PENERAPAN ALGORITMA *TEXTRANK* PADA PERINGKASAN TEKS BERITA BERBAHASA INDONESIA DENGAN *WORD2VEC* DAN LSA

SKRIPSI

Oleh :
PRANA WIJAYA PRATAMA NANDANA
NIM. 210605110120

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

PENERAPAN ALGORITMA *TEXTRANK* PADA PERINGKASAN TEKS BERITA BERBAHASA INDONESIA DENGAN *WORD2VEC* DAN LSA

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
PRANA WIJAYA PRATAMA NANDANA
NIM. 210605110120

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

HALAMAN PERSETUJUAN

PENERAPAN ALGORITMA *TEXTRANK* PADA PERINGKASAN TEKS BERITA BERBAHASA INDONESIA DENGAN *WORD2VEC* DAN LSA

SKRIPSI

Oleh :
PRANA WIJAYA PRATAMA NANDANA
NIM. 210605110120

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 14 April 2025

Pembimbing I,	Pembimbing II,
Prof. Dr. Muhammad Faisal, M.T	Syahiduz Zaman, M.Kom
NIP. 19740510 200501 1 007	NIP. 19700502 200501 1 005

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

PENERAPAN ALGORITMA *TEXTRANK* PADA PERINGKASAN TEKS BERITA BERBAHASA INDONESIA DENGAN *WORD2VEC* DAN LSA

SKRIPSI

Oleh :
PRANA WIJAYA PRATAMA NANDANA
NIM. 210605110120

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 06 Mei 2025

Susunan Dewan Penguji

Ketua Penguji	: A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )
Anggota Penguji I	: Fajar Rohman Hariri, M.Kom NIP. 19890515 201801 1 001	( )
Anggota Penguji II	: Prof. Dr. Muhammad Faisal, M.T NIP. 19740510 200501 1 007	( )
Anggota Penguji III	: Syahiduz Zaman, M.Kom NIP. 19700502 200501 1 005	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Prana Wijaya Pratama Nandana
NIM	: 210605110120
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Penerapan Algoritma *TextRank* pada Peringkasan Teks Berita Berbahasa Indonesia dengan *Word2Vec* & LSA

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 06 Mei 2025
Yang membuat pernyataan,

Prana Wijaya Pratama Nandana
NIM.210605110120

MOTTO

... Jalani, Nikmati, Syukuri ...

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT dan terima kasih yang mendalam, penulis mempersembahkan karya sederhana ini kepada kedua orang tua tercinta, keluarga yang selalu mendukung, para dosen pembimbing, para sahabat, serta seluruh pihak yang telah memberikan dukungan moral, doa, dan semangat selama proses penyusunan skripsi ini, hingga akhirnya dapat terselesaikan dengan baik dan tepat waktu.

KATA PENGANTAR

Assalamualaikum Wr.Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat dan kesehatan, sehingga penulis mampu menyelesaikan skripsi ini dengan baik. Penulis menyampaikan ucapan Terima kasih kepada semua pihak yang pernah terlibat langsung maupun tidak langsung dalam menyelesaikan skripsi ini, bukan hanya karena usaha keras dari penulis sendiri, akan tetapi karena adanya dukungan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan M.MT., IPU selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Prof. Dr. Muhammad Faisal, M.T selaku Dosen Pembimbing 1 yang telah membimbing serta memberikan arahan serta motivasi dalam penulisan skripsi dari awal hingga akhir.
5. Syahiduz Zaman, M.Kom selaku Dosen Pembimbing 2 yang telah memberikan bimbingan, arahan serta bantuan dalam terwujudnya karya tulis skripsi ini dari awal hingga akhir.

6. A’la Syauqi, M.Kom selaku penguji I dan Fajar Rohman Hariri, M.Kom selaku penguji II yang telah meluangkan waktunya untuk menguji dan dengan sabar memberi arahan dan saran dalam menyelesaikan skripsi ini.
7. Segenap civitas akademik Program Studi Teknik Informatika, dan seluruh dosen yang telah memberikan ilmu serta arahan semasa kuliah.
8. Ibu Eka Sari Setyowati selaku orangtua penulis yang selalu memberikan dukungan dalam segala hal sehingga penulis dapat menyelesaikan skripsi dengan baik.
9. Teman-teman yang telah memberikan bantuan yang sangat besar dalam penyusunan skripsi ini

Penulis menyadari bahwa skripsi ini masih belum sempurna dan kemungkinan masih terdapat kekurangan di dalamnya. Oleh sebab itu, penulis sangat mengharapkan masukan serta saran yang membangun guna menyempurnakan penelitian ini agar dapat memberikan manfaat yang lebih luas, baik bagi penulis sendiri maupun bagi para pembaca.

Malang, 06 Mei 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Nandana, Prana Wijaya Pratama. 2025. **Penerapan Algoritma *TextRank* pada Peringkasan Teks Berita Berbahasa Indonesia dengan *Word2Vec* dan LSA**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Muhammad Faisal, M.T (II) Syahiduz Zaman, M.Kom.

.

Kata Kunci: Peringkasan, *TextRank*, Word2Vec, LSA.

Dalam era digital yang dipenuhi informasi, masyarakat sering kesulitan menyaring berita yang relevan karena banyaknya konten yang tersedia. Penelitian ini bertujuan untuk merancang sistem peringkasan otomatis untuk berita online berbahasa Indonesia dengan pendekatan ekstraktif menggunakan kombinasi algoritma *TextRank*, *Word2Vec*, dan *Latent Semantic Analysis* (LSA). *TextRank* digunakan untuk menentukan kalimat penting, *Word2Vec* mengubah kata menjadi vektor angka yang dapat mewakili maknanya, dan LSA mereduksi dimensi vektor agar analisis semantik antar kalimat lebih efisien. Dataset yang digunakan adalah Indosum yang berisi 5000 artikel berita dari berbagai topik. Sistem ini dievaluasi menggunakan metrik ROUGE dan penilaian manual oleh validator. Hasil pengujian menunjukkan bahwa tingkat kompresi 30% menghasilkan skor ROUGE-1 sebesar 0.4808, ROUGE-2 sebesar 0.3433, dan ROUGE-L sebesar 0.4675 yang merupakan hasil tertinggi dibandingkan tingkat kompresi lainnya. Penilaian manual juga menunjukkan bahwa ringkasan pada kompresi 30% paling informatif dan koheren, membuktikan bahwa kombinasi ketiga metode tersebut mampu meningkatkan kualitas ringkasan secara signifikan.

ABSTRACT

Nandana, Prana Wijaya Pratama. 2025. **The Implementation of the TextRank Algorithm for Summarizing Indonesian News Texts Using Word2Vec and LSA**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang. Promoter: (I) Prof. Dr. Muhammad Faisal, M.T (II) Syahiduz Zaman, M.Kom.

Keywords: *Summarization*, *TextRank, Word2Vec*, LSA.

In the digital era filled with abundant information, people often struggle to filter relevant news due to the overwhelming amount of content available. This study aims to develop an automatic summarization system for Indonesian online news using an extractive approach by combining the TextRank algorithm, Word2Vec, and Latent Semantic Analysis (LSA). TextRank is used to identify the most important sentences, Word2Vec converts words into numerical vectors that represent their meanings, and LSA reduces the dimensions of these vectors to make semantic analysis between sentences more efficient. The dataset used is Indosum, consisting of 5,000 news articles across various topics. The system is evaluated using ROUGE metrics and manual assessment by a validator. The experimental results show that a 30% compression level yields the highest performance, with ROUGE-1 score of 0.4808, ROUGE-2 of 0.3433, and ROUGE-L of 0.4675. Manual evaluation also indicates that the summaries at 30% compression are the most informative and coherent, proving that the combination of these three methods significantly improves summarization quality.

مستخلص البحث

ناندانا، برانا ويجايا براتاما. 2025. تطبيق خوارزمية TextRank في تلخيص نصوص الأخبار الإندونيسية باستخدام Word2Vec وLSA. أُطرُوحَة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج. المشرف: (أ) الأستاذ الدكتور محمد فيصل، م.ت. (ب) سيادوز زمان، م.كوم.

الكلمات المفتاحية: التلخيص، TextRank، Word2Vec، LSA

في العصر الرقمي المليء بالمعلومات، غالبًا ما يواجه الأشخاص صعوبة في تصفية الأخبار ذات الصلة بسبب الكمية الكبيرة من المحتوى المتاح. تهدف هذه الدراسة إلى تصميم نظام تلخيص تلقائي للأخبار عبر الإنترنت باللغة الإندونيسية باستخدام Lastent Semantic Analysisو Word2Vecو TextRank نهج استخلاصي باستخدام مزيج من خوارزميات بتحويل الكلمات إلى متجهات رقمية يمكنها Word2Vec لتحديد الجمل المهمة، ويقوم TextRank يتم استخدام .(LSA) بتقليل أبعاد المتجهات لجعل التحليل الدلالي بين الجمل أكثر كفاءة. مجموعة البيانات المستخدمة هي LSA تمثيل معناها، ويقوم والتقييم ROUGE والتي تحتوي على 5000 مقالة إخبارية من مواضيع مختلفة. تم تقييم النظام باستخدام مقاييس Indosum -ROUGEتبلغ 0.4808، و ROUGE-1 اليدوي من قبل المحققين. وتظهر نتائج الاختبار أن مستوى ضغط 30 %ينتج درجة تبلغ 0.4675، وهي أعلى النتائج مقارنة بمستويات الضغط الأخرى. وأظهر التقييم اليدوي ROUGE-Lتبلغ 0.3433، و 2 أيضًا أن الملخص عند ضغط 30% كان الأكثر إفادة وتماسكًا، مما يثبت أن الجمع بين الأساليب الثلاثة كان قادرًا على تحسين .جودة الملخص بشكل كبير

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berita online merupakan laporan terbaru mengenai berbagai peristiwa yang dipublikasikan melalui platform digital dan dapat diakses kapan saja serta di mana saja selama terdapat koneksi internet (Sukmono, 2021). Berita online telah menjadi salah satu sumber informasi utama masyarakat Indonesia berdasarkan survei dari *Reuters Institute* dengan presentase 84% pada tahun 2023. Namun, banyaknya jumlah berita yang diterbitkan setiap hari menimbulkan tantangan baru, yakni kesulitan bagi pembaca dalam memilah berita yang relevan dan kredibel, di tengah maraknya fenomena berita palsu yang mengancam kualitas informasi yang diterima oleh publik (Majerczak & Strzelecki, 2022). Seiring dengan pesatnya perkembangan teknologi dan jumlah informasi yang terus meningkat, kemampuan untuk menyampaikan inti berita secara cepat menjadi semakin penting . Untuk mengatasi kesulitan dalam memahami informasi dari teks berita yang panjang dan kompleks, salah satu solusi yang dapat dilakukan untuk mengatasi permasalahan ini adalah teknologi peringkasan teks otomatis, yang mampu menyajikan inti informasi secara ringkas dan efisien, sehingga pembaca dapat memperoleh informasi penting dengan cepat dan efisien (Juan-Manuel & Torres-Moreno, 2014)

Menanggapi tantangan tersebut, berbagai penelitian telah dilakukan untuk mengembangkan teknologi peringkasan teks otomatis guna membantu pembaca memahami isi berita secara lebih efisien. Salah satunya adalah penelitian yang

dilakukan oleh (Wijaya & Girsang, 2024), dalam penelitian tersebut mengusulkan metode hibrida yang menggabungkan algoritma *LexRank* dan YAKE untuk peringkasan teks otomatis pada berita Indonesia. Metode ini bertujuan meningkatkan akurasi dan relevansi ringkasan dengan menggabungkan penilaian pentingnya kalimat menggunakan *LexRank* dan kata kunci melalui YAKE. Pengujian dilakukan pada 5000 artikel berita Indonesia dari dataset Indosum, dengan tujuan meringkas artikel berita menjadi 30% dari panjang asli teksnya. Hasil evaluasi untuk metode *LexRank* dan YAKE menunjukkan nilai *f1-score* mencapai 0,453, yang mencerminkan peningkatan performa dalam mengidentifikasi dan merangkum konten secara lebih relevan dan lugas dibandingkan metode lainnya seperti *LexRank* maupun TextRank + GA.

Meskipun hasil evaluasi pada penelitian sebelumnya menunjukkan adanya peningkatan dalam performa peringkasan, nilai *f1-score* yang diperoleh masih menunjukkan adanya potensi untuk perbaikan lebih lanjut. Hal ini menunjukkan bahwa sistem peringkasan yang ada belum sepenuhnya mampu menangkap esensi informasi dengan tingkat akurasi yang tinggi. Oleh karena itu, perlu dilakukan pendekatan yang lebih mendalam dalam memahami konteks dan hubungan antar kalimat agar ringkasan yang dihasilkan semakin akurat dan relevan.

Pentingnya peningkatan kualitas ringkasan ini tidak hanya berkaitan dengan aspek teknis semata, tetapi juga memiliki dampak sosial yang signifikan. Dalam masyarakat modern, informasi menyebar dengan sangat cepat dan sering kali tanpa proses verifikasi yang memadai (Rahmadhany et al. 2021). Oleh karena itu, kemampuan untuk menyajikan ringkasan yang padat, jelas, dan tetap akurat

menjadi sangat krusial. Hal ini sejalan dengan nilai yang tercantum dalam QS Al-Hujurat:6 sebagai berikut.

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِنْ جَاۤءَكُمْ فَاسِقٌۢ بِنَبَاٍ فَتَبَيَّنُوْٓا اَنْ تُصِيْبُوْا قَوْمًاۢ بِجَهَالَةٍ فَتُصْبِحُوْا عَلٰى مَا فَعَلْتُمْ نٰدِمِيْنَ ۝٦

"*Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.". (QS al-Hujurat:6)*

Melalui ayat tersebut Allah SWT mengingatkan pentingnya memverifikasi informasi untuk mencegah kesalahpahaman yang dapat berdampak negatif. Oleh karena itu, teknologi peringkasan teks otomatis yang lebih akurat tidak hanya membantu dari segi efisiensi, tetapi juga mendukung penyebaran informasi yang benar dan terpercaya.

Sebagai respon terhadap kebutuhan tersebut, penelitian ini berupaya meningkatkan kualitas ringkasan dibandingkan penelitian sebelumnya dengan menggabungkan metode *TextRank*, *Word2Vec*, dan *Latent Semantic Analysis* (LSA) dalam sistem peringkasan teks otomatis untuk berita berbahasa Indonesia. *TextRank* digunakan untuk menentukan kalimat-kalimat paling penting dalam teks melalui perhitungan peringkat berbasis keterkaitan antar kalimat. *Word2Vec* berperan dalam mengubah kata-kata menjadi vektor angka sehingga memungkinkan analisis hubungan dan kesamaan makna antar kata secara matematis. Sementara itu, penambahan metode LSA diharapkan mampu menangkap pola makna antar kalimat dengan lebih baik, sehingga ringkasan yang dihasilkan menjadi lebih akurat, koheren, dan sesuai dengan konteks aslinya.

1.2 Rumusan Masalah

Apakah penerapan kombinasi metode *Word2Vec*, *Latent Semantic Analysis* (LSA), dan *TextRank* pada sistem peringkasan teks berita berbahasa Indonesia dapat meningkatkan kualitas ringkasan dibandingkan dengan metode pada penelitian sebelumnya?

1.3 Batasan Masalah

Batasan masalah yang di tetapkan dalam penelitian ini adalah :

1. Data yang digunakan pada penelitian hanya data berita yang berasal dari dataset Indosum.
2. Pendekatan peringkasan yang digunakan dalam penelitian ini adalah peringkasan secara ekstraktif, sedangkan metode yang digunakan adalah Algoritma *TextRank*, dengan integrasi model *Word2Vec* dan *Latent Semantic Analysis* (LSA).
3. Hasil peringkasan akan dievaluasi menggunakan metrik ROUGE untuk mengukur kualitas ringkasan berdasarkan nilai *precision*, *recall*, dan *f1-score* serta oleh seorang validator untuk penilaian secara manualnya.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk menguji efektivitas integrasi metode *TextRank*, *Word2Vec*, dan LSA dalam menghasilkan ringkasan berita yang lebih akurat dan koheren dibandingkan metode pada penelitian sebelumnya.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah menyediakan sistem peringkasan teks otomatis yang mampu menyajikan ringkasan berita secara cepat, akurat, dan sesuai konteks, sehingga memudahkan pembaca untuk menyaring informasi penting di tengah banyaknya berita yang tersedia, membantu penyebaran informasi yang lebih terpercaya dengan ringkasan yang informatif, serta berkontribusi pada pengembangan teknologi pemrosesan bahasa alami dalam bahasa Indonesia yang lebih efisien dan relevan.

BAB II

STUDI PUSTAKA

2.1 *Text Summarization*

Text summarization atau peringkasan teks adalah proses mereduksi teks panjang menjadi teks yang lebih singkat tanpa menghilangkan makna atau informasi penting yang ada pada teks asli. Tujuannya adalah memberikan gambaran umum mengenai isi dokumen yang membantu pengguna memahami inti informasi secara cepat tanpa harus membaca keseluruhan teks (Vikas et al. 2020). Peringkasan teks dapat membantu dalam menyajikan informasi secara ringkas, terutama ketika pengguna ingin memahami inti informasi tanpa harus membaca seluruh dokumen. Dalam konteks ini, peringkasan teks dapat dilakukan melalui dua pendekatan utama yaitu peringkasan ekstraktif dan peringkasan abstraktif. Peringkasan ekstraktif memilih kalimat atau frasa penting secara langsung dari teks, sedangkan peringkasan abstraktif menghasilkan ringkasan dengan menyusun ulang informasi menggunakan kata-kata baru, mirip dengan cara manusia meringkas (Kirmani et al. 2024)).

Penggunaan peringkasan teks terdapat di berbagai bidang, mulai dari jurnalistik untuk menyajikan berita dalam bentuk *headline* yang ringkas, hingga sektor akademik untuk membantu peneliti dan pelajar memahami artikel ilmiah dengan lebih efisien (Thange et al. 2023). Tantangan utama dari peringkasan teks adalah memastikan isi ringkasan tetap menyatu, jelas, dan tetap menyampaikan informasi penting secara utuh dan mudah dipahami. (Abdolahi & Zahed, 2019).

Penelitian oleh (El-Kassas et al. 2021) menunjukkan bahwa peringkasan teks mampu mengurangi waktu yang dihabiskan dalam memahami dokumen kompleks hingga lebih dari separuhnya, serta meningkatkan efisiensi dan produktivitas. Sementara itu, (Vanisha et al. 2022) juga menjelaskan mengenai manfaat dari adanya peringkasan teks antara lain mengurangi waktu membaca dengan menyajikan informasi kunci secara padat, mengubah informasi mentah yang besar menjadi data yang berguna dan mudah dipahami, serta membantu dalam menangani jumlah data yang sangat besar dengan mengekstrak dan mempertahankan informasi penting dari dokumen. Dengan demikian, peringkasan teks memudahkan akses dan pemahaman terhadap informasi yang relevan bagi masyarakat luas.

