SCIENCE AND INFORMATION TECHNOLOGY
SINTECH
JOURNAL

SINTECH JOURNAL | ISSN 2598-7305 | E-ISSN 2598-9642
Vol. 8 No 1 – April 2025 | https://s.id/sintechjournal
DOI: https://doi.org/10.31598
Publishing: Prahasta Publisher

Peringkasan Teks Putusan Pengadilan Berbahasa Indonesia Menggunakan Algoritma TextRank

Miftah Adha[1], Mohammad Nasucha[2]

[1]Department of Informatics, Universitas Pembangunan Jaya
[2]Department of Informatics and Center for Urban Studies, Universitas Pembangunan Jaya
Jl. Cendrawasih Raya Blok B7/P, Sawah Baru, Kec. Ciputat, Kota Tangerang Selatan, Indonesia

e-mail: miftah.adha@student.upj.ac.id[1], mohammad.nasucha@upj.ac.id[2*]

Received : December, 2024	Accepted : March, 2025	Published : April, 2025

Abstract

Court decisions are important legal documents that are often long and complex, which may cause difficulties for readers in understanding the meaning and may imply a lenghty reading time. This research raises the issue of how to develop a web-based text auto summarization application for court decision documents in Indonesian languange. This research uses mixed methods. As for the software development life cycle, we use the Agile methodology. Within the development we use a natural language processing approach that includes tokenization, text cleaning, stemming, term frequency-inverse document frequency (TF-IDF) calculation, and application of cosine similarity to measure the similarity between sentences before applying the TextRank algorithm. We use the Python programming language and the Flask web framework, utilizing libraries such as PyPDF2 for PDF processing, Sastrawi for the stemming process, and NetworkX for the implementation of the TextRank algorithm. The dataset used consists of 50 court decision documents. Evaluation of the application is carried out using precision, recall, and F-measure metrics, comparing the application's summarization results against the reference summary made by an expert. The test shows the highest precision value of 0.62 at a compression rate of 75%, demonstrating the application's ability to produce informative summaries. This research is expected to contribute to the development of text auto summarization applications on court decision documents, as well as to open up opportunities for further research.

Keywords: automatic text summarization, cosine similarity, court decisions, natural language processing, textrank, tf-idf.

Abstrak

Putusan pengadilan merupakan dokumen hukum penting yang sering kali panjang dan kompleks, sehingga dapat mengakibatkan kesulitan dalam pemahaman dokumen dan waktu baca yang lama. Penelitian ini mengangkat masalah bagaimana mengembangkan aplikasi peringkasan teks otomatis berbasis web untuk dokumen putusan pengadilan berbahasa Indonesia**.** Penelitian ini menerapkan metode campuran. Metodologi pengembangan yang digunakan adalah *Agile.* Penelitian ini menerapkan pendekatan pemrosesan bahasa alami yang mencakup tokenisasi, pembersihan teks, *stemming*, perhitungan *term frequency-inverse document frequency* (TF-IDF), dan penerapan c*osine similarity* untuk mengukur kesamaan antar kalimat sebelum menerapkan algoritma *TextRank*. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dan *framework* web Flask, dengan memanfaatkan pustaka seperti PyPDF2 untuk pemrosesan PDF, Sastrawi untuk proses *stemming*, dan NetworkX untuk implementasi algoritma *TextRank*. *Dataset* yang digunakan terdiri dari 50 dokumen putusan pengadilan. Evaluasi aplikasi dilakukan dengan menggunakan metrik *precision*, *recall*, dan *F-measure*, membandingkan hasil peringkasan aplikasi terhadap ringkasan pembanding yang dibuat oleh pakar ilmu

Hukum. Hasil pengujian menunjukkan nilai *precision* tertinggi sebesar 0.62 pada *compression rate* 75%, mendemonstrasikan kemampuan aplikasi dalam menghasilkan ringkasan yang informatif. Penelitian ini diharapkan memberikan kontribusi kepada pengembangan aplikasi peringkasan teks otomatis pada dokumen putusan pengadilan, serta membuka peluang penelitian selanjutnya.

Kata Kunci: ***cosine similarity, nlp (pemrosesan bahasa alami), peringkasan teks otomatis, putusan pengadilan, textrank, tf-idf.***

1. PENDAHULUAN

Putusan pengadilan merupakan keputusan resmi yang dikeluarkan oleh pengadilan dalam perkara gugatan berdasarkan sengketa atau perselisihan. Putusan hakim sebagai pejabat negara dengan wewenang memutuskan perkara sesuai hukum yang berlaku dapat menghasilkan tiga kemungkinan hasil: hukuman pidana, keputusan bebas, dan keputusan pengeluaran. Selain itu, jenis putusan ini juga dapat diklasifikasikan sebagai putusan akhir yang bersifat material dan putusan yang tidak termasuk dalam kategori putusan akhir [1].

