “Text Summarization pada Artikel Berita Menggunakan Vector Space Model dan Cosine Similarity”, *Dinamika Informatika : Jurnal Ilmiah Teknologi Informasi*, vol. 14, no. 1, pp. 11-24, 2022. doi: 10.35315/informatika.v14i1.9163
- [3] S. Chamira, “Implementasi Metode Text Mining Frequency-Invers Document Frequency (Tf-Idf) untuk Monitoring Diskusi Online”, *Journal of Informatics, Electrical and Electronics Engineering*, vol. 1, no. 3, pp. 97-102, 2022. doi: 10.47065/jieee.v1i3.353
- [4] A. Z. Z. Abidin and A. Sukmadinata, “Sistem Deteksi Kerusakan pada Sistem Operasi

- Menggunakan Metode TF-IDF dan Cosine Similarity”, *Jurnal Ilmiah Informatika*, vol. 8, no. 2, pp. 107–112, 2020. doi: 10.33884/jif.v8i02.1968
- [5] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, “Movie Recommendation System Using Cosine Similarity and KNN”, *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 5, pp. 556-559, 2020. doi: 10.35940/ijeat.E9666.069520
- [6] A. Kurniawan and M. I. Humaidy, “Penerapan Algoritma Maximum Marginal Relevance dalam Pe-ringkasan Teks secara Otomatis”, *Bulletin of Data Science*, vol. 1, no. 2, pp. 49-56, 2022.
- [7] Y. A. Kresna, I. Cholissodin, and I. Indriati, “Peringkasan Teks Menggunakan Metode Maximum Marginal Relevance terhadap Artikel Berita Terkait COVID-19”, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 9, pp. 3901-3907, 2021.
- [8] M. Ramezani, M. S. Shahryari, A. R. Feizi-Derakhshi, and M. R. Feizi-Derakhshi, “Unsupervised Broadcast News Summarization; a Comparative Study on Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA)”, *28th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1-7, 2023. doi: 10.1109/CSICC58665.2023.10105403.
- [9] D. Sari, D. Hirdanti, N. S. Nasution, and F. P. Gurning, “Systematic Literatur Review: Analisis Manfaat Digital Mobile JKN dalam Pembiayaan Kesehatan”, *ZAHRA: Journal of Health and Medical Research*, vol. 4, no. 1, pp. 1-9, 2024.
- [10] A. Z. Z. Abidin and E. Nurjanah, “Sistem Peringkasan Teks Otomatis Multi Dokumen Kliping Artikel Berita Gempa Menggunakan Metode TF-IDF”, *Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, pp. 52-69, 2020. doi: 10.47561/a.v13i1.166
- [11] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, “Improving Text Pre-processing for Student Complaint Document Classification Using Sastrawi”, *IOP Conference Series: Materials Science and Engineering*, vol. 871, no. 1, pp. 1-6, 2020. doi: 10.1088/1757-899X/874/1/012017
- [12] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest”, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 9, pp. 4305-4313, 2022.
- [13] Halimah, S. Agustian, and S. Ramadhani, “Peringkasan Teks Otomatis (Automated Text Summarization) pada Artikel Berbahasa Indonesia Menggunakan Algoritma Lexrank”, *Jurnal Computer Science and Information Technology (CoSciTech)*, vol. 3, no. 3, pp. 371-381, 2022. doi: 10.37859/coscitech.v3i3.4300
- [14] Y. Findawati and M. A. Rosid, *Buku Ajar Text Mining*. Sidoarjo: UMSIDA PRESS, 2021
- [15] D. Septiani and I. Isabela, “Analisis Term Frequency Inverse Document Frequency (TF-IDF) dalam Temu Kembali Informasi pada Dokumen Teks”, *Sistem dan Teknologi Informasi Indonesia (SINTESIA)*, vol. 1, no. 2, pp. 81-88, 2022.
- [16] A. Apriani, H. Zakiyudin, and K. Marzuki, “Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta”, *Jurnal Bumigora Information Technology (BITE)*, vol. 3, no. 1, pp. 19-27, 2021. doi: 10.30812/bite.v3i1.1110
- [17] D. A. Louis, S. Rostianingsih, and L. W. Santoso, “Implementasi Text Summarization pada Review Aplikasi Super di Google Play Store Menggunakan Metode Maximum Marginal Relevance”, *Jurnal Infra*, vol. 10, no. 2, pp. 179-183, 2022.
- [18] V. S. Ginting, K. Kusriani, and E. Taufiq, “Implementasi Algoritma C4.5 untuk Memprediksi Keterlambatan Pembayaran Sumbangan Pembangunan Pendidikan Sekolah Menggunakan Python”, *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 36-44, 2020. doi: 10.35585/inspir.v10i1.2535