Hal ini menjadi bukti bahwa peringkasan teks memungkinkan pemahaman informasi secara cepat dan efektif, memfasilitasi akses cepat ke informasi yang relevan dan mendukung efisiensi dalam berbagai bidang. Dengan demikian, peringkasan teks memainkan peran penting dalam mengelola volume informasi yang besar dan membantu penyebaran informasi yang lebih cepat dan akurat di berbagai bidang.

2.2 *Text Processing*

Pemrosesan teks atau *text processing* adalah sebuah metode dalam bidang NLP (*Natural Language Processing*) yang berfokus pada pengolahan data teks mentah agar menjadi lebih terstruktur dan siap untuk dianalisis. Proses ini mencakup berbagai langkah seperti pembersihan teks, normalisasi, tokenisasi, dan perubahan ke bentuk akar kata. Pemrosesan teks bertujuan untuk menghilangkan

elemen-elemen yang tidak diperlukan seperti tanda baca, angka, atau karakter khusus yang dapat mengganggu analisis, sehingga menghasilkan data teks yang lebih konsisten (MR ADEPU RAJESH & DR TRYAMBAK HIWARKAR, 2023). Proses ini sangat penting dalam pengolahan big data, terutama ketika teks berasal dari sumber yang beragam seperti berita, media sosial, atau laporan.

Dalam berbagai studi, pemrosesan teks terbukti meningkatkan kualitas analisis dan interpretasi data. Seperti pada penelitian yang dilakukan oleh (Juna & Hayaty, 2023) menunjukkan bahwa strategi pra-pemrosesan yang tepat, khususnya kombinasi pembersihan data dan pengubahan huruf, secara signifikan meningkatkan kualitas ringkasan teks otomatis. Dengan penerapan teknik-teknik ini, sistem ringkasan dapat menghasilkan ringkasan yang lebih akurat dan relevan, memudahkan pengguna dalam memahami informasi penting dari jumlah data yang besar dan terus berkembang. Hal ini ditunjukkan dengan hasil dari evaluasi sistem yang menggunakan teknik pemrosesan data dengan nilai ROUGE-1: 0.78, ROUGE-2: 0.60, dan ROUGE-L: 0.68.

Dari penelitian diatas, terbukti bahwa proses pembersihan teks memainkan peran penting dalam memastikan data teks lebih bersih dan siap untuk dianalisis, serta mendukung peningkatan performa model dalam tugas-tugas NLP yang kompleks khususnya dibidang peringkasan teks.

2.3 *Word2Vvec*

Word2Vec adalah model pembelajaran mesin yang digunakan untuk mengubah kata-kata dalam teks menjadi vektor numerik atau angka yang mewakili makna kata , metode *Word2Vec* yang dikembangkan oleh Tomas Mikolov dan

timnya di Google pada 2013 pada penelitian, untuk menghasilkan representasi numerik (vektor) dari kata-kata dalam sebuah teks. Model ini menjadi populer di bidang pemrosesan bahasa alami (NLP) karena kemampuannya untuk menangkap makna semantik kata berdasarkan konteks kalimat. *Word2Vec* mengubah setiap kata dalam teks menjadi serangkaian angka (vektor) yang mewakili makna kata tersebut. Vektor ini berada dalam ruang yang memiliki banyak dimensi, artinya setiap kata direpresentasikan dengan angka-angka dalam ruang yang sangat luas. Semakin banyak dimensi yang digunakan, semakin detail representasi kata tersebut. Dalam ruang vektor ini, kata-kata yang sering muncul dalam konteks yang sama akan memiliki posisi yang saling berdekatan, sehingga kata-kata dengan makna serupa akan terletak lebih dekat satu sama lain. *Word2Vec* menggunakan dua pendekatan utama, yaitu *Continuous Bag of Words* (CBOW) dan *Skip-gram*, yang masing-masing memprediksi kata target dengan memanfaatkan kata-kata di sekitarnya, atau sebaliknya (Goyal et al. 2018).

Keunggulan *Word2Vec* dibandingkan dengan metode tradisional seperti *Bag of Words* (BoW) atau *Term Frequency-Inverse Document Frequency* (TF-IDF) terletak pada kemampuannya untuk mempertimbangkan konteks kata (Dai et al. 2024), bukan hanya frekuensi kata dalam teks. *Word2Vec* menghasilkan vektor yang menggambarkan hubungan semantik antar kata, seperti operasi vektor "raja – laki-laki + perempuan" yang menghasilkan kata "ratu". Model ini biasanya menghasilkan vektor dengan dimensi antara 100 hingga 300, yang menggambarkan karakteristik kata secara lebih detail. Selain itu, model ini tidak memerlukan data berlabel, sehingga dapat diterapkan pada berbagai bahasa dan domain tanpa

pelatihan khusus. Dengan vektor-vektor ini, *Word2Vec* dapat digunakan dalam berbagai aplikasi NLP, seperti analisis sentimen, peringkasan teks otomatis, dan klasifikasi teks.

Beberapa penelitian menegaskan potensi dan keunggulan *Word2Vec* dalam berbagai pengaplikasian. (Wazery et al. 2022) dalam penelitiannya menunjukkan keunggulan model *skip-gram* dari *Word2Vec* dalam meningkatkan efektivitas rangkuman teks Bahasa Arab secara abstraktif. Dalam studi ini, *Word2Vec* digunakan untuk menghasilkan embedding kata yang lebih dalam, memungkinkan model *sequence-to-sequence* dengan perhatian global menangkap makna semantik secara akurat dan menyeluruh. Dibandingkan dengan metode *Continuous Bag of Words* (CBOW), *skip-gram Word2Vec* menunjukkan performa yang lebih tinggi dalam menyampaikan konteks dan hubungan antar kata, sehingga menghasilkan rangkuman yang lebih relevan dan koheren. Model yang paling unggul dalam penelitian ini adalah BiLSTM tiga lapis yang dipadukan dengan skip-gram *Word2Vec* dan praproses AraBERT, mencapai nilai evaluasi ROUGE-1 F1 sebesar 51.49, ROUGE-2 F1 sebesar 12.27, ROUGE-L F1 sebesar 34.37, dan BLEU sebesar 0.41. Hasil ini menunjukkan bahwa *Word2Vec*, khususnya *skip-gram*, secara signifikan memperkuat kinerja model dalam menangkap struktur semantik yang kompleks, menjadikannya metode unggul dalam aplikasi rangkuman teks yang membutuhkan pemahaman konteks mendalam.

Selain itu, penelitian (Haider et al. 2020) berjudul "*Automatic Text Summarization Using Gensim Word2Vec and K-Means Clustering Algorithm*" mengevaluasi efektivitas *Word2Vec* dalam menghasilkan ringkasan teks secara

ekstraktif menggunakan algoritma *K-Means*. Hasil penelitian ini menunjukkan bahwa *Word2Vec skip-gram* mampu menangkap konteks semantik antar kalimat dengan baik, yang membantu dalam pengelompokan kalimat berdasarkan kesamaan makna untuk merangkum teks secara efektif. Model ini diuji pada berbagai kategori artikel berita, dan hasil terbaik dicapai pada kategori bisnis dengan BLEU *score* kumulatif tertinggi sebesar 0.894. Studi ini menyimpulkan bahwa kombinasi *Word2Vec* dan *K-Means* menghasilkan ringkasan teks yang relevan dan informatif, terutama untuk teks dengan struktur kalimat yang jelas dan mengandung banyak informasi penting.

2.4 *Latent Semantic Analysis* (LSA)

Latent Semantic Analysis (LSA) adalah metode dalam pengolahan bahasa alami (NLP) yang digunakan untuk mencari hubungan makna tersembunyi antara kata-kata dalam teks. LSA bekerja dengan membangun sebuah matriks yang menunjukkan hubungan antara dokumen dan kata-kata. Kemudian, teknik yang disebut *Singular Value Decomposition* (SVD) digunakan untuk menyederhanakan data dengan mengurangi dimensi dan menghilangkan informasi yang tidak penting. Hasil dari proses ini adalah representasi kata dan dokumen dalam bentuk yang lebih sederhana, yang memudahkan untuk menemukan kata-kata yang memiliki makna yang mirip meskipun tidak selalu muncul bersama dalam teks. Dengan kemampuannya menyederhanakan representasi kata dan dokumen, LSA dapat digunakan dalam berbagai tugas yang melibatkan pemahaman makna teks (Surianto et al. 2022).

Beberapa penelitian sebelumnya telah memanfaatkan metode *Latent Semantic Analysis* (LSA) dalam berbagai aplikasi, termasuk dalam meringkas teks berita. Dalam penelitian yang dilakukan oleh (Ramezani et al. 2023) yang berjudul "*Unsupervised Broadcast News Summarization; A Comparative Study on Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA)*", LSA digunakan untuk meringkas transkripsi berita siaran bahasa Persia. Hasil evaluasi menunjukkan bahwa LSA memiliki keunggulan dalam *generic summarization*. Metrik evaluasi seperti ROUGE-1 *F-score*, *Precision*, dan *Recall* menunjukkan bahwa LSA lebih efektif dalam menghasilkan ringkasan yang koheren dan informatif. Pada evaluasi *generic summarization*, LSA yang menggunakan metode Murray et al. menghasilkan ROUGE-1 *F-score* sebesar 0.5930 dengan *Precision* 53.35% dan *Recall* 56.61%, sedangkan MMR hanya mencatatkan ROUGE-1 F-score 0.3957 dengan *Precision* 35.68% dan *Recall* 37.91%. Hal ini menunjukkan bahwa LSA lebih unggul dalam menangkap makna utama dari teks tanpa mengorbankan kualitas informasi yang relevan, serta lebih efektif dalam mengurangi redundansi dibandingkan dengan MMR. Penelitian ini juga menyoroti bahwa LSA, meskipun tidak berbasis kueri, mampu menghasilkan ringkasan yang lebih baik dalam konteks *generic summarization* dibandingkan dengan MMR yang lebih cocok untuk *query-based summarization*.

Selanjutnya, penelitian oleh (Giri et al. 2023) mengevaluasi efektivitas metode SVD dalam LSA dan *Fuzzy Logic* untuk meringkas teks Bahasa Marathi, menemukan bahwa masing-masing pendekatan memiliki keunggulan tersendiri. Dalam *single document summarization*, *Fuzzy Logic* menunjukkan performa lebih

baik dengan nilai evaluasi ROUGE-1 F1 sebesar 0.623, ROUGE-2 F1 sebesar 0.546, dan ROUGE-L F1 sebesar 0.64, lebih tinggi dibandingkan SVD yang mencatat ROUGE-1 F1 sebesar 0.612, ROUGE-2 F1 sebesar 0.512, dan ROUGE-L F1 sebesar 0.626. Namun, untuk *multi-document summarization*, SVD unggul dengan nilai *Precision* sebesar 0.705, *Recall* sebesar 0.693, dan F1 sebesar 0.682, mengungguli *Fuzzy Logic* yang memiliki *Precision* sebesar 0.625, *Recall* sebesar 0.655, dan F1 sebesar 0.63. Hasil ini menunjukkan bahwa SVD lebih efektif untuk multi-document summarization, sementara *Fuzzy Logic* lebih tepat untuk *single document summarization*, terutama karena kemampuannya dalam mempertimbangkan panjang dan posisi kalimat sebagai faktor pemeringkatan.

2.5 Algoritma *TextRank*

TextRank adalah sebuah algoritma berbasis graf yang digunakan untuk tugas ekstraksi informasi dari teks, seperti peringkasan teks dan ekstraksi kata kunci Algoritma ini pertama kali diperkenalkan oleh Mihalcea dan Tarau pada tahun 2004 yang terinspirasi dari algoritma *PageRank* yang digunakan oleh Google untuk menentukan peringkat halaman web. *TextRank* bekerja dengan membangun graf dari dokumen teks, di mana simpul-simpul dalam graf ini dapat berupa kata atau kalimat. Setiap simpul dihubungkan oleh sebuah tepi (*edge*) yang menunjukkan tingkat kesamaan atau keterkaitan antara simpul-simpul tersebut (Jane C. Patosa et al. 2022). Misalnya, dalam peringkasan teks, setiap kalimat dianggap sebagai simpul, dan kemiripan antara dua kalimat menentukan bobot tepi yang menghubungkan simpul-simpul tersebut.

Penelitian terkait penggunaan *TextRank* telah menunjukkan keefektifan algoritma ini dalam berbagai aplikasi NLP, terutama dalam konteks peringkasan teks otomatis. Dalam penelitian oleh (Rani & Bidhan, 2021), yang membandingkan tiga teknik peringkasan teks TF-IDF, *TextRank*, dan *Latent Dirichlet Allocation* (LDA) pada dataset berita, hasilnya menunjukkan bahwa *TextRank* mendominasi dalam hal *recall*, *precision*, dan *f-measure* pada dataset berita. Untuk nilai *recall*, *TextRank* mencapai skor rata-rata 0,8038, yang menunjukkan kemampuannya untuk mengidentifikasi kalimat-kalimat penting secara lebih efisien dibandingkan dengan kedua metode lainnya. Selain itu, *TextRank* juga mengungguli TF-IDF dan LDA dalam hal *precision* dan *f-measure*, dengan skor masing-masing 0,5687 dan 0,6520. Hal ini menegaskan bahwa *TextRank* lebih unggul dalam merangkum artikel berita, terutama dalam memilih kalimat-kalimat yang relevan dan menyusun ringkasan yang lebih representatif untuk teks berita yang padat informasi .

Selain itu, penelitian oleh (Yulianti et al. 2023) menunjukkan bahwa penggunaan *TextRank* yang dipadukan dengan *weighted word embedding* menghasilkan kinerja yang lebih baik dalam peringkasan teks dibandingkan dengan *TextRank* asli. Dalam penelitian tersebut, *TextRank* menggunakan *word embedding* tanpa pembobot TF-IDF menunjukkan peningkatan kinerja yang signifikan. Misalnya, sistem yang menggunakan *Word2Vec* memperoleh ROUGE-1 sebesar 0.3822, ROUGE-2 sebesar 0.2693, dan ROUGE-L sebesar 0.3646, yang lebih tinggi dibandingkan *TextRank* standar dengan nilai ROUGE-1 0.3479, ROUGE-2 0.2339, dan ROUGE-L 0.3302. Demikian pula, sistem yang menggunakan *FastText* dan IndoBERT juga menunjukkan peningkatan dalam ketiga metrik ROUGE.

Terlebih lagi, ketika *word embedding* diberi bobot menggunakan TF-IDF, peningkatan kinerja lebih terlihat lagi. Misalnya, *TextRank* + *Word2Vec* + TF-IDF memperoleh ROUGE-1 0.4082, ROUGE-2 0.3041, dan ROUGE-L 0.3926, yang menunjukkan peningkatan signifikan, khususnya pada ROUGE-2 dengan peningkatan sebesar 12.92% dibandingkan dengan *TextRank* tanpa bobot. Sistem berbasis BERT juga menunjukkan peningkatan, dengan *TextRank* + *IndoBERTlarge* + TF-IDF memperoleh ROUGE-1 0.4044, ROUGE-2 0.2982, dan ROUGE-L 0.3884, meskipun peningkatannya lebih kecil dibandingkan dengan model *Word2Vec* dan *FastText*. Berdasarkan hasil tersebut, dapat disimpulkan bahwa meskipun *TextRank* yang menggunakan BERT memberikan kinerja yang baik dalam menangkap konteks kata, sistem yang menggabungkan *TextRank* dengan *Word2Vec* dan *FastText* yang diberi bobot TF-IDF menunjukkan peningkatan terbesar, terutama pada ROUGE-2, menjadikannya metode yang lebih unggul dalam menghasilkan ringkasan yang lebih akurat dan relevan.

2.6 Evaluasi Hasil Peringkasan

Penilaian hasil peringkasan teks otomatis merupakan bagian penting dalam evaluasi kualitas sistem peringkasan. Untuk memastikan kualitas hasil peringkasan, pada penilaian ini akan dilakukan dengan dua teknik penilaian, yakni: penilaian otomatis dan penilaian manual. Penilaian otomatis akan menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L, sedangkan penilaian manual akan dilakukan oleh seorang validator.

2.6.1 Evaluasi Otomatis

Penilaian otomatis akan menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L, sedangkan penilaian manual akan dilakukan oleh seorang validator. Evaluasi otomatis dilakukan dengan menggunakan ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*), atau lebih tepatnya ROUGE-1, ROUGE-2, dan ROGE-L . Metrik ini digunakan untuk mengukur tingkat kesamaan antara n-gram dalam ringkasan otomatis dengan referensi yang ada, baik itu teks asli atau ringkasan yang dibuat secara manual (Barbella & Tortora, 2022). Pada penelitian ini ROUGE metrik yang diukur adalah *recall*, *precision*, dan *f1-score*. Meskipun penilaian otomatis dapat memberikan hasil berupa angka, namun metode ini tidak sepenuhnya efektif dalam menilai beberapa aspek penting seperti kelengkapan, koherensi, dan keterbacaan dari ringkasan yang dihasilkan.

2.6.2 Evaluasi Manual

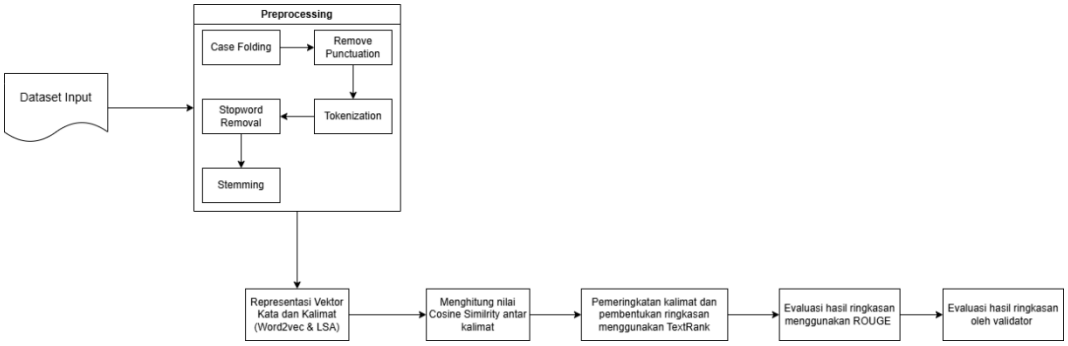
Penilaian manual terhadap hasil peringkasan teks otomatis dilakukan oleh validator yang menilai kualitas ringkasan berdasarkan kriteria seperti kelengkapan informasi, keterbacaan, dan koherensi. Untuk memastikan representativitas sampel dan mengurangi bias, metode *Simple Random Sampling* (SRS) digunakan dalam pemilihan data untuk evaluasi. Dengan SRS, setiap unit dalam populasi memiliki peluang yang sama untuk dipilih sebagai sampel, sehingga hasil penilaian dapat digeneralisasi ke populasi secara keseluruhan. Metode ini telah terbukti efektif dalam memberikan representasi yang tidak bias dan probabilitas seleksi yang setara untuk semua anggota populasi, terutama dalam populasi yang seragam.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Sistem

Desain sistem pada Gambar 3.1 yang dirancang dalam penelitian ini memiliki beberapa tahapan utama yang dimulai dari input dataset hingga evaluasi hasil peringkasan.