Putusan pengadilan berperan sebagai dokumen hukum penting yang memuat elemen-elemen seperti identitas pihak-pihak, pertimbangan hukum, bukti, dan amar putusan. Meskipun penting dalam sistem hukum, putusan pengadilan di Indonesia sering kali terlalu panjang, menggunakan istilah hukum yang kompleks, serta sulit dipahami oleh masyarakat umum maupun praktisi hukum [2][3]. Selain itu, elemen-elemen yang berulang, seperti objek gugatan dan alat bukti, sering menyebabkan dokumen semakin panjang tanpa memberikan informasi baru [3]. Hal ini mengakibatkan proses memahami isi putusan menjadi lambat dan kurang efisien, terutama bagi praktisi hukum yang harus menangani banyak kasus dalam waktu singkat.

Untuk mengatasi masalah tersebut, peringkasan teks otomatis dapat menjadi solusi yang efektif. Peringkasan teks adalah teknik pemrosesan bahasa alami yang memadatkan dokumen panjang menjadi versi ringkas tanpa kehilangan informasi utama [4]. Peringkasan teks memiliki beragam aplikasi, seperti cuplikan mesin pencari, pembuatan judul berita, dan peringkasan teks biomedis. Terdapat dua pendekatan utama dalam peringkasan teks, yaitu metode ekstraktif dan metode abstraktif [5]. Metode ekstraktif memilih kalimat-kalimat penting dari teks asli, sementara metode abstraktif menghasilkan kalimat-kalimat baru.Penelitian sebelumnya telah menerapkan model berbasis *Bidirectional Encoder Representations* from *Transformers* (BERT) dan metode *Latent Semantic Analysis* (LSA) pada teks putusan pengadilan atau dokumen hukum berbahasa Indonesia. Model seperti IndoBERT-Lite-Base menunjukkan performa terbaik dengan nilai ROUGE-1 hingga 1.00 pada dokumen tertentu [6], sementara LSA mencapai performa *F-measure* 61% pada compression rate 25% [7]. Namun, model-model ini memiliki kelemahan signifikan, yaitu ketidakmampuan menangani elemen tidak relevan seperti *footer, header,* dan *watermark* secara otomatis. Penghilangan elemen ini masih membutuhkan proses manual, yang mengurangi efisiensi dan memperpanjang waktu pemrosesan.

Penelitian ini bertujuan untuk mengatasi keterbatasan pendekatan berbasis BERT dalam menghilangkan elemen tidak relevan melalui pengembangan model peringkasan teks otomatis berbasis algoritma TextRank. Dengan mengintegrasikan *rule-based regular expression* pada tahap *preprocessing*, penelitian ini memungkinkan penghilangan elemen seperti *footer*, *header*, dan *watermark* secara otomatis, yang sebelumnya memerlukan intervensi manual dalam pendekatan BERT. Setelah preprocessing, algoritma *TextRank* digunakan untuk memilih kalimat-kalimat penting dari teks asli berdasarkan hubungan semantik antar kalimat dalam graf.

Selain itu, penelitian ini juga membandingkan hasil peringkasan *TextRank* dengan pendekatan *Latent Semantic Analysis* (LSA), karena keduanya berbasis metode non-*deep learning*. Analisis ini bertujuan untuk mengevaluasi efisiensi dan kualitas ringkasan dari kedua algoritma, terutama dalam konteks dokumen hukum yang panjang dan kompleks.

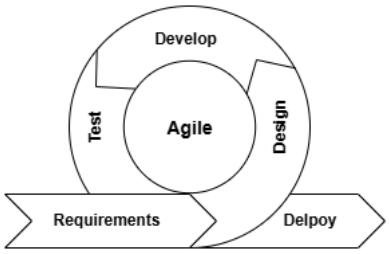
Dengan mengotomatisasi peringkasan teks putusan pengadilan, penelitian ini diharapkan dapat mempercepat proses peringkasan dan meningkatkan efisiensi, sehingga membantu mempercepat pemahaman terhadap inti dari putusan pengadilan yang panjang dan kompleks tanpa mengurangi esensi dari isi putusan. Peringkasan teks otomatis menjadi sangat krusial di era digital saat ini, di mana jumlah data tekstual yang tersedia meningkat secara pesat sehingga menyulitkan pemrosesan secara manual. Oleh karena itu, penelitian ini berkontribusi dalam menghadirkan solusi peringkasan teks otomatis yang lebih adaptif dan efisien untuk dokumen hukum di Indonesia.

