Peringkasan Teks Otomatis pada Dokumen Berita Berbahasa Indonesia dengan Metode Maximum Marginal Relevance dengan Improved Sqrt-Cosine Similarity

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Nama: Kevin Aryo Wicaksono
NIM: 175150201111039

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021

PENGESAHAN

Peringkasan Teks Otomatis pada Dokumen Berita Berbahasa Indonesia dengan Metode Maximum Marginal Relevance dengan Improved Sqrt-Cosine Similarity

SKRIPSI

Diajukan untuk memenuhi Sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Kevin Aryo Wicaksono
NIM: 175150201111039

Skripsi ini telah diuji dan dinyatakan lulus pada
02 Juli 2021
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Puta Pandu Adkara, S.Kom., M.Kom.	Prima Zulvarina, S.S., M.Pd.
NIP: 198507252008121002	NIK:2016078507012001

Mengetahui,
Ketua Jurusan Teknik Informatika

Achmad Basuki, S.T., M.MG., Ph.D.
NIP: 19741182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi in tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Juni 2021

Kevin Aryo Wicaksono

NIM: 175150201111039

ABSTRAK

Kevin Aryo Wicaksono, Peringkasan Teks Otomatis untuk Dokumen Berita Berbahasa Indonesia dengan Metode *Maximum Marginal Relevance* dengan *Improved Sqrt-Cosine Similarity*

Pembimbing: Putra Pandu Adikara, S.Kom., M.Kom. dan Prima Zulvarina, S.S., M.Pd.

Peringkasan teks otomatis pada dasarnya merupakan proses menyederhanakan suatu teks dengan cara membuat ringkasan yang mewakilkan isi dari teks tersebut. Sumber informasi utama tentang kejadian yang sudah terjadi atau sedang terjadi adalah berita, berita seringkali diharapkan dengan isi yang terlalu panjang sehingga tidak efektif bagi pembacanya. Hal tersebut menyebabkan kurang efisiennya waktu yang diperlukan untuk mendapatkan informasi dari suatu teks berita, sehingga diperlukan adanya ringkasan yang mewakili isi dari teks berita tersebut dalam bentuk yang lebih sederhana sehingga lebih mudah dan cepat dipahami. Penelitian ini diawali dengan melakukan preprocessing pada teks berita yang akan diringkas. Proses selanjutnya adalah mencari bobot masing-masing token dengan metode pembobotan TFIDF, dengan memanfaatkan nilai TFIDF maka dapat dicari similarity antar kalimat dengan metode ISC Similarity. Ringkasan dibuat berdasarkan nilai MMR dari masing-masing kalimat mulai dari yang tertinggi, nilai MMR yang tinggi menandakan bahwa kalimat tersebut mirip dengan query yang digunakan. Query dalam pembuatan ringkasan didapatkan dari judul berita yang diuji. Hasil pengujian ringkasan oleh sistem dengan metode ROUGE-L mendapatkan nilai rata-rata tertinggi pada ringkasan dengan persentase 10% dengan nilai rata-rata precision 0,8743, recall 0,7678, dan f-measure 0,7678.

Kata kunci: ringkasan, berita, maximum marginal relevance, isc similarity

ABSTRACT

Kevin Aryo Wicaksono, Automatic Teks Summarization for Indonesian Text News Using Maximum Marginal Relevance with Improved Sqrt-Cosine Similarity

***Advisors*: Putra Pandu Adikara, S.Kom., M.Kom. and Prima Zulvarina, S.S., M.Pd.**

Automatic text summarization in general can be defined as a process to simplify a text by making a summary that represent the contents of the text. In real life, the main source for knowing an event that is happening or already happened is news, but frequently news' text presented in a long story. A long news' text often led to an efficiency issue in time used for understanding the content of the news, as for the solution a text summary is needed to save more time used in understanding the content of a news. This research began with implementing preprocessing to the news' text that is going to summarized. The next process is implementing TFIDF weighting method to get the weight of each token, with ISC Similarity method the TFIDF value was then used to get the similarity value between every sentence in the text. The creation of the text summary is based on MMR value, higher the value mean that the sentence is most likely represent the content of the text. The title of the text is used as the query for text summarization process. The highest average test results of this research using the ROUGE-L method is found at 10% summary with the value of precision 0,8743, recall 0,7678, and f-measure 0,7678.

Keyword: summary, news, maximum marginal relevance, isc similarity

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada era modern, jumlah dokumen tekstual meningkat secara pesat begitu juga dengan kebutuhan informasi. Sumber utama dari sebuah informasi adalah berita, artikel-artikel berita sudah banyak yang diunggah di situs-situs internet. Proses pencarian dan pengambilan informasi dari teks berita dituntut selalu cepat dan akurat, hal tersebut dapat diatasi dengan cara membuat ringkasan teks sehingga pemerolehan informasi dapat dilakukan dengan efektif dan efisien. Hal tersebut membawa kita ke dalam masalah baru yaitu, bagaimana bisa membuat sebuah ringkasan dari suatu dokumen berbasis teks dengan cepat dan akurat. Untuk membuat ringkasan yang bisa mewakili isi dokumen kita harus terlebih dahulu membaca teks tersebut secara keseluruhan, oleh karena itu dibutuhkan sebuah solusi yang dapat meringkas teks secara otomatis.

Penelitian yang dilakukan Widyassari (2020) menyatakan bahwa ringkasan merupakan sebuah penyajian singkat dari sebuah dokumen teks asli yang masih mempertahankan informasi-informasi penting dari dokumen teks aslinya. Terdapat dua macam ringkasan yaitu, ringkasan abstraktif dan ringkasan ekstraktif. Ringkasan abstraktif adalah sebuah ringkasan yang terbentuk dari kalimat-kalimat baru yang tidak terdapat dalam teks asli namun tetap dapat mencakup informasi dari teks aslinya, sedangkan ringkasan ekstraktif merupakan sebuah ringkasan yang terdiri dari kalimat-kalimat yang telah diekstraksi dari dokumen teks aslinya.

Peringkasan teks otomatis secara abstraktif dinilai lebih sulit untuk diterapkan secara otomatis, maka banyak penelitian lebih memilih peringkasan teks otomatis secara ekstraktif. Peringkasan teks otomatis secara ekstraktif dapat diartikan sebagai sebuah pengklasifikasian kalimat-kalimat dari teks asli menjadi dua kelompok: ringkasan dan bukan ringkasan (Anh, et al., 2019). Keunggulan dari peringkasan teks otomatis secara ekstraktif adalah lebih cepat dan mudah dibandingkan secara abstraktif, cara ini juga terbukti memberi hasil yang akurat karena pembaca dapat secara langsung mengerti apa yang dimaksud oleh teks aslinya. Kerugiannya adalah peringkasan teks otomatis secara ekstraktif menghasilkan ringkasan yang cenderung panjang jika dibandingkan dengan ringkasan yang dibuat manusia, kurangnya kesinambungan antar kalimat dan juga terdapat redundansi pada beberapa kalimat ringkasan (El-Kassas, et al., 2020).

Ada beberapa metode yang bisa diterapkan pada sistem peringkasan teks otomatis seperti Support Vector Machine (SVM), BM25, Maximum Marginal Relevance (MMR), dan Non-Negative Matrix Factorization (NMF). Pada penelitian sebelumnya tentang peringkasan teks otomatis menggunakan metode NMF terhadap 100 dokumen Bahasa Indonesia, evaluasi menunjukkan ringkasan sistem mempunyai rata-rata presisi dan *recall* masing-masing 0,19724 dan 0,34085 (Ridok, 2014). Penelitian selanjutnya pada topik peringkasan teks

otomatis menggunakan metode SVM dengan 10 data uji, menghasilkan nilai *recall* sebesar 0,428, nilai *precision* sebesar 0,604, dan nilai *f-measure* sebesar 0,496 (Somantri, et al., 2018). Adapun penelitian mengenai peringkasan teks otomatis menggunakan metode MMR dengan 20 data uji menghasilkan rata-rata nilai *precision* sebesar 0,624 dan *recall* sebesar 0,736 (Indriani, 2014). Penelitian lain yang juga menggunakan metode MMR untuk peringkasan teks otomatis pada 30 berita mendapatkan hasil nilai rata-rata precision 0,70, recall 0,75, f-measure 0,70, dan akurasi 74,17 (Saraswati, et al., 2018).

Hasil evaluasi dari metode MMR dinilai cukup bagus untuk melakukan peringkasan teks otomatis sehingga peneliti memutuskan untuk menggunakan metode MMR untuk ringkasan ekstraktif pada penelitian ini. Pada dua penelitian dengan metode yang sama sebelumnya, perhitungan similarity antar kalimat menggunakan metode cosine similarity yang merupakan fungsi untuk menghitung kemiripan antara dua vektor (Saraswati, et al., 2018). Dalam penelitian ini peneliti mencoba meningkatkan nilai hasil evaluasi dengan mengganti metode cosine similarity dengan metode Improved Sqrt-Cosine Similarity. Keputusan tersebut diambil karena dalam penelitian yang membahas perbandingan metode penghitung kemiripan teks, didapatkan hasil rata-rata akurasi Improved Sqrt-Cosine sebesar 0,3563, Cosine sebesar 0,3370 dan Gaussian sebesar 0,2949 (Sohangir & Wang, 2017).

1.2 Rumusan Masalah

Berdasar uraian latar belakang yang sudah dirumuskan, dapat dirumuskan 2 masalah sebagai berikut.

1. Bagaimana perancangan peringkasan teks otomatis untuk suatu dokumen teks menggunakan metode MMR?
2. Bagaimana hasil evaluasi dari penerapan metode MMR dan Improved Sqrt-Cosine Similarity untuk peringkasan teks otomatis?

1.3 Tujuan

Tujuan dari penelitian peringkasan teks otomatis ini adalah sebagai berikut.

1. Menjelaskan penerapan metode MMR dalam melakukan peringkasan teks otomatis untuk suatu dokumen teks, sehingga mempermudah dan mempercepat proses pencarian informasi
2. Mendeskripsikan hasil evaluasi dari penerapan metode MMR dan *Improved Sqrt-Cosine Similarity* untuk peringkasan teks otomatis

1.4 Manfaat

Manfaat yang bisa diambil dari penelitian ini adalah sebagai berikut.

1. Mempermudah dan mempercepat proses pemerolehan informasi dari suatu dokumen dengan ringkasan yang dihasilkan

2. Mengetahui hasil evaluasi dari penerapan metode MMR dan Improved Sqrt-Cosine Similarity dalam membuat ringkasan

1.5 Batasan Masalah

Batasan masalah yang ditetapkan dalam penelitian ini sebagai berikut.

1. Penelitian ini hanya terbatas pada dokumen teks berbahasa Indonesia.
2. Dokumen berita yang digunakan berjumlah 10 berita yang diambil dari beberapa situs portal berita.
3. *Query* yang digunakan berasal dari judul dari teks berita sebagai data uji.

1.6 Sistematika Pembahasan

Penelitian mengenai peringkasan secara otomatis berita berbahasa Indonesia ini ditulis berdasarkan susunan sistematika pembahasan yang terdiri dari beberapa bab yaitu:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang dari penelitianini, rumusan masalah yang diangkat dalam penelitian ini, tujuan dan manfaat yang bisa diambil, batasan masalah yang ditetapkan dan sistematika pembahasan pada penelitian peringkasan teks otomatis pada dokumen berita berbahasa Indonesia dengan metode *Maximum Marginal Relevance* dengan *Improved Sqrt-Cosine Similarity*.

BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepustakaan terdiri dari kajian pustaka dan dasar teori. Dalam kajian pustaka akan berisi tentang referensi dari penelitian-penelitian yang sudah dilakukan sebelumnya yang berhubungan dengan peringkasan teks otomatis pada dokumen berita berbahasa Indonesia dengan metode Maximum Marginal Relevance dengan Improved Sqrt-Cosine Similarity. Dasar teori akan berisi uraian penjelasan mengenai teori, konsep, metode yang digunakan, atau sistem dari penelitian.

BAB 3 METODOLOGI

Bab metodologi penelitian terdiri beberapa bagian yaitu, tipe penelitian, strategi penelitian yang digunakan, lokasi dilaksanakannya penelitian, jadwal penelitian, metode yang digunakan untuk mengumpulkan data, teknik untuk analisis hasil pengujian, peralatan pendukung, dan perancangan algoritme.

BAB 4 PERANCANGAN

Bab perancangan terdiri dari perancangan algoritme dari metode yang digunakan, manualisasi , perancangan uji coba serta evaluasi sistem peringkasan teks otomatis pada dokumen berita berbahasa Indonesia dengan metode Maximum Marginal Relevance dengan Improved Sqrt-Cosine Similarity.

BAB 5 IMPLEMENTASI

Bab implementasi terdiri dari implementasi/penerapan sistem peringkasan teks otomatis pada dokumen berita berbahasa Indonesia dengan metode Maximum Marginal Relevance dengan Improved Sqrt-Cosine Similarity.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan membahas tentang hasil pengujian penelitian dan hasil analisa terhadap hasil dari penelitian yang dilakukan.

BAB 7 PENUTUP

Bab penutup terdiri dari kesimpulan dan saran untuk penelitian selanjutnya. Kesimpulan merupakan rangkuman hasil penelitian yang didapatkan berdasarkan tujuan yang dapat menjawab rumusan masalah dari penelitian. Saran adalah masukan dari penulis untuk penelitian selanjutnya untuk dapat memperbaiki dan mengembangkan penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepustakaan tersusun atas kajian pustaka dan uraian serta penjelasan mengenai teori yang digunakan dalam penelitian ini. Bagian kajian pustaka dalam bab ini berisi referensi dari penelitian-penelitian sebelumnya dengan topik yang berhubungan.

2.1 Kajian Pustaka

Terdapat beberapa penelitian yang dilakukan sebelumnya mengenai peringkasan teks otomatis dengan berbagai metode dan pendekatan. Penelitian pertama mengulas tentang teknik dan metode yang digunakan dalam peringkasan teks otomatis. Terdapat dua jenis ringkasan yaitu ekstraktif dan abstraktif, ringkasan ekstraktif merupakan ringkasan yang dibentuk dari beberapa kalimat yang diambil langsung dari teks asli sedangkan ringkasan abstraktif lebih kompleks karena terbentuk oleh kalimat-kalimat baru yang tidak berasal dari teks asli (Widyassari, et al., 2020). Dalam penelitian selanjutnya yang masih membahas teknik dan metode peringkasan teks otomatis, disebutkan juga terdapat ringkasan campuran. Ringkasan campuran merupakan kombinasi dari ringkasan ekstraktif dan abstraktif yang mana memberi keuntungan dari kedua jenis ringkasan baik ekstraktif maupun abstraktif (El-Kassas, et al., 2020).

Penelitian ketiga menggunakan metode Maximum Marginal Relevance untuk peringkasan sinopsis buku berbahasa Indonesia. Dalam penelitian ini menggunakan 20 data uji coba yang menghasilkan nilai rata-rata *precision* dan *recall* sebesar 62,4% dan 73,6%. Hasil ringkasan yang didapat dari pnelitian ini menerapkan metode pembobotan TF-IDF (Indriani, 2014).

Penelitian keempat meggunakan metode Maximum Marginal Relevance untuk peringkasan teks berita otomatis. Data yang digunakan sebagai data uji dalam penelitian ini diambil dari situs berita online Tempo Interaktif yang dipilih secara acak dan diunduh dari bulan Januari 2009 hingga Juni 2009 sebanyak 30 berita. Hasil yang didapatkan dari penelitian ini adalah rata-rata nilai *recall* sebesar 60%, *precision* sebesar 77% dan *f-measure* 66% (Mustaqhfiri, et al., 2011).

Penelitian kelima tentang peringkasan teks otomatis dengan metode MMR pada hasil sistem temu kembali informasi. Metode MMR digunakan untuk menghitung *similarity* antar kalimat dengan kalimat lainnya dan antara kalimat dengan *query*. Dalam penelitian ini menggunakan pengujian *precision@k* dan *precision, recall, f-measure* dan akurasi, yang mana didapatkan hasil *precision@k* sebesar 0,96 untuk hasil system temu kembali dan hasil rata-rata *precision sebesar 0,70, recall sebesar 0,75, f-measure sebesar 0,70* dan akurasi sebesar 74,17 (Saraswati, et al., 2018).

Penelitian keenam tentang perbandingan metode penghitung kemiripan, yang membandingkan metode Improved Sqrt-Cosine Similarity, Cosine Similarity dan Gaussian-based Similarity. Pada penerapan kasus klasifikasi dokumen

didapatkan nilai akurasi ISC sebesar 0,7079, Cosine sebesar 0,6476 dan Gaussian sebesar 0,4606. Kemudian pada kasus klustering dokumen didapat hasil nilai akurasi ISC sebesar 0,3354, Cosine sebesar 0,3115 dan Gaussian sebesar 0,3005 (Sohangir & Wang, 2017).

Dari hasil peneltian sebelumnya yang dikaji, dapat diambil kesimpulan bahwa metode Maximum Marginal Relevance (MMR) memiliki hasil yang bagus untuk peringkasan teks otomatis. Perhitungan similarity dengan menggunakan metode Improved Sqrt-Cosine Similarity juga terbukti mendapat hasil yang paling baik diantara metode-metode penghtiung similarity lainnya yang diuji dalam penelitian oleh Sohangir dan Wang (2017). Hal tersebut mendasari keputusan penulis untuk menggunakan kedua metode tersebut dalam penelitian ini.