Gambar 3.1 Desain Sistem

Seperti yang terdapat pada Gambar 3.1, desain sistem peringkasan teks otomatis ini terdiri dari beberapa tahap utama yang dimulai dari input dataset, di mana data teks berita dimasukkan dan diproses melalui tahap *preprocessing*. Pada tahap ini, dilakukan serangkaian proses seperti *case folding*, penghapusan tanda baca, tokenisasi, penghapusan *stopword*, dan *stemming* untuk membersihkan teks dan mengubahnya menjadi format yang siap digunakan. Setelah pra-pemrosesan, sistem melakukan ekstraksi fitur menggunakan kombinasi *Word2Vec* dan *Latent Semantic Analysis* (LSA) untuk menghasilkan representasi vektor dari teks. Tahap selanjutnya adalah pemberian skor pada kalimat menggunakan metode *cosine*

similarity untuk menghitung nilai kesamaan hubungan antar kalimat. Hasil dari perhitungan ini akan digunakan dalam perhitungan algoritma *TextRank*. Algoritma *TextRank* berfungsi untuk memberikan peringkat pada kalimat-kalimat berdasarkan skor yang diperoleh. Kalimat dengan skor tertinggi akan dipilih dan disusun menjadi ringkasan. Langkah terakhir adalah evaluasi hasil, pada penelitian ini akan menggunakan 2 jenis penilaian evaluasi hasil ringkasan, yakni: penilaian secara otomatis dengan menggunakan metrik ROUGE, dan penilaian secara manual yang akan dinilai oleh seorang validator. Kemudian dari hasil kedua penilaian ini akan dibandingkan untuk memastikan hasil yang lebih dapat terpercaya untuk penilaian hasil ringkasannya.

3.2 Pengumpulan Data

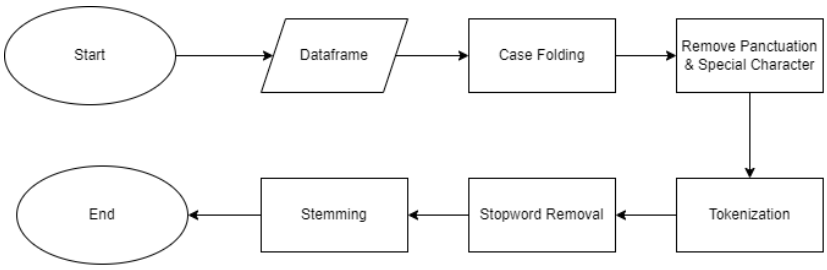
Data pada penelitian ini menggunakan 5000 data berita dari dataset Indosum. Dataset Indosum adalah kumpulan data yang dirancang khusus untuk mendukung penelitian dalam bidang pemrosesan bahasa alami, terutama untuk tugas penyusunan ringkasan otomatis. Dataset ini mencakup berbagai tema seperti politik, ekonomi, kesehatan, teknologi, olahraga, dan budaya. Sumber berita yang digunakan dalam pembuatan dataset Indosum berasal dari berbagai portal berita terkemuka di Indonesia, seperti Kompas, Detik, CNN Indonesia, dan Tempo, untuk memastikan bahwa data yang dikumpulkan akurat, terpercaya, dan representatif. dengan struktur data yang terorganisir dan anotasi yang tepat, serta setiap artikel dilengkapi dengan ringkasan abstraktif yang ditulis secara manual oleh dua penutur asli bahasa Indonesia (Kurniawan & Louvan, 2018).

	category	gold_labels	id	paragraphs	source	source_url	summary
14261	tajuk utama	[[True], [True, True], [True, False], [False, ...	1494387000-bongkar-pungli-rutan-bungkuk-polisi...	[[[Merdeka.com, -, Direktorat, Reserse, Krimin...	merdeka	https://www.merdeka.com/peristiwa/bongkar-pung...	[[Direktorat, Reserse, Kriminal, Khususnya, Po...
14262	tajuk utama	[[True, True], [False, False], [False, False],...	1503822625-implementasi-het-beras-kementan-pun...	[[[Merdeka.com, -, Kementerian, Perdagangan, t...	merdeka	https://www.merdeka.com/uang/implementasi-het-...	[[Kementerian, Perdagangan, telah, menetapkan,...
14263	tajuk utama	[[True, True], [True, False], [False], [False]]	1491261317-menhub-panggil-manajemen-lion-air	[[[Rimanews, -, Menteri, Perhubungan, Budi, Ka...	rimanews	http://rimanews.com/nasional/peristiwa/read/20...	[[Menteri, Perhubungan, Budi, Karya, Sumadi, a...
14264	olahraga	[[True], [True, True], [False], [False], [Fals...	1502994600-conor-mcgregor-lebih-siap-daripada-...	[[[JUARA.net, -, Duel, antara, Floyd, Mayweath...	juara.net	http://juara.bolasport.com/read/ragam/ragam/18...	[[Sejumlah, persipan, jelang, duel, antara, Fl...
14265	hiburan	[[True], [False], [False, True], [True], [Fals...	1503286200-pendaki-gunung-tambora-diperkirakan...	[[[Mataram, (, ANTARA, News,), -, Kepala, Bal...	antaranews	http://www.antaranews.com/berita/647551/pendak...	[[Kepala, Balai, Taman, Nasional, Tambora, Bud...

Gambar 3.2 Dataset Indosum

3.3 *Text Processing*

Data yang telah didapatkan perlu untuk melalui proses *preprocessing*. Data perlu dipreproses untuk membersihkan dan menyiapkan informasi sehingga analisis atau pemodelan dapat dilakukan dengan lebih efektif dan akurat. Tahapan *preprocessing* dapat dilihat pada gambar 3.3



Gambar 3.3 Alur *Preprocessing*

Tahap *preprocessing* dalam sistem ini dimulai dengan mengubah data input menjadi bentuk *dataframe* menggunakan *library* pandas, yang memudahkan pengolahan data dalam struktur tabel. Setelah data berada dalam bentuk *dataframe*, langkah pertama yang dilakukan adalah *case folding*, yaitu mengubah semua huruf dalam teks menjadi huruf kecil untuk menyamakan format dan menghindari perbedaan yang disebabkan oleh penggunaan huruf kapital. Selanjutnya, dilakukan proses *remove punctuation* & *special characters*, di mana tanda baca dan karakter khusus dihapus dari teks untuk membersihkan data dari simbol yang tidak relevan.

Setelah itu, teks dipecah menjadi unit-unit kata melalui proses *tokenization*. Setelah teks dipecah menjadi token, dilakukan *stopword removal*, yaitu menghapus kata-kata yang tidak memberikan makna penting dalam pemrosesan, seperti "dan", "di", "ke", dan lain-lain. Tahap terakhir dalam *preprocessing* adalah *stemming*, yaitu mengubah kata-kata yang telah di-tokenisasi menjadi bentuk dasarnya atau akar kata. Proses ini memastikan bahwa variasi dari kata-kata dengan makna yang sama disatukan dalam satu bentuk kata dasar. Setelah semua tahapan selesai, hasilnya adalah teks yang telah diproses dan siap digunakan untuk tahap-tahap pemrosesan berikutnya dalam sistem peringkasan teks otomatis. Berikut adalah contoh hasil dari tahapan *preprocessing*.

Tabel 3.1 Contoh Hasil *Preprocessing*

Tahapan	**Hasil Teks**
Teks Asli	Jakarta , CNN Indonesia - - Timnas Argentina kembali gagal meraih kemenangan tanpa kehadiran Lionel Messi . Kali ini tim Tango ditahan imbang Peru 2 - 2 di Stadion Nasional , Lima , Kamis (6 / 10) malam waktu setempat atau Jumat pagi WIB . Argentina membuang keunggulan dua kali saat menghadapi Peru . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke - 16 . Bek Everton itu memanfaatkan kemelut di kotak penalti Peru lewat skema sepak pojok . Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Di babak kedua , tepatnya menit ke - 58 , tuan rumah berhasil menyamakan kedudukan lewat kapten tim Jose Paolo Guerrero . Usai menerima umpan Miguel Trauco , Guerrero berhasil menahan bola dengan dada dan lolos dari kawalan Funes Mori . Mantan penyerang Bayern Munich itu kemudian melepaskan tendangan mendatar yang tidak bisa dihentikan kiper Sergio Romero . Argentina kembali unggul lewat Gonzalo Higuain pada menit ke - 78 . El Pipita berhasil meneruskan umpan terobosan Pablo Zabaleta dan mencungkil bola melewati kiper Pedro Gallese . Sayang , Argentina yang ditangani Edgardo Bauza tidak mampu mempertahankan keunggulan . Pada menit ke - 84 , Peru mendapat tendangan penalti yang dieksekusi dengan sempurna oleh Christian Cueva setelah Mori menjatuhkan Guerrero . Hasil imbang 2 - 2 membuat Argentina gagal naik dari peringkat lima klasemen sementara kualifikasi Piala Dunia 2018 zona Conmebol . Argentina kini mengoleksi 16 poin , tertinggal tiga poin dari Uruguay di puncak klasemen . Argentina untuk kali kedua beruntun meraih hasil imbang . Sebelumnya , Argentina ditahan imbang Venezuela 2 - 2 , 6 September lalu . Menariknya , di kedua laga tersebut Messi tidak tampil karena cedera . Messi saat ini sedang menjalani pemulihan cedera pangkal paha selama tiga pekan yang didapatnya

	saat melawan Atletico Madrid . Penyerang Barcelona itu juga dipastikan absen ketika Argentina menghadapi Paraguay di Cordoba , 11 Oktober mendatang . (har)
Case folding	jakarta , cnn indonesia - - timnas argentina kembali gagal meraih kemenangan tanpa kehadiran lionel messi . kali ini tim tango ditahan imbang peru 2 - 2 di stadion nasional , lima , kamis (6 / 10) malam waktu setempat atau jumat pagi wib . argentina membuang keunggulan dua kali saat menghadapi peru . ramiro funes mori membawa argentina unggul lewat gol pada menit ke - 16 . bek everton itu memanfaatkan kemelut di kotak penalti peru lewat skema sepak pojok . keunggulan satu gol argentina bertahan hingga jeda babak pertama . di babak kedua , tepatnya menit ke - 58 , tuan rumah berhasil menyamakan kedudukan lewat kapten tim jose paolo guerrero . usai menerima umpan miguel trauco , guerrero berhasil menahan bola dengan dada dan lolos dari kawalan funes mori . mantan penyerang bayern munich itu kemudian melepaskan tendangan mendatar yang tidak bisa dihentikan kiper sergio romero . argentina kembali unggul lewat gonzalo higuain pada menit ke - 78 . el pipita berhasil meneruskan umpan terobosan pablo zabaleta dan mencungkil bola melewati kiper pedro gallese . sayang , argentina yang ditangani edgardo bauza tidak mampu mempertahankan keunggulan . pada menit ke - 84 , peru mendapat tendangan penalti yang dieksekusi dengan sempurna oleh christian cueva setelah mori menjatuhkan guerrero . hasil imbang 2 - 2 membuat argentina gagal naik dari peringkat lima klasemen sementara kualifikasi piala dunia 2018 zona conmebol . argentina kini mengoleksi 16 poin , tertinggal tiga poin dari uruguay di puncak klasemen . argentina untuk kali kedua beruntun meraih hasil imbang . sebelumnya , argentina ditahan imbang venezuela 2 - 2 , 6 september lalu . menariknya , di kedua laga tersebut messi tidak tampil karena cedera . messi saat ini sedang menjalani pemulihan cedera pangkal paha selama tiga pekan yang didapatnya saat melawan atletico madrid . penyerang barcelona itu juga dipastikan absen ketika argentina menghadapi paraguay di cordoba , 11 oktober mendatang . (har)
Removing Punctuations, and Special Characters	jakarta cnn indonesia timnas argentina kembali gagal meraih kemenangan tanpa kehadiran lionel messi kali ini tim tango ditahan imbang peru 2 2 di stadion nasional lima kamis 6 10 malam waktu setempat atau jumat pagi wib argentina membuang keunggulan dua kali saat menghadapi peru ramiro funes mori membawa argentina unggul lewat gol pada menit ke 16 bek everton itu memanfaatkan kemelut di kotak penalti peru lewat skema sepak pojok keunggulan satu gol argentina bertahan hingga jeda babak pertama di babak kedua tepatnya menit ke 58 tuan rumah berhasil menyamakan kedudukan lewat kapten tim jose paolo guerrero usai menerima umpan miguel trauco guerrero berhasil menahan bola dengan dada dan lolos dari kawalan funes mori mantan penyerang bayern munich itu kemudian melepaskan tendangan mendatar yang tidak bisa dihentikan kiper sergio romero argentina kembali unggul lewat gonzalo higuain pada menit ke 78 el pipita berhasil meneruskan umpan terobosan pablo zabaleta dan mencungkil bola melewati kiper pedro gallese sayang argentina yang ditangani edgardo bauza tidak mampu mempertahankan keunggulan pada menit ke 84 peru mendapat tendangan penalti yang dieksekusi dengan sempurna oleh christian cueva setelah mori menjatuhkan

	guerrero hasil imbang 2 2 membuat argentina gagal naik dari peringkat lima klasemen sementara kualifikasi piala dunia 2018 zona conmebol argentina kini mengoleksi 16 poin tertinggal tiga poin dari uruguay di puncak klasemen argentina untuk kali kedua beruntun meraih hasil imbang sebelumnya argentina ditahan imbang venezuela 2 2 6 september lalu menariknya di kedua laga tersebut messi tidak tampil karena cedera messi saat ini sedang menjalani pemulihan cedera pangkal paha selama tiga pekan yang didapatnya saat melawan atletico madrid penyerang barcelona itu juga dipastikan absen ketika argentina menghadapi paraguay di cordoba 11 oktober mendatang har
Tokenization	['jakarta', 'cnn', 'indonesia', 'timnas', 'argentina', 'kembali', 'gagal', 'meraih', 'kemenangan', 'tanpa', 'kehadiran', 'lionel', 'messi', 'kali', 'ini', 'tim', 'tango', 'ditahan', 'imbang', 'peru', '2', '2', 'di', 'stadion', 'nasional', 'lima', 'kamis', '6', '10', 'malam', 'waktu', 'setempat', 'atau', 'jumat', 'pagi', 'wib', 'argentina', 'membuang', 'keunggulan', 'dua', 'kali', 'saat', 'menghadapi', 'peru', 'ramiro', 'funes', 'mori', 'membawa', 'argentina', 'unggul', 'lewat', 'gol', 'pada', 'menit', 'ke', '16', 'bek', 'everton', 'itu', 'memanfaatkan', 'kemelut', 'di', 'kotak', 'penalti', 'peru', 'lewat', 'skema', 'sepak', 'pojok', 'keunggulan', 'satu', 'gol', 'argentina', 'bertahan', 'hingga', 'jeda', 'babak', 'pertama', 'di', 'babak', 'kedua', 'tepatnya', 'menit', 'ke', '58', 'tuan', 'rumah', 'berhasil', 'menyamakan', 'kedudukan', 'lewat', 'kapten', 'tim', 'jose', 'paolo', 'guerrero', 'usai', 'menerima', 'umpan', 'miguel', 'trauco', 'guerrero', 'berhasil', 'menahan', 'bola', 'dengan', 'dada', 'dan', 'lolos', 'dari', 'kawalan', 'funes', 'mori', 'mantan', 'penyerang', 'bayern', 'munich', 'itu', 'kemudian', 'melepaskan', 'tendangan', 'mendatar', 'yang', 'tidak', 'bisa', 'dihentikan', 'kiper', 'sergio', 'romero', 'argentina', 'kembali', 'unggul', 'lewat', 'gonzalo', 'higuain', 'pada', 'menit', 'ke', '78', 'el', 'pipita', 'berhasil', 'meneruskan', 'umpan', 'terobosan', 'pablo', 'zabaleta', 'dan', 'mencungkil', 'bola', 'melewati', 'kiper', 'pedro', 'gallese', 'sayang', 'argentina', 'yang', 'ditangani', 'edgardo', 'bauza', 'tidak', 'mampu', 'mempertahankan', 'keunggulan', 'pada', 'menit', 'ke', '84', 'peru', 'mendapat', 'tendangan', 'penalti', 'yang', 'dieksekusi', 'dengan', 'sempurna', 'oleh', 'christian', 'cueva', 'setelah', 'mori', 'menjatuhkan', 'guerrero', 'hasil', 'imbang', '2', '2', 'membuat', 'argentina', 'gagal', 'naik', 'dari', 'peringkat', 'lima', 'klasemen', 'sementara', 'kualifikasi', 'piala', 'dunia', '2018', 'zona', 'conmebol', 'argentina', 'kini', 'mengoleksi', '16', 'poin', 'tertinggal', 'tiga', 'poin', 'dari', 'uruguay', 'di', 'puncak', 'klasemen', 'argentina', 'untuk', 'kali', 'kedua', 'beruntun', 'meraih', 'hasil', 'imbang', 'sebelumnya', 'argentina', 'ditahan', 'imbang', 'venezuela', '2', '2', '6', 'september', 'lalu', 'menariknya', 'di', 'kedua', 'laga', 'tersebut', 'messi', 'tidak', 'tampil', 'karena', 'cedera', 'messi', 'saat', 'ini', 'sedang', 'menjalani', 'pemulihan', 'cedera', 'pangkal', 'paha', 'selama', 'tiga', 'pekan', 'yang', 'didapatnya', 'saat', 'melawan', 'atletico', 'madrid', 'penyerang', 'barcelona', 'itu', 'juga', 'dipastikan', 'absen', 'ketika', 'argentina', 'menghadapi', 'paraguay', 'di', 'cordoba', '11', 'oktober', 'mendatang', 'har']
Stopwords	['jakarta', 'cnn', 'indonesia', 'timnas', 'argentina', 'gagal', 'meraih', 'kemenangan', 'kehadiran', 'lionel', 'messi', 'kali', 'tim', 'tango', 'ditahan', 'imbang', 'peru', '2', '2', 'stadion', 'nasional', 'kamis', '6', '10', 'malam', 'jumat', 'pagi', 'wib', 'argentina', 'membuang', 'keunggulan', 'kali', 'menghadapi', 'peru', 'ramiro', 'funes', 'mori', 'membawa', 'argentina', 'unggul', 'gol', 'menit', '16', 'bek', 'everton',