2. METODE PENELITIAN

Penelitian ini menerapkan metode campuran (*mixed methods*) dengan pendekatan kualitatif dan kuantitatif untuk memastikan analisis yang komprehensif dalam peringkasan teks putusan pengadilan. Metode kualitatif digunakan pada tahap pengumpulan data dan pengolahan teks, khususnya dalam proses *text preprocessing*. Tahapan ini meliputi konversi dokumen dari format PDF ke teks (*parsing PDF*), tokenisasi, penghapusan *stopwords*, *stemming*, penghapusan elemen yang tidak penting, serta deteksi dan normalisasi singkatan agar tidak dianggap sebagai akhir kalimat. Proses-proses ini dilakukan melalui pengkodean dengan teknik pemrograman yang membutuhkan analisis mendalam untuk memastikan bahwa teks yang dihasilkan relevan dan sesuai dengan tujuan penelitian.

Di sisi lain, metode kuantitatif diterapkan pada komputasi numerik, mencakup penghitungan bobot dengan *Term Frequency-Inverse Document Frequency* (TF-IDF), pengukuran kesamaan teks menggunakan *Cosine Similarity* dan *Latent Semantic Analysis (LSA),* penerapan algoritma *TextRank* untuk memilih kalimat-kalimat penting, serta pengujian algoritma melalui evaluasi metrik. Dengan pendekatan ini, metode kualitatif memastikan relevansi dan kualitas data yang diproses, sementara metode kuantitatif memberikan kerangka matematis untuk analisis dan evaluasi yang sistematis. Kombinasi kedua metode ini memastikan hasil peringkasan yang tidak hanya akurat dan efisien tetapi juga tetap relevan dalam konteks hukum.

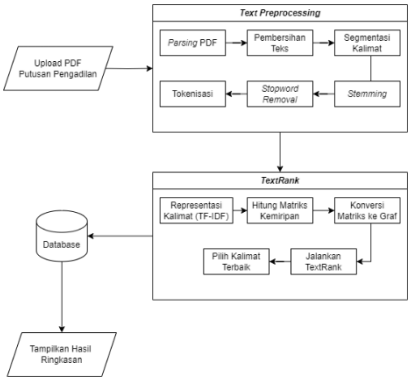
Pada penelitian ini, metodologi pengembangan yang digunakan adalah *Agile*. *Agile software development* adalah metode yang mendukung perencanaan adaptif, perkembangan evolusi, pengiriman awal, dan perbaikan terus-menerus [8]. Metode ini diterapkan dengan pendekatan iteratif dan inkremental, di mana pengembangan dilakukan dalam siklus-siklus pendek. Setiap siklus mencakup tahapan seperti *requirements* (kebutuhan), *design* (perancangan), *develop* (pengembangan), *test* (pengujian), dan *deploy* (penerapan) [9], seperti yang ditunjukkan dalam Gambar 1. Pendekatan ini memungkinkan fleksibilitas dan adaptasi cepat terhadap perubahan kebutuhan atau *feedback* selama proses pengembangan.