2.2 Dasar Teori

2.2.1 Peringkasan Teks Otomatis

Peringkasan Teks Otomatis adalah pembuatan sebuah ringkasan/rangkuman dari satu atau banyak sumber teks secara otomatis dengan bantuan sistem yang dijalankan pada komputer. Cara kerja peringkasan teks otomatis secara sederhana adalah sistem diberi *input*(masukan) berupa teks, kemudian akan dilakukan proses peringkasan oleh sistem, dan sistem akan menghasilkan *output*(keluaran) berupa ringkasan yang mencakup poin-poin penting dari sumber teks. Menurut Widyassari (2020) pada peringkasan teks terdapat dua jenis pendekatan yaitu ringkasan ekstraktif yang merupakan sebuah ringkasan dengan terdiri dari kalimat-kalimat yang diambil secara langsung dari teks sumber, sedangkan ringkasan abstraktif merupakan ringkasan yang terdiri dari kalimat-kalimat baru yang disebut paraphrase yang dibuat menggunakan kata-kata yang tidak ada pada teks sumber. Menurut Wafaa S. El-Kassas (2020), ringkasan memiliki 3 jenis pendekatan, selain ekstraktif dan abstraktif juga disebutkan pendekatan secara *hybrid*(campuran) yang merupakan penggabungan antara ekstraktif dan abstraktif.

2.2.2 Berita

Dalam pengertian umumnya berita adalah sebuah informasi yang sifatnya fakta tentang kejadian yang sudah atau tengah terjadi yang disampaikan dengan perantara media baik elektronik maupun cetak. Berita merupakan sumber informasi tercepat dari suatu kejadian yang bersifat fakta penting dan menarik bagi sebagian besar pembaca (M. Romli, 1999). Dalam penelitian ini berita yang digunakan merupakan berita berbahasa Indonesia pada media elektronik.

2.2.3 Text Preprocessing

Proses yang dilakukan untuk menyiapkan data berupa teks agar menjadi data yang dapat diolah di tahapan berikutnya disebut *text preprocessing*. Tahap ini sangat diperlukan karena tidak semua bagian dalam teks dapat dimanfaatkan dalam proses *text mining* sehingga perlu dilakukan *preprocessing* untuk mendapatkan token/term yang bisa mewakili isi teks untuk dapat diolah.

Tahapan dalam *preprocessing* yaitu *case folding, cleaning,* tokenisasi, *stopword removal/filtering*, dan *stemming.*

2.2.3.1 Case Folding

Case Folding merupakan proses mengubah kata yang ada pada kalimat menjadi huruf kecil (Khairunnisa, et al., 2021). Pada penelitian ini, peneliti memilih untuk menyamakan menjadi huruf kecil semua (*lowercase*).

2.2.3.2 Cleaning

Cleaning adalah sebuah proses menghilangkan angka, tanda baca, dan simbol dari teks berita (Khairunnisa, et al., 2021). Penghilangan ini dilakukan karena angka dan tanda baca dianggap tidak memiliki pengaruh terhadap informasi yang terdapat dalam teks berita. Pada dokumen berita salah satu contoh yang dihilangkan adalah tanggal.

2.2.3.3 Tokenisasi

Tokenisasi adalah proses memisahkan teks menjadi beberapa bagian yang dinamakan dengan token. Kumpulan token yang memiliki karakter yang sama disebut type. Type yang sudah dinormalisasikan disebut dengan term (Budiman & Widjaja, 2020). Dalam penelitian ini dilakukan dua tahap tokenisasi, yaitu memecah dokumen menjadi kalimat dan memecah kalimat menjadi kata.

2.2.3.4 Filtering

Filtering merupakan langkah selanjutnya yang dilakukan setelah tokenisasi. *Filtering* atau bisa disebut juga sebagai tahap *stop word removal* yaitu proses untuk menghilangkan beberapa kata yang tidak penting dalam kalimat dan hanya menyimpan kata yang dianggap penting saja (Prabowo, et al., 2016). Stop word berisi kata-kata yang sering dijumpai di dalam teks tanpa mewakili isi teks tersebut seperti kata ganti, kata hubung, dan lain-lain. *Stopword* yang digunakan dalam peneitian ini diambil dari *library Sastrawi* versi 1.0.1.

2.2.3.5 Stemming

Stemming adalah proses mengubah suatu kata menjadi bentuk kata dasarnya. Tujuan dilakukannya stemming adalah untuk menghilangkan imbuhan, untuk mengurangi jumlah kata, untuk menyamakan kata dengan kata dasarnya, untuk menghemat waktu dan memori (Vijayarani, et al., 2020). Dalam penelitian ini proses stemming dilakukan menggunakan *library* Sastrawi untuk menghilangkan imbuhan yang ada pada kata/term sehingga menjadi bentuk kata dasarnya. Tahap stemming dalam penelitian ini dilakukan dengan menggunakan library Sastrawi versi 1.0.1.

2.2.4 TF-IDF (*Term Frequency-Inverse Document Frequency*)

Untuk dapat digunakan dalam proses selanjutnya setelah *preprocessing* maka, data harus diubah menjadi bentuk nuumerik dengan pembobotan TF-IDF. Metode Term Frequency Invers Document Frequency adalah metode yang

berfungsi untuk menentukan tingkat keterhubungan antar kata pada dokumen dengan memberi bobot pada setiap kata. Metode TF-IDF memanfaatkan dua konsep yaitu *term frequency* (TF) dan *inverse document frequency* (IDF) (Herwijayanti, et al., 2018).

Perhitungan TF dapat dilakukan dengan Persamaan 2.1.

$$TF = \begin{cases} 1 + \log(f_{t,d}), f_{t,d} > 0 \\ 0, f_{t,d} = 0 \end{cases} \quad (2.1)$$

Perhitungan IDF dapat dilakukan dengan Persamaan 2.2.

$$IDF = \log\left(\frac{N}{df}\right) \quad (2.2)$$

Keterangan:

$f_{t,d}$ = jumlah *term t* pada dokumen *d*

N = total jumlah dokumen

df = banyaknya dokumen yang mengandung term

2.2.5 Improved Sqrt-Cosine Similiarity

Improved Sqrt-Cosine Similarity adalah sebuah penyempurnaan dari metode Sqrt-Cosine Similarity. Sqrt-Cosine Similarity merupakan metode penghitung kemiripan yang mencoba menggunakan keuntungan dari Hellinger Distance. Daripada menggunakan normalisasi L_1, metode Improved Sqrt-Cosine Similarity menggunakan akar dari normalisasi L_1 (Sohangir & Wang, 2017). Persamaan *Improved Sqrt-Cosine Similarity* dapat dilihat pada Persamaan 2.3.

$$ISC(x, y) = \frac{\sum_{i=1}^{m} \sqrt{x_i y_i}}{\sqrt{(\sum_{i=1}^{m} x_i)} \sqrt{(\sum_{i=1}^{m} y_i)}} \quad (2.3)$$

Keterangan:

x = vektor

y = vektor

x_i = bobot term *i* pada blok x_i

y_i = bobot term *i* pada blok y_i

i = banyaknya *term* dalam suatu kalimat

m = jumlah vektor

2.2.6 Maximum Marginal Relevance

Metode *Maximum Marginal Relevance* (MMR) adalah suatu metode yang digunakan untuk membuat ringkasan pada dokumen tunggal atau multi dokumen. Metode MMR meringkas suatu dokumen dengan cara menghitung *siimilarity* antara teks dengan *query* yang diberikan (Saraswati, et al., 2018). Pada peringkasan teks otomatis menggunakan metode MMR, dilakukan proses

pemecahan dokumen menjadi kalimat dan dikelompokkan berdasarkan kalimat yang mirip dan kalimat yang tidak mirip dengan *query*. MMR digunakan dengan mengolah matriks *similiarity* kalimat untuk memberi peringkat pada kalimat-kalimat sebagai tanggapan pada *query* yang diberikan (Mustaqhfiri et al, 2011). Peringkasan *small* dokumen seperti berita atau sinopsis umumnya menggunakan nilai parameter λ = 0,7 atau λ = 0,8, karena akan menghasilkan ringkasan yang baik (Indriani, 2014). Perhitungan MMR dinyatakan dalam Persamaan 2.4.

$$MMR = argmax\ [\lambda * Sim1(S_i, Q) - (1 - \lambda) * \max Sim2(S_i, S')] \quad (2.4)$$

Keterangan:

S_i = vektor bobot kata yang menjadi kandidat

S' = vektor bobot kata selain kandidat

λ = parameter yang memengaruhi tingkat relevansi

Q = vektor bobot *query*

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

$Sim2(S_i,S')$ = nilai *simiiarity* antar kalimat ke-*i* dengan kalimat lainnya

2.2.7 Evaluasi

Metode Recall-Oriented Understudy for Gisting Evaluation (ROUGE) dipilih sebagai metode evaluasi untuk hasil pengujian dari penelitian ini. Proses evaluasi dalam penelitian ini dilakukan dengan membandingkan ringkasan yang dibuat oleh sistem dengan ringkasan oleh pakar secara manual. Ada 5 macam metode ROUGE yang ada yaitu, ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, dan ROUGE-SU, namun yang digunakan dalam penelitian ini hanya ROUGE-L saja. ROUGE-L merupakan salah satu jenis ROUGE yang menggunakan Longest Common Subsequence (LCS) untuk menghitung nilai *precision*, *recall*, dan *f-measure* (Lin, 2004).

Precision adalah perbandingan antara jumlah dari keseluruhan dokumen yang ditampilkan. *Recall* merupakan perbandingan antara banyaknya dokumen relevan yang ditampilkan dengan jumlah keseluruhan dokumen relevan (Indriani, 2014). *F-measure* merupakan perhitungan evaluasi dalam *information retrieval* yang memanfaatkan nilai *recall* dan *precision* (Erwin, et al., 2019). Persamaan *precision*, *recall,* dan *f-measure* ditampilkan pada Persamaan 2.5, Persamaan 2.6, dan Persamaan 2.7.

a. *Precision*

$$precision = \frac{LCS(X,Y)}{m} \quad (2.5)$$

b. *Recall*

$$recall = \frac{LCS(X,Y)}{n} \quad (2.6)$$

c. *F-measure*

$$f - measure = \frac{(1+\beta^2) \times Precision \times Recall}{Recall + \beta^2 \times Precision} \quad (2.7)$$

Keterangan:

$LCS(X,Y)$ = nilai LCS dari vektor *X* dan *Y*

β = sebuah parameter yang dalam *Document Understanding Conference* (DUC) memiliki nilai yang tinggi (8) (Lin, 2004).

m = panjang vektor X

n = panjang vektor Y

Persamaan untuk menghitung nilai Longest Common Subsequence (LCS) akan ditampilkan pada Persamaan 2.8.

$$LCS[i,j] = \begin{cases} 0 & if\ i = 0\ or\ j = 0 \\ LCS[i-1, j-1] + 1 & if\ X[i] = Y[j] \\ \max\{LCS[i-1,j], LCS[i,j-1]\} & if\ X[i] \neq Y[j] \end{cases} \quad (2.8)$$

Contoh penerapan dari algoritma LCS dengan *sequence-1* = "ABCDAF" dan *sequence-2* = "ACBCF" akan ditampilkan pada Tabel 2.1.

Tabel 2.1 Contoh Penerapan Algoritma LCS

		A	B	C	D	A	F
	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1
C	0	1	1	2	2	2	2
B	0	1	2	2	2	2	2
C	0	1	2	3	3	3	3
F	0	1	2	3	3	3	4

Berdasarkan Tabel 2.1 dapat diambil kesimpulan bahwa nilai LCS dari *sequence-1* dan *sequence-2* adalah 4 dengan susunan huruf "ABCF". Adapun langkah-langkah yang diterapkan adalah sebagai berikut (Saleh, 2020).

1. Cari keberadaan kata pada tiap baris, jika tidak ketemu ubah cara baca.
2. Cocokan kata pada tiap baris, jika tidak ketemu cocokan kata tiap kolom, jika tidak ketemu ubah cara baca.
3. Cocokan kata pada tiap kolom, jika tidak ketemu cocokan kata secara diagonal dari kiri bawah ke kanan atas, jika tidak ketemu ubah cara baca.
4. Cocokan kata secara diagonal dari kiri bawah ke kanan atas, jika tidak ketemu lakukan pencocokan kata ke diagonal kebalikannya $\geq$ Panjang kata yang dicari, jika tidak ketemu ubah cara baca.
5. Cocokan kata secara diagonal dari kiri atas ke kanan bawah, bawah yang panjang diagonalnya > panjang kata yang dicari.

BAB 3 METODOLOGI

Dalam bab ini akan dijelaskan langkah-langkah yang digunakan oleh penulis dalam melaksanakan penelitian ini.

3.1 Tipe Penelitian

Penelitian ini merupakan penelitian nonimplementatif-analitik yang artinya produk yang dihasilkan dari penelitian ini berupa hasil analisis yang berhubungan sesuai dengan topik yang diteliti. Pada penelitian ini, metode yang dipakai untuk menghasilkan suatu produk ringkasan teks adalah metode pemeringkatan kemiripan untuk mencari kalimat dengan kemiripan tertinggi dengan *query* sehingga bisa menghasilkan ringkasan yang relevan.

3.2 Strategi Penelitian

Strategi penelitian yang dilakukan untuk pada penelitian ini adalah penelitian eksperimen. Penelitian eksperimen merupakan penelitian yang berusaha menentukan apakah suatu perlakuan mempengaruhi hasil sebuah penelitian. Eksperimen merupakan salah satu prosedur dimana terdapat satu atau lebih faktor yang bisa dimanipulasi dengan syarat semua faktor tersebut konstan (Siyoto & Sodik, 2015). Metode yang digunakan dalam penelitian ini bertujuan untuk mencari kalimat yang paling mirip dengan *query* yang ada sehingga dari kalimat tersebut dapat disusun untuk menjadi sebuah ringkasan.

3.3 Lokasi Penelitian

Penelitian ini akan dilaksanakan di laboratorium riset Fakultas Ilmu Komputer Universitas Brawijaya Kota Malang, Jawa Timur.

3.4 Metode Pengumpulan Data

Metode pengumpulan data merupakan cara yang diterapkan penulis untuk mengumpulkan semua data yang nantinya akan digunakan dalam penelitian ini. Data yang digunakan pada penelitian ini diambil dari beberapa situs berita online berbahasa Indonesia. Penulis akan mengambil judul dan isi teks berita yang kemudian akan disimpan dalam bentuk *plain text*. Data yang digunakan dibatasi dengan hanya menggunakan berita berbahasa Indonesia dengan memanfaatkan judul berita sebagai *query*, teks berita yang digunakan berjumlah 10 berita. Pengujian akan dilakukan dengan cara membandingkan ringkasan yang dibuat oleh sistem dengan ringkasan yang dibuat secara manual oleh pakar. Pakar yang membuat ringkasan manual dalam penelitan ini adalah Ibu Prima Zulvarina, S.S., M.Pd yang merupakan dosen Bahasa Indonesia di Fakultas Ilmu Komputer Universitas Brawijaya.

3.5 Teknik Analisis Data

Teknik analisis data pada hasil peringkasan otomatis berupa pengujian hasil menggunakan metode ROUGE-L dengan menghitung nilai *precision, recall,* dan *f-measure*. Hasil penelitian akan diuji tingkat relevansinya antara ringkasan manual dan ringkasan oleh sistem.

3.6 Peralatan Pendukung

Dalam melakukan penelitian ini penulis memerlukan beberapa peralatan pendukung untuk membantu menjalankan penelitian dari awal hingga akhir, peralatan pendukung tersebut meliputi perangkat lunak (*software*) dan perangkat keras (*hardware*).

3.6.1 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan yaitu:

1. Sistem operasi Windows 10 Home 64-bit
2. Bahasa pemrograman Python 3.7.4 (64-bit)
3. Microsoft Office 2019

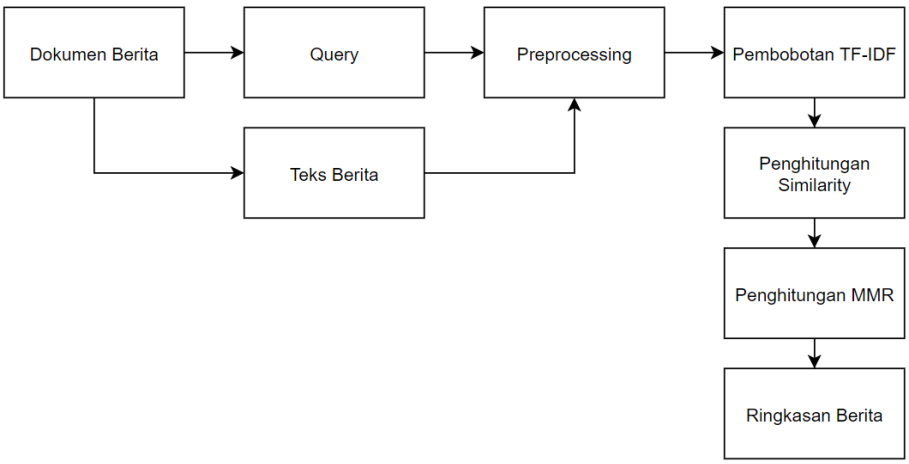
3.6.2 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan yaitu:

1. Intel® Core™ i7-8750H
2. Intel® UHD Graphics 630 dan NVIDIA GeForce GTX 1050 Ti
3. Memori RAM 8 GB DDR4
4. *Solid State Drive* (SSD) 256 GB dan *Solid State Hybrid Drive* (SSHD) 1 TB

3.7 Perancangan Algoritma

Pada bagian perancangan algoritma ini penulis akan menjabarkan komponen-komponen yang dibutuhkan untuk proses pembuatan sistem perancangan otomatis. Proses implementasi algoritma diawali dengan melakukan *preprocessing* pada dokumen berita. Setelah melalui tahap *preprocessing* akan dilakukan pembobotan menggunakan TF-IDF untuk setiap kata dan *query*. Kemudian dilakukan perhitungan kemiripan antara *query* dengan setiap kalimat dan juga antara kalimat yang satu dengan kalimat lainnya dengan metode Improved Sqrt-Cosine Similarity. Selanjutnya dilakukan perhitungan MMR untuk dapat melakukan proses selanjutnya yaitu ekstraksi ringkasan. Tahapan-tahapan tersebuut akan digambarkan dalam diagram alir pada Gambar 3.1.