	'memanfaatkan', 'kemelut', 'kotak', 'penalti', 'peru', 'skema', 'sepak', 'pojok', 'keunggulan', 'gol', 'argentina', 'bertahan', 'jeda', 'babak', 'babak', 'tepatnya', 'menit', '58', 'tuan', 'rumah', 'berhasil', 'menyamakan', 'kedudukan', 'kapten', 'tim', 'jose', 'paolo', 'guerrero', 'menerima', 'umpan', 'miguel', 'trauco', 'guerrero', 'berhasil', 'menahan', 'bola', 'dada', 'lolos', 'kawalan', 'funes', 'mori', 'mantan', 'penyerang', 'bayern', 'munich', 'melepaskan', 'tendangan', 'mendatar', 'dihentikan', 'kiper', 'sergio', 'romero', 'argentina', 'unggul', 'gonzalo', 'higuain', 'menit', '78', 'el', 'pipita', 'berhasil', 'meneruskan', 'umpan', 'terobosan', 'pablo', 'zabaleta', 'mencungkil', 'bola', 'melewati', 'kiper', 'pedro', 'gallese', 'sayang', 'argentina', 'ditangani', 'edgardo', 'bauza', 'mempertahankan', 'keunggulan', 'menit', '84', 'peru', 'tendangan', 'penalti', 'dieksekusi', 'sempurna', 'christian', 'cueva', 'mori', 'menjatuhkan', 'guerrero', 'hasil', 'imbang', '2', '2', 'argentina', 'gagal', 'peringkat', 'klasemen', 'kualifikasi', 'piala', 'dunia', '2018', 'zona', 'conmebol', 'argentina', 'mengoleksi', '16', 'poin', 'tertinggal', 'poin', 'uruguay', 'puncak', 'klasemen', 'argentina', 'kali', 'beruntun', 'meraih', 'hasil', 'imbang', 'argentina', 'ditahan', 'imbang', 'venezuela', '2', '2', '6', 'september', 'menariknya', 'laga', 'messi', 'tampil', 'cedera', 'messi', 'menjalani', 'pemulihan', 'cedera', 'pangkal', 'paha', 'pekan', 'didapatnya', 'melawan', 'atletico', 'madrid', 'penyerang', 'barcelona', 'absen', 'argentina', 'menghadapi', 'paraguay', 'cordoba', '11', 'oktober', 'har']
Stemming	jakarta cnn indonesia timnas argentina gagal raih menang hadir lionel messi kali tim tango tahan imbang peru 2 2 stadion nasional kamis 6 10 malam jumat pagi wib argentina buang unggul kali hadap peru ramiro funes mori bawa argentina unggul gol menit 16 bek everton manfaat kemelut kotak penalti peru skema sepak pojok unggul gol argentina tahan jeda babak babak tepat menit 58 tuan rumah hasil sama duduk kapten tim jose paolo guerrero terima umpan miguel trauco guerrero hasil tahan bola dada lolos kawal funes mori mantan serang bayern munich lepas tendang datar henti kiper sergio romero argentina unggul gonzalo higuain menit 78 el pipita hasil terus umpan terobos pablo zabaleta cungkil bola lewat kiper pedro gallese sayang argentina tangan edgardo bauza tahan unggul menit 84 peru tendang penalti eksekusi sempurna christian cueva mori jatuh guerrero hasil imbang 2 2 argentina gagal peringkat klasemen kualifikasi piala dunia 2018 zona conmebol argentina koleksi 16 poin tinggal poin uruguay puncak klasemen argentina kali untun raih hasil imbang argentina tahan imbang venezuela 2 2 6 september tarik laga messi tampil cedera messi jalan pulih cedera pangkal paha pekan dapat lawan atletico madrid serang barcelona absen argentina hadap paraguay cordoba 11 oktober har

3.4 Representasi Vektor

Word2Vec dan *Latent Semantic Analysis* LSA) digunakan untuk merepresentasikan vektor kata dan kalimat dalam program peringkasan teks pada penelitian ini. Dalam penelitian ini, metode *Word2Vec* diimplementasikan

menggunakan *library* Python bernama Gensim, sedangkan untuk metode LSA akan diimplementasikan menggunakan *library* python yaitu *TruncatedSVD*.

3.4.1 *Word2Vec*

Gambar 3. 4 Flowchart Word2Vec

Pada gambar 3.4, alur kerja model *Word2Vec* dengan metode *Skipgram* dimulai dengan pengumpulan teks yang diproses melalui prapemrosesan, seperti tokenisasi dan penghapusan *stopwords*. Selanjutnya, dibangunlah *vocabulary* yang berisi kata-kata unik, dan model membuat pasangan kata target dan konteks berdasarkan ukuran *window* yang ditentukan. Bobot awal untuk kata target dan

konteks diinisialisasi secara acak, dan pada tahap pelatihan, model menghitung probabilitas konteks, kemudian menghitung error dan memperbarui bobot menggunakan metode gradient descent. Proses ini diulang hingga model mencapai hasil yang optimal. Jika hasilnya belum memadai, iterasi berikutnya dilakukan dengan memperbarui bobot dan menghitung error ulang.

Selanjutnya, untuk langkah perhitungan yang lebih detail pada model *Word2Vec Skipgram*,vberikut ini adalah langkah-langkah representasi vektor dari *Word2Vec*:

1. Pembentukan Kata C*ontext-Target*

   Tahap pertama setiap kata dalam teks diubah menjadi pasangan *context-target* berdasarkan jendela kata (*window size*). Pada penelitian nilai dari window size nya sebanyak 6. Contohnya seperti jika kata targetnya “jakarta” maka kata konteksnya adalah 'cnn', 'indonesia', 'timnas', 'argentina', 'gagal', 'raih'.

2. Representasi *One Hot Encode*

   Tahap kedua setiap kata dalam kosa kata (*vocab*) direpresentasikan sebagai vektor *one-hot*, yaitu vektor dengan nilai 1 pada indeks kata tertentu dan 0 di tempat lain. Berikut adalah contoh dari representasi *one-hot encode*.

   jakarta = [1,0,0,0,0,0,0,0,0,0]
   cnn = [0,1,0,0,0,0,0,0,0,0]
   indonesia = [0,0,1,0,0,0,0,0,0,0]
   timnas = [0,0,0,1,0,0,0,0,0,0]
   argentina = [0,0,0,0,1,0,0,0,0,0]
   gagal = [0,0,0,0,0,1,0,0,0,0]
   raih = [0,0,0,0,0,0,1,0,0,0]

3. Proyeksi ke Ruang Embedding

Tahap ketiga kata dalam representasi *one-hot* diproyeksikan ke ruang embedding menggunakan matriks bobot input (W_{in}), yang memiliki dimensi [*vocab size* × *vector size*]. Representasi embedding (h) untuk kata dihitung sebagai:

$$h = W_{in}^{T} \cdot (one - hot\ vector) \quad (3.1)$$

Dimana :
h : Vektor embedding kata yang dihasilkan dengan panjang sesuai *vector_size*.
W_{in} : Matriks bobot input dengan dimensi [*vocab size* × *vector size*].
one-hot vector : Representasi *one-hot* dari kata input.

4. Perhitungan Skor Output

Tahap keempat vektor embedding h digunakan untuk menghitung skor keluaran (u) untuk setiap kata target menggunakan matriks bobot output (W_{out}), yang memiliki dimensi [*vector size* × *vocab size*]:

$$u = w_{out}^{T} \cdot h \quad (3.2)$$

Dimana:
U : Skor keluaran untuk setiap kata dalam *vocab*.
W_{out} : Matriks bobot output dengan dimensi [*vector size* × *vocab size*].
H : Vektor embedding kata dari langkah sebelumnya.

5. *Softmax* untuk Probabilitas

Tahap kelima skor u diubah menjadi probabilitas menggunakan fungsi *softmax* dengan rumus (3.3):

$$P(target\ |context) = \frac{\exp\ (u_i)}{\sum_{j=1}^{v} \exp\ (u_j)} \quad (3.3)$$

Dimana:

P(target | context)	: Probabilitas kata target muncul dalam konteks kata input
u_i	: Skor keluaran untuk kata target.
V	: Ukuran kosa kata (*vocab*).
$\sum_{j=1}^{v} \exp(u_j)$	: Penjumlahan eksponensial dari skor untuk semua kata dalam *vocab*.

6. Fungsi *Loss* dan *Backpropagation*

Tahap keenam fungsi kerugian *negative log-likelihood* (3.4) digunakan untuk mengukur kesalahan prediksi:

$$Loss = -\log(P(target \mid context)) \tag{3.4}$$

Dimana:

Loss	: Nilai kerugian model untuk *context-target* pasangan tertentu.
P(*target* | *context*)	: Probabilitas yang dihitung dengan *softmax*.

$$W_{in} = W_{in} - \eta \cdot \frac{\partial Loss}{\partial W_{in}} \tag{3.5}$$

$$W_{out} = W_{out} - \eta \cdot \frac{\partial Loss}{\partial W_{out}} \tag{3.6}$$

Dimana:

η	: Laju pembelajaran (*learning rate*).
$\frac{\partial Loss}{\partial W_{in}}$	: Gradien loss terhadap bobot W_{in}.
$\frac{\partial Loss}{\partial W_{out}}$	: Gradien loss terhadap bobot W_{out}.

7. Output Vektor Embedding

Setelah pelatihan, setiap kata akan memiliki vektor embedding unik yang diambil dari matriks W_{in}. Misalnya, kata "jakarta" dapat memiliki vektor seperti [-0.0096 0.1192 0.1907 …], sementara kata "cnn" memiliki vektor yang berbeda.

Berikut ini adalah contoh hasil vektor kata *Word2Vec* pada data pertama

Tabel 3. 2 Contoh Hasil Vektor Kata *Word2Vec*

Kata	**Vektor**
jakarta	[-0.0096 0.1192 0.1907 … 0.0304 0.0473 0.1250]
cnn	[-0.1093 0.6967 0.0199 … 0.0784 -0.1382 -0.0297]
indonesia	[0.3706 0.2307 0.0261 … -0.2323 -0.0850 -0.0507]
timnas	[0.3322 0.0905 -0.1944 … 0.0631 0.2099 0.1140]
argentina	[-0.1834 -0.3350 0.1361 … 0.1755 0.2503 0.1862]
gagal	[0.2040 0.2528 -0.0639 … 0.1238 -0.3021 0.0490]
raih	[0.0921 0.3142 0.1186 … 0.0265 0.5982 0.0502]

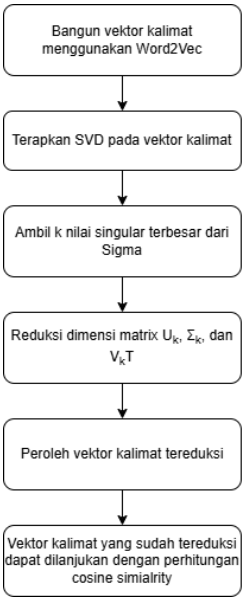
Setelah model *Word2Vec* menghasilkan vektor kata, vektor kalimat dibentuk dengan menghitung rata-rata vektor kata yang ada dalam kalimat tersebut. Sistem akan memilih kata-kata dalam kalimat yang terdapat dalam *vocabulary* model, kemudian menghitung rata-rata vektor kata tersebut untuk menghasilkan satu vektor kalimat. Berikut adalah contoh untuk hasil dari vektor kalimat yang telah dibentuk oleh *Word2Vec* pada data pertama.

Tabel 3. 3 Contoh Hasil Vektor Kalimat *Word2Vec*

Kalimat	**Vektor Kalimat**
Kalimat 1	[0.2301 0.0663 0.0589 … 0.0817 0.0517 0.0040]
Kalimat 2	[-0.1174 0.0572 0.0576 … 0.0517 0.1000 -0.0785]
Kalimat 3	[0.0536 -0.0479 0.0121 … 0.0503 0.2219 0.0777]
…	….
Kalimat 19	[1.8386 0.2663 0.8520 … -0.0538 0.0610 0.0634]
Kalimat 20	[1.8998 0.4048 1.0861 … -0.0852 0.0879 -0.2875]
Kalimat 21	[2.7700 0.4853 1.1723 … 0.1692 0.0888 -0.0090]

3.4.2 *Latent Semantic Analysis* (LSA)

Setelah proses training model *Word2Vec* selesai, langkah selanjutnya adalah menggunakan LSA (*Latent Semantic Analysis*) untuk mereduksi dimensi vektor hasil dari *Word2Vec* yang awalnya 300 menjadi 180. Berikut adalah alur kerja dari LSA.



Gambar 3. 5 *Flowchart LSA*

Pada gambar 3.5, *Latent Semantic Analysis* (LSA) digunakan untuk mereduksi dimensi vektor kalimat yang dihitung dari vektor kata yang dihasilkan oleh *Word2Vec*. Proses dimulai dengan melatih *Word2Vec* untuk menghasilkan vektor kata yang merepresentasikan makna semantik setiap kata dalam kalimat. Kemudian, vektor-vektor kata tersebut akan di rata-rata untuk menghasilkan vektor kalimat. Setelah itu, LSA diterapkan menggunakan *TruncatedSVD* untuk mereduksi dimensi vektor kalimat, sehingga vektor kalimat menjadi lebih padat dan efisien. Vektor kalimat yang telah tereduksi dimensi ini kemudian digunakan untuk menghitung *cosine similarity* antar kalimat, yang mengukur seberapa mirip kalimat satu dengan kalimat lainnya. Hasil perhitungan kesamaan ini digunakan oleh *TextRank* untuk memberi skor pada setiap kalimat, dan kalimat-kalimat dengan skor tertinggi dipilih untuk membentuk ringkasan.

Berikut ini adalah langkah-langkah dalam perhitungan reduksi dimensi vektor menggunakan LSA:

Langkah pertama adalah *Word2Vec* akan membangun vektor kalimat yang didapat dari perhitungan rata rata vektor katanya, setelah perhitungan rata-rata tersebut maka akan didapat vektor kalimat yang berdimensi 300, yang mana akan direduksi menggunakan LSA menjadi 180.

Setelah vektor kalimat *Word2Vec* didapat, maka langkah selanjutnya adalah perhitungan LSA dengan melakukan *Singular Value Decomposi*tion (SVD), menggunakan persamaan (3.7) (Srivastava & Sahami, 2009)

$$A = U \cdot \sum \cdot V^T \tag{3.7}$$

Di mana:
U : matriks yang berisi vektor kata (terdapat di bagian kiri),
$\sum$: matriks diagonal yang berisi nilai singular (faktor yang digunakan untuk mereduksi dimensi),
V^T : matriks yang berisi konteks atau kata-kata lainnya.

Dalam langkah reduksi dimensi, yaitu memilih k nilai singular terbesar dari matriks sigma, dan hanya mengambil kolom dan baris pertama dari U dan V untuk menghasilkan matriks yang lebih kecil. Matriks tereduksi A_k dapat dihitung dengan rumus (3.8) (Srivastava & Sahami, 2009):

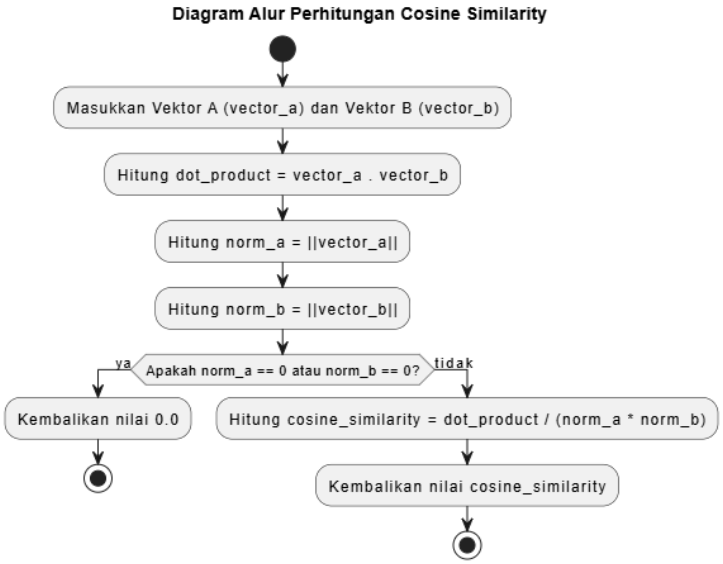
$$A_k = U_k \cdot \sum_k \cdot V_k^T \tag{3.8}$$

Setelah memperoleh matriks tereduksi, selanjutnya dapat menggunakan vektor kalimat yang telah direduksi untuk menghitung nilai *cosine similarity* antar kalimat.

Tabel 3. 4 Contoh Hasil Vektor Kalimat LSA

Kalimat	Vektor Kalimat
Kalimat 1	[1.5743 0.3699 0.7336 … -0.0244 0.0415 -0.0728]
Kalimat 2	[1.6187 -0.1239 0.5287 … -0.0577 0.000086 -0.163]
Kalimat 3	[1.7528 0.4653 0.7231 … -0.1283 -0.0890 -0.2087]
…	….
Kalimat 19	[1.8386 0.2663 0.8520 … -0.0538 0.0610 0.0634]
Kalimat 20	[1.8998 0.4048 1.0861 … -0.0852 0.0879 -0.2875]
Kalimat 21	[2.7700 0.4853 1.1723 … 0.1692 0.0888 -0.0090]

3.5 *Cosine Similarity*



Gambar 3. 6 *Flowchart Cosine Similarity*

Fungsi *cosine similarity* digunakan untuk mengukur kemiripan antara dua vektor kalimat yang dihitung menggunakan model *Word2Vec* dan LSA. Untuk Langkah-langkah dari perhitungan *cosine similarity* adalah, pertama-tama dua vektor, vektor_a dan vektor_b akan digunakan untuk menghitung nilai *dot product* antara kedua vektor tersebut dengan rumus:

$$dot\_product = \sum_{i=0}^{n} a_i b_i \tag{3.8}$$

Di mana a_i dan b_i adalah elemen-elemen dari *vector_a* dan *vector_b*, dan n adalah jumlah dimensi dari vektor-vektor tersebut. Setelah itu, dihitung juga norma (panjang) dari masing-masing vektor menggunakan rumus norma Euclidean (3.9) dan (3.10):

$$norm\_a = \sqrt{\sum_{i=1}^{n} a_i^2} \tag{3.9}$$

$$norm\_b = \sqrt{\sum_{i=1}^{n} b_i^2} \tag{3.10}$$

Norma ini digunakan untuk menormalkan perhitungan agar hasilnya dapat dibandingkan meskipun vektor-vektor tersebut memiliki panjang yang berbeda. Selanjutnya, jika salah satu norma bernilai nol, maka *cosine similarity* dianggap 0.0 karena ini menunjukkan bahwa salah satu vektor tidak memiliki informasi yang valid atau tidak terdefinisi. Jika hasil tidak 0, maka *cosine similarity* dihitung dengan rumus (3.11):

$$cosine\_similarity = \frac{dot\_product}{norm\_a \times norm\_b} \tag{3.11}$$

Hasil *cosine similarity* ini akan berada dalam rentang -1 hingga 1, yang menggambarkan sejauh mana kedua vektor tersebut memiliki arah yang sama (nilai

1 berarti sangat mirip, nilai 0 berarti tidak ada kesamaan, dan nilai -1 berarti berlawanan arah). Sebagai contoh, berikut ini adalah perhitungan *cosine similarity* untuk kalimat pertama dan kedua sebagai berikut:

1. Hitung *Dot Product*

   Dot product untuk kedua kalimat ini adalah sebagai berikut:

   $$A \cdot B = (1.57 \times 1.62) + (0.37 \times -0.11) + (0.75 \times 0.52) + \cdots = 3.498$$