Gambar 1. Metode Pengembangan *Agile*

2.1. Arsiktektur Aplikasi

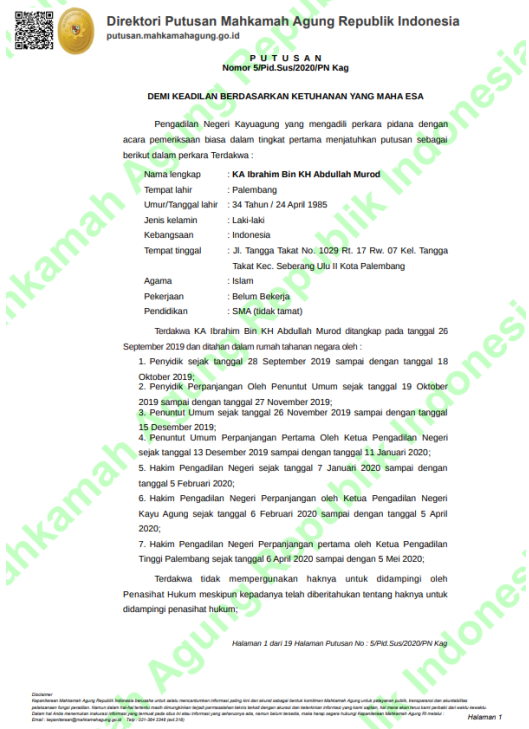
Penelitian ini menerapkan aplikasi peringkasan otomatis berbasis web untuk dokumen putusan pengadilan. Prosesnya dimulai dengan pengguna mengunggah dokumen PDF putusan pengadilan melalui antarmuka web. Selanjutnya, aplikasi melakukan tahap *preprocessing* pada dokumen, yang mencakup *parsing* PDF, pembersihan teks, segmentasi kalimat, tokenisasi, *stemming* dan menghilangkan *Stopwords*. Hasil *preprocessing* kemudian diproses menggunakan metode peringkasan, yang meliputi representasi kalimat (TF-IDF), perhitungan matriks kemiripan (*Cosine Similarity*), konversi matriks ke graf, dan penerapan algoritma *TextRank*. Tahap akhir adalah pemilihan kalimat terbaik berdasarkan peringkat tertinggi yang dihasilkan algoritma *TextRank*. Hasil peringkasan disimpan dalam database dan kemudian ditampilkan kepada pengguna melalui antarmuka web. Arsitektur keseluruhan aplikasi penelitian ini, termasuk integrasi database untuk penyimpanan dan pengambilan hasil, ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Aplikasi

2.2. *Dataset*

Kumpulan data (*Dataset)* yang dipakai dalam penelitian ini diperoleh dari repositori GitHub, yang berisi kumpulan dokumen putusan pengadilan. *Dataset* ini merupakan komponen dari proyek penelitian yang dikerjakan oleh Sheila Fitria dengan judul "Automatic Summarization of Court Decision Documents over Narcotic Cases Using BERT." Dalam penelitian ini, *dataset* tersebut dimanfaatkan untuk pengujian aplikasi peringkasan teks otomatis yang dikembangkan. *Dataset* ini terdiri dari 50 dokumen dalam format PDF yang diproses aplikasi yang dibangun untuk penelitian ini. Selain itu, juga terdapat *dataset* ringkasan referensi yang dibuat oleh pakar untuk membandingkan hasil peringkasan. *Dataset* dapat diakses melalui repositori GitHub berikut [10]. Meskipun dataset ini mencakup 50 dokumen putusan pengadilan dari repositori GitHub, ukuran dataset ini relatif kecil untuk menggambarkan berbagai macam putusan pengadilan yang ada di Indonesia. Selain itu, dataset ini hanya mencakup dokumen dalam format PDF, yang mungkin tidak merepresentasikan semua variasi format dokumen hukum. Potensi bias dapat terjadi karena dokumen berasal dari satu sumber, sehingga gaya penulisan dan struktur dokumen cenderung seragam.



Direktori Putusan Mahkamah Agung Republik Indonesia
putusan.mahkamahagung.go.id

P U T U S A N
Nomor 5/Pid.Sus/2020/PN Kag

DEMI KEADILAN BERDASARKAN KETUHANAN YANG MAHA ESA

Pengadilan Negeri Kayuagung yang mengadili perkara pidana dengan acara pemeriksaan biasa dalam tingkat pertama menjatuhkan putusan sebagai berikut dalam perkara Terdakwa :

Nama lengkap	: **KA Ibrahim Bin KH Abdullah Murod**
Tempat lahir	: Palembang
Umur/Tanggal lahir	: 34 Tahun / 24 April 1985
Jenis kelamin	: Laki-laki
Kebangsaan	: Indonesia
Tempat tinggal	: Jl. Tangga Takat No. 1029 Rt. 17 Rw. 07 Kel. Tangga Takat Kec. Seberang Ulu II Kota Palembang
Agama	: Islam
Pekerjaan	: Belum Bekerja
Pendidikan	: SMA (tidak tamat)

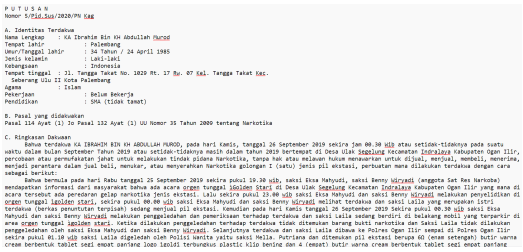
Terdakwa KA Ibrahim Bin KH Abdullah Murod ditangkap pada tanggal 26 September 2019 dan ditahan dalam rumah tahanan negara oleh :