Gambar 3.1 Arsitektur Peringkasan Teks Otomatis

BAB 4 PERANCANGAN

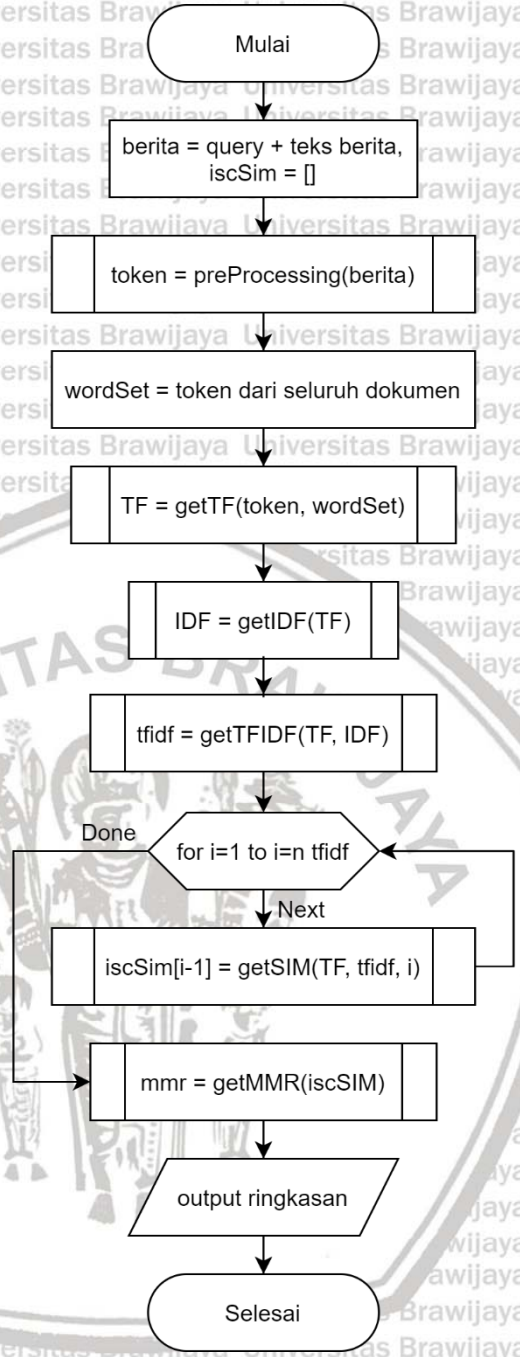
Bab perancangan terdiri dari perancangan algoritme dari metode yang digunakan, manualisasi , perancangan uji coba serta evaluasi sistem peringkasan teks otomatis. Selain itu, dalam bab ini juga menampilkan diagram alir proses dari setiap algoritme yang digunakan dalam penelitian ini.

4.1 Deskripsi Umum Sistem

Pada penelitian ini, sistem yang dirancang merupakan sistem peringkasan teks otomatis untuk berita berbahasa Indonesia dengan menggunakan metode Maximum Marginal Relevance. Sistem dibangun dengan menggunakan Bahasa pemrograman Python dan teks editor Visual Studio Code sebagai alat bantu untuk menjalankan kode program yang akan dibuat. Dataset yang digunakan berupa teks berita yang disimpan dalam file berekstensi txt yang berjumlah 10 berita yang diambil dari beberapa situs portal berita. Langkah pertama yang dilakukan oleh sistem adalah melakukan *preprocessing* pada teks berita, *preprocessing* yang dilakukan berupa *case folding, cleaning,* tokenisasi*, filtering* dan *stemming*. Langkah selanjutnya yaitu menghitung tingkat kesamaan antara query dengan kalimat dan *similarity* antar kalimat. Hasil perhitungan *similarity* tersebut akan digunakan untuk menghitung nilai MMR untuk membuat ringkasan.

4.2 Perancangan Algoritme

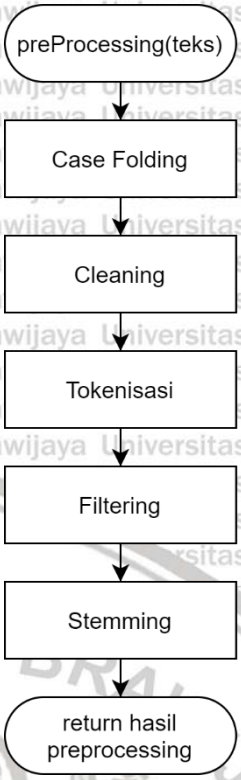
Pada proses perancangan system peringkasan teks otomatis untuk berita berbahasa Indonesia dengan metode MMR, terdapat dua masukan yaitu *query* yang berasal dari judul berita dan teks berita yang ingin diringkas. Langkah pertama yang dilakukan adalah melakukan preprocessing pada teks berita. Langkah kedua yaitu melakukan pembobotan TF-IDF untuk masing-masing *term*, nilai TF-IDF tersebut akan digunakan untuk menghitung similarity antara *query* dengan kalimat dan similarity antar kalimat. Langkah selanjutnya yaitu menghitung nilai MMR menggunakan nilai *similarity* yang telah didapat, nilai MMR tersebut kemudian akan diurutkan mulai dari yang terbesar. Ringkasan dibuat berdasarkan kalimat dengan nilai MMR tertinggi dan seterusnya. Tahapan proses secara keseluruhan dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Alir Sistem

4.3 Preprocessing

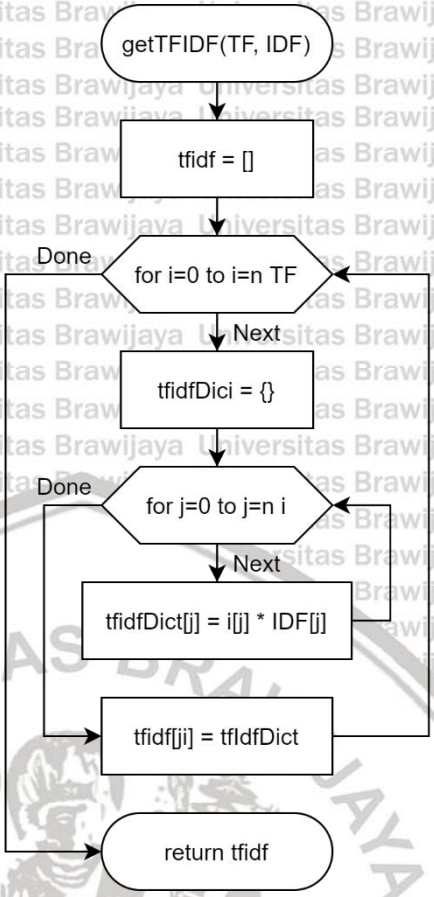
Langkah pertama dari penelitian ini yaitu melakukan *preprocessing* pada dokumen berita. *Preprocessing* digunakan untuk mendapatkan term yang akan digunakan untuk proses pembobotan TF-IDF. Tahap *preprocessing* yang diterapkan terdiri dari 5 langkah yaitu, *case folding, cleaning, tokenisasi, filtering* dan *stemming*. Diagram alir *preprocessing* akan ditampilkan pada Gambar 4.2.



Gambar 4.2 Diagram Alir *Preprocessing*

4.4 Pembobotan TF-IDF

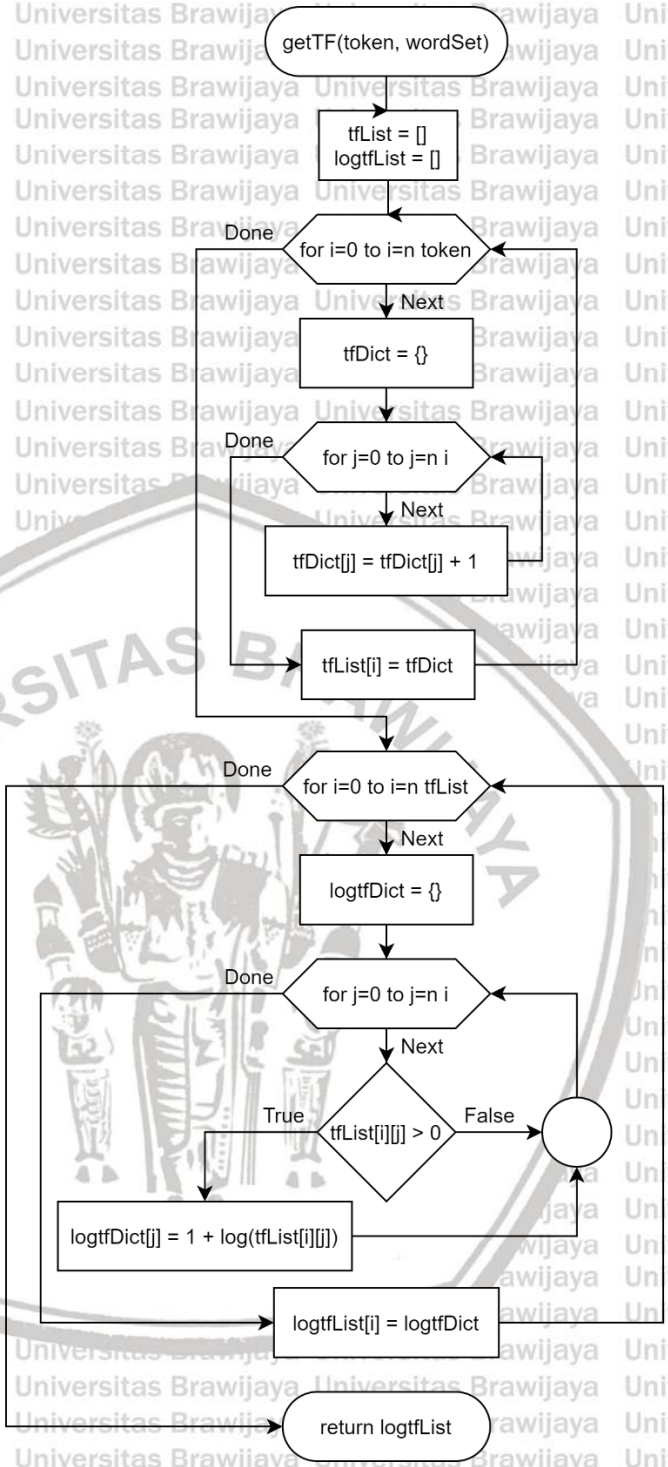
Dalam penelitian ini metode pembobotan yang digunakan adalah TF-IDF. Metode TF-IDF diawali dengan menghitung *term frequency* (TF) merupakan jumlah kemunculan masing-masing kata dalam setiap dokumen, selanjutnya dilakukan penghtiungan *document frequency* (DF) yang merupakan jumlah dokumen yang mengandung *term* tertentu. *Document frequency* tersebut selanjutnya akan digunakan untuk menghitung *invers document frequency* (IDF). Setelah nilai TF dan IDF didapatkan, penghitungan pembobotan TF-IDF baru bisa dilakukan dengan cara mengalikan nilai TF dan IDF masing-masing *term*. Diagram alir pembobotan TF-IDF dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram Alir Pembobotan TF-IDF

4.4.1 Term Frequency

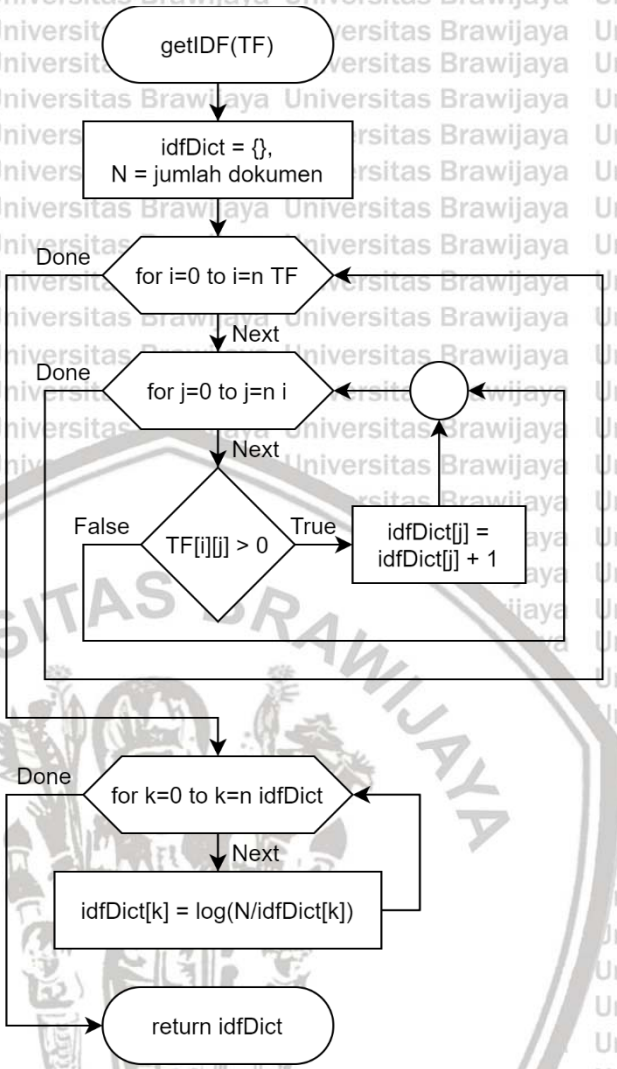
Pada proses *Term Frequency* sistem akan menghitung frekuensi kemunculan masing-masing term pada setiap dokumen yang mana dalam penelitian ini yang dimaksud dokumen adalah setiap kalimat dari teks berita yang diringkas. Perhitungan *term frequency* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir Perhitungan *Term Frequency*

4.4.2 Inverse Document Frequency

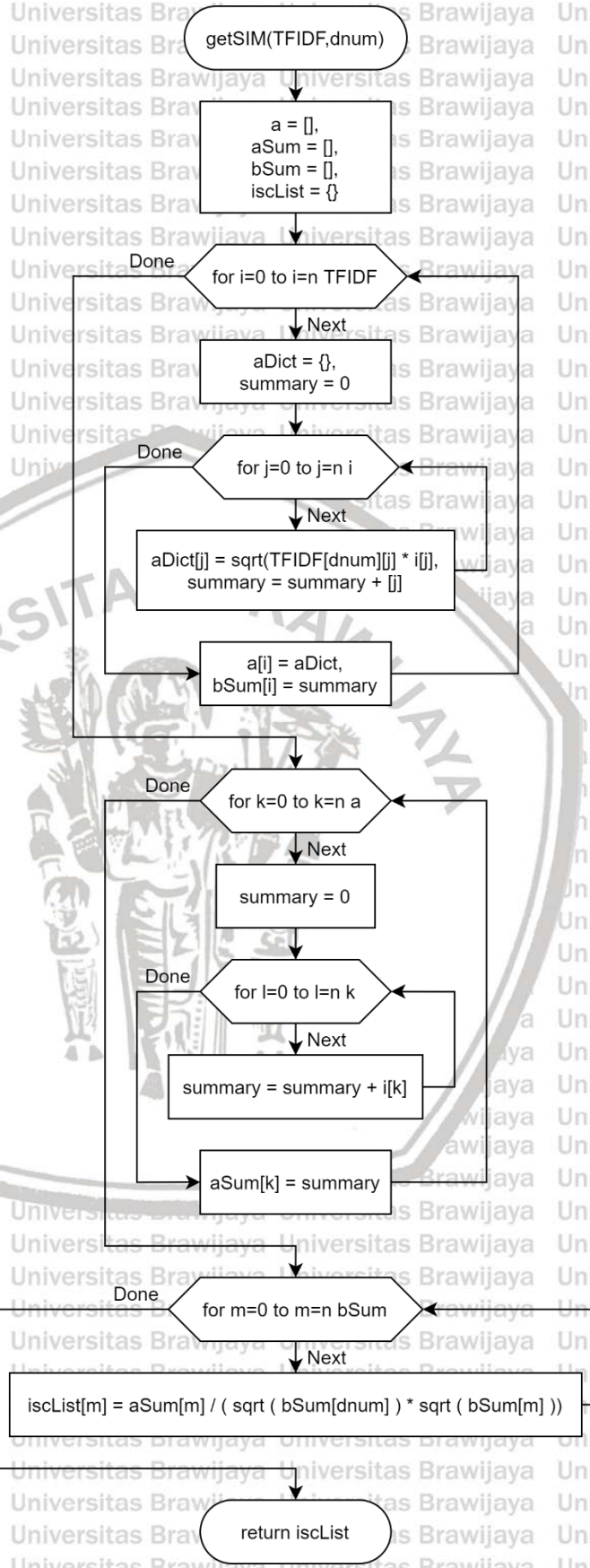
Pada proses Inverse Document Frequency sistem akan menghitung nilai IDF dengan menggunakan nilai DF yang telah didapatkan. Perhitungan IDF didapatkan dengan melakukan perhitungan logaritma pada nilai yang dihasilkan dari pembagian antara jumlah seluruh dokumen dengan nilai DF. Perhitungan *inverse document frequency* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram Alir Perhitungan *Inverse Document Frequency*

4.5 Perhitungan *Similarity*

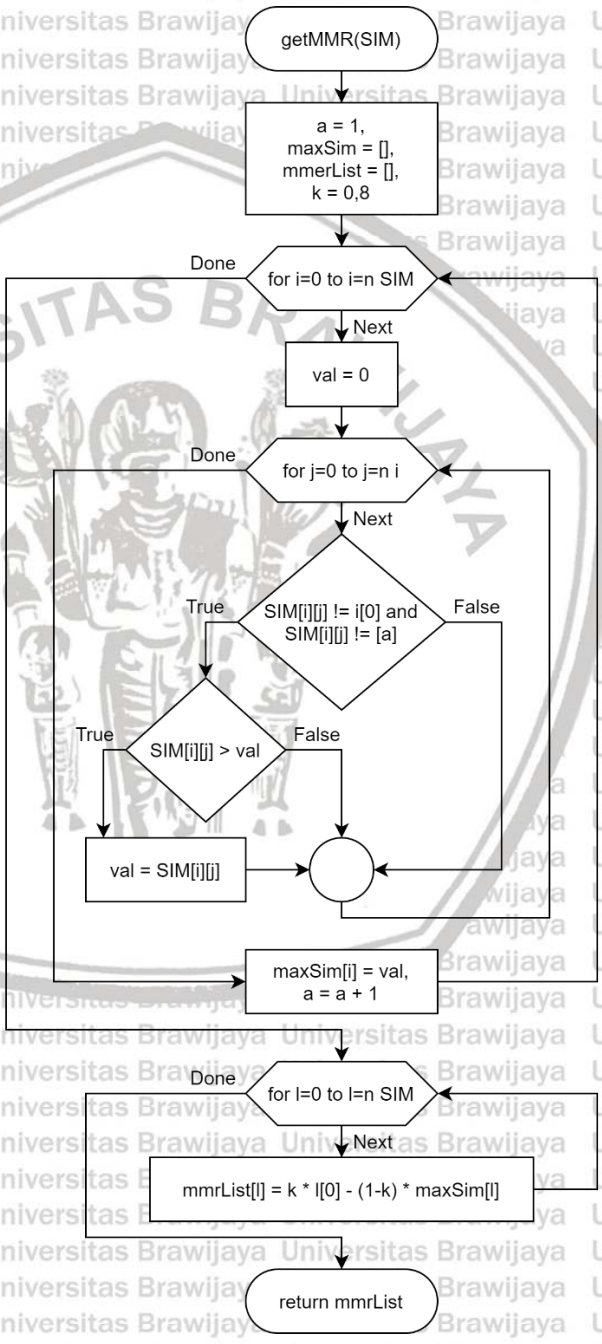
Dalam tahap perhitungan *similarity*, masing-masing kalimat akan dihitung tingkat kemiripannya dengan kalimat lainnya. Dalam penelitian ini perhitungan similarity kalimat dilakukan dengan menggunakan metode *Improved Sqrt-Cosine Similarity*, metode ini memanfaatkan nilai TF-IDF yang telah didapat sebelumnya. Diagram alir perhitungan *similarity* kalimat dapat dilihat pada Gambar 4.6.