   Hasil *dot productnya* adalah $A \cdot B$ = 3.4983

2. Hitung Nilai Magnitude

   $$||A|| = \sqrt{1.5719^2 + 0.3650^2 + 0.7453^2 + \cdots} = 2.3576$$

   $$||B|| = \sqrt{1.6164^2 + (-0.1069)^2 + 0.5253^2 + \cdots} = 2.2836$$

   Hasil dari magnitude dari Kalimat 1 adalah $||A|| = 2.3576$ dan hasil dari dari magnitude dari Kalimat 1 adalah $||B|| = 2.2836$

3. Hitung *Cosine Similarity*

   $$cosine\_similarity = \frac{3.4983}{2.3576 \times 2.2836} = 0.6498$$

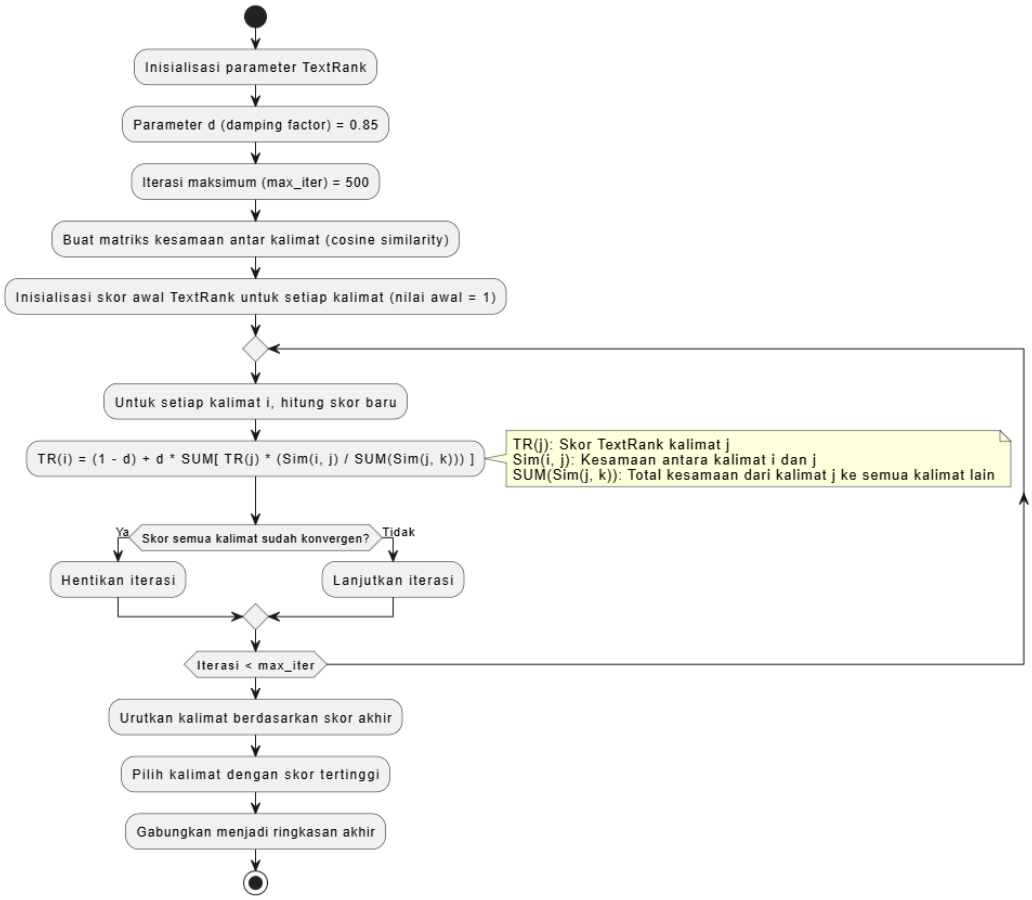
   Nilai *cosine similarity* yang dihitung ini kemudian digunakan dalam *similarity matrix* untuk mengukur kemiripan antar kalimat dalam proses pembuatan ringkasan menggunakan algoritma *TextRank*. *Similarity matrix* ini membantu menentukan relevansi antar kalimat, yang pada akhirnya memungkinkan sistem untuk memilih kalimat-kalimat yang memiliki nilai paling tinggi untuk dimasukkan dalam ringkasan akhir. Dengan demikian, hasil dari perhitungan *cosine similarity* ini mencerminkan kemiripan

semantik antar kalimat, yang sangat penting dalam menentukan kualitas ringkasan yang dihasilkan.

Tabel 3. 5 Contoh Hasil *Cosine Similarity*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0.6	0.7	0.7	0.6	0.7	0.6	0.6	0.7	0.6	0.6	0.7	0.6	0.7	0.7	0.8	0.6	0.8	0.7	0.7	0
1	0.6	0	0.8	0.6	0.6	0.6	0.7	0.6	0.6	0.4	0.5	0.6	0.6	0.7	0.6	0.7	0.7	0.6	0.6	0.6	0.6
2	0.7	0.8	0	0.8	0.7	0.7	0.6	0.7	0.6	0.6	0.6	0.8	0.7	0.7	0.6	0.7	0.7	0.6	0.6	0.7	0.7
3	0.7	0.6	0.8	0	0.7	0.8	0.7	0.8	0.7	0.7	0.6	0.8	0.8	0.7	0.6	0.7	0.6	0.7	0.6	0.7	0.7
4	0.6	0.6	0.7	0.7	0	0.7	0.7	0.8	0.7	0.6	0.7	0.6	0.8	0.6	0.5	0.6	0.5	0.6	0.6	0.6	0.6
5	0.7	0.6	0.7	0.8	0.7	0	0.8	0.7	0.7	0.8	0.7	0.8	0.7	0.8	0.7	0.8	0.7	0.7	0.6	0.7	0.7
6	0.6	0.7	0.6	0.7	0.7	0.8	0	0.8	0.7	0.6	0.7	0.7	0.7	0.7	0.6	0.7	0.6	0.7	0.6	0.6	0.6
7	0.6	0.6	0.7	0.8	0.8	0.7	0.8	0	0.7	0.6	0.8	0.7	0.8	0.6	0.5	0.6	0.6	0.6	0.6	0.6	0.6
8	0.7	0.6	0.6	0.7	0.7	0.7	0.7	0.7	0	0.7	0.8	0.7	0.7	0.6	0.6	0.6	0.6	0.7	0.7	0.7	0.7
9	0.6	0.4	0.6	0.7	0.6	0.8	0.6	0.6	0.7	0	0.6	0.7	0.6	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.6
10	0.6	0.5	0.6	0.6	0.7	0.7	0.7	0.8	0.8	0.6	0	0.7	0.7	0.5	0.5	0.6	0.5	0.6	0.6	0.6	0.6
11	0.7	0.6	0.8	0.8	0.6	0.8	0.7	0.7	0.7	0.7	0.7	0	0.7	0.7	0.7	0.8	0.7	0.6	0.6	0.7	0.7
12	0.6	0.6	0.7	0.8	0.8	0.7	0.7	0.8	0.7	0.6	0.7	0.7	0	0.6	0.5	0.6	0.5	0.6	0.6	0.6	0.6
13	0.7	0.7	0.7	0.7	0.6	0.8	0.7	0.6	0.6	0.6	0.5	0.7	0.6	0	0.8	0.8	0.7	0.6	0.6	0.7	0.7
14	0.7	0.6	0.6	0.6	0.5	0.7	0.6	0.5	0.6	0.6	0.5	0.7	0.5	0.8	0	0.7	0.6	0.6	0.5	0.6	0.7
15	0.8	0.7	0.7	0.7	0.6	0.8	0.7	0.6	0.6	0.6	0.6	0.8	0.6	0.8	0.7	0	0.7	0.6	0.6	0.7	0.8
16	0.6	0.7	0.7	0.6	0.5	0.7	0.6	0.6	0.6	0.5	0.5	0.7	0.5	0.7	0.6	0.7	0	0.5	0.5	0.7	0.6
17	0.8	0.6	0.6	0.7	0.6	0.7	0.7	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.5	0	0.8	0.7	0.8
18	0.7	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.5	0.6	0.5	0.8	0	0.7	0.7

3.6 *TextRank*



Gambar 3. 7 *Flowchart TextRank*

Algoritma *TextRank* berfungsi untuk menghasilkan ringkasan teks dengan cara menghitung skor pentingnya setiap kalimat dalam teks, berdasarkan hubungan antar kalimat. Proses dimulai dengan inisialisasi parameter, yaitu *damping factor* (d) yang diatur ke 0.85 dan iterasi maksimum (*max_iter*) yang ditentukan hingga 500. *Damping factor* mengatur seberapa besar pengaruh skor kalimat lain terhadap skor kalimat yang sedang dihitung, dengan nilai 0.85 yang berarti bahwa 85% skor kalimat dipengaruhi oleh kalimat-kalimat yang terhubung. Selanjutnya, program membangun matriks kesamaan antar kalimat menggunakan cosine similarity untuk

mengukur kemiripan semantik antara kalimat-kalimat dalam teks. Setelah itu, setiap kalimat diberikan skor awal yang sama (untuk penelitian ini adalah 1) yang akan diperbarui dalam iterasi berikutnya. Perhitungan skor untuk setiap kalimat dilakukan menggunakan rumus *TextRank* (3.12) sebagai berikut:

$$S(V_i) = (1 - \mathrm{d}) + \mathrm{d} \cdot \sum_{V_j \in I\,n(V_i)} \frac{w_{ji}}{\sum_{V_j \in I\,n(V_i)} w_{jk}} S(V_j) \qquad (3.12)$$

Dimana :

$S(V_i)$: Skor untuk simpul i
d : Faktor penyesuaian (pada penelitian ini 0.85)
V_j : Simpul tetangga dari simpul i
w_{ji} : Bobot antara simpul j dan i
$\sum w_{jk}$: Jumlah bobot keluar dari simpul j.
$S(V_j)$: Skor untuk simpul j

Untuk lebih jelasnya berikut ini adalah langkah-langkah perhitungan dari *TextRank*.

1. Menentukan Skor Awal untuk Setiap Kalimat

   Langkah pertama adalah memberikan skor awal 1 untuk setiap kalimat, karena setiap kalimat dianggap setara pada iterasi pertama. Skor awal 1 dalam algoritma *TextRank* digunakan sebagai titik awal yang setara untuk semua kalimat, mempermudah perhitungan dan memastikan penilaian yang adil berdasarkan hubungan *similarity* antar kalimat, yang kemudian diperbarui melalui iterasi untuk mencerminkan tingkat kepentingan kalimat dalam konteks keseluruhan.

2. Menghitung Normalisasi

   Pada tahap ini akan dilakukan perhitungan untuk nilai normalisasi kalimat satu terhadap kalimat lainnya, yaitu dengan cara menjumlahkan semua nilai

similarity matrix pada kalimat pertama. Berikut contohnya untuk kalimat pertama.

$$total\ similarity\ (\boldsymbol{v_1}) = 0.6498 + 0.6639 + 0.6838 + \cdots = 13.3460$$

Perhitungan normalisai juga berlaku untuk nilai *similarity* kalimat lainnya.

$$total\ similarity\ (\boldsymbol{v_2}) = 12.3651$$

$$total\ similarity\ (\boldsymbol{v_3}) = 13.0104$$

...

$$total\ similarity\ (\boldsymbol{v_{21}}) = 6.794985$$

3. Menghitung Kontribusi dari Kalimat Lainnya Terhadap v_1

Langkah selanjutnya adalah mengalikan *damping factor (d*) dengan *cosine similarity* antara v_1 dan kalimat lainnya. Berikut adalah contoh perhitungan untuk kalimat 1 iterasi 1.

$$score(v_1) = (1 - 0.85) + 0.85 \times \left(\frac{0.6498}{12.37} \times 1 + \frac{0.6639}{13.01} \times 1 + \cdots + \frac{0.4915}{6.795} \times 1\right) = 0.99$$

Jadi, skor awal untuk kalimat pertama setelah iterasi pertama adalah 0.999. perhitungan yang sama akan dilakukan untuk tiap kalimat, kemudian jika setiap kalimat telah mendapat nilai maka akan dilakukan iterasi kedua. Proses ini diulang dalam beberapa iterasi hingga maksimum iterasi tercapai atau sampai skor kalimat konvergen, yaitu tidak ada lagi perubahan signifikan dalam skor antar iterasi. Setelah iterasi selesai, kalimat-kalimat diurutkan berdasarkan skor mereka, dan kalimat dengan skor tertinggi dipilih untuk membentuk ringkasan teks. Hasil akhirnya adalah ringkasan yang terdiri dari kalimat-kalimat yang paling penting dan relevan dalam teks asli, yang dipilih berdasarkan keterhubungannya dengan kalimat lainnya.

Berikut ini adalah contoh hasil dari perhtiungan *TextRank*.

Tabel 3. 6 Contoh Hasil *TextRank*

Kalimat	Nilai
Kalimat 1	1.05158861
Kalimat 2	0.98368959
Kalimat 3	1.02681064
Kalimat 4	1.08315842
Kalimat 5	1.00553736
Kalimat 6	1.09085639
Kalimat 7	1.04552844
Kalimat 8	1.04331471
Kalimat 9	1.03010294
Kalimat 10	0.95953304
Kalimat 11	0.96033243
Kalimat 12	1.06712305
Kalimat 13	1.01549635
Kalimat 14	1.04523994
Kalimat 15	0.96287691
Kalimat 16	1.05470046
Kalimat 17	0.95737344
Kalimat 18	1.00680971
Kalimat 19	0.97127992
Kalimat 20	1.03187464
Kalimat 21	0.60677304

Kemudian berdasarkan hasil dari scoring kalimat tersebut, akan di urutkan berdasarkan nilai tertinggi ke terendah, setelah di urutkan nilianya maka ringkasan akan mengambil kalimat dengan nilai tertinggi, untuk contoh akan menggunakan 2 kalimat dengan nilai tertinggi untuk dijadikan ringkasan dan berikut ini adalah hasilnya.

Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke – 16.

Gambar 3. 8 Contoh Hasil RIngkasan

3.7 Evaluasi Hasil

Untuk mendapatkan hasil evaluasi yang optimal pada penelitian ini akan menggunakan 2 cara penilaian evaluasi, yaitu yang pertama menggunakan evaluasi

secara otomatis dengan menggunakan metrik ROUGE, dan yang kedua adalah evaluasi secara manual oleh seorang validator.

3.7.1 Evaluasi Otomatis

Evaluasi hasil secara otomatis akan menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L yang berfokus pada pengukuran *presisi*, *recall*, dan *F1-score* untuk menilai kinerja ringkasan otomatis dalam menangkap informasi penting dari teks referensi. ROUGE-1 mengukur *presisi* dan *recall* berdasarkan unigram (kata tunggal), sedangkan ROUGE-2 memperhitungkan bigram (dua kata berurutan) untuk melihat sejauh mana urutan kata dapat dipertahankan. Sementara itu, ROUGE-L mengukur kelengkapan dan koherensi keseluruhan dengan mempertimbangkan urutan kata dalam kalimat, menggunakan panjang urutan terpanjang yang ditemukan dalam kedua ringkasan sebagai acuan. Rumus untuk *presisi*, *recall*, dan *F1-score* adalah:

$$Precision = \frac{Jumlah\ kata\ yang\ ada\ di\ ringkasan\ otomatis\ dan\ referensi}{Jumlah\ kata\ di\ ringkasan\ otomatis} \quad (3.13)$$

$$Recall = \frac{Jumlah\ kata\ yang\ ada\ di\ ringkasan\ otomatis\ dan\ referensi}{Jumlah\ kata\ di\ referensi} \quad (3.14)$$

$$F1 - SCORE = 2 \times \frac{Precison\ \times\ Recall}{Precison\ +\ Recall} \quad (3.15)$$

3.7.2 Evaluasi Manual

Evaluasi manual dilakukan oleh seorang validator dalam rentang waktu 1 bulan mulai dari 23 Maret 2025 hingga 23 April 2025 dengan menggunakan data yang diambil secara acak melalui metode *Simple Random Sampling* serta rumus

Slovin untuk memperoleh jumlah sampel yang digunakan. *Simple Random Sampling* adalah teknik pengambilan sampel di mana setiap elemen dalam populasi memiliki kesempatan yang sama untuk dipilih sebagai sampel, sehingga sampel yang diambil dari 5.000 data dilakukan secara acak tanpa mempertimbangkan karakteristik khusus lainnya. Hal ini bertujuan untuk memastikan hasil yang objektif dan representatif. Kemudian untuk penentuan jumlah data sample dengan menggunakan rumus Slovin (Majdina dkk., 2024) (3.16).

$$n = \frac{N}{1 + Ne^2} \tag{3.16}$$

Dimana:
n = ukuran sampel yang dibutuhkan
N = jumlah populasi (5.000 data)
e^2 = margin of *error* (pada penelitian ini menggunakan 0.05)

Dengan menggunakan rumus di atas, ukuran sampel dihitung sebagai berikut:

$$n = \frac{5000}{1 + 5000 \times 0.05^2} = \frac{5000}{1 + 5000 \times 0.0025} = \frac{5000}{13.5} = 370$$

Dari perhitungan diatas diperoleh ukuran sampel sebanyak 370 data dari 5.000 data berita.

Selanjutya validator akan diberikan lembar penilaian serta data sample yang telah dipilih. Poin-poin yang akan dinilai oleh validator adalah sebagai berikut :

a. Konten : Sejauh mana ringkasan mencakup informasi dari teks asli.
b. Relevansi : Sejauh mana relevansi isi ringkasan dengan teks asli.
c. Redundansi : Seberapa banyak kata yang terulang dalam ringkasan dibandingkan dengan teks asli.
d. Spesifisitas : Berdasarkan teks asli yang telah anda baca, apakah

ringkasan ini terlalu umum atau terlalu rinci dalam menggambarkan informasi yang dibahas dalam teks asli.

Setiap poin akan diberikan nilai penilaian 1-7, yang mana nilai 1 adalah nilai paling buruk dan nilai 7 adalah nilai paling baik. Skor tersebut kemudian dianalisis untuk menilai kualitas keseluruhan dari hasil peringkasan yang dilakukan.