1. Penyidik sejak tanggal 28 September 2019 sampai dengan tanggal 18 Oktober 2019;
2. Penyidik Perpanjangan Oleh Penuntut Umum sejak tanggal 19 Oktober 2019 sampai dengan tanggal 27 November 2019;
3. Penuntut Umum sejak tanggal 26 November 2019 sampai dengan tanggal 15 Desember 2019;
4. Penuntut Umum Perpanjangan Pertama Oleh Ketua Pengadilan Negeri sejak tanggal 13 Desember 2019 sampai dengan tanggal 11 Januari 2020;
5. Hakim Pengadilan Negeri sejak tanggal 7 Januari 2020 sampai dengan tanggal 5 Februari 2020;
6. Hakim Pengadilan Negeri Perpanjangan oleh Ketua Pengadilan Negeri Kayu Agung sejak tanggal 6 Februari 2020 sampai dengan tanggal 5 April 2020;
7. Hakim Pengadilan Negeri Perpanjangan pertama oleh Ketua Pengadilan Tinggi Palembang sejak tanggal 6 April 2020 sampai dengan 5 Mei 2020;

Terdakwa tidak mempergunakan haknya untuk didampingi oleh Penasihat Hukum meskipun kepadanya telah diberitahukan tentang haknya untuk didampingi penasihat hukum;

Disclaimer
Kepaniteraan Mahkamah Agung Republik Indonesia berusaha untuk selalu mencantumkan informasi paling kini dan akurat sebagai bentuk komitmen Mahkamah Agung untuk pelayanan publik, transparansi dan akuntabilitas pelaksanaan fungsi peradilan. Namun dalam hal hal tertentu masih ditunggunkan kejadi permasalahan teknis terkait dengan akurasi dan ketertinian informasi yang kami sajikan, hal mana akan terus kami perbaiki dari waktu kewaktu.
Dalam hal Anda menemukan inakurasi informasi yang termuat pada situs ini atau informasi yang seharusnya ada, namun belum tersedia, maka harap segera hubungi Kepaniteraan Mahkamah Agung RI melalui :
Email : kepaniteraan@mahkamahagung.go.id Telp : 021-384 3348 (ext.318)

Gambar 1. Contoh *Dataset* Dokumen Putusan Pengadilan



P U T U S A N
Nomor 5/Pid.Sus/2020/PN Kag

A. Identitas Terdakwa
Nama Lengkap : KA Ibrahim Bin KH Abdullah Murod
Tempat lahir : Palembang
Umur/Tanggal lahir : 34 Tahun / 24 April 1985
Jenis kelamin : Laki-laki
Kebangsaan : Indonesia
Tempat tinggal : Jl. Tangga Takat No. 1029 Rt. 17 Rw. 07 Kel. Tangga Takat Kec. Seberang Ulu II Kota Palembang
Agama : Islam
Pekerjaan : Belum Bekerja
Pendidikan : SMA (tidak tamat)

B. Pasal yang didakwakan
Pasal 114 Ayat (1) Jo Pasal 132 Ayat (1) UU Nomor 35 Tahun 2009 tentang Narkotika

C. Ringkasan Dakwaan
Bahwa terdakwa KA IBRAHIM BIN KH ABDULLAH MUROD, pada hari kamis, tanggal 26 September 2019 sekira jam 00.30 wib atau setidak-tidaknya pada suatu waktu dalam bulan September Tahun 2019 atau setidak-tidaknya masih dalam tahun 2019 bertempat di Desa Ulak Segelung kecamatan Indralaya Kabupaten Ogan Ilir, percobaan atau permufakatan jahat untuk melakukan tindak pidana Narkotika, tanpa hak atau melawan hukum menawarkan untuk dijual, menjual, membeli, menerima, menjadi perantara dalam jual beli, menukar, atau menyerahkan Narkotika golongan I (satu) jenis pil ekstasi, perbuatan mana dilakukan terdakwa dengan cara sebagai berikut:
Bahwa bermula pada hari Rabu tanggal 25 September 2019 sekira pukul 19.30 wib, saksi Eksa Mahyudi, saksi Benny Wirayati (anggota Sat Res Narkoba) mendapatkan informasi dari masyarakat bahwa ada acara organ tunggal Igolden Starz di Desa Ulak Segelung kecamatan Indralaya Kabupaten Ogan Ilir yang mana di acara tersebut ada peredaran gelap narkotika jenis ekstasi. Lalu sekira pukul 23.00 wib saksi Eksa Mahyudi dan saksi Benny Wirayati melakukan penyelidikan di organ tunggal Igolden starz, sekira pukul 00.00 wib saksi Eksa Mahyudi dan saksi Benny Wirayati melihat terdakwa dan saksi Laila yang merupakan istri terdakwa (berkas penuntutan terpisah) sedang menjual pil ekstasi. kemudian pada hari kamis tanggal 26 September 2019 Sekira pukul 00.30 wib saksi Eksa Mahyudi dan saksi Benny Wirayati melakukan penggeledahan dan pemeriksaan terhadap terdakwa dan saksi Laila sedang berdiri di belakang mobil yang terparkir di area organ tunggal Igolden starz. Ketika dilakukan penggeledahan terhadap terdakwa tidak ditemukan barang bukti narkotika dan Saksi Laila tidak dilakukan penggeledahan oleh saksi Eksa Mahyudi dan saksi Benny Wirayati. Selanjutnya terdakwa dan saksi Laila dibawa ke Polres Ogan Ilir sekira pukul 01.10 wib saksi Laila digeledah oleh Polisi wanita yaitu saksi Hella. Putriana dan ditemukan pil ekstasi berupa 60 (enam setengah) butir warna cream berbentuk tablet sedi empat panjang logo lebih terbungkus plastic klip bening dan 4 (empat) butir warna cream berbentuk tablet sedi empat panjang