Gambar 4.6 Diagram Alir Perhitungan *Similarity*

4.6 Pembentukan Ringkasan Berita

Dalam penelitian ini ringkasan berita didapatkan dengan melihat nilai perhitungan MMR dari masing-masing kalimat. Nilai MMR yang tinggi menandakan bahwa kalimat tersebut memiliki tingkat relevansi yang tinggi terhadap query dan dianggap dapat mewakili informasi yang ingin disampaikan. Perhitungan MMR dilakukan dengan memanfaatkan nilai similarity dari query dengan kalimat dan nilai similarity tertinggi antara kalimat tersebut dengan kalimat lainnya. Diagram alir perhitungan MMR dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Perhitungan *Maximum Marginal Relevance*

4.7 Perhitungan Manual

4.7.1 Data Uji

Data uji yang digunakan untuk perhitungan manualisasi merupakan sebuah berita yang diambil dari situs berita cnnindonesia.com. Teks berita yang digunakan terdiri dari 14 kalimat berita dan 1 kalimat judul. Data uji akan ditampilkan pada Tabel 4.1.

Tabel 4.1 Data Uji

Judul	Teks Berita
Fakta Temuan Sinovac Kurang Ampuh Lawan Mutasi Corona Brasil	Ahli biologi molekuler Ahmad Rusdan Handoyo menyatakan studi yang menyatakan vaksin Covid-19 buatan Sinovac kurang ampuh melawan varian P1 yang ditemukan pertama kali di Brasil masih bersifat sebagian. Menurutnya, studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian P1 yang telah menerima vaksin Sinovac. "Studi ini melihat satu dari dua aspek kekebalan tubuh yaitu antibodi," ujar Ahmad kepada CNNIndonesia.com, Selasa (9/3). Ahmad menuturkan studi itu memperlihatkan antibodi yang muncul baik dari orang yang terinfeksi alami maupun tervaksinasi kurang efektif dalam memblok infeksi virus varian P1 yang ditemukan di Brazil. Hasil studi itu, kata dia bisa menjelaskan mengapa terjadi banyak kasus reinfeksi di Brasil meski hampir 70 persen populasi kota Manaus, Brazil, tempat studi dilakukan sudah membentuk antibodi terhadap varian lama (SARS-CoV-2). "Namun data di laboratorium ini memang diakui penelitiannya belum menguji aspek imunitas kedua, yaitu status aktivasi seluler yaitu respon sel T," ujarnya. Ahmad berkata sel T ini penting untuk mengenali sel manusia yang telah terinfeksi virus. Ketika antibodi, imunitas humoral gagal mencegah infeksi pada sel, dia menyebut masih ada sel T yang akan membasmi sel yg terinfeksi. "Maka perlu studi lanjutan terhadap aspek sel T pada orang yg telah tervaksinasi," ujar Ahmad. Berdasarkan studi itu pula, Ahmad mengingatkan pentingnya melakukan post vaccination surveillance pada orang yang telah tervaksinasi dan menghitung kejadian kasus covid bergejala berat antara komunitas yang divaksin dengan yang belum divaksin. "Apabila terkonfirmasi terjadi kasus Covid gejala berat pada orang yang sudah divaksin maka wajib hukumnya untuk mengirim sampel sisa PCR ke tim Genome Surveilans besutan Kemkes dan Kemenristekdikti untuk dianalisa genom virusnya," ujarnya. Ahmad menambahkan tidak ada batasan 'minimal' sampel untuk meneliti keampuhan sebuah vaksin selama dalam pembahasan hasil tidak ada overclaim. Lebih dari itu, dia menyebut perlunya studi lanjutan terkait jumlah titer antibodi, misalnya apakah delapan orang itu mewakili jumlah antibodi yang sama karena jumlah antibodi yang terbentuk tentu ada kontribusi. "Mengingat jumlah antibodi yang muncul dua minggu setelah vaksin tidak sama dibanding setelah dua bulan usai vaksinasi misalnya," ujar Ahmad.

4.7.2 Perhitungan Manual *Preprocessing*

Pada bagian ini akan dijelaskan tahapan perhitungan manual dari proses *preprocessing* teks berita pada Tabel 4.1. Tabel proses *preprocessing* akan ditampilkan secara lengkap pada lampiran A.

4.7.2.1 *Case Folding*

Tahapan pertama pada proses *preprocessing* adalah *case folding*. Dalam tahap ini setiap huruf pada data uji disamakan kapitalisasinya menjadi huruf kecil semua. Hasil penerapan *case folding* pada data uji dapat dilihat pada Tabel 4.2.

Tabel 4.2 Perhitungan Manual *Case Folding*

Setelah proses *case folding*	
Q	fakta temuan sinovac kurang ampuh lawan mutasi corona brasil
D1	ahli biologi molekuler ahmad rusdan handoyo menyatakan studi yang menyatakan vaksin covid-19 buatan sinovac kurang ampuh melawan varian p1 yang ditemukan pertama kali di brasil masih bersifat sebagian.
D2	menurutnya, studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian p1 yang telah menerima vaksin sinovac.
D3	"studi ini melihat satu dari dua aspek kekebalan tubuh yaitu antibodi," ujar ahmad kepada cnnindonesia.com, selasa (9/3).
...	...
D14	"mengingat jumlah antibodi yang muncul dua minggu setelah vaksin tidak sama dibanding setelah dua bulan usai vaksinasi misalnya," ujar ahmad.

4.7.2.2 *Cleaning*

Tahapan selanjutnya adalah melakukan tahap *cleaning* pada data uji. Pada tahap *cleaning*, sistem akan menghapus semua tanda baca/karakter dan angka selain huruf. Hasil dari proses *cleaning* terhadap data uji akan ditampilkan pada Tabel 4.3.

Tabel 4.3 Perhitungan Manual *Cleaning*

Setelah proses *cleaning*	
Q	fakta temuan sinovac kurang ampuh lawan mutasi corona brasil
D1	ahli biologi molekuler ahmad rusdan handoyo menyatakan studi yang menyatakan vaksin covid buatan sinovac kurang ampuh melawan varian p yang ditemukan pertama kali di brasil masih bersifat sebagian
D2	menurutnya studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian p yang telah menerima vaksin sinovac
D3	studi ini melihat satu dari dua aspek kekebalan tubuh yaitu antibodi ujar ahmad kepada cnnindonesiacom selasa
...	...
D14	mengingat jumlah antibodi yang muncul dua minggu setelah vaksin tidak sama dibanding setelah dua bulan usai vaksinasi misalnya ujar ahmad

4.7.2.3 Tokenisasi

Tahap selanjutnya adalah melakukan tokenisasi. Dalam tahap tokenisasi, kalimat-kalimat dalam data uji akan dipisah-pisahkan dengan parameter pemisah spasi sehingga menjadi token. Hasil dari tahap tokenisasi terhadap data uji dapat dilihat pada Tabel 4.4.

Tabel 4.4 Perhitungan Manual Tokenisasi

Setelah proses tokenisasi	
Q	'fakta', 'temuan', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'menyatakan', 'studi', 'yang', 'menyatakan', 'vaksin', 'covid', 'buatan', 'sinovac', 'kurang', 'ampuh', 'melawan', 'varian', 'p', 'yang', 'ditemukan', 'pertama', 'kali', 'di', 'brasil', 'masih', 'bersifat', 'sebagian'
D2	'menurutnya', 'studi', 'itu', 'baru', 'sekedar', 'meneliti', 'antibodi', 'pada', 'orang', 'terinfeksi', 'varian', 'p', 'yang', 'telah', 'menerima', 'vaksin', 'sinovac'
D3	'studi', 'ini', 'melihat', 'satu', 'dari', 'dua', 'aspek', 'kekebalan', 'tubuh', 'yaitu', 'antibodi', 'ujar', 'ahmad', 'kepada', 'cnnindonesiacom', 'selasa'
...	...
D14	'mengingat', 'jumlah', 'antibodi', 'yang', 'muncul', 'dua', 'minggu', 'setelah', 'vaksin', 'tidak', 'sama', 'dibanding', 'setelah', 'dua', 'bulan', 'usai', 'vaksinasi', 'misalnya', 'ujar', 'ahmad'

4.7.2.4 Filtering

Hasil dari tahap tokenisasi yang berupa token akan masuk ke tahap *filtering*. Dalam tahap *filtering*, token yang termasuk dalam daftar *stopword* akan dihapus atau dibuang. Hasil dari tahap *filtering* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Perhitungan Manual *Filtering*

Setelah proses *filtering*	
Q	'fakta', 'temuan', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'menyatakan', 'studi', 'menyatakan', 'vaksin', 'covid', 'buatan', 'sinovac', 'kurang', 'ampuh', 'melawan', 'varian', 'ditemukan', 'pertama', 'kali', 'brasil', 'bersifat', 'sebagian'
D2	'menurutnya', 'studi', 'baru', 'sekedar', 'meneliti', 'antibodi', 'orang', 'terinfeksi', 'varian', 'menerima', 'vaksin', 'sinovac'
D3	'studi', 'melihat', 'satu', 'aspek', 'kekebalan', 'tubuh', 'antibodi', 'ujar', 'ahmad', 'cnnindonesiacom', 'selasa'
...	...
D14	'mengingat', 'jumlah', 'antibodi', 'muncul', 'minggu', 'vaksin', 'sama', 'dibanding', 'bulan', 'usai', 'vaksinasi', 'misalnya', 'ujar', 'ahmad'

4.7.2.5 Stemming

Tahap terakhir dalam *preprocessing* adalah *stemming*. Dalam tahap *stemming*, setiap token yang telah lolos dari tahap *filtering* akan dihilangkan imbuhannya sehingga menjadi ke bentuk kata dasarnya. Hasil penerapan tahap *stemming* terhadap data uji dapat dilihat pada Tabel 4.6.

Tabel 4.6 Perhitungan Manual *Stemming*

Setelah proses *stemming*	
Q	'fakta', 'temu', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'nyata', 'studi', 'nyata', 'vaksin', 'covid', 'buat', 'sinovac', 'kurang', 'ampuh', 'lawan', 'varian', 'temu', 'pertama', 'kali', 'brasil', 'sifat', 'bagi'
D2	'turut', 'studi', 'baru', 'dar', 'teliti', 'antibodi', 'orang', 'infeksi', 'varian', 'terima',

	'vaksin', 'sinovac'
D3	'studi', 'lihat', 'satu', 'aspek', 'kebal', 'tubuh', 'antibodi', 'ujar', 'ahmad', 'cnnindonesiacom', 'selasa'
...	...
D14	'ingat', 'jumlah', 'antibodi', 'muncul', 'minggu', 'vaksin', 'sama', 'banding', 'bulan', 'usai', 'vaksinasi', 'misal', 'ujar', 'ahmad'

4.7.3 Perhitungan Manual Pembobotan

Pada tahap perhitungan manual pembobotan akan dijelaskan perhitungan manual untuk tahapan yang ada pada pembobotan dengan metode TF-IDF. Tahap pertama dimulai dengan mencari *term frequency* (TF) kemudian dilanjutkan mencari *inverse document frequency* (IDF). Nilai bobot term didapat dengan mengalikan nilai TF dengan nilai IDF yang didapatkan. Tabel perhitungan manual pembobotan secara lengkap akan ditampilkan pada lampiran B.

4.7.3.1 Term Frequency (TF)

Tahap pertama untuk dapat menghitung pembobotan yaitu mencari nilai TF setiap term. Nilai TF didapat dengan menghitung jumlah kemunculan masing-masing term pada setiap dokumen. Nilai TF yang sudah didapat kemudian akan dinormalisasi untuk meminimalkan jarak antar nilai TF. Hasil perhitungan normalisasi nilai TF dapat dilihat pada Tabel 4.7.

Tabel 4.7 Perhitungan Manual *Term Frequency*

Token	Q	D1	D2	D3	...	D14
ahli	0	1	0	0	...	0
ahmad	0	1	0	1	...	1
aktivasi	0	0	0	0	...	0
aku	0	0	0	0	...	0
alami	0	0	0	0	...	0
ampuh	1	1	0	0	...	0
antibodi	0	0	1	1	...	1
apabila	0	0	0	0	...	0
aspek	0	0	0	1	...	0
bagi	0	1	0	0	...	0
bahas	0	0	0	0	...	0
baik	0	0	0	0	...	0
banding	0	0	0	0	...	1
banyak	0	0	0	0	...	0
baru	0	0	1	0	...	0
basmi	0	0	0	0	...	0
batas	0	0	0	0	...	0
bentuk	0	0	0	0	...	0
berat	0	0	0	0	...	0
besut	0	0	0	0	...	0
biologi	0	1	0	0	...	0
blok	0	0	0	0	...	0
brasil	1	1	0	0	...	0
brazil	0	0	0	0	...	0
buah	0	0	0	0	...	0
buat	0	1	0	0	...	0

bulan	0	0	0	0	...	1
cegah	0	0	0	0	...	0
...	...	...	...	...	...	...
uji	0	0	0	0	...	0
usai	0	0	0	0	...	1
vaccination	0	0	0	0	...	0
vaksin	0	1	1	0	...	1
vaksinasi	0	0	0	0	...	1
varian	0	1	1	0	...	0
virus	0	0	0	0	...	0
wajib	0	0	0	0	...	0
wakil	0	0	0	0	...	0

4.7.3.2 Inverse Document Frequency (IDF)

Tahap selanjutnya setelah mencari nilai TF setiap term adalah menghitung nilai IDF, tapi sebelumnya akan dicari nilai document frequency (DF) terlebih dahulu. Nilai DF didapat dengan cara menghitung jumlah dokumen yang memiliki term tertentu, setelah nilai DF didapatkan langkah selanjutnya yaitu menghitung IDF. Hasil nilai DF dan hasil perhitungan IDF dapat dilihat pada Tabel 4.8. Berikut adalah contoh perhitungan IDF dengan Persamaan 2.2.

$$IDF = \log\left(\frac{N}{df}\right)$$
$$= \log\left(\frac{15}{8}\right)$$
$$= \log(1{,}875)$$
$$= 0{,}273001$$

Tabel 4.8 Perhitungan Manual *Inverse Document Frequency*

Token	DF	IDF
ahli	1	1,176091
ahmad	8	0,273001
aktivasi	1	1,176091
aku	1	1,176091
alami	1	1,176091
ampuh	3	0,69897
antibodi	7	0,330993
apabila	1	1,176091
aspek	3	0,69897
bagi	1	1,176091
bahas	1	1,176091
baik	1	1,176091
banding	1	1,176091
banyak	1	1,176091
baru	1	1,176091
basmi	1	1,176091
batas	1	1,176091
bentuk	2	0,875061
berat	2	0,875061

besut	1	1,176091
biologi	1	1,176091
blok	1	1,176091
brasil	3	0,69897
brazil	2	0,875061
buah	1	1,176091
buat	1	1,176091
bulan	1	1,176091
cegah	1	1,176091
...	...	...
uji	1	1,176091
usai	1	1,176091
vaccination	1	1,176091
vaksin	6	0,39794
vaksinasi	4	0,574031
varian	4	0,574031
virus	3	0,69897
wajib	1	1,176091
wakil	1	1,176091

4.7.4 Perhitungan Manual *Similarity* Kalimat

Nilai *similarity* pada tabel hanya ditampilkan nilai pada segitiga atasnya saja karena nilai pada tabel tersebut simetris. Nilai pada bagian segitiga bawah sama seperti nilai pada segitiga atas. Nilai similarity antar kalimat dapat dilihat pada Tabel 4.9. Tabel perhitungan manual *similarity* antar kalimat secara lengkap akan disajikan dalam lampiran C. Berikut adalah contoh perhitungan Similarity dengan Persamaan 2.3.