3.8 Skenario Ujicoba

Skenario uji coba dalam penelitian ini akan dilakukan dengan menguji dataset dari Indosum menggunakan metode gabungan *TextRank*, *Latent Semantic Analysis* (LSA), dan *Word2Vec*. Pengujian akan dilakukan pada tiga tingkat kompresi yang berbeda, yaitu:

a. Tingkat kompresi 10%
b. Tingkat kompresi 20%
c. Tingkat kompresi 30%

Selanjutnya akan dilakukan evaluasi kinerja menggunakan metrik ROUGE-1, ROUGE-2, dan ROUGE-L untuk menilai kualitas ringkasan yang dihasilkan pada setiap tingkat kompresi. Hasil ringkasan juga akan dievaluasi secara manual oleh seorang validator, yaitu dosen Bahasa Indonesia bernama Rifqi Rohmanul Khakim, M.Pd, untuk memberikan penilaian kualitatif terhadap akurasi, kelengkapan, dan keterbacaan ringkasan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tahap Pengujian

Untuk mendapatkan nilai hasil evaluasi yang diinginkan dari pengujian sistem peringkasan teks, terdapat tahap-tahap yang harus dilakukan untuk mendapatkan hasil yang diharapkan, seperti:

1. Melakukan *Preprocessing* pada Data

   Pada dataset yang digunakan tedapat kolom "*paragraphs*" yang berisi teks asli berita yang akan diringkas, data pada kolom ini yang akan dilakukan preprocessing dan diproses untuk diringkas, sedangkan kolom satunya adalah kolom "*summary*" yaitu kolom yang berisi ringkasan manual dari teks asli.

	paragraphs	summary	processed_paragraphs
0	Jakarta , CNN Indonesia - - Timnas Argentina k...	Timnas Argentina ditahan imbang ketika berhada...	jakarta cnn indonesia timnas argentina gagal r...
1	Anda mendambakan bermain game RPG dengan grafi...	Pooka : Magic and Mischief adalah game petuala...	damba main game rpg grafis colorful cocok main...
2	Kebakaran melanda kawasan pemukiman di Jalan K...	Kebakaran melanda kawasan pemukiman di Jalan K...	bakar landa kawasan mukim jalan kapuk raya vil...
3	Jakarta , CNN Indonesia - - Bek Juventus , Gio...	Bek Juventus , Giorgio Chiellini , memuji Gonz...	jakarta cnn indonesia bek juventus giorgio chi...
4	Jakarta , CNN Indonesia - - Predikat Film Terb...	Tahun ini pendaftar kategori Film Animasi Terb...	jakarta cnn indonesia predikat film baik acade...

Gambar 4. 1 Data Setelah *Preprocessing*

2. *Training* untuk Model *Word2Vec*

   Model akan dilatih dengan parameter *vector_size*=300 untuk mendapatkan makna semantik yang kaya dan *window*=6 untuk konteks yang luas. Kata yang muncul kurang dari dua kali diabaikan (*min_count*=2), dengan negative sampling (*negative*=15) untuk optimasi. Pelatihan menggunakan 4

worker threads, l*earning rate* = 0.03 (*alpha*=0.03), dan berlangsung sebanyak 100 epoch untuk memastikan embedding yang stabil.

```
word2vec_model = Word2Vec(
    sentences,
    vector_size=300,
    window=6,
    min_count=2,
    workers=4,
    sg=1,
    negative=15,
    alpha=0.03,
    epochs=100
)
```

Gambar 4. 2 Parameter *Word2Vec*

3. Reduksi Dimensi dengan LSA

Tahap berikutnya model *Latent Semantic Analysis* (LSA) dalam penelitian ini menggunakan TruncatedSVD dengan jumlah komponen utama sebanyak 180 (*n_components*=180) untuk mereduksi dimensi vektor kata tanpa kehilangan informasi penting. Reduksi ini bertujuan meningkatkan efisiensi komputasi dan menghilangkan noise pada representasi teks.

```
lsa = TruncatedSVD(n_components=180)
lsa_vectors = lsa.fit_transform(word_vectors)
```

Gambar 4. 3 Parameter LSA

4. TextRank

Kemudian pada bagian metode *TextRank* menghitung *cosine similarity* antar kalimat berbasis vektor *Word2Vec* yang direduksi dengan LSA, Jika ditemukan vektor nol (*zero vector*), maka nilai *similarity* diatur menjadi 0 untuk mencegah terjadinya error dalam perhitungan. Untuk parameter pada penelitian ini algoritma menggunakan damping factor 0.85, serta maksimum iterasi adalah 500.

5. Menguji Performa Sistem

   Menguji performa sistem peringkasan teks yang telah dibuat, terdapat skenario untuk tingkat kompresi ringkasan terhadap teks asli dengan tingkat kompresi 10%, 20%, dan 30%. Dari ketiga tingkat kompresi tersebut masing masing dinilai menggunakan 2 cara yakni evaluasi secara otomatis menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L dengan menampilkan nilai *recall*, *precision*, dan *f1-score* untuk setiap jenis metrik ROUGE. Selanjutnya, cara kedua adalah dengan mengevaluasi secara manual yang akan dilakukan oleh Rifqi Rohmanul Khakim, M.Pd, selaku dosen Bahasa Indonesia di Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang juga berperan sebagai validator.".

6. Analisi dan Perbandingan

   Menganalisis dan membandingkan hasil dari ketiga sampel ujicoba, untuk mengetahui tingkat kompresi manakah yang memiliki hasil ringkasan dan evaluasi paling baik diantara ketiganya.

4.2 Hasil Pengujian

Pengujian sistem dilakukan berdasarkan tahapan yang telah dijelaskan sebelumnya pada sub bab 4.1, mulai dari preprocessing, pelatihan model Word2Vec, reduksi dimensi menggunakan LSA, hingga penerapan algoritma TextRank untuk menentukan kalimat-kalimat yang akan diringkas. Pada tahap akhir, sistem dievaluasi baik secara otomatis menggunakan metrik ROUGE maupun secara manual oleh validator. Evaluasi dilakukan terhadap tiga skenario tingkat kompresi ringkasan, yaitu 10%, 20%, dan 30%, untuk mengetahui

konfigurasi mana yang menghasilkan ringkasan paling optimal berdasarkan kualitas dan kelengkapan informasi.

Pada Tabel 4.1 menunjukkan contoh hasil ringkasan untuk setiap skenario.

Tabel 4. 1 Contoh Hasil Ringkasan

Jenis Teks Berita	**Hasil Ringkasan**
Teks berita asli	Jakarta , CNN Indonesia - - Timnas Argentina kembali gagal meraih kemenangan tanpa kehadiran Lionel Messi . Kali ini tim Tango ditahan imbang Peru 2 - 2 di Stadion Nasional , Lima , Kamis (6 / 10) malam waktu setempat atau Jumat pagi WIB . Argentina membuang keunggulan dua kali saat menghadapi Peru . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke - 16 . Bek Everton itu memanfaatkan kemelut di kotak penalti Peru lewat skema sepak pojok . Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Di babak kedua , tepatnya menit ke - 58 , tuan rumah berhasil menyamakan kedudukan lewat kapten tim Jose Paolo Guerrero . Usai menerima umpan Miguel Trauco , Guerrero berhasil menahan bola dengan dada dan lolos dari kawalan Funes Mori . Mantan penyerang Bayern Munich itu kemudian melepaskan tendangan mendatar yang tidak bisa dihentikan kiper Sergio Romero . Argentina kembali unggul lewat Gonzalo Higuain pada menit ke - 78 . El Pipita berhasil meneruskan umpan terobosan Pablo Zabaleta dan mencungkil bola melewati kiper Pedro Gallese . Sayang , Argentina yang ditangani Edgardo Bauza tidak mampu mempertahankan keunggulan . Pada menit ke - 84 , Peru mendapat tendangan penalti yang dieksekusi dengan sempurna oleh Christian Cueva setelah Mori menjatuhkan Guerrero . Hasil imbang 2 - 2 membuat Argentina gagal naik dari peringkat lima klasemen sementara kualifikasi Piala Dunia 2018 zona Conmebol . Argentina kini mengoleksi 16 poin , tertinggal tiga poin dari Uruguay di puncak klasemen . Argentina untuk kali kedua beruntun meraih hasil imbang . Sebelumnya , Argentina ditahan imbang Venezuela 2 - 2 , 6 September lalu . Menariknya , di kedua laga tersebut Messi tidak tampil karena cedera . Messi saat ini sedang menjalani pemulihan cedera pangkal paha selama tiga pekan yang didapatnya saat melawan Atletico Madrid . Penyerang Barcelona itu juga dipastikan absen ketika Argentina menghadapi Paraguay di Cordoba , 11 Oktober mendatang . (har)
Ringkasan sistem (Skenario 1)	Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke - 16 .
Ringkasan sistem (Skenario 2)	Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke - 16 . Sayang , Argentina yang ditangani Edgardo Bauza tidak mampu mempertahankan keunggulan . Jakarta , CNN Indonesia - - Timnas Argentina kembali gagal meraih kemenangan tanpa kehadiran Lionel Messi .
Ringkasan sistem (Skenario 3)	Keunggulan satu gol Argentina bertahan hingga jeda babak pertama . Ramiro Funes Mori membawa Argentina unggul lewat gol pada menit ke - 16 . Sayang , Argentina yang ditangani Edgardo Bauza tidak mampu mempertahankan keunggulan . Jakarta , CNN Indonesia - - Timnas Argentina kembali gagal meraih kemenangan tanpa

	kehadiran Lionel Messi . Argentina untuk kali kedua beruntun meraih hasil imbang . Usai menerima umpan Miguel Trauco , Guerrero berhasil menahan bola dengan dada dan lolos dari kawalan Funes Mori .

Pada Tabel 4.2 menunjukkan perbedaan yang signifikan untuk rata-rata jumlah kata dan kalimat untuk teks asli berita, dibandingkan dengan ketiga skenario ujicoba ringkasan pada keseluruhan data berdasarkan tingkat kompresi.

Tabel 4. 2 Hasil Perbedaan Jumlah Kata dan Kalimat

Skenario	Jumlah Kata	Jumlah Kalimat
Teks berita asli	344	20
Ringkasan sistem (skenario 1)	39	3
Ringkasan sistem (skenario 2)	81	4
Ringkasan sistem (skenario 3)	120	6

Setelah ringkasan berhasil dibuat, maka langkah selanjutnya adalah melakukan evaluasi secara manual dengan menggunakan validator dan secara otomatis menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L dengan menampilkan nilai *recall*, *precision*, dan *f1-score* untuk ketiga skenario ujicoba.

4.2.1 Tingkat Kompresi 10%

Pada Tabel 4.3 menunjukkan hasil evaluasi secara otomatis pada skenario pengujian pertama dengan tingkat kompresi ringkasan terhadap teks asli sebesar 10%.

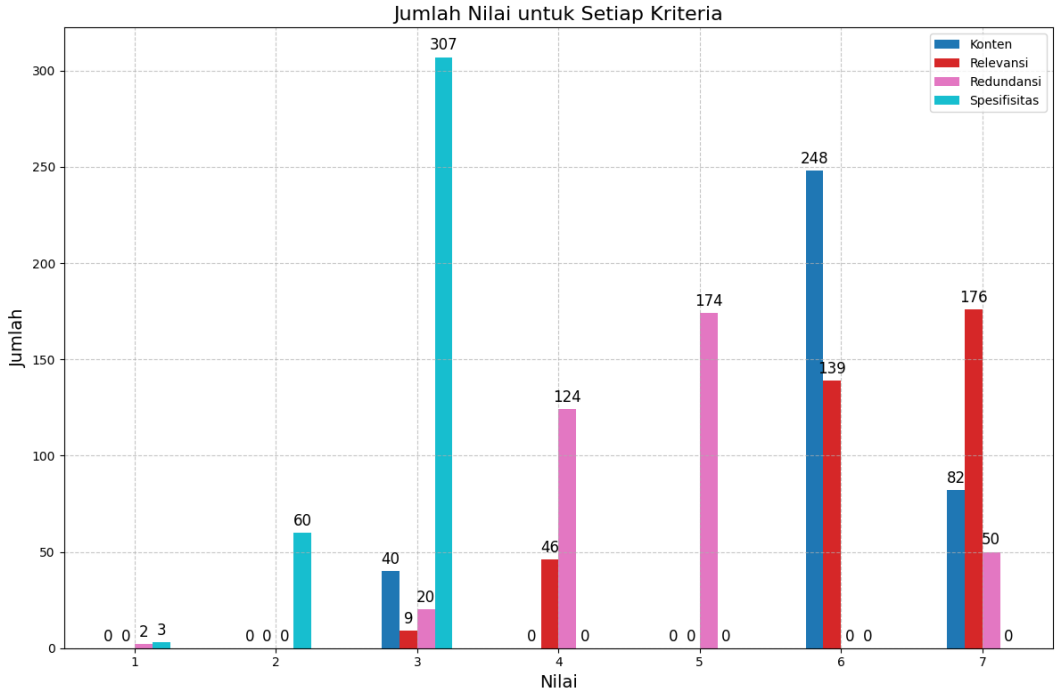
Tabel 4. 3 Hasil Evaluasi Skenario 1

	Precision	Recall	F1-Score
ROUGE-1	0.5386	0.3122	0.3796
ROUGE-2	0.3732	0.1980	0.2459
ROUGE-L	0.5097	0.2951	0.3591

Hasil evaluasi menunjukkan bahwa pada ROUGE-1, *precision* memiliki nilai 0.5386, *recall* 0.3122, dan *f1-score* 0.3796, yang menunjukkan kinerja yang

cukup baik dalam mencocokkan kata-kata. Namun, pada ROUGE-2, *precision* menurun menjadi 0.3732, *recall* 0.1980, dan *f1-score* 0.2459, yang menunjukkan penurunan kinerja dalam mencocokkan bigram. Di sisi lain, pada ROUGE-L, *precision* sedikit lebih rendah dengan nilai 0.5097, *recall* 0.2951, dan *f1-score* 0.3591, yang menunjukkan penurunan kecil dibandingkan dengan ROUGE-1 tetapi tetap lebih baik daripada ROUGE-2. Secara keseluruhan, *precision*, *recall*, dan *f1-score* mengalami penurunan pada ROUGE-2, sementara ROUGE-1 dan ROUGE-L menunjukkan kinerja yang lebih konsisten.

Gambar 4.4 menunjukkan hasil penilaian yang dilakukan oleh validator.



Gambar 4. 4 Distribusi Nilai Validator (Skenario 1)

Gambar di atas menunjukkan diagram batang dari sebaran data hasil penilaian oleh validator terhadap ringkasan dengan tingkat kompresi 10%. Pada kriteria konten, mayoritas penilaian terkonsentrasi pada nilai 6 (sebanyak 248) dan

nilai 7 (sebanyak 82), yang mengindikasikan bahwa isi ringkasan secara umum dianggap cukup lengkap dan informatif. Kriteria relevansi juga menunjukkan distribusi positif, dengan dominasi pada nilai 7 (176) dan nilai 6 (139), menandakan bahwa isi ringkasan relevan terhadap topik utama. Untuk redundansi, distribusi tertinggi berada pada nilai 5 (174) dan 4 (124), yang menunjukkan bahwa sebagian ringkasan masih mengandung informasi berulang namun dalam batas yang wajar. Sementara itu, kriteria spesifisitas memperlihatkan kecenderungan kuat pada nilai 3 (307) dan 2 (60), yang menandakan bahwa detail dalam ringkasan cenderung bersifat umum dan belum cukup spesifik. Secara keseluruhan, penilaian menunjukkan bahwa ringkasan memiliki kualitas yang baik dalam hal konten dan relevansi, namun masih perlu ditingkatkan dari segi spesifisitas.

4.2.2 Tingkat Kompresi 20%

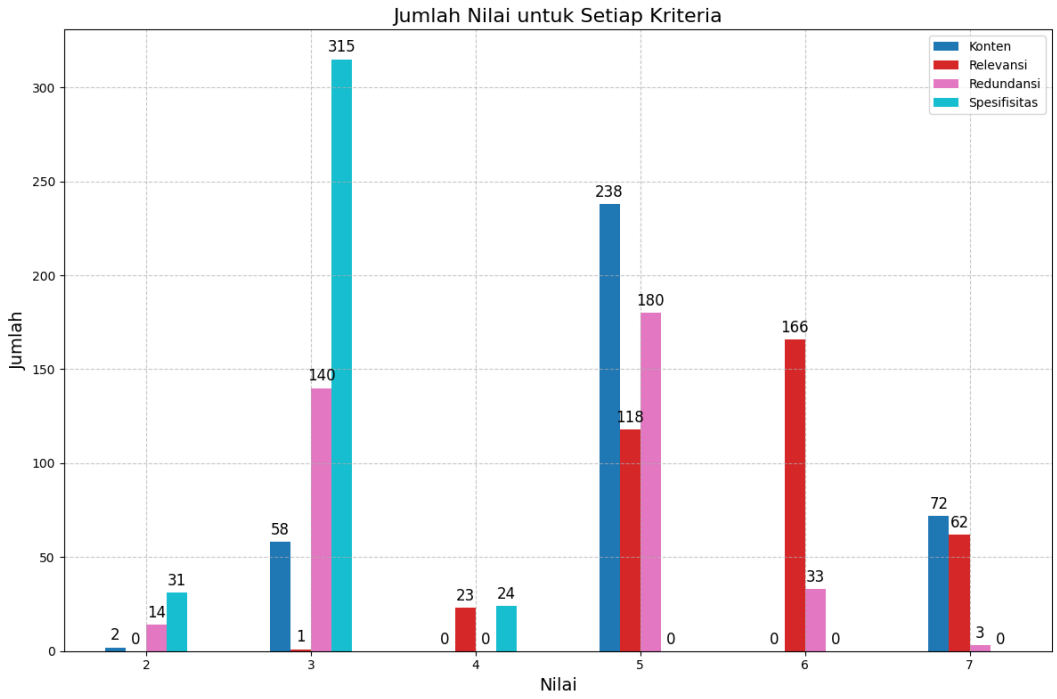
Pada Tabel 4.4 menunjukkan hasil evaluasi secara otomatis pada skenario pengujian kedua dengan tingkat kompresi ringkasan terhadap teks asli sebesar 20%.

Tabel 4. 4 Hasil Evaluasi Skenario 2

	Precision	**Recall**	**F1-Score**
ROUGE-1	0.4674	0.4827	0.4558
ROUGE-2	0.3245	0.3423	0.3154
ROUGE-L	0.4495	0.4643	0.4385

Hasil evaluasi pada Tabel 4.4 menunjukkan hasil pada ROUGE-1, *precision* tercatat sebesar 0.4674, *recall* 0.4827, dan *f1-score* 0.4558, yang menunjukkan kinerja yang cukup baik dalam mencocokkan kata-kata dari ringkasan dengan teks asli. Sementara pada ROUGE-2, *precision* menurun menjadi 0.3245, *recall* 0.3423, dan *f1-score* 0.3154, yang menunjukkan penurunan performa dalam mencocokkan bigram. Pada ROUGE-L, *precision* sebesar 0.4495, *recall* 0.4643, dan *f1-score*

0.4385, sedikit lebih rendah dibandingkan dengan ROUGE-1, namun masih memberikan hasil yang lebih baik dibandingkan dengan ROUGE-2. Secara keseluruhan, meskipun hasilnya bervariasi, ROUGE-1 memberikan kinerja terbaik, diikuti oleh ROUGE-L dan ROUGE-2 yang sedikit lebih rendah.