Gambar 2. Contoh *Dataset* Referensi Ringkasan oleh Pakar.

2.3. *Text Preprocessing*

Sebelum teks diproses dan dianalisis, diperlukan tahap *preprocessing* untuk mempersiapkan data agar siap digunakan dalam aplikasi. Beberapa langkah *preprocessing* yang dilakukan dalam aplikasi ini meliputi:

a. *Parsing* Teks dari PDF

Dalam aplikasi ini, proses *parsing* teks dari PDF dilakukan menggunakan pustaka PyPDF2. PyPDF2 memungkinkan pengambilan teks dari halaman-halaman PDF dan mengubahnya menjadi teks mentah yang dapat diproses lebih lanjut.

b. Pembersihan Teks

Setelah file PDF berhasil diubah menjadi teks, terdapat beberapa elemen yang perlu dihilangkan, seperti *watermark* (misalnya, pengulangan teks "Mahkamah Agung" sebanyak

lima kali), *header*, dan *footer*. Proses ini menggunakan *rule-based regular expression* untuk mendeteksi dan menghapus elemen-elemen tersebut secara otomatis. Dengan cara ini, teks yang dianalisis menjadi bersih dan siap untuk proses peringkasan lebih lanjut.

c. Segmentasi Kalimat

Segmentasi kalimat adalah proses memisahkan teks menjadi kalimat-kalimat individual dengan mendeteksi batas kalimat, yaitu awal dan akhir setiap kalimat [11]. Proses ini dilakukan dengan mendeteksi tanda baca seperti tanda tanya (?), tanda seru (!), dan titik (.) Tantangan muncul karena tanda baca tersebut dapat digunakan dalam konteks lain, seperti singkatan. Untuk mengatasi hal ini, aplikasi yang dikembangkan menyertakan daftar singkatan guna mencegah salah identifikasi sebagai akhir kalimat. Singkatan seperti "Dr.", "Kec.", dan "Jln." tidak dianggap sebagai akhir kalimat, melainkan sebagai bagian dari kalimat yang sedang berjalan.

d. Tokenisasi

Tokenisasi kata adalah proses memecah kalimat menjadi kata-kata atau unit-unit yang lebih kecil, disebut token. Ini dilakukan dengan memisahkan kalimat berdasarkan spasi dan tanda baca.

e. *Stemming*

Stemming merupakan langkah untuk mengurangi kata-kata yang memiliki imbuhan menjadi bentuk dasar. Pada penelitian ini, proses *stemming* diimplementasikan dengan memanfaatkan Sastrawi, yaitu pustaka *stemmer* bahasa Indonesia yang dapat diakses secara publik. Pustaka ini didasarkan pada penelitian dari berbagai sumber dan menggunakan kamus dari kateglo.com dengan beberapa perubahan kecil [12].

f. *Stopwords Removal*

Proses penghilangan *stopwords* dilakukan untuk menghapus kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan kontribusi signifikan terhadap analisis atau pemahaman dokumen. Stopwords seperti 'dan', 'atau', 'yang', 'di', dan 'ke' cenderung tidak memiliki bobot informasi tinggi dalam konteks dokumen hukum. Dengan mengeliminasi kata-kata ini, proses analisis teks dapat lebih fokus pada kata-kata yang bermakna dan relevan, sehingga meningkatkan efisiensi.