$$ISC(x,y) = \frac{\sum_{i=1}^{m}\sqrt{x_i y_i}}{\sqrt{(\sum_{i=1}^{m} x_i)}\sqrt{(\sum_{i=1}^{m} y_i)}}$$

$$= \frac{1{,}943943}{\sqrt{19{,}8779}\sqrt{8{,}650242}}$$

$$= 0{,}148246$$

Tabel 4.9 Perhitungan Manual *Similarity* Kalimat

Sim	Q	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14
Q	1	0,349	0,085	0	0,129	0,054	0	0	0	0	0	0	0,073	0	0
D1		1	0,148	0,041	0,146	0,077	0	0,024	0	0,055	0,097	0,050	0,090	0,016	0,045
D2			1	0,069	0,196	0,089	0,061	0,077	0,116	0,102	0,097	0,055	0,109	0,095	0,074
D3				1	0,153	0,047	0,102	0,036	0,039	0,259	0,047	0,033	0,027	0,059	0,109
D4					1	0,117	0	0,166	0,096	0,175	0,100	0,058	0,021	0,072	0,159
D5						1	0	0,075	0,025	0,030	0,141	0,061	0,110	0,089	0,021
D6							1	0,067	0,140	0,201	0	0,025	0,052	0	0,037

D7								1	0,198	0,164	0,115	0,056	0,031	0	0,032
D8									1	0,108	0	0	0	0,115	0,034
D9										1	0,172	0,079	0,036	0,281	0,177
D10											1	0,241	0,054	0,044	0,165
D11												1	0,075	0,021	0,053
D12													1	0	0,059
D13														1	0,251
D14															1

4.7.5 Perhitungan Manual *Maximum Marginal Relevance*

Perhitungan terakhir yang dilakukan untuk membuat ringkasan pada penelitian ini adalah menghitung nilai MMR. Nilai MMR dapat dihitung dengan memanfaatkan nilai similarity antar kalimat dengan nilai parameter λ = 0,8. Hasil perhitungan nilai MMR dapat dilihat pada Tabel 4.10. Berikut contoh perhitungan nilai MMR dengan Persamaan 2.4.

$$MMR = argmax\ [\lambda * Sim1(S_i, Q) - (1 - \lambda) * \max Sim2(S_i, S')]$$
$$= 0{,}8 * 0{,}348758 - (1 - 0{,}8) * 0{,}148246$$
$$= 0{,}249357$$

Tabel 4.10 Perhitungan Manual *Maximum Marginal Relevance*

Dokumen	Nilai MMR
D1	0,249357
D2	0,028416
D3	-0,0518
D4	0,06369
D5	0,01464
D6	-0,04023
D7	-0,03963
D8	-0,03963
D9	-0,05626
D10	-0,04829
D11	-0,04829
D12	0,036045
D13	-0,05626
D14	-0,05018

4.7.6 Peringkasan Dokumen

Data teks berita yang digunakan untuk perhitungan manual terdiri dari 14 kalimat. Peringkasan sebanyak 50% dilakukan terhadap data uji tersebut sehingga menghasilkan ringkasan dengan 7 kalimat. Ringkasan dibuat

berdasarkan nilai MMR masing-masing kalimat tanpa mengubah urutan kalimat dalam berita. Ringkasan akan ditampilkan pada tabel 4.11.

Tabel 4.11 Ringkasan 50% terhadap Data Uji

Dokumen	Nilai MMR	Kalimat
D1	0,249357	Ahli biologi molekuler Ahmad Rusdan Handoyo menyatakan studi yang menyatakan vaksin Covid-19 buatan Sinovac kurang ampuh melawan varian P1 yang ditemukan pertama kali di Brasil masih bersifat sebagian.
D2	0,028416	Menurutnya, studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian P1 yang telah menerima vaksin Sinovac.
D4	0,06369	Ahmad menuturkan studi itu memperlihatkan antibodi yang muncul baik dari orang yang terinfeksi alami maupun tervaksinasi kurang efektif dalam memblok infeksi virus varian P1 yang ditemukan di Brazil.
D5	0,01464	Hasil studi itu, kata dia bisa menjelaskan mengapa terjadi banyak kasus reinfeksi di Brasil meski hampir 70 persen populasi kota Manaus, Brazil, tempat studi dilakukan sudah membentuk antibodi terhadap varian lama (SARS-CoV-2).
D7	-0,03963	Ahmad berkata sel T ini penting untuk mengenali sel manusia yang telah terinfeksi virus.
D8	-0,03963	Ketika antibodi, imunitas humoral gagal mencegah infeksi pada sel, dia menyebut masih ada sel T yang akan membasmi sel yg terinfeksi.
D12	0,036045	Ahmad menambahkan tidak ada batasan 'minimal' sampel untuk meneliti keampuhan sebuah vaksin selama dalam pembahasan hasil tidak ada overclaim.

4.8 Perancangan Pengujian

Dalam penelitian ini ada 3 ringkasan yang akan dihasilkan sistem, masing-masing dengan persentase ringkasan 10%, 25% dan 50%. Pengujian dilakukan untuk mengetahui tingkat keberhasilan metode MMR terhadap sistem peringkasan teks otomatis. Pengujian dilakukan dengan membandingkan ringkasan yang dibuat manual oleh pakar dengan ringkasan yang dibuat oleh sistem, kualitas ringkasan akan diukur dengan menghitung nilai *precision, recall,* dan *f-measure* dari metode ROUGE-L : Longest Common Subsequence.

BAB 5 IMPLEMENTASI

Pada bab ini akan menjelaskan tentang bagaimana implementasi atau penerapan dari perancangan algoritma *Maximum Marginal Relevance* terjadi. Terdapat beberapa kode program untuk peringkasan teks berita otomatis yaitu proses *preprocessing* teks, pembobotan TFIDF, penghitungan Improved Sqrt-Cosine Similarity, dan penghitungan nilai Maximum Marginal Relevance untuk membuat ringkasan.

5.1 Implementasi Fungsi *Main*

Dalam fungsi main, semua fungsi yang dibuat sebelumnya dipanggil berurutan sesuai dengan algoritma sistem. Implementasi kode program dapat dilihat pada Kode Program 5.1.

Main

```
import re
import math
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize

stopWordFactory = StopWordRemoverFactory()
stopwords = stopWordFactory.get_stop_words()

stemFactory = StemmerFactory()
stemmer = stemFactory.create_stemmer()

def main(beritaKe, persentaseRingkasan):
    fileBerita = open("data/"+beritaKe+".txt", "r")
    berita = fileBerita.read()

    temp = re.split("\n+", berita)
    teksBerita = re.split("(?<!\w\.\w.)(?<![A-Z][a-
z]\.)(?<=\.|\?)\s", temp[1])
    teks = []
    teks.append(temp[0])
    for kalimat in teksBerita:
        teks.append(kalimat)

    token = preProcessing(berita)
    wordSet = set().union(*token)
    TF = getTF(token,wordSet)
    IDF = getIDF(TF)
    tfidf = getTFIDF(TF,IDF)
    iscSim = []
    for i in range(1,len(tfidf)):
        iscSim.append(getSIM(tfidf,i))
    mmr = getMMR(iscSim)

    print(berita, "\n")
    getRingkasan(mmr,teks,persentaseRingkasan)

```

```
40  if __name__ == "__main__":
41      main("manualisasi", 50)
```

Kode Program 5.1 Implementasi Fungsi *Main*

Penjelasan dari Kode Program 5.1 mengenai implementasi *preprocessing* yaitu.

1. Baris 1-7 merupakan proses *import* untuk *re*, *math*, *stopword* dan *stemmer* dari *library Sastrawi* dan juga *library NLTK*
2. Baris 9-10 merupakan proses memasukkan *stopword list* dari *library* Sastrawi ke dalam variabel *stopwords*
3. Baris 12-13 merupakan proses memasukkan *stemmer* dari *library* Sastrawi ke dalam variabel *stemmer*
4. Baris 15 merupakan deklarasi fungsi main dengan parameter *beritaKe*, *presentaseRingkasan*.
5. Baris 16-17 merupakan proses membuka dan membaca *file* berita kemudian disimpan dalam variabel *berita*.
6. Baris 19-25 merupakan proses untuk memecah teks berita menjadi masing-masing kalimat yang akan disimpan dalam *array teks*.
7. Baris 27-35 merupakan serangkaian proses memanggil fungsi mulai dari fungsi *preProcessing*, fungsi *getTF*, fungsi *getIDF*, fungsi *getTFIDF*, fungsi *getSIM*, dan fungsi *getMMR*. Hasil akhir keluaran dari serangkaian fungsi tersebut merupakan *array* berisi nilai MMR masing-masing kalimat yang kemudian disimpan dalam variabel *mmr*.
8. Baris 37-38 merupakan proses mencetak berita dan memanggil fungsi *getRingkasan* dengan parameter *mmr*, *teks*, *persentaseRingkasan*.
9. Baris 40-41 merupakan proses memanggil fungsi *main* dengan isi parameter berupa "*manualisasi*" dan *50*.

5.2 Implementasi *Preprocessing*

Proses *preprocessing* melewati beberapa tahap dimulai dari tahap *case folding*, *data cleaning*, tokenisasi, *filtering*, dan *stemming*. Setiap tahap dalam proses *preprocessing* disatukan dalam satu fungsi. Implementasi kode program dapat dilihat pada Kode Program 5.2.

Preprocessing

```
1  def preProcessing(berita):
2      hasilPreprocessing = []
3
4      temp = re.split("\n+", berita)
5      teksBerita = re.split("(?<!\w\.\w.)(?<![A-Z][a-
6  z]\.)(?<=\.|\?)\s", temp[1])
7      teks = []
8      teks.append(temp[0])
9      for kalimat in teksBerita:
```

```
10          teks.append(kalimat)
11
12      for kalimat in teks:
13          stem = []
14          lowerCase = kalimat.lower()
15
16          numRemove = re.sub(r'\d+', ' ', lowerCase)
17          puncRemove = re.sub(r'[^\w\s]', ' ', numRemove)
18
19          token = word_tokenize(puncRemove)
20
21          filteredToken = [word for word in token if word
22  not in stopwords]
23          filteredToken = [word for word in filteredToken if
24  len(word) > 2]
25
26          for word in filteredToken:
27              stem.append(stemmer.stem(word))
28
29          hasilPreprocessing.append(stem)
30
31      return hasilPreprocessing
```

Kode Program 5.2 Implementasi *Preprocessing*

Penjelasan dari Kode Program 5.2 mengenai implementasi *preprocessing* yaitu.

1. Baris 1-10 merupakan proses untuk memecah teks berita menjadi masing-masing kalimat yang akan disimpan dalam *array teks*.
2. Baris 12 merupakan proses untuk melakukan perulangan untuk setiap kalimat pada *array teks*.
3. Baris 14 merupakan proses untuk *case folding* dengan fungsi *lower*() yang kemudian disimpan dalam variabel *lowercase*.
4. Baris 16-17 merupakan proses untuk *data cleaning* menggunakan regex dengan cara mengganti karakter selain huruf menjadi spasi kosong.
5. Baris 19 merupakan proses untuk melakukan tokenisasi pada kalimat dengan menggunakan fungsi *word_tokenize*().
6. Baris 21-24 merupakan proses untuk melakukan *filtering* dengan membuang kata yang termasuk dalam *stopword list* dan kata yang kurang dari 3 huruf.
7. Baris 26-29 merupakan proses untuk melakukan *stemming* pada masing-masing token dengan menggunakan *stemmer* dari sastrawi. Hasil dari proses *stemming* kemudian dimasukkan ke dalam *array hasilPreprocessing*.

5.3 Implementasi Pembobotan TFIDF

Proses menghitung bobot TFIDF dari masing-masing token terdiri dari beberapa tahap dimulai dari menghitung nilai TF dilanjutkan menghitung nilai IDF. Hasil nilai TF dan IDF akan digunakan dalam proses perhitungan bobot TFIDF

masing-masing token. Setiap tahap dalam proses pembobotan disajikan dalam fungsi. Implementasi kode program dapat dilihat pada Kode Program 5.3.

Pembobotan TFIDF

```
def getTF(token, wordSet):
    tfList = []
    logtfList = []

    for kalimat in token:
        tfDict = dict.fromkeys(wordSet, 0)
        for kata in kalimat:
            tfDict[kata] += 1

        tfList.append(tfDict)

    for kalimat in tfList:
        logtfDict = dict.fromkeys(wordSet, 0)
        for kata, val in kalimat.items():
            if val > 0:
                logtfDict[kata] = 1 +
math.log10(float(val))

        logtfList.append(logtfDict)

    return logtfList

def getIDF(TF):
    idfDict = {}
    idfDict = dict.fromkeys(TF[0].keys(), 0)
    N = len(TF)

    for dok in TF:
        for kata, val in dok.items():
            if val > 0:
                idfDict[kata] += 1

    for kata, val in idfDict.items():
        idfDict[kata] = math.log10(N / float(val))

    return idfDict

def getTFIDF(TF, IDF):
    tfidfList = []

    for dok in TF:
        tfidfDict = {}
        tfidfDict = dict.fromkeys(TF[0].keys(), 0)

        for kata in dok:
            tfidfDict[kata] = dok[kata] * IDF[kata]

        tfidfList.append(tfidfDict)

    return tfidfList
```

Kode Program 5.3 Implementasi Pembobotan TFIDF

Penjelasan dari Kode Program 5.3 mengenai implementasi pembobotan TFIDF yaitu.

1. Baris 1 merupakan deklarasi fungsi bernama *getTF* dengan parameter *token* dan *wordSet*.
2. Baris 2-3 merupakan deklarasi *array tfList* dan *logtfList*.
3. Baris 5-10 merupakan proses menghitung nilai raw tf dari masing-masing token kemudian dimasukkan ke dalam *array tfList*.
4. Baris 12-21 merupakan proses menghitung nilai normalisasi dari raw tf masing-masing token, kemudian dimasukkan ke dalam *array logtfList*. *Return* dari fungsi *getTF* berupa array *logtfList*.
5. Baris 23 merupakan deklarasi fungsi dengan nama *getIDF* berparameter *TF*.
6. Baris 24-26 merupakan deklarasi *dictionary idfDict* dan variabel *N*.
7. Baris 28-36 merupakan proses menghitung nilai IDF yang kemudian dimasukkan ke dalam *idfDict* dengan menyesuaikan *key* dengan *value*-nya. *Return* dari fungsi *getIDF* berupa *dictionary idfDIct*.
8. Baris 38 merupakan deklarasi fungsi dengan nama *getTFIDF* dengan parameter *TF* dan *IDF*.
9. Baris 39 merupakan deklarasi *array tfidfList*.
10. Baris 41-50 merupakan proses menghitung pembobotan TFIDF dengan mengkalikan nilai TF setiap token dengan nilai IDF-nya, kemudian dimasukkan dalam *array tfidfList*. *Return* dari fungsi *getTFIDF* berupa *array tfidfList*.

5.4 Implementasi Penghitungan *Improved Sqrt-Cosine Similarity*

Proses menghitung similarity dengan menggunakan metode *Improved Sqrt-Cosine Similarity* dalam penelitian ini dibagi menjadi 2 yaitu menghitung pembilang dan *similarity*. Proses penghitungan pembilang yang akan digunakan untuk menghitung *similarity* kalimat disajikan didalam satu fungsi yang sama. Implementasi kode program dapat dilihat pada Kode Program 5.3.

Penghitungan Improved Sqrt-Cosine Similarity

```
1  def getSIM(TFIDF, dnum):
2      a = []
3      aSum = []
4      tfidfSum = []
5      iscSim = []
6      i = 0
7
8      for dok in TFIDF:
9          aDict = {}
10         aDict = dict.fromkeys(TFIDF[0].keys(), 0)
11         summary = 0
12
13         for kata in dok:
14             aDict[kata] =
15 math.sqrt(TFIDF[dnum][kata]*dok[kata])
```

```
16              summary += dok[kata]
17
18          a.append(aDict)
19          tfidfSum.append(summary)
20
21      for dok in a:
22          summary = 0
23
24          for kata, val in dok.items():
25              summary += val
26
27          aSum.append(summary)
28
29      for val in tfidfSum:
30          iscSim.append(aSum[i]/(math.sqrt(tfidfSum[dnum])*m
31          ath.sqrt(val)))
32          i += 1
33
34      return iscSim
```

Kode Program 5.4 Implementasi Penghitungan Improved Sqrt-Cosine Similarity

Penjelasan dari Kode Program 5.4 mengenai implementasi penghitungan Improved Sqrt-Cosine Similarity yaitu.

1. Baris 1 merupakan deklarasi fungsi dengan nama *getSIM* yang memiliki parameter *TFIDF* dan *dnum*.
2. Baris 2-6 merupakan deklarasi dari *array* dengan nama *a*, *aSum*, *tfidfSum*, dan *iscSim* dan juga variabel *i*.
3. Baris 8-27 merupakan proses menghitung pembilang untuk diguankan dalam rumus menghitung *similarity*, nilai pembilang untuk masing-masing dokumen disimpan dalam *array aSum*.
4. Baris 29-34 merupakan proses menghitung *similarity* dengan metode *Improved Sqrt-Cosine Similarity* yang kemudian disimpan dalam *array iscSim*. *Return* dari fungsi *getSIM* berupa *array iscSim*.

5.5 Implementasi Penghitungan *Maximum Marginal Relevance*

Proses menghitung nilai MMR dari masing-masing kalimat memanfaatkan nilai *similarity* antar kalimat yang dihitung sebelumnya. Penghitungan *Maximum Marginal Relevance* disajikan dalam bentuk fungsi. Implementasi kode program akan ditampilkan pada Kode Program 5.5.