Gambar 4. 5 Distribusi Nilai Validator (Kompresi 20%)

Gambar 4.5 menyajikan distribusi penilaian validator terhadap ringkasan dengan tingkat kompresi 20%. Pada aspek konten, sebagian besar penilaian berkumpul pada nilai 5 (238) dan nilai tinggi seperti 7 (72), yang menunjukkan bahwa meskipun diringkas secara signifikan, informasi inti tetap dapat dipertahankan dengan baik. relevansi juga mendapatkan penilaian tinggi, khususnya pada nilai 6 (166) dan 7 (62), menandakan bahwa isi ringkasan tetap sesuai dan berhubungan erat dengan topik aslinya. Sementara itu, redundansi terlihat masih cukup terkendali, dengan penilaian tertinggi pada nilai 5 (180) dan 3

(140), mengindikasikan bahwa pengulangan informasi tidak terlalu dominan. Namun, pada aspek spesifisitas, penilaian paling banyak berada pada nilai 3 (315), yang mencerminkan bahwa detail dalam ringkasan masih belum cukup tajam atau khusus. Secara keseluruhan, ringkasan dengan kompresi 20% dinilai cukup informatif dan relevan, namun masih memiliki ruang untuk peningkatan dalam penyampaian rincian yang lebih spesifik.

4.2.3 Tingkat Kompresi 30%

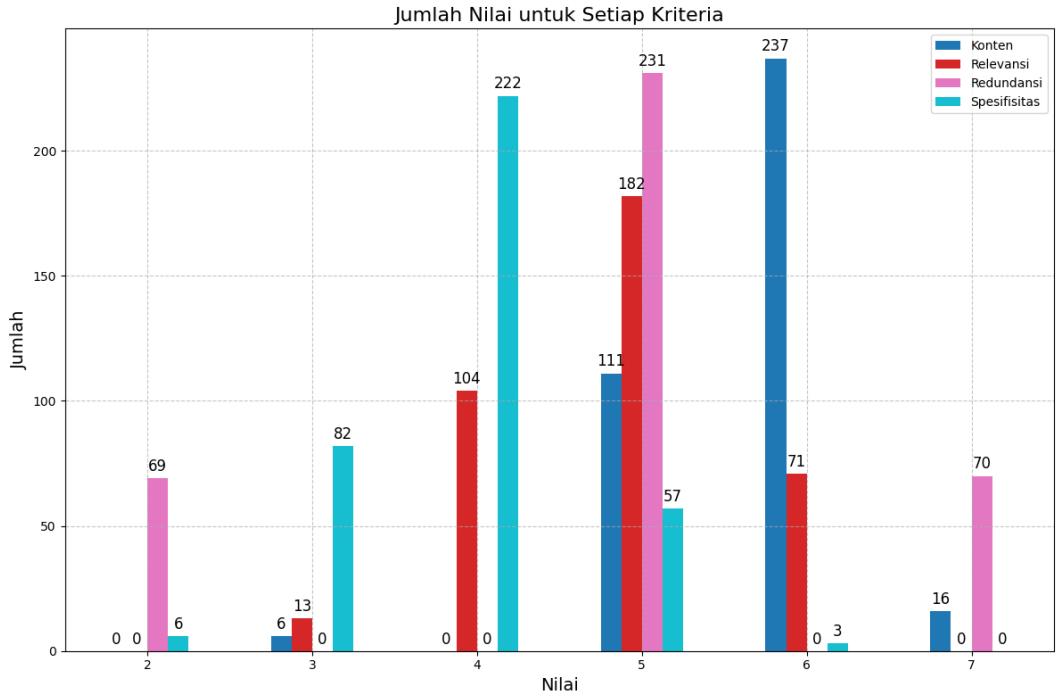
Pada Tabel 4.5 menunjukkan hasil evaluasi secara otomatis pada skenario pengujian pertama dengan tingkat kompresi ringkasan terhadap teks asli sebesar 30%.

Tabel 4. 5 Hasil Evaluasi Skenario 3

	Precision	**Recall**	**F1-Score**
ROUGE-1	0.4248	0.6013	0.4808
ROUGE-2	0.2968	0.4570	0.3433
ROUGE-L	0.4129	0.5848	0.4675

Berdasarkan hasil evaluasi yang tedapat pada tabel 4.5, untuk ROUGE-1, *precision* tercatat sebesar 0.4248, *recall* 0.6013, dan *f1-score* 0.4808, yang menunjukkan kinerja yang cukup baik dalam mencocokkan kata-kata antara ringkasan dan teks asli. Pada ROUGE-2, *precision* mengalami penurunan menjadi 0.2968, *recall* 0.4570, dan *f1-score* 0.3433, yang mencerminkan penurunan kemampuan dalam mencocokkan bigram. Sementara itu, pada ROUGE-L, *precision* tercatat 0.4129, *recall* 0.5848, dan *f1-score* 0.4675, yang sedikit lebih rendah dibandingkan dengan ROUGE-1, namun tetap menunjukkan kinerja yang lebih baik dibandingkan dengan ROUGE-2. Secara keseluruhan, meskipun terdapat

variasi dalam hasilnya, ROUGE-1 menunjukkan kinerja terbaik, diikuti oleh ROUGE-L dan ROUGE-2 yang sedikit lebih rendah.



Gambar 4. 6 Distribusi Nilai Validator (Kompresi 30%)

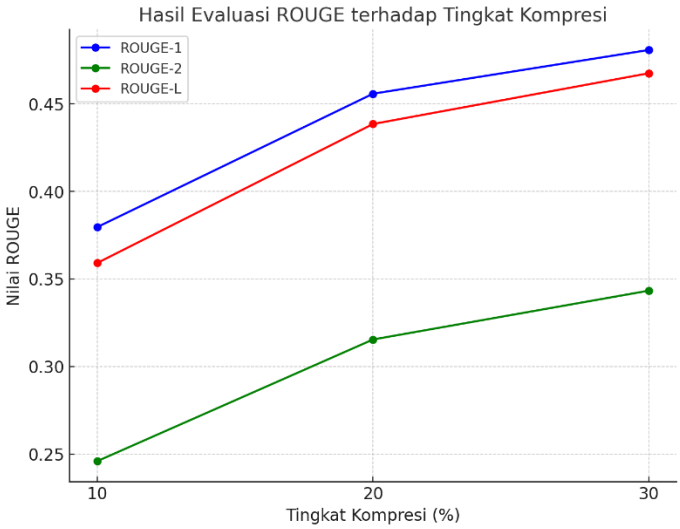
Gambar 4.6 menampilkan distribusi penilaian dari validator terhadap ringkasan dengan tingkat kompresi 30%. Penilaian pada aspek konten menunjukkan dominasi pada nilai 6 (237) dan 5 (111), menandakan bahwa sebagian besar ringkasan dianggap cukup informatif meskipun diringkas lebih padat. Kriteria relevansi juga mendapatkan banyak penilaian tinggi pada nilai 5 (182) dan 6 (71), yang mencerminkan bahwa inti dari informasi masih relevan dengan topik utama. Pada aspek redundansi, nilai tertinggi tercatat pada angka 5 (231), menunjukkan bahwa sebagian besar ringkasan dinilai cukup ringkas dan tidak bertele-tele. Sementara itu, spesifisitas memperlihatkan kecenderungan tinggi pada nilai 4 (222)

dan 3 (82), yang menunjukkan bahwa detail informasi dalam ringkasan masih belum sepenuhnya spesifik.

4.3 Pembahasan

Penelitian ini menggunakan metrik ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) untuk mengevaluasi secara otomatis kualitas ringkasan teks yang dihasilkan oleh model peringkasan. ROUGE mengukur kesamaan antara ringkasan otomatis dengan teks referensi (biasanya di ringkas oleh manusia), dengan fokus pada tiga aspek utama: ROUGE-1, yang mengukur kesamaan unigram (kata tunggal) ROUGE-2, yang mengukur kesamaan bigram (dua kata berturut-turut) dan ROUGE-L digunakan untuk menilai kesamaan antara ringkasan dan teks aslinya dengan melihat urutan kata terpanjang yang muncul secara berurutan pada keduanya. Metrik ini berguna untuk menilai seberapa baik model dapat memilih kata-kata yang relevan (*precision*), mencakup informasi yang cukup (*recall*), serta mencapai keseimbangan antara keduanya (*f1-Score*).

Sebagai kelanjutan dari penggunaan metrik ini, dilakukan pengujian terhadap ringkasan dengan tiga variasi tingkat kompresi, yaitu 10%, 20%, dan 30%, untuk menilai sejauh mana tingkat pemadatan informasi memengaruhi kualitas ringkasan. Evaluasi ini menghasilkan nilai ROUGE yang menunjukkan pola peningkatan seiring naiknya tingkat kompresi, sebagaimana ditampilkan pada grafik pada Gambar 4.8 berikut, yang memperlihatkan tren nilai ROUGE-1, ROUGE-2, dan ROUGE-L pada setiap skenario kompresi yang diuji.



Gambar 4. 7 Grafik Perbandingan Hasil Evaluasi

Secara umum, grafik diatas memperlihatkan bahwa semakin tinggi tingkat kompresi, semakin baik pula skor ROUGE yang diperoleh. Hal ini mengindikasikan bahwa ringkasan dengan isi yang lebih panjang mampu menangkap lebih banyak informasi penting dari teks asli. Tabel 4.6 akan merinci hasil evaluasi dari metrik ROUGE pada ketiga skenario ujicoba..

Tabel 4. 6 Perbandingan Rata-rata Skor Pengujian

Tingkat Kompresi	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
10%	0.3796	0.2459	0.3591
20%	0.4558	0.3154	0.4385
30%	0.4808	0.3433	0.4675

Hasil evaluasi yang ditunjukkan dalam Tabel 4.6 menggambarkan nilai ROUGE untuk tiga tingkat kompresi (10%, 20%, dan 30%) yang diuji pada sistem. Dari hasil yang terlihat, dapat dilihat bahwa seiring dengan meningkatnya tingkat kompresi, nilai evaluasi juga mengalami peningkatan, khususnya pada metrik ROUGE-1, ROUGE-2, dan ROUGE-L. Pada tingkat kompresi 10%, nilai ROUGE-1 tercatat sebesar 0.3796, ROUGE-2 sebesar 0.2459, dan ROUGE-L sebesar

0.3591. Ketika tingkat kompresi ditingkatkan menjadi 20%, terdapat kenaikan signifikan di ROUGE-1 (0.4558), ROUGE-2 (0.3154), dan ROUGE-L (0.4385). Pada tingkat kompresi tertinggi, yaitu 30%, nilai evaluasi kembali meningkat pada ketiga metrik tersebut, dengan ROUGE-1 sebesar 0.4808, ROUGE-2 sebesar 0.3433, dan ROUGE-L mencapai 0.4675.

Selain evaluasi otomatis menggunakan metrik ROUGE, kualitas ringkasan juga dinilai melalui penilaian subjektif oleh validator berdasarkan empat aspek: Konten, Relevansi, Redundansi, dan Spesifisitas. Evaluasi ini bertujuan untuk mengamati sejauh mana ringkasan yang dihasilkan benar-benar relevan dan berkualitas dari sudut pandang manusia. Berdasarkan ketiga grafik distribusi penilaian terhadap ringkasan dengan tingkat kompresi 10%, 20%, dan 30%, terlihat adanya perbedaan dalam persepsi kualitas dari masing-masing ringkasan. Pada tingkat kompresi 10%, penilaian tinggi tampak pada aspek konten dan relevansi, namun redundansi dan spesifisitas cenderung rendah, yang mengindikasikan bahwa meskipun informatif, ringkasan masih memuat informasi berulang dan kurang spesifik. Kompresi 20% menunjukkan perbaikan dalam hal relevansi serta terdapat penurunan di aspek konten dan redundansi, sedangkan nilai spesifisitas masih rendah. Sementara itu, pada tingkat kompresi 30%, grafik menunjukkan distribusi nilai yang lebih merata dan tinggi pada hampir semua kriteria, termasuk redundansi dan spesifisitas yang mengalami peningkatan. Hal ini mengindikasikan bahwa ringkasan dengan kompresi 30% dinilai paling baik oleh validator karena mampu menyajikan informasi yang ringkas namun tetap relevan, tidak bertele-tele, serta

mengandung informasi yang lebih spesifik dan fokus dibandingkan dua tingkat kompresi lainnya.

Dengan mempertimbangkan hasil dari evaluasi otomatis dan manual, dapat disimpulkan bahwa tingkat kompresi 30% memberikan performa terbaik secara keseluruhan. Ringkasan pada tingkat ini tidak hanya mencapai skor ROUGE tertinggi, tetapi juga dinilai paling berkualitas oleh validator karena mampu menyampaikan informasi yang padat, relevan, dan lebih fokus dibandingkan dua tingkat kompresi lainnya..

4.4 Integrasi Islam

Sistem peringkasan teks otomatis ini sejalan dengan dua konsep muamalah, yaitu muamalah ma'a Allah dan muamalah ma'a an-nas. Muamalah ma'a Allah mencakup segala aspek yang berkaitan dengan hubungan manusia dengan Tuhan, sementara muamalah ma'a an-nas berkaitan dengan interaksi antar sesama manusia. Dalam konteks muamalah ma'a Allah, peringkasan teks dapat merefleksikan pentingnya efisiensi dan pengelolaan waktu yang bijaksana.. Sementara itu, pada aspek muamalah ma'a an-nas, sistem peringkasan teks ini memfasilitasi komunikasi yang lebih jelas dan efektif antar pengguna, serta membantu dalam menyampaikan informasi secara ringkas dan akurat.

4.4.1 Muamalah Ma'a Allah

Dalam dunia informasi yang terus berkembang, banyak artikel berita yang isi artikelnya disajikan dengan panjang lebar, bahkan seringkali judulnya kurang sesuai dengan isi. Akan tetapi masalah tersebut dapat dipecahkan dengan adanya

sistem peringkasan teks yang dapat membantu pembaca memahami informasi utama dari artikel berita tanpa kehilangan informasi penting didalamnya. Dalam konteks Islam, prinsip ini dapat dikaitkan dengan konsep muamalah ma'a Allah, yaitu berinteraksi dengan Allah melalui amal-amal yang baik, termasuk dalam cara menyampaikan informasi. Hal ini selaras dengan agama Islam yang menajarkan akan pentingnya menyampaikan perkataan secara jelas, padat, dan tidak berbelit-belit, seperti yang terdapat dalam firman Allah Surah Al-Isra ayat 36:

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهٖ عِلْمٌۗ اِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ اُولٰۤىِٕكَ كَانَ عَنْهُ مَسْـُٔوْلًا ﴿٣٦﴾

"Dan janganlah kamu mengikuti apa yang kamu tidak mempunyai pengetahuan tentangnya. Sesungguhnya pendengaran, penglihatan dan hati, semuanya itu akan diminta pertanggungan jawabnya" (QS. Al-Isra'/17:36)

Menurut Tafsir As-Sa'di, ayat ini menekankan bahwa setiap manusia akan dimintai pertanggungjawaban atas semua yang ia dengar, lihat, dan pikirkan. Seorang hamba yang sadar akan hal ini seharusnya memanfaatkan seluruh anggota tubuh yang diberikan Allah semata-mata untuk beribadah kepada-Nya. Ini mencakup tindakan menjaga perkataan agar tidak sembarangan, menggunakan akal untuk mempertimbangkan manfaat, serta menahan diri dari hal-hal yang dibenci oleh Allah. Berdasarkan tafsir tersebut menekankan tentang pentingnya selektivitas dalam menyampaikan informasi. Karena seringkali artikel berita itu tidak sinkron antara judul dan isinya, serta memiliki teks yang panjang dan sulit untuk dimengerti. Dengan adanya sistem peringkasan teks pembaca dapat langsung mengetahui informasi penting yang ada di berita tanpa harus membaca teks berita yang panjang.

Hadis Nabi Muhammad shallallahu 'alaihi wasallam juga menegaskan pentingnya komunikasi yang ringkas namun bermakna. Rasulullah bersabda:

عَنْ وَاصِلِ بْنِ حَيَّانَ قَالَ: قَالَ أَبُو وَائِلٍ: خَطَبَنَا عَمَّارٌ فَأَوْجَزَ وَأَبْلَغَ فَلَمَّا نَزَلَ قُلْنَا يَا أَبَا الْيَقْظَانِ لَقَدْ أَبْلَغْتَ وَأَوْجَزْتَ فَلَوْ كُنْتَ تَنَفَّسْتَ فَقَالَ: إِنِّي سَمِعْتُ رَسُولَ الله يَقُولُ إِنَّ طُولَ صَلاَةِ الرَّجُلِ وَقِصَرَ خُطْبَتِهِ مَئِنَّةٌ مِنْ فِقْهِهِ فَأَطِيلُوا الصَّلاَةَ وَاقْصُرُوا الْخُطْبَةَ وَإِنَّ مِنَ الْبَيَانِ لَسِحْرًا.

"Dari Washil bin Hayyan, dia berkata, Abu Wa'il berkata, 'Ammar pernah memberi khutbah kepada kami dengan singkat dan padat isinya. Dan ketika turun, kami katakan kepadanya, 'Wahai Abu Yaqzhan, sesungguhnya Engkau telah menyampaikan dan menyingkat khutbah, kalau saja Engkau memanjangkannya.'" Maka dia menjawab, sesungguhnya aku pernah mendengar Rasulullah shallallahu 'alaihi wasallam bersabda, 'Sesungguhnya panjangnya salat seseorang dan pendek khutbahnya itu menjadi ciri pemahaman yang baik dalam agama. Oleh karena itu, perpanjanglah salat dan perpendeklah khutbah. Dan sesungguhnya di antara bagian dari penjelasan itu mengandung daya tarik." (H.R. Muslim)

Dari hadis ini, kita dapat mengambil pelajaran bahwa keberhasilan dalam menyampaikan pesan terletak pada kemampuan menjelaskan inti masalah tanpa bertele-tele. Hal ini sejalan dengan fungsi sistem peringkasan teks yang dapat membantu menyajikan informasi dengan singkat dan jelas, namun tetap mempertahankan substansi.