2.4. TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan metode statistik yang dipakai untuk menilai tingkat kepentingan suatu kata dalam dokumen atau sekumpulan dokumen [13]. Metode ini mengintegrasikan dua konsep, yaitu frekuensi kata (*Term Frequency)* dan inversi dari frekuensi dokumen yang mencakup kata tersebut (*Inverse Document Frequency*), untuk menentukan bobot pentingnya kata tersebut dalam konteks keseluruhan. Metode ini diperkenalkan oleh Salton [14] sebagai suatu kombinasi yang mampu meningkatkan kinerja, terutama dalam meningkatkan nilai *recall* dan *precision* [15]. Berikut adalah perhitungannya:

a. *Term Frequency (TF)*

$$TF(t,s) = \frac{f(t,s)}{len(s)} \quad (1)$$

Di mana:

- $f(t,s)$ adalah frekuensi kemunculan kata t dalam kalimat s.
- $len(s)$ adalah total kata dalam kalimat s.

b. *Inverse Document Frequency (IDF)*

$$IDF(t) = \log(\frac{N}{DF+1}) \quad (2)$$

Di mana:

- N adalah total keseluruhan kalimat (atau dokumen).
- DF adalah total kalimat yang mengandung kata ($term$).

c. TF-IDF

$$TF - IDF(t,s) = TF(t,s) \times IDF(t) \quad (3)$$

Dalam aplikasi ini, TF-IDF digunakan untuk membangun representasi vektor dari setiap kalimat dalam dokumen. Setiap kalimat diubah menjadi vektor berdasarkan bobot kata yang dihitung dengan metode TF-IDF, yang kemudian digunakan dalam langkah selanjutnya untuk menghitung kesamaan antar kalimat menggunakan *cosine similarity*.

2.5. *Cosine Similarity*

Cosine Similarity adalah teknik yang digunakan untuk menilai tingkat kesamaan antara dua vektor dokumen dengan cara menghitung kosinus sudut antara kedua vektor tersebut. Teknik ini sering digunakan untuk mengukur

kemiripan antar dokumen berbasis teks. Jika dua vektor memiliki arah yang sama, maka dokumen tersebut dianggap mirip, dengan nilai kemiripan yang mendekati 1. Sebaliknya, semakin besar sudut di antara dua vektor, semakin kecil nilai kemiripannya, dengan nilai minimum 0 [16].

Dalam aplikasi ini, *Cosine Similarity* digunakan untuk menghitung kemiripan antar kalimat setelah bobot kata dalam setiap kalimat dihitung menggunakan TF-IDF. Dengan menghitung kesamaan antar kalimat, algoritma dapat menentukan kalimat-kalimat yang memiliki hubungan semantik yang kuat, yang akan digunakan dalam algoritma *TextRank* untuk menentukan kalimat yang paling penting. Persamaan (1) merupakan contoh penulisan persamaan untuk menghitung nilai kemiripan antara dua dokumen menggunakan *Cosine Similarity*. Pada persamaan (1), *Cosine Similarity* merupakan nilai kemiripan antara dua vektor dokumen, sedangkan A dan B adalah vektor dari dua dokumen yang ingin dibandingkan.

$$Cosine\ Similarity = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (4)$$

Pada persamaan ini:

- $A \cdot B$ adalah hasil perkalian dot (*inner product*) antara vektor dokumen A dan B,
- $||A||$ dan $||B||$ adalah magnitudo (panjang) dari masing-masing vektor dokumen.

2.6 *Latent Semantic Analysis*

*Latent Semantic Analysis (*LSA) adalah metode statistik aljabar tanpa pengawasan yang kuat untuk mengembangkan penggambaran implisit semantik teks berdasarkan kemunculan bersama kata. Berbagai aplikasi seperti kategorisasi dokumen, pencarian dan penyaringan informasi, dan yang paling penting, ringkasan teks otomatis menggunakan LSA. [17]

Proses LSA dimulai dengan membangun matriks *term-by-sentences* A , di mana setiap kolom mewakili vektor frekuensi term yang terberat dari kalimat dalam dokumen [18]. SVD dari matriks A didefinisikan sebagai:

$$A = U\Sigma V^T \quad (5)$$

Di mana :

- A adalah matriks TF-IDF / *term-by-sentences*
- U adalah matriks vektor singular kiri
- Σ adalah matriks diagonal nilai singular
- V^T adalah transpose matriks vektor singular kanan