Penghitungan Maximum Marginal Relevance

```
1   def getMMR(SIM):
2       i = 1
3       maxSim = []
4       mmrList = []
5       k = 0.8
6
7       for dok in SIM:
8           val = 0
```

```
9           for kata in dok:
10              if kata != dok[0] and kata != dok[i]:
11                  if kata > val:
12                      val = kata
13          maxSim.append(val)
14          i += 1
15
16      j = 0
17      for dok in SIM:
18          mmrVal = k*dok[0]-(1-k)*maxSim[j]
19          mmrList.append(mmrVal)
20          j += 1
21
22      return mmrList
```

Kode Program 5.5 Implementasi Penghitungan *Maximum Marginal Relevance*

Penjelasan dari Kode Program 5.4 mengenai implementasi penghitungan *Maximum Marginal Relevance* yaitu.

1. Baris 1 merupakan deklarasi fungsi bernama *getMMR* dengan parameter *SIM*.
2. Baris 2-5 merupakan deklarasi dari *array* dengan nama *maxSim* dan *mmrList*, kemudian inisialisasi variabel *i* bernilai 1 dan variabel *k* bernilai 0,8.
3. Baris 7-14 merupakan proses mencari nilai *maximum* dari nilai *similarity* kalimat dengan kalimat lainnya.
4. Baris 16 merupakan inisialisasi variabel *j* bernilai 0.
5. Baris 17-22 merupakan proses menghitung nilai *Maximum Marginal Relevance* masing-masing kalimat, kemudian disimpan dalam *array mmrList*. *Return* dari fungsi *getMMR* berupa *array* mmrList.

5.6 Implementasi Pembentukan Ringkasan

Pembentukan ringkasan memanfaatkan nilai MMR yang dicari sebelumnya. Jumlah kata yang diambil sebagai ringkasan mengacu pada persentase yang ditentukan yaitu 10%, 25%, dan 50%. Implementasi kode program dapat dilihat pada Kode Program 5.6.

```
Pembentukan Ringkasan
1   def getRingkasan(MMR, teks, persentase):
2       jumlahKalimat = round((persentase*(len(teks)-1))/100)
3       dokList = []
4       for i in range(1, len(teks)):
5           dokList.append(i)
6
7       mmrDict = {}
8       mmrDict = dict.fromkeys(dokList, 0)
9       for i in range(1, len(teks)):
10          mmrDict[i] = MMR[i-1]
11
12      ringkasan = dict(sorted(mmrDict.items(), key=lambda x:
```

```
13      x[1], reverse=True))
14          ringkasanList = []
15          i = 0
16          for key, value in ringkasan.items():
17              if i == jumlahKalimat:
18                  break
19              ringkasanList.append(key)
20              i += 1
21          ringkasanList.sort()
22
23          print("Ringkasan", persentase, "%")
24          print("Query :", teks[0])
25          for indeks in ringkasanList:
26              print(teks[indeks])
```

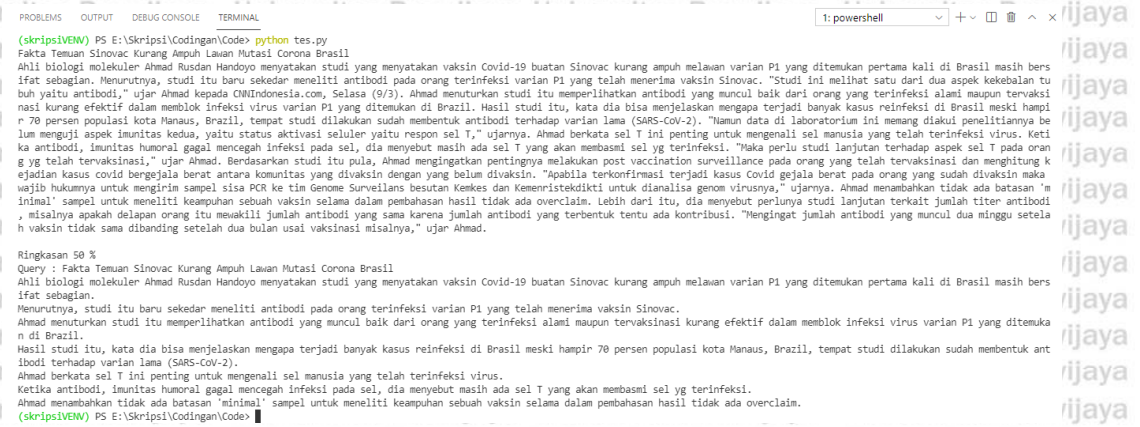
Kode Program 5.6 Implementasi Pembentukan Ringkasan

Penjelasan dari Kode Program 5.6 mengenai implementasi pembobotan TFIDF yaitu.

1. Baris 1 merupakan deklarasi fungsi dengan nama *getRingkasan* dengan parameter *MMR*, *teks*, dan *persentase*.
2. Baris 2 merupakan proses menghitung jumlah kalimat yang membentuk ringkasan berdasarkan persentase yang ditentukan.
3. Baris 3-5 merupakan proses membuat *array dokList* yang berisi kalimat-kalimat dari teks berita selain kalimat *query*.
4. Baris 7-10 merupakan proses membuat *dictionary mmrDict* yang berisi nilai MMR kalimat pertama sampai kalimat terakhir.
5. Baris 12-13 merupakan proses mengurutkan *mmrDict* berdasarkan *value*-nya dari yang terbesar hingga terkecil.
6. Baris 14-21 merupakan proses membuat *array ringkasanList* yang berisi indeks kalimat yang termasuk dalam ringkasan.
7. Baris 23-26 merupakan proses mencetak ringkasan dengan cara mencetak isi dari teks berdasarkan indeks dari *array ringkasanList*.

5.7 Tampilan Program

Hasil keluaran dari sistem yaitu mencetak judul dan teks berita secara lengkap kemudian mencetak persentase yang digunakan untuk membuat ringkasan. *Query* dan ringkasan dicetak perbaris untuk masing-masing kalimatnya.



Gambar 5.1 Hasil Keluaran Sistem Peringkasan Otomatis

BAB 6 PENGUJIAN DAN ANALISIS

Data yang digunakan dalam penelitian ini berupa teks berita dengan jumlah 10 teks berita. Masing-masing teks berita akan diringkas oleh sistem dengan 3 macam persentase yaitu 10%, 25%, dan 50%. Ringkasan dibuat berdasarkan kalimat dalam teks berita yang mirip dengan *query* yang merupakan judul dari masing-masing teks berita. Pada bab ini akan dijelaskan tentang proses pengujian dan analisis dari hasil uji pembuatan ringkasan dengan metode MMR. Pengujian dilakukan dengan cara membandingkan ringkasan yang dibuat oleh sistem dengan ringkasan yang dibuat secara manual oleh pakar.

6.1 Pengujian Ringkasan dengan ROUGE-L

Pada bagian ini, ringkasan oleh sistem akan diuji dengan menggunakan metode ROUGE-L. Data uji yang ditampilkan diambil dari salah satu dari 10 data berita yang digunakan dalam penelitian ini. Data uji yang digunakan merupakan dokumen ke-7 dengan 16 kalimat, ditampilkan dalam Tabel 6.1.

Tabel 6.1 Data Uji

Judul	Teks Berita
ODHA Rentan Terkena Tuberkulosis	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). Data dari lembaga pengelola HIV/AIDS Rumah Sakit Umum Daerah (RSUD) Abdul Muluk menyebutkan sampai bulan November ada 7.366 orang yang melakukan tes HIV, hasilnya sebanyak 297 pengidap HIV baru. "Tahun ini kami melakukan pemeriksaan terhadap 766 odha dan hasilnya sebanyak 42 orang terkena TB," kata Pengelola HIV/AIDS Provinsi Lampung Otta Nur Kirana pada Kamis (1/12/2016). Dia menjelaskan, kekebalan tubuh yang rendah bisa menyebabkan seseorang mudah terserang TB. Karena itu, ODHA termasuk kelompok yang sangat terserang. "Untuk itu ODHA di Lampung wajib mengikuti screening Tuberkulosis sebulan sekali," kata Otta. Petugas akan mengajukan pertanyaan apakah pasien batuk, mengeluarkan keringat pada malam hari, berat badan menurun dan adakah pembesaran kelenjar. "Jika satu dari indikasi yang kami tanyakan itu ada, kami akan melakukan penegakan diagnosa TB," kata dia lagi. Memperingati hari AIDS sedunia, para aktivis HIV/AIDS mengajak masyarakat agar tidak mengasingkan ODHA, untuk mencegah penularan lebih luas lagi. Jika masyarakat menerima ODHA, diharapkan perilaku ODHA jadi lebih baik. Sehingga, penyebaran HIV/AIDS bisa dicegah. Menurut Otta, untuk mengidentifikasi dan melakukan pengendalian HIV/AIDS pada populasi kunci lebih mudah, ketimbang pada populasi jembatan. "Tetapi kami kesulitan untuk mengindentifikasi populasi jembatan (pengguna seks). Mereka tidak ada wadah yang memudahkan kami untuk meminta keterangan," ujarnya. Kelompok populasi jembatan inilah yang menurutnya patut untuk diwaspadai, karena sebagian besar ibu rumah tangga sudah banyak menjadi korban. Termasuk kasus HIV di Lampung, 800 ODHA yang aktif memeriksakan kesehatannya 40 persen adalah ibu rumah tangga yang tertular dari suaminya sendiri.

6.1.1. Pengujian Ringkasan dengan Persentase 10%

Ringkasan dengan persentase 10% yang dihasilkan dari data uji pada Tabel 6.1 berjumlah 2 kalimat, jumlah kalimat tersebut merupakan pembulatan dari nilai 1,6. Pengujian dilakukan dengan cara membandingkan ringkasan yang dibuat sistem dan ringkasan yang dibuat secara manual oleh pakar. Ringkasan oleh pakar dan sistem untuk persentase 10% akan ditampilkan dalam Tabel 6.2.

Tabel 6.2 Ringkasan Berita dengan Persentase 10% untuk Dokumen ke-7

	Teks Ringkasan
Ringkasan Pakar	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). Dia menjelaskan, kekebalan tubuh yang rendah bisa menyebabkan seseorang mudah terserang TB.
Ringkasan Sistem	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). "Untuk itu ODHA di Lampung wajib mengikuti screening Tuberkulosis sebulan sekali," kata Otta.

Pengujian terhadap ringkasan sistem dan ringkasan manual dengan metode ROUGE-L menghasilkan nilai *precision, recall*, dan *f-measure* yang ditampilkan pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian Ringkasan dengan Persentase 10%

Dokumen	*Precision*	*Recall*	*F-Measure*
1	0,521	0,351	0,352
2	0,992	1	0,998
3	0,997	1	0,999
4	0,992	1	0,999
5	0,992	1	0,999
6	0,819	0,296	0,299
7	0,641	0,652	0,652
8	0,801	0,379	0,382
9	0,994	1	0,999
10	0,994	1	0,999
Rata-Rata	0,8743	0,7678	0,7678

6.1.2. Pengujian Ringkasan dengan Persentase 25%

Ringkasan dengan persentase 25% yang dihasilkan dari data uji pada Tabel 6.1 berjumlah 4 kalimat. Pengujian dilakukan dengan melakukan perbandingan antara ringkasan yang dibuat sistem dan ringkasan yang dibuat secara manual oleh pakar. Ringkasan oleh pakar dan sistem untuk persentase 25% akan ditampilkan dalam Tabel 6.4.

Tabel 6.4 Ringkasan Sistem dengan Persentase 25% untuk Dokumen ke-7

	Teks Ringkasan
Ringkasan Pakar	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). Dia menjelaskan, kekebalan tubuh yang rendah bisa menyebabkan seseorang mudah terserang TB.

	Memperingati hari AIDS sedunia, para aktivis HIV/AIDS mengajak masyarakat agar tidak mengasingkan ODHA, untuk mencegah penularan lebih luas lagi. Menurut Otta, untuk mengidentifikasi dan melakukan pengendalian HIV/AIDS pada populasi kunci lebih mudah, ketimbang pada populasi jembatan.
Ringkasan Sistem	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). "Tahun ini kami melakukan pemeriksaan terhadap 766 odha dan hasilnya sebanyak 42 orang terkena TB," kata Pengelola HIV/AIDS Provinsi Lampung Otta Nur Kirana pada Kamis (1/12/2016). Karena itu, ODHA termasuk kelompok yang sangat terserang. "Untuk itu ODHA di Lampung wajib mengikuti screening Tuberkulosis sebulan sekali," kata Otta.

Pengujian terhadap ringkasan sistem dan ringkasan manual dengan metode ROUGE-L menghasilkan nilai *precision, recall*, dan *f-measure* yang ditampilkan pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian Ringkasan dengan Persentase 25%

Dokumen	*Precision*	*Recall*	*F-Measure*
1	0,708	0,579	0,581
2	0,653	0,551	0,552
3	0,66	0,865	0,861
4	0,667	0,577	0,578
5	0,733	0,687	0,687
6	0,768	0,747	0,748
7	0,549	0,494	0,495
8	0,697	0,426	0,428
9	0,639	0,673	0,672
10	0,971	0,447	0,45
Rata-Rata	0,7045	0,6046	0,6052

6.1.3. Pengujian Ringkasan dengan Persentase 50%

Ringkasan dengan persentase 50% yang dihasilkan dari data uji pada Tabel 6.1 berjumlah 8 kalimat. Pengujian dilakukan dengan membandingkan ringkasan yang dibuat sistem dan ringkasan yang dibuat secara manual oleh pakar. Ringkasan oleh pakar dan sistem untuk persentase 25% akan ditampilkan dalam Tabel 6.6.

Tabel 6.6 Ringkasan Sistem dengan Persentase 50% untuk Dokumen ke-7

	Teks Ringkasan
Ringkasan Pakar	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). Dia menjelaskan, kekebalan tubuh yang rendah bisa menyebabkan seseorang mudah terserang TB. Karena itu, ODHA termasuk kelompok yang sangat terserang. Memperingati hari AIDS sedunia, para aktivis HIV/AIDS mengajak masyarakat agar tidak mengasingkan ODHA, untuk mencegah penularan lebih luas lagi. Jika masyarakat menerima ODHA,

	diharapkan perilaku ODHA jadi lebih baik. Sehingga, penyebaran HIV/AIDS bisa dicegah. Menurut Otta, untuk mengidentifikasi dan melakukan pengendalian HIV/AIDS pada populasi kunci lebih mudah, ketimbang pada populasi jembatan. Termasuk kasus HIV di Lampung, 800 ODHA yang aktif memeriksakan kesehatannya 40 persen adalah ibu rumah tangga yang tertular dari suaminya sendiri.
Ringkasan Sistem	Orang dengan HIV/AIDS (ODHA) rentan tertular penyakit Tuberkulosis (TB). "Tahun ini kami melakukan pemeriksaan terhadap 766 odha dan hasilnya sebanyak 42 orang terkena TB," kata Pengelola HIV/AIDS Provinsi Lampung Otta Nur Kirana pada Kamis (1/12/2016). Karena itu, ODHA termasuk kelompok yang sangat terserang. "Untuk itu ODHA di Lampung wajib mengikuti screening Tuberkulosis sebulan sekali," kata Otta. Petugas akan mengajukan pertanyaan apakah pasien batuk, mengeluarkan keringat pada malam hari, berat badan menurun dan adakah pembesaran kelenjar. Memperingati hari AIDS sedunia, para aktivis HIV/AIDS mengajak masyarakat agar tidak mengasingkan ODHA, untuk mencegah penularan lebih luas lagi. Jika masyarakat menerima ODHA, diharapkan perilaku ODHA jadi lebih baik. Termasuk kasus HIV di Lampung, 800 ODHA yang aktif memeriksakan kesehatannya 40 persen adalah ibu rumah tangga yang tertular dari suaminya sendiri.

Pengujian terhadap ringkasan sistem dan ringkasan manual dengan metode ROUGE-L menghasilkan nilai *precision, recall*, dan *f-measure* yang ditampilkan pada Tabel 6.7.

Tabel 6.7 Hasil Pengujian Ringkasan dengan Persentase 50%

Dokumen	*Precision*	*Recall*	*F-Measure*
1	0,571	0,441	0,442
2	0,597	0,563	0,563
3	0,598	0,742	0,739
4	0,454	0,613	0,609
5	0,714	0,706	0,706
6	0,808	0,695	0,696
7	0,598	0,712	0,709
8	0,844	0,767	0,768
9	0,565	0,683	0,681
10	0,841	0,849	0,849
Rata-Rata	0,659	0,6771	0,6762

6.2 Analisis Pengujian

Berdasarkan hasil pengujian pada Tabel 6.3 dokumen ke-1, dokumen ke-6, dan dokumen ke-8 mendapat hasil yang rendah dibandingkan dengan dokumen lain pada persentase yang sama. Hasil yang rendah tersebut dikarenakan pada

persentase 10% hanya mengambil 1 atau 2 kalmat untuk membentuk ringkasan, sehingga saat ringkasan dibandingkan maka terdapat hasil yang jauh berbeda antara yang mirip dan tidak mirip dengan ringkasan manual. Pada ringkasan yang mirip akan mendapat nilai precision, recall, dan f-measure yang tinggi tapi pada ringkasan yang tidak mirip akan mendapat hasil yang rendah namun tidak sampai sama dengan 0. Metode ROUGE-L melakukan evaluasi pada ringkasan dengan membandingkan susunan huruf dari masing-masing ringkasan sehingga sangat kecil kemungkinan mendapat hasil sama dengan 0 pada saat membandingkan kedua ringkasan.