Sistem peringkasan teks otomatis dapat dimaknai sebagai bentuk implementasi nilai-nilai Islam dalam teknologi modern. Dengan meringkas teks, kita tidak hanya membantu manusia menghemat waktu dan energi, tetapi juga melaksanakan anjuran Islam untuk menyampaikan sesuatu dengan tepat sasaran. Sebagaimana disebutkan hadist yang diriwayatkan oleh Imam Ahmad bin Hanbal:

حَدَّثَنَا قُرَيْشُ بْنُ إِبْرَ عْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ إِنَّ طُولَ صَلَاةِ الرَّجُلِ وَقِصَرَ خُطْبَتِهِ مَئِنَّةٌ اهِيمَ قَالَ حَدَّثَنَا عَبْدُ الرَّحْمَنِ بْنُ عَبْدِ الْمَلِكِ بْنِ أَبْجَرَ عَنْ أَبِيهِ عَنْ وَاصِلِ بْنِ حَيَّانَ قَالَ قَالَ أَبُو وَائِلٍ

خَطَبَنَا عَمَّارٌ فَأَبْلَغَ وَأَوْجَزَ فَلَمَّا نَزَلَ قُلْنَا يَا أَبَا الْيَقْظَانِ لَقَدْ أَبْلَغْتَ وَأَوْجَزْتَ فَلَوْ كُنْتَ تَنَفَّسْتَ قَالَ إِنِّي سَمِ مِنْ فِقْهِهِ فَأَطِيلُوا الصَّلَاةَ وَأَقْصِرُوا الْخُطْبَةَ فَإِنَّ مِنْ الْبَيَانِ لَسِحْرًا

"Telah menceritakan kepada kami [Quraisy bin Ibrahim] telah menceritakan kepada kami [Abdurrahman bin Abdul Malik bin Abjar] dari [Bapaknya] dari [Washil bin Hayyan] ia berkata, [Abu Wa`il] berkata, "Ammar pernah berkhutbah di hadapan kami dengan khutbah yang singkat menyentuh hati. Saat ia turun, maka kami berkata, "Wahai Abu Yaqzhan, sungguh kamu telah menyampaikan khutbah yang menyentuh namun ringkas, sekiranya kamu mau memangjangkan (khutbah)." Ammar lalu berkata, "Aku mendengar Rasulullah shallallahu 'alaihi wasallam bersabda: "Sungguh, panjangnya shalat dan ringkasnya khutbah yang disampaikan oleh seseorang adalah tanda kefakihannya. Maka panjangkanlah shalat dan ringkaskanlah khutbah. Dan sungguh di antara indahnya penjelasan adalah bagian dari sihir." (H.R. Ahmad no 17598)

Sihir yang dimaksud dalam hadis ini adalah kemampuan untuk menarik perhatian dengan penjelasan yang jelas dan indah, tanpa perlu menggunakan kata-kata yang berlebihan. Dengan demikian, sistem peringkasan teks dapat menjadi sarana dalam menjalankan muamalah ma'a Allah, mengingat semua tindakan manusia pada akhirnya akan dipertanggungjawabkan di hadapan-Nya.

4.4.2 Muamalah Ma'a An-Nas

Muamalah ma'a an-nas, yang berarti hubungan atau interaksi dengan sesama manusia, mengajarkan kita untuk memberikan manfaat dan meringankan beban orang lain. Dalam konteks ini, sistem peringkasan teks dirancang untuk membantu pembaca memahami inti informasi dari teks yang panjang dengan lebih cepat dan efisien. Hal ini sejalan dengan prinsip Islam yang mendorong umatnya untuk saling membantu dalam hal-hal yang mendatangkan kebaikan dan manfaat. Sebagaimana firman Allah dalam Surah Al-Qashash ayat 77:

وَابْتَغِ فِيْمَآ اٰتٰىكَ اللّٰهُ الدَّارَ الْاٰخِرَةَ وَلَا تَنْسَ نَصِيْبَكَ مِنَ الدُّنْيَا وَاَحْسِنْ كَمَآ اَحْسَنَ اللّٰهُ اِلَيْكَ وَلَا تَبْغِ الْفَسَادَ فِى الْاَرْضِۗ اِنَّ اللّٰهَ لَا يُحِبُّ الْمُفْسِدِيْنَ ٧٧

"Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan bahagianmu dari (kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan." (QS. Al-Qashash/28:77)

Tafsir As-Sa'di terhadap Surah Al-Qashash ayat 77 menekankan pentingnya memanfaatkan karunia Allah untuk kebaikan dunia dan akhirat. Sistem peringkasan teks merupakan wujud nyata dari perintah untuk "berbuat baik kepada hamba-hamba Allah" dengan mempermudah pengguna mendapatkan inti informasi dari teks yang panjang, sehingga waktu mereka dapat dimanfaatkan lebih baik untuk aktivitas yang mendekatkan diri kepada Allah. Selain itu, ayat ini juga mengingatkan untuk tidak melupakan bagian duniawi, yang berarti teknologi dapat digunakan secara bijak untuk meningkatkan produktivitas tanpa merusak nilai agama.

Hadis Rasulullah SAW yang diriwayatkan oleh Imam Ahmad dalam musnadnya juga menegaskan pentingnya saling membantu karena Allah:

حَدَّثَنَا وَكِيعٌ حَدَّثَنَا شُعْبَةُ عَنْ أَبِي الْفَيْضِ عَنْ سُلَيْمِ بْنِ عَامِرٍ قَالَ كَانَ بَيْنَ مُعَاوِيَةَ وَبَيْنَ قَوْمٍ مِنْ الرُّومِ عَهْدٌ فَخَرَجَ مُعَاوِيَةُ قَالَ فَجَعَلَ يَسِيرُ فِي أَرْضِهِمْ حَتَّى يَنْقُضُوا فَيُغِيرَ عَلَيْهِمْ فَإِذَا رَجُلٌ يُنَادِي فِي نَاحِيَةِ النَّاسِ وَفَاءٌ لَا غَدْرٌ فَإِذَا هُوَ عَمْرُو بْنُ عَبَسَةَ فَقَالَ سَمِعْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ مَنْ كَانَ بَيْنَهُ وَبَيْنَ قَوْمٍ عَهْدٌ فَلَا يَشُدَّ عُقْدَةً وَلَا يَحُلَّ حَتَّى يَمْضِيَ أَمَدُهَا أَوْ يَنْبِذَ إِلَيْهِمْ عَلَى سَوَا

"Telah menceritakan kepada kami [Waki'] Telah menceritakan kepada kami [Syu'bah] dari [Abu Al Faidl] dari [Sulaim bin Amir] ia berkata: Antara

Mu'awiyah dan orang-orang Romawi terdapat suatu perjanjian. Mu'awiyah pun keluar dan berjalan menelusuri perkampungan mereka, hingga jika mereka sampai melanggar perjanjian, maka mereka diserang. Tiba-tiba seorang laki-laki yang berada di kerumunan manusia menyerukan, "Penuhilah janji dan hendaklah tidak ada pengkhianatan." Dan ternyata ia adalah [Amru bin Abasah]. Lalu ia berkata: Saya mendengar Rasulullah shallallahu 'alaihi wa sallam bersabda: "Barangsiapa yang antara ia dan suatu kaum terdapat suatu perjanjian, maka janganlah ia memperberat perjanjian itu (dengan menambah syarat-syaratnya) atau mencederai ikatannya, hingga batas waktunya berakhir atau hingga mereka sama-sama membatalkannya." (H.R. Ahmad no 18621)

Hadist ini mengingatkan bahwa saling membantu dalam kebaikan merupakan bentuk ibadah yang mendatangkan cinta Allah. Oleh karena itu, sistem peringkasan teks dapat dianggap sebagai bentuk implementasi dari nilai-nilai ini, di mana teknologi digunakan untuk memudahkan orang lain dalam memperoleh informasi, menghemat waktu, dan mendukung produktivitas dalam pekerjaan maupun pembelajaran.

Sistem peringkasan ini tidak hanya memberikan manfaat praktis, tetapi juga membantu pengguna untuk memanfaatkan waktu mereka dengan lebih baik dalam memenuhi kewajiban terhadap Allah SWT. Dengan mempermudah akses terhadap informasi, sistem ini menjadi sarana untuk mempercepat ikhtiar dalam memahami ilmu dan kebenaran, yang pada akhirnya bermuara pada pencapaian kebahagiaan dunia dan akhirat. Ini juga termasuk bagian dari berbuat baik kepada orang lain sebagaimana yang diajarkan dalam Islam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil menerapkan algoritma *TextRank* dalam peringkasan teks berita berbahasa Indonesia dengan menggunakan metode *Word2Vec* dan LSA untuk reduksi dimensi. Hasil evaluasi menggunakan metrik ROUGE-1, ROUGE-2, dan ROUGE-L menunjukkan bahwa sistem yang dikembangkan mampu menghasilkan ringkasan dengan kualitas yang baik pada tingkat kompresi paling optimal adalah 30% dibandingkan dengan dua skenario lainnya. Dengan ROUGE-1 mencapai nilai 0.4808, ROUGE-2 0.3433, dan ROUGE-L 0.4675, sistem ini mampu menangkap informasi penting dari teks asli secara efektif, meskipun ada penurunan kinerja pada ROUGE-2 yang mengukur kesamaan bigram. Secara keseluruhan, hasil ini menunjukkan bahwa kombinasi *TextRank* dengan *Word2Vec* dan LSA dapat meningkatkan kualitas ringkasan teks berita berbahasa Indonesia.

Hal ini ditunjukkan bahwa penelitian ini berhasil mencapai tujuan untuk memperoleh hasil yang lebih baik dibandingkan dengan penelitian sebelumnya. Berdasarkan hasil evaluasi, sistem yang dikembangkan dalam penelitian ini menunjukkan performa yang lebih baik. Pada tingkat kompresi 30%, *precision* dari metode yang dikembangkan mencapai 0. 4248, *recall* mencapai 0.6013, dengan ROUGE-1 sebesar 0.4808, sementara pada penelitian sebelumnya, metode *LexRank* dan YAKE hanya mencatat *precision* 0.386, *recall* 0.586, dan ROUGE-1 0.453. Tidak hanya itu, sistem yang dikembangkan dalam penelitian ini mampu

memberikan keseimbangan yang lebih baik dengan nilai 0.3433 pada ROUGE-2 dan ROUGE-L mencapai 0.4675, menunjukkan kualitas yang lebih seimbang dan optimal. Dengan demikian, penelitian ini berhasil menghasilkan sistem peringkasan teks yang lebih baik, lebih optimal, dan lebih relevan untuk teks berita berbahasa Indonesia.

5.2 Saran

Penelitian peringkasan teks berita dengan menggabungkan metode *TextRank*, LSA, dan *Word2Vec* ini masih memiliki banyak kekurangan, oleh karena itu untuk mendapatkan hasil ringkasan yang lebih baik lagi terdapat saran untuk penelitian kedepannya:

1 Pengujian pada dataset lain serta pengujian pada dataset yang ukurannya lebih besar lagi untuk menguji kefektifan metode dalam meringkas selain dataset dari Indosum.

2 Menggunakan metode peringkasan lain yang sudah terbukti baik untuk diterapkan dalam peringkasan teks seperti BERT, GPT, Transformers, dan lain-lain.

3 Menggunakan metode ekstraksi fitur lain seperti *Glove*, *Word2Vec* CBOW, dan lain-lain.

DAFTAR PUSTAKA

Abdolahi, M., & Zahed, M. (2019). Textual Coherence Improvement of Extractive Document Summarization Using Greedy Approach and Word Vecto. *International Journal of Modern Education and Computer Science*, *11*(4), 23–31. https://doi.org/10.5815/ijmecs.2019.04.03

Barbella, M., & Tortora, G. (2022). Rouge Metric Evaluation for Text Summarization Techniques. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4120317

Dai, S., Li, K., Luo, Z., Zhao, P., Hong, B., Zhu, A., & Liu, J. (2024). AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks. *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023*, *5*(1), 13–21. https://doi.org/10.60087/jaigs.v5i1.149

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679. https://doi.org/https://doi.org/10.1016/j.eswa.2020.113679

Giri, V. V, Math, M. M., & Kulkarni, U. P. (2023). Computational Intelligence and Machine Learning Marathi Extractive Text Summarization using Latent Semantic Analysis and Fuzzy Algorithms. *Computational Intelligence and Machine Learning*, *4*(1), 33–38. https://doi.org/10.36647/CIML/04.02.A001

Goyal, P., Pandey, S., & Jain, K. (2018). Deep Learning for Natural Language Processing. Dalam *Deep Learning for Natural Language Processing*. Apress. https://doi.org/10.1007/978-1-4842-3685-7

Haider, M. M., Hossin, Md. A., Mahi, H. R., & Arif, H. (2020). Automatic Text Summarization Using Gensim Word2Vec and K-Means Clustering Algorithm. *2020 IEEE Region 10 Symposium (TENSYMP)*, 283–286. https://doi.org/10.1109/TENSYMP50017.2020.9230670

Jane C. Patosa, K., James M. Hernandez, M., Agustin, V. A., Regala, R., Mata, K. E., Michael A. Cortez, D., Mahusay, L., Bitancor, A., & Caubang, P. (2022). Enhancement of TextRank Algorithm using Coreference Resolution. *International Journal of Research Publications*, *101*(1). https://doi.org/10.47119/ijrp1001011520223190

Juan-Manuel, & Torres-Moreno. (2014). *Automatic Text Summarization*. https://doi.org/10.1002/9781119004752.ch1

Juna, M. F., & Hayaty, M. (2023). The observed preprocessing strategies for doing automatic text summarizing. *Computer Science and Information Technologies*, *4*(2), 119–126. https://doi.org/10.11591/csit.v4i2.pp119-126

Kirmani, M., Kaur, G., & Mohd, M. (2024). Analysis of Abstractive and Extractive Summarization Methods. *International Journal of Emerging Technologies in Learning (iJET)*, *19*(01), 86–96. https://doi.org/10.3991/ijet.v19i01.46079

Kurniawan, K., & Louvan, S. (2018). IndoSum: A New Benchmark Dataset for Indonesian Text Summarization. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 215–220. https://doi.org/10.1109/IALP.2018.8629109

Majdina, N. I., Pratikno, B., & Tripena, A. (2024). PENENTUAN UKURAN SAMPEL MENGGUNAKAN RUMUS. *Jurnal Ilmiah Matematika dan Pendidikan Matematika (JMP)*, *16*, 73–84. https://doi.org/10.20884/1.jmp.2024.16.1.11230

Majerczak, P., & Strzelecki, A. (2022). Trust, Media Credibility, Social Ties, and the Intention to Share Information Verification in an Age of Fake News. *Behavioral Sciences*, *12*(2). https://doi.org/10.3390/bs12020051

MR ADEPU RAJESH, & DR TRYAMBAK HIWARKAR. (2023). Exploring Preprocessing Techniques for Natural LanguageText: A Comprehensive Study Using Python Code. *international journal of engineering technology and management sciences*, *7*(5), 390–399. https://doi.org/10.46647/ijetms.2023.v07i05.047

Rahmadhany, A., Aldila Safitri, A., & Irwansyah, I. (2021). Fenomena Penyebaran Hoax dan Hate Speech pada Media Sosial. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, *3*(1), 30–43. https://doi.org/10.47233/jteksis.v3i1.182

Ramezani, M., Shahryari, M.-S., Feizi-Derakhshi, A.-R., & Feizi-Derakhshi, M.-R. (2023). Unsupervised Broadcast News Summarization; a Comparative Study on Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA). *2023 28th International Computer Conference, Computer Society of Iran (CSICC)*, 1–7. https://doi.org/10.1109/CSICC58665.2023.10105403

Rani, U., & Bidhan, K. (2021). Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA. *Journal of scientific research*, *65*(01), 304–311. https://doi.org/10.37398/jsr.2021.650140

Srivastava, A., & Sahami, M. (2009). Text Mining: Classification, Clustering, and Applications. *Boca Raton*. https://doi.org/10.1201/9781420059458

Sukmono, N. D. (2021). Clickbait Judul Berita Online dalam Pemberitaan Covid-19. *Transformatika: Jurnal Bahasa, Sastra, dan Pengajarannya*, *5*, 1–13. https://doi.org/10.31002/transformatika.v%vi%i.3643

Surianto, D. F., Kadir, R. A. P., Syafaat, F., Fakhri, M. M., & Rifqie, D. M. (2022). Implementasi Metode Latent Semantic Analysis Pada Peringkasan Artikel Bahasa Indonesia Menggunakan Pendekatan Steinberger Jezek. *JURIKOM (Jurnal Riset Komputer)*, *9*(4), 894. https://doi.org/10.30865/jurikom.v9i4.4620

Thange, S., Dange, J., Karjule, V., & Sase, J. (2023). A Survey on Text Summarization Techniques. *International Journal of Scientific and Research Publications*, *13*(11), 528–535. https://doi.org/10.29322/ijsrp.13.11.2023.p14355

Vanisha, R., Akanksha, S., Mahalakshmi, V. V., & Sirisha, R. (2022). *Text Summarization Using Deep Learning*. https://doi.org/10.48047/IJIEMR/V11/I06/37

Vikas, A., Pradyumna, G. V. N., & Shaik, T. A. (2020). Text Summarization. *International Journal Of Engineering And Computer Science*, *9*, 24940–24947. https://doi.org/10.18535/ijecs/v9i02.4437

Wazery, Y. M., Saleh, M. E., Alharbi, A., & Ali, A. A. (2022). Abstractive Arabic Text Summarization Based on Deep Learning. *Computational Intelligence and Neuroscience, 2022*. https://doi.org/10.1155/2022/1566890

Wijaya, J., & Girsang, A. S. (2024). Indonesian News Extractive Summarization using Lexrank and YAKE Algorithm. *Statistics, Optimization and Information Computing*, *12*(6), 1973–1983. https://doi.org/10.19139/soic-2310-5070-1976

Yulianti, E., Pangestu, N., & Jiwanggi, M. A. (2023). Enhanced TextRank using weighted word embedding for text summarization. *International Journal of Electrical and Computer Engineering*, *13*(5), 5472–5482. https://doi.org/10.11591/ijece.v13i5.pp5472-5482

LAMPIRAN-LAMPIRAN

```python
# Train Word2Vec Model (Skip-Gram)
word2vec_model = Word2Vec(
    sentences,
    vector_size=300,      # Dimensi vektor Word2Vec
    window=6,             # Jumlah window Word2Vec
    min_count=2,          # Jumlah minimal kata yang di proses
    workers=4,            # Jumlah thread
    sg=1,                 # Skip-gram model
    negative=15,          # Jumlah negative sampling
    alpha=0.03,           # Learning rate
    epochs=100            # Jumlah epoch
)
word_vectors = np.array([word2vec_model.wv[word] for word in word2vec_model.wv.index_to_key])
```

```python
# Fungsi LSA
lsa = TruncatedSVD(n_components=180, random_state=42)  # Menentukan jumlah dimensi hasil reduksi
lsa_vectors = lsa.fit_transform(word_vectors)
```

```python
# Fungsi Cosine Similarity
def cosine_similarity_manual(vector_a, vector_b):
    dot_product = np.dot(vector_a, vector_b)
    norm_a = np.linalg.norm(vector_a)
    norm_b = np.linalg.norm(vector_b)
    if norm_a == 0 or norm_b == 0:
        return 0.0
    return dot_product / (norm_a * norm_b)
```

```python
# Fungsi peringkasan utama
def generate_summary(row, word2vec_model, lsa, d=0.85, max_iter=500, debug=False):
```

```
# Iterasi perhitungan TextRank
for _ in range(max_iter):
    new_scores = np.ones(len(sentences))
    for i in range(len(sentences)):
        score_sum = 0
        for j in range(len(sentences)):
            if i != j:
                sum_sim_j = np.sum(sim_matrix[j])
                if sum_sim_j > 0:
                    score_sum += sim_matrix[i][j] * scores[j] / sum_sim_j
        new_scores[i] = (1 - d) + d * score_sum
    if np.allclose(new_scores, scores, atol=1e-6):
        break
    scores = new_scores

# Perankingan kalimat berdasarkan nilai TextRank
ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)

# Pilih kalimat yang akan digunakan sebagai ringkasan
summary_length = max(1, int(len(sentences) * 0.10)) # jumlah ringkasan 20% dari teks asli
summary = ' '.join([ranked_sentences[i][1] for i in range(min(len(ranked_sentences), summary_length))])
```