Dalam konteks ringkasan teks, LSA memungkinkan pemilihan kalimat yang memiliki kontribusi tertinggi pada dimensi *latent* utama. Misalnya, untuk dokumen hukum, LSA dapat menangkap hubungan semantik antara kalimat-kalimat yang membahas topik serupa. Kelebihan LSA adalah kemampuannya menangkap hubungan sinonim tanpa memerlukan pelatihan model, meskipun hasilnya terbatas pada hubungan linear dan memerlukan parameter optimal seperti n_components. Kombinasi LSA dengan *TextRank* diharapkan dapat meningkatkan kualitas ringkasan dengan mengintegrasikan hubungan semantik dan struktur graf antar kalimat.

2.6. *TextRank*

TextRank berfungsi sebagai metode ekstraksi teks otomatis tanpa pengawasan yang tidak memerlukan pengetahuan linguistik atau pengetahuan domain tertentu sebelumnya [13]. *TextRank* merupakan algoritma peringkat berbasis graf yang diusulkan oleh Mihalcea dan Tarau untuk pemrosesan teks otomatis yang terinspirasi dari algoritma *PageRank* [19]. Algoritma ini mengadaptasi konsep PageRank ke dalam domain teks, dengan memanfaatkan struktur graf untuk mengidentifikasi informasi penting dalam dokumen. Dalam penelitian ini, implementasi algoritma *TextRank* dilakukan menggunakan pustaka NetworkX, yang memungkinkan pengelolaan graf secara efisien. Rumusnya adalah sebagai berikut:

$$S(V_i) = (1-d) + d \times \sum_{V_J \in In(V_i)} \frac{S(V_J)}{L(V_J)} \quad (5)$$

Di mana:

- $S(V_i)$ adalah skor simpul V_i (dalam hal ini, kalimat)
- d adalah *damping factor* (faktor peredam, biasanya d = 0.85, yang mengontrol probabilitas lompatan acak dalam graf)
- $In(V_i)$ adalah kumpulan simpul yang menunjuk ke V_i (yaitu kalimat lain yang terhubung ke V_i)
- $L(V_J)$ adalah jumlah simpul yang terhubung keluar dari V_J (jumlah kalimat yang memiliki hubungan dengan V_J).

Dalam rumus ini, skor simpul $S(V_i)$ dihitung dengan menggabungkan dua komponen: *damping factor* sebesar $1-d$, yang mewakili probabilitas berpindah secara acak ke

simpul lain, dan komponen berbobot sebesar d, yang memperhitungkan skor simpul-simpul yang terhubung ke V_i. Komponen berbobot ini dihitung sebagai jumlah dari skor setiap simpul yang menunjuk ke V_i $S(V_i))$, dibagi dengan jumlah total hubungan keluar dari simpul tersebut $L(V_j)$. Skor akhir setiap simpul dihitung secara iteratif, hingga skor stabil (konvergen). Nilai *damping factor* biasanya diset ke 0,85, yang berarti 85% perhitungan dipengaruhi oleh hubungan antar simpul, dan 15% sisanya diizinkan untuk berpindah ke simpul lain secara acak. Hasil dari perhitungan ini menentukan simpul mana yang paling penting dalam teks berdasarkan hubungan keseluruhan antar simpul.

2.7. Metrik Evaluasi

Metrik Evaluasi digunakan untuk menilai performa aplikasi peringkasan teks yang dikembangkan. Tiga metrik utama yang digunakan dalam penelitian ini adalah *Precision*, *Recall*, dan *F-measure*.

a. *Precision*

Precision didefinisikan sebagai rasio antara jumlah kalimat yang relevan yang berhasil ditemukan dengan total keseluruhan kalimat yang ditemukan oleh aplikasi. Rumus untuk menghitung *precision* dinyatakan dalam persamaan (6) sebagai berikut:

$$Precision = \frac{tp}{tp+fp} \quad (6)$$

Di mana:

- tp (*true positive*) adalah kalimat yang terdapat dalam ringkasan manual dan juga muncul dalam hasil ringkasan aplikasi.
- fp (*false positive*) adalah kalimat yang hanya muncul dalam ringkasan aplikasi tetapi tidak ada dalam ringkasan manual.

b. *Recall*

Recall merupakan perbandingan antara jumlah kalimat relevan yang berhasil ditemukan kembali dengan total keseleruhan kalimat yang seharusnya ditemukan. Rumus untuk menghitung *recall* dinyatakan dalam persamaan (7) sebagai