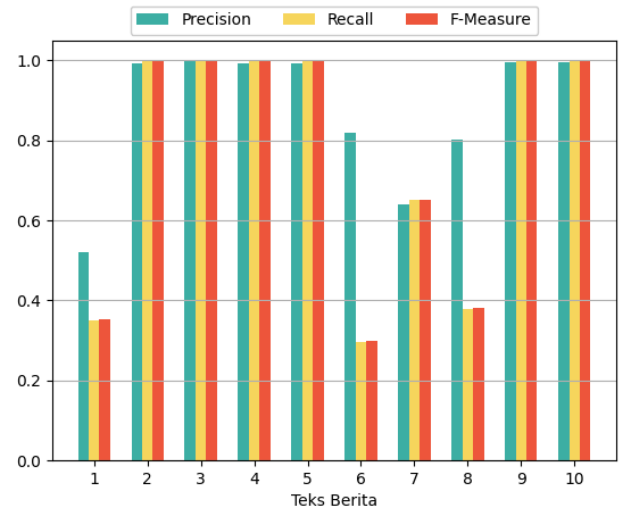
Pada Tabel 6.5 dan Tabel 6.7, nilai precision, recall, dan f-measure yang didapatkan memiliki jarak yang tidak terlalu jauh antar dokumennya. Hal tersebut disebabkan karena pada ringkasan dengan persentase 25% dan persentase 50% jumlah kalimat yang menyusun ringkasan lebih dari 4 kalimat. Semakin banyak jumlah kalimat yang dibandingkan maka semakin tinggi juga kemungkinan adanya susunan huruf yang mirip dari kedua ringkasan yang dibandingkan.

Pengujian yang dilakukan terhadap 10 teks berita dengan menggunakan metode ROUGE-L menghasilkan hasil berupa nilai *precision, recall,* dan *f-measure*. Hasil rata-rata nilai *precision*, *recall*, dan *f-measure* akan ditampilkan pada Tabel 6.8.

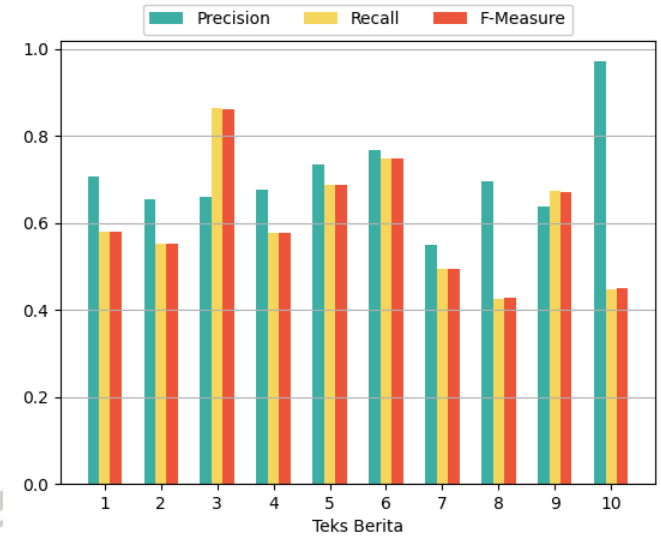
Tabel 6.8 Hasil Rata-Rata Pengujian dari 10 Teks Berita yang Diuji

Ringkasan	*Precision*	*Recall*	*F-Measure*
Persentase 10%	0,8743	0,7678	0,7678
Persentase 25%	0,7045	0,6046	0,6052
Persentase 50%	0,659	0,6771	0,6762

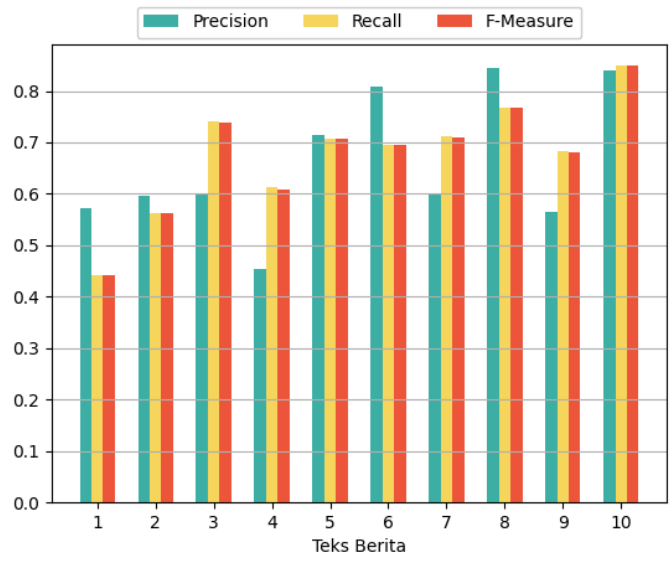
Nilai *precision, recall,* dan *f-measure* untuk masing-masing ringkasan pada setiap persentasenya akan ditampilkan dalam grafik pada Gambar 6.1, Gambar 6.2, dan Gambar 6.3.



Gambar 6.1 Grafik hasil uji ringkasan persentase 10%



Gambar 6.2 Grafik hasil uji ringkasan persentase 25%



Gambar 6.3 Grafik hasil uji ringkasan persentase 50%

Berdasarkan grafik pada Gambar 6.1 terdapat 6 ringkasan berita yang menghasilkan nilai *precision, recall,* dan *f-measure* mendekati 1 yang mengartikan bahwa kedua ringkasan yang dibandigkan memiliki kemiripan yang tinggi. Grafik pada Gambar 6.2 dan Gambar 6.3 untuk ringkasan persentase 25% dan 50% menampilkan nilai *precision, recall,* dan *f-measure* yang lebih rendah dibandingkan dengan hasil pengujian pada ringkasan persentase 10%, dengan kata lain dapat diartikan bahwa saat semakin banyak jumlah kalimat dalam ringkasan yang dibandingkan maka lebih besar kemungkinan ringkasan tersebut mendapatkan hasil *precision, recall,* dan *f-measure* yang rendah. Berdasarkan

analisis tersebut dapat ditarik kesimpulan bahwa metode ROUGE-L merupakan metode yang baik digunakan sebagai metode evaluasi ringkasan dengan jumlah kalimat yang sedikit dan kurang baik untuk ringkasan dengan jumlah kalimat yang banyak.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, ada 3 hal yang dapat disimpulkan yaitu.

1. Dalam melakukan peringkasan teks otomatis untuk suatu dokumen teks, metode Maximum Marginal Relevance terbilang sangat dipengaruhi oleh beberpa faktor seperti *query* dan juga metode untuk mencari nilai *similarity*. Proses peringkasan teks diawali dengan cara melakukan *preprocessing* terhadap teks berita hingga mendapatkan beberapa term yang akan dihitung bobotnya dengan metode TF-IDF. Setelah mendapat bobot masing-masing *term*, setiap kalimat akan dihitung nilai kemiripannya dengan metode Improved Sqrt-Cosine Similarity. Nilai kemiripan antar kalimat tersebut akan digunakan untuk menghitung nilai Maximum Marginal Relevance, semakin besar nilai MMR maka kalimat tersebut dianggap semakin mirip atau mewakili query yang digunakan begitu juga sebaliknya.
2. Hasil uji pada ringkasan dengan persentase 10% mendapat nilai yang tinggi (mendekati 1). Hal tersebut dikarenakan pada persentase 10% kebanyakan ringkasan yang dihasilkan hanya terdiri dari 1 kalimat saja sehingga saat kalimat yang dibandingkan sama maka akan menghasilkan nilai yang tinggi namun pada saat kalimat yang dibandingkan berbeda hasil yang didapat tidak sama dengan 0 karena metode pengujian ROUGE-L melakukan perbandingan untuk setiap runtutan huruf yang menyusun sebuah kalimat. Tingkat kemiripan antara ringkasan oleh sistem dengan ringkasan oleh pakar secara manual terbilang rendah karena ringkasan sistem terdiri dari kalimat yang dianggap mirip dengan *query* sedangkan ringkasan oleh pakar dibuat berdasarkan kalmat berhubungan dengan kalimat utama atau inti dari teks berita tersebut. Hasil rata-rata nilai *precision*, *recall*, dan *f-measure* dari pengujian yang dilakukan akan ditampilkan pada Tabel 7.1.

Tabel 7.1 Hasil Rata-Rata Nilai Precision, Recall, dan F-Measure

Ringkasan	*Precision*	*Recall*	*F-Measure*
Persentase 10%	0,8743	0,7678	0,7678
Persentase 25%	0,7045	0,6046	0,6052
Persentase 50%	0,659	0,6771	0,6762

7.2 Saran

Penelitian peringkasan teks otomatis dengan menggunakan metode Maximum Marginal Relevance masih memiliki beberapa kekurangan yang harus diperbaiki. Beberapa saran yang penulis dapat sampaikan untuk penelitian selanjutnya dengan topik seperti penelitian ini yaitu.

1. Menggunakan berita dengan jumlah kalimat yang lebih banyak untuk menghindari adanya ringkasan yang hanya memiliki 1 kalimat didalamnya.
2. Menggunakan kalimat utama sebagai query sebagai dasar membentuk ringkasan oleh sistem, karena kalimat utama lebih mewakili isi dari teks yang ingin diringkas.
3. Meminta lebih dari 1 pakar untuk membuat ringkasan manual untuk mengurangi tingkat subjektivitas dari ringkasan yang dibuat.

DAFTAR REFERENSI

Anh, B. T. M., My, N. T. & Trang, N. T. T., 2019. Enhanced Genetic Algorithm for Single Document Extractive Summarization.

Budiman, A. E. & Widjaja, A., 2020. Analisis Pengaruh Teks Preprocessing Terhadap Deteksi Plagiarisme pada Dokumen Tugas Akhir.

El-Kassas, W. S., Salama, C. R., Rafea, A. A. & Mohamed, H. K., 2020. Automatic Text Summarization: A Comprehensive Survey.

Erwin, I. M., R., Prakasa, E. & Sugiarto, B., 2019. KAYU7NET: Identifikasi dan Evaluasi F-Measure Citra Kayu Berbasis Deep Convolution Neural Network (DCNN).

Herwijayanti, B., Ratnawati, D. E. & Muflikhah, L., 2018. Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similiarity.

Indriani, A., 2014. Maximum Marginal Relevance untuk Peringkasan Teks Otomatis Sinopsis Buku Berbahasa Indonesia.

Khairunnisa, S., A. & Faraby, S. A., 2021. Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19).

Lin, C.-Y., 2004. Looking for a Few Good Metrics: Rouge and its Evaluation.

Lin, C.-Y., 2004. Rouge: A Package for Automatic Evaluation of Summaries.

M. Romli, A. S., 1999. *Jurnalistik Praktis Untuk Pemula.* s.l.:Penerbit PT Remaja Rosdakarya.

Mustaqhfiri, M., Abidin, Z. & Kusumawati, R., 2011. Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance.

Prabowo, D. A. et al., 2016. TF-IDF-Enhanced Genetic Algorithm untuk Extractive Automatic Text Summarization.

Ridok, A., 2014. Peringkasan Dokumen Bahasa Indonesia Berbasis Non-Negative Matrix Factorization (NMF).

Saleh, P. N., 2020. Implementasi Algoritma Longest Common Subsequence dengan Algoritma Genetika pada Permainan Word Search Puzzle.

Saraswati, N. F., I. & Perdana, R. S., 2018. Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia.

Siyoto, S. & Sodik, M. A., 2015. *Dasar Metodologi Penelitian.* s.l.:Literasi Media Publishing.

Sohangir, S. & Wang, D., 2017. Improved Sqrt-Cosine Similarity Measurement.

Somantri, P. G., Komarudin, A. & Ilyas, R., 2018. Peringkasan Teks Otomatis Berita Berdasarkan Klasifikasi Kalimat Menggunakan Support Vector Machine.

Vijayarani, S., Ilamathi, J. & N., 2020. Preprocessing Techniques for Text Mining - An Overview.

Widyassari, A. P. et al., 2020. Review of automatic text summarization techniques & methods.

LAMPIRAN

LAMPIRAN A HASIL PREPROCESSING

Proses	Dokumen	Hasil
Case Folding	Q	fakta temuan sinovac kurang ampuh lawan mutasi corona brasil
	D1	ahli biologi molekuler ahmad rusdan handoyo menyatakan studi yang menyatakan vaksin covid-19 buatan sinovac kurang ampuh melawan varian p1 yang ditemukan pertama kali di brasil masih bersifat sebagian.
	D2	menurutnya, studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian p1 yang telah menerima vaksin sinovac.
	D3	"studi ini melihat satu dari dua aspek kekebalan tubuh yaitu antibodi," ujar ahmad kepada cnnindonesia.com, selasa (9/3).
	D4	ahmad menuturkan studi itu memperlihatkan antibodi yang muncul baik dari orang yang terinfeksi alami maupun tervaksinasi kurang efektif dalam memblok infeksi virus varian p1 yang ditemukan di brazil.
	D5	hasil studi itu, kata dia bisa menjelaskan mengapa terjadi banyak kasus reinfeksi di brasil meski hampir 70 persen populasi kota manaus, brazil, tempat studi dilakukan sudah membentuk antibodi terhadap varian lama (sars-cov-2).
	D6	"namun data di laboratorium ini memang diakui penelitiannya belum menguji aspek imunitas kedua, yaitu status aktivasi seluler yaitu respon sel t," ujarnya.
	D7	ahmad berkata sel t ini penting untuk mengenali sel manusia yang telah terinfeksi virus.
	D8	ketika antibodi, imunitas humoral gagal mencegah infeksi pada sel, dia menyebut masih ada sel t yang akan membasmi sel yg terinfeksi.
	D9	"maka perlu studi lanjutan terhadap aspek sel t pada orang yg telah tervaksinasi," ujar ahmad.
	D10	berdasarkan studi itu pula, ahmad mengingatkan pentingnya melakukan post vaccination surveillance pada orang yang telah tervaksinasi dan menghitung kejadian kasus covid bergejala berat antara komunitas yang divaksin dengan yang belum divaksin.
	D11	"apabila terkonfirmasi terjadi kasus covid gejala berat pada orang yang sudah divaksin maka wajib hukumnya untuk mengirim sampel sisa pcr ke tim genome surveilans besutan kemkes dan kemenristekdikti untuk dianalisa genom virusnya," ujarnya.
	D12	ahmad menambahkan tidak ada batasan 'minimal' sampel untuk meneliti keampuhan sebuah vaksin

		selama dalam pembahasan hasil tidak ada overclaim.
	D13	lebih dari itu, dia menyebut perlunya studi lanjutan terkait jumlah titer antibodi, misalnya apakah delapan orang itu mewakili jumlah antibodi yang sama karena jumlah antibodi yang terbentuk tentu ada kontribusi.
	D14	"mengingat jumlah antibodi yang muncul dua minggu setelah vaksin tidak sama dibanding setelah dua bulan usai vaksinasi misalnya," ujar ahmad.
	Q	fakta temuan sinovac kurang ampuh lawan mutasi corona brasil
	D1	ahli biologi molekuler ahmad rusdan handoyo menyatakan studi yang menyatakan vaksin covid buatan sinovac kurang ampuh melawan varian p yang ditemukan pertama kali di brasil masih bersifat sebagian
	D2	menurutnya studi itu baru sekedar meneliti antibodi pada orang terinfeksi varian p yang telah menerima vaksin sinovac
	D3	studi ini melihat satu dari dua aspek kekebalan tubuh yaitu antibodi ujar ahmad kepada cnnindonesia com selasa
	D4	ahmad menuturkan studi itu memperlihatkan antibodi yang muncul baik dari orang yang terinfeksi alami maupun tervaksinasi kurang efektif dalam memblok infeksi virus varian p yang ditemukan di brazil
	D5	hasil studi itu kata dia bisa menjelaskan mengapa terjadi banyak kasus reinfeksi di brasil meski hampir persen populasi kota manaus brazil tempat studi dilakukan sudah membentuk antibodi terhadap varian lama sars cov
Cleaning	D6	namun data di laboratorium ini memang diakui penelitiannya belum menguji aspek imunitas kedua yaitu status aktivasi seluler yaitu respon sel t ujarnya
	D7	ahmad berkata sel t ini penting untuk mengenali sel manusia yang telah terinfeksi virus
	D8	ketika antibodi imunitas humoral gagal mencegah infeksi pada sel dia menyebut masih ada sel t yang akan membasmi sel yg terinfeksi
	D9	maka perlu studi lanjutan terhadap aspek sel t pada orang yg telah tervaksinasi ujar ahmad
	D10	berdasarkan studi itu pula ahmad mengingatkan pentingnya melakukan post vaccination surveillance pada orang yang telah tervaksinasi dan menghitung kejadian kasus covid bergejala berat antara komunitas yang divaksin dengan yang belum divaksin
	D11	apabila terkonfirmasi terjadi kasus covid gejala berat pada orang yang sudah divaksin maka wajib hukumnya untuk mengirim sampel sisa pcr ke tim genome surveilans besutan kemkes dan

		kemenristekdikti untuk dianalisa genom virusnya ujarnya
	D12	ahmad menambahkan tidak ada batasan minimal sampel untuk meneliti keampuhan sebuah vaksin selama dalam pembahasan hasil tidak ada overclaim
	D13	lebih dari itu dia menyebut perlunya studi lanjutan terkait jumlah titer antibodi misalnya apakah delapan orang itu mewakili jumlah antibodi yang sama karena jumlah antibodi yang terbentuk tentu ada kontribusi
	D14	mengingat jumlah antibodi yang muncul dua minggu setelah vaksin tidak sama dibanding setelah dua bulan usai vaksinasi misalnya ujar ahmad
Tokenisasi	Q	'fakta', 'temuan', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
	D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'menyatakan', 'studi', 'yang', 'menyatakan', 'vaksin', 'covid', 'buatan', 'sinovac', 'kurang', 'ampuh', 'melawan', 'varian', 'p', 'yang', 'ditemukan', 'pertama', 'kali', 'di', 'brasil', 'masih', 'bersifat', 'sebagian'
	D2	'menurutnya', 'studi', 'itu', 'baru', 'sekedar', 'meneliti', 'antibodi', 'pada', 'orang', 'terinfeksi', 'varian', 'p', 'yang', 'telah', 'menerima', 'vaksin', 'sinovac'
	D3	'studi', 'ini', 'melihat', 'satu', 'dari', 'dua', 'aspek', 'kekebalan', 'tubuh', 'yaitu', 'antibodi', 'ujar', 'ahmad', 'kepada', 'cnnindonesia', 'com', 'selasa'
	D4	'ahmad', 'menuturkan', 'studi', 'itu', 'memperlihatkan', 'antibodi', 'yang', 'muncul', 'baik', 'dari', 'orang', 'yang', 'terinfeksi', 'alami', 'maupun', 'tervaksinasi', 'kurang', 'efektif', 'dalam', 'memblok', 'infeksi', 'virus', 'varian', 'p', 'yang', 'ditemukan', 'di', 'brazil'
	D5	'hasil', 'studi', 'itu', 'kata', 'dia', 'bisa', 'menjelaskan', 'mengapa', 'terjadi', 'banyak', 'kasus', 'reinfeksi', 'di', 'brasil', 'meski', 'hampir', 'persen', 'populasi', 'kota', 'manaus', 'brazil', 'tempat', 'studi', 'dilakukan', 'sudah', 'membentuk', 'antibodi', 'terhadap', 'varian', 'lama', 'sars', 'cov'
	D6	'namun', 'data', 'di', 'laboratorium', 'ini', 'memang', 'diakui', 'penelitiannya', 'belum', 'menguji', 'aspek', 'imunitas', 'kedua', 'yaitu', 'status', 'aktivasi', 'seluler', 'yaitu', 'respon', 'sel', 't', 'ujarnya'
	D7	'ahmad', 'berkata', 'sel', 't', 'ini', 'penting', 'untuk', 'mengenali', 'sel', 'manusia', 'yang', 'telah', 'terinfeksi', 'virus'
	D8	'ketika', 'antibodi', 'imunitas', 'humoral', 'gagal', 'mencegah', 'infeksi', 'pada', 'sel', 'dia', 'menyebut', 'masih', 'ada', 'sel', 't', 'yang', 'akan', 'membasmi', 'sel', 'yg', 'terinfeksi'
	D9	'maka', 'perlu', 'studi', 'lanjutan', 'terhadap', 'aspek', 'sel', 't', 'pada', 'orang', 'yg', 'telah', 'tervaksinasi',

		'ujar', 'ahmad'
	D10	'berdasarkan', 'studi', 'itu', 'pula', 'ahmad', 'mengingatkan', 'pentingnya', 'melakukan', 'post', 'vaccination', 'surveillance', 'pada', 'orang', 'yang', 'telah', 'tervaksinasi', 'dan', 'menghitung', 'kejadian', 'kasus', 'covid', 'bergejala', 'berat', 'antara', 'komunitas', 'yang', 'divaksin', 'dengan', 'yang', 'belum', 'divaksin'
	D11	'apabila', 'terkonfirmasi', 'terjadi', 'kasus', 'covid', 'gejala', 'berat', 'pada', 'orang', 'yang', 'sudah', 'divaksin', 'maka', 'wajib', 'hukumnya', 'untuk', 'mengirim', 'sampel', 'sisa', 'pcr', 'ke', 'tim', 'genome', 'surveilans', 'besutan', 'kemkes', 'dan', 'kemenristekdikti', 'untuk', 'dianalisa', 'genom', 'virusnya', 'ujarnya'
	D12	'ahmad', 'menambahkan', 'tidak', 'ada', 'batasan', 'minimal', 'sampel', 'untuk', 'meneliti', 'keampuhan', 'sebuah', 'vaksin', 'selama', 'dalam', 'pembahasan', 'hasil', 'tidak', 'ada', 'overclaim'
	D13	'lebih', 'dari', 'itu', 'dia', 'menyebut', 'perlunya', 'studi', 'lanjutan', 'terkait', 'jumlah', 'titer', 'antibodi', 'misalnya', 'apakah', 'delapan', 'orang', 'itu', 'mewakili', 'jumlah', 'antibodi', 'yang', 'sama', 'karena', 'jumlah', 'antibodi', 'yang', 'terbentuk', 'tentu', 'ada', 'kontribusi'
	D14	'mengingat', 'jumlah', 'antibodi', 'yang', 'muncul', 'dua', 'minggu', 'setelah', 'vaksin', 'tidak', 'sama', 'dibanding', 'setelah', 'dua', 'bulan', 'usai', 'vaksinasi', 'misalnya', 'ujar', 'ahmad'
Filtering	Q	'fakta', 'temuan', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
	D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'menyatakan', 'studi', 'menyatakan', 'vaksin', 'covid', 'buatan', 'sinovac', 'kurang', 'ampuh', 'melawan', 'varian', 'ditemukan', 'pertama', 'kali', 'brasil', 'bersifat', 'sebagian'
	D2	'menurutnya', 'studi', 'baru', 'sekedar', 'meneliti', 'antibodi', 'orang', 'terinfeksi', 'varian', 'menerima', 'vaksin', 'sinovac'
	D3	'studi', 'melihat', 'satu', 'aspek', 'kekebalan', 'tubuh', 'antibodi', 'ujar', 'ahmad', 'cnnindonesia', 'com', 'selasa'
	D4	'ahmad', 'menuturkan', 'studi', 'memperlihatkan', 'antibodi', 'muncul', 'baik', 'orang', 'terinfeksi', 'alami', 'maupun', 'tervaksinasi', 'kurang', 'efektif', 'memblok', 'infeksi', 'virus', 'varian', 'ditemukan', 'brazil'
	D5	'hasil', 'studi', 'kata', 'menjelaskan', 'terjadi', 'banyak', 'kasus', 'reinfeksi', 'brasil', 'meski', 'hampir', 'persen', 'populasi', 'kota', 'manaus', 'brazil', 'tempat', 'studi', 'dilakukan', 'membentuk', 'antibodi', 'varian', 'lama', 'sars', 'cov'
	D6	'data', 'laboratorium', 'memang', 'diakui', 'penelitiannya', 'menguji', 'aspek', 'imunitas',

		'kedua', 'status', 'aktivasi', 'seluler', 'respon', 'sel', 'ujarnya'
	D7	'ahmad', 'berkata', 'sel', 'penting', 'mengenali', 'sel', 'manusia', 'terinfeksi', 'virus'
	D8	'antibodi', 'imunitas', 'humoral', 'gagal', 'mencegah', 'infeksi', 'sel', 'menyebut', 'sel', 'membasmi', 'sel', 'terinfeksi'
	D9	'perlu', 'studi', 'lanjutan', 'aspek', 'sel', 'orang', 'tervaksinasi', 'ujar', 'ahmad'
	D10	'berdasarkan', 'studi', 'ahmad', 'mengingatkan', 'pentingnya', 'melakukan', 'post', 'vaccination', 'surveillance', 'orang', 'tervaksinasi', 'menghitung', 'kejadian', 'kasus', 'covid', 'bergejala', 'berat', 'komunitas', 'divaksin', 'divaksin'
	D11	'apabila', 'terkonfirmasi', 'terjadi', 'kasus', 'covid', 'gejala', 'berat', 'orang', 'divaksin', 'wajib', 'hukumnya', 'mengirim', 'sampel', 'sisa', 'pcr', 'tim', 'genome', 'surveilans', 'besutan', 'kemkes', 'kemenristekdikti', 'dianalisa', 'genom', 'virusnya', 'ujarnya'
	D12	'ahmad', 'menambahkan', 'batasan', 'minimal', 'sampel', 'meneliti', 'keampuhan', 'sebuah', 'vaksin', 'selama', 'pembahasan', 'hasil', 'overclaim'
	D13	'lebih', 'menyebut', 'perlunya', 'studi', 'lanjutan', 'terkait', 'jumlah', 'titer', 'antibodi', 'misalnya', 'delapan', 'orang', 'mewakili', 'jumlah', 'antibodi', 'sama', 'jumlah', 'antibodi', 'terbentuk', 'kontribusi'
	D14	'mengingat', 'jumlah', 'antibodi', 'muncul', 'minggu', 'vaksin', 'sama', 'dibanding', 'bulan', 'usai', 'vaksinasi', 'misalnya', 'ujar', 'ahmad'
Stemming	Q	'fakta', 'temu', 'sinovac', 'kurang', 'ampuh', 'lawan', 'mutasi', 'corona', 'brasil'
	D1	'ahli', 'biologi', 'molekuler', 'ahmad', 'rusdan', 'handoyo', 'nyata', 'studi', 'nyata', 'vaksin', 'covid', 'buat', 'sinovac', 'kurang', 'ampuh', 'lawan', 'varian', 'temu', 'pertama', 'kali', 'brasil', 'sifat', 'bagi'
	D2	'turut', 'studi', 'baru', 'dar', 'teliti', 'antibodi', 'orang', 'infeksi', 'varian', 'terima', 'vaksin', 'sinovac'
	D3	'studi', 'lihat', 'satu', 'aspek', 'kebal', 'tubuh', 'antibodi', 'ujar', 'ahmad', 'cnnindonesia', 'com', 'selasa'
	D4	'ahmad', 'tutur', 'studi', 'lihat', 'antibodi', 'muncul', 'baik', 'orang', 'infeksi', 'alami', 'maupun', 'vaksinasi', 'kurang', 'efektif', 'blok', 'infeksi', 'virus', 'varian', 'temu', 'brazil'
	D5	'hasil', 'studi', 'kata', 'jelas', 'jadi', 'banyak', 'kasus', 'reinfeksi', 'brasil', 'meski', 'hampir', 'persen', 'populasi', 'kota', 'manaus', 'brazil', 'tempat', 'studi', 'laku', 'bentuk', 'antibodi', 'varian', 'lama', 'sars', 'cov'
	D6	'data', 'laboratorium', 'memang', 'aku', 'teliti', 'uji', 'aspek', 'imunitas', 'dua', 'status', 'aktivasi', 'seluler', 'respon', 'sel', 'ujar'
	D7	'ahmad', 'kata', 'sel', 'penting', 'nali', 'sel', 'manusia',

		'infeksi', 'virus'
	D8	'antibodi', 'imunitas', 'humoral', 'gagal', 'cegah', 'infeksi', 'sel', 'sebut', 'sel', 'basmi', 'sel', 'infeksi'
	D9	'perlu', 'studi', 'lanjut', 'aspek', 'sel', 'orang', 'vaksinasi', 'ujar', 'ahmad'
	D10	'dasar', 'studi', 'ahmad', 'ingat', 'penting', 'laku', 'post', 'vaccination', 'surveillance', 'orang', 'vaksinasi', 'hitung', 'jadi', 'kasus', 'covid', 'gejala', 'berat', 'komunitas', 'vaksin', 'vaksin'
	D11	'apabila', 'konfirmasi', 'jadi', 'kasus', 'covid', 'gejala', 'berat', 'orang', 'vaksin', 'wajib', 'hukum', 'kirim', 'sampel', 'sisa', 'pcr', 'tim', 'genome', 'surveilans', 'besut', 'kemkes', 'kemenristekdikti', 'dianalisa', 'genom', 'virus', 'ujar'
	D12	'ahmad', 'tambah', 'batas', 'minimal', 'sampel', 'teliti', 'ampuh', 'buah', 'vaksin', 'lama', 'bahas', 'hasil', 'overclaim'
	D13	'lebih', 'sebut', 'perlu', 'studi', 'lanjut', 'kait', 'jumlah', 'titer', 'antibodi', 'misal', 'delapan', 'orang', 'wakil', 'jumlah', 'antibodi', 'sama', 'jumlah', 'antibodi', 'bentuk', 'kontribusi'
	D14	'ingat', 'jumlah', 'antibodi', 'muncul', 'minggu', 'vaksin', 'sama', 'banding', 'bulan', 'usai', 'vaksinasi', 'misal', 'ujar', 'ahmad'

LAMPIRAN B PEMBOBOTAN TFIDF

Token	TF															DF	IDF
	Q	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14		
ahli	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
ahmad	0	1	0	1	1	0	0	1	0	1	1	0	1	0	1	8	0,273001
aktivasi	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
aku	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
alami	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
ampuh	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0,69897
antibodi	0	0	1	1	1	1	0	0	1	0	0	0	0	1,477121	1	7	0,330993
apabila	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
aspek	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	3	0,69897
bagi	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
bahas	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176091
baik	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
banding	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1,176091
banyak	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
baru	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
basmi	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1,176

																	091
batas	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176091
bentuk	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	2	0,875061
berat	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	0,875061
besut	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
biologi	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
blok	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
brasil	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	3	0,69897
brazil	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	2	0,875061
buah	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176091
buat	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
bulan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1,176091
cegah	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1,176091
cnnindonesiacom	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
corona	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
covid	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	3	0,69897
dar	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
dasar	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176091

data	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
delapan	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091
dianalisa	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
dua	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
efektif	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
fakta	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
gagal	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1,176091
gejala	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	0,875061
genom	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
genome	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
hampir	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
handoyo	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
hasil	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	2	0,875061
hitung	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176091
hukum	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
humoral	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1,176091
imunitas	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	2	0,875061
infeksi	0	0	1	0	1,301	0	0	1	1,301	0	0	0	0	0	0	4	0,574

					03				03								031
ingat	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2	0,875061
jadi	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	3	0,69897
jelas	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
jumlah	0	0	0	0	0	0	0	0	0	0	0	0	0	1,477121	1	2	0,875061
kait	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091
kali	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
kasus	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	3	0,69897
kata	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	2	0,875061
kebal	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
kemenristekdikti	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
kemkes	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
kirim	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
komunitas	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176091
konfirmasi	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
kontribusi	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091
kota	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
kurang	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0,69897

laboratorium	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
laku	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0,875061
lama	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	2	0,875061
lanjut	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	2	0,875061
lawan	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0,875061
lebih	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091
lihat	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	2	0,875061
manaus	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
manusia	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1,176091
maupun	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
memang	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
meski	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
minggu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1,176091
minimal	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176091
misal	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	0,875061
molekuler	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
muncul	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2	0,875061
mutasi	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176

																	091
nali	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1,176 091
nyata	0	1,301 03	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176 091
orang	0	0	1	0	1	0	0	0	0	1	1	1	0	1	0	6	0,397 94
overclaim	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176 091
pcr	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176 091
penting	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	2	0,875 061
perlu	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	2	0,875 061
persen	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176 091
pertama	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176 091
populasi	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176 091
post	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176 091
reinfeksi	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176 091
respon	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176 091
rusdan	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176 091
sama	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	0,875 061
sampel	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	2	0,875 061
sarscov	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176 091

satu	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
sebut	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	2	0,875061
selasa	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
sel	0	0	0	0	0	0	1	1,30103	1,477121	1	0	0	0	0	0	4	0,574031
seluler	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
sifat	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
sinovac	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0,69897
sisa	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
status	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
studi	0	1	1	1	1	1,30103	0	0	0	1	1	0	0	1	0	8	0,273001
surveilans	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
surveillance	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176091
tambah	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1,176091
teliti	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	3	0,69897
tempat	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1,176091
temu	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0,69897
terima	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
tim	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176

																	091
titer	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091
tubuh	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
turut	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1,176091
tutur	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1,176091
ujar	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1	5	0,477121
uji	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1,176091
usai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1,176091
vaccination	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1,176091
vaksin	0	1	1	0	0	0	0	0	0	0	1,30103	1	1	0	1	6	0,39794
vaksinasi	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	4	0,574031
varian	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	4	0,574031
virus	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	3	0,69897
wajib	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1,176091
wakil	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1,176091

LAMPIRAN C SIMILARITY KALIMAT

Sim	Q	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14
Q	1	0,349	0,085	0	0,129	0,054	0	0	0	0	0	0	0,073	0	0
D1	0,349	1	0,148	0,041	0,146	0,077	0	0,024	0	0,055	0,097	0,050	0,090	0,016	0,045
D2	0,085	0,148	1	0,069	0,196	0,089	0,061	0,077	0,116	0,102	0,097	0,055	0,109	0,095	0,074
D3	0	0,041	0,069	1	0,153	0,047	0,102	0,036	0,039	0,259	0,047	0,033	0,027	0,059	0,109
D4	0,129	0,146	0,196	0,153	1	0,117	0	0,166	0,096	0,175	0,100	0,058	0,021	0,072	0,159
D5	0,054	0,077	0,089	0,047	0,117	1	0	0,075	0,025	0,030	0,141	0,061	0,110	0,089	0,021
D6	0	0	0,061	0,102	0	0	1	0,067	0,140	0,201	0	0,025	0,052	0	0,037
D7	0	0,024	0,077	0,036	0,166	0,075	0,067	1	0,198	0,164	0,115	0,056	0,031	0	0,032
D8	0	0	0,116	0,039	0,096	0,025	0,140	0,198	1	0,108	0	0	0	0,115	0,034
D9	0	0,055	0,102	0,259	0,175	0,030	0,201	0,164	0,108	1	0,172	0,079	0,036	0,281	0,177
D10	0	0,097	0,097	0,047	0,100	0,141	0	0,115	0	0,172	1	0,241	0,054	0,044	0,165
D11	0	0,050	0,055	0,033	0,058	0,061	0,025	0,056	0	0,079	0,241	1	0,075	0,021	0,053
D12	0,073	0,090	0,109	0,027	0,021	0,110	0,052	0,031	0	0,036	0,054	0,075	1	0	0,059
D13	0	0,016	0,095	0,059	0,072	0,089	0	0	0,115	0,281	0,044	0,021	0	1	0,251
D14	0	0,045	0,074	0,109	0,159	0,021	0,037	0,032	0,034	0,177	0,165	0,053	0,059	0,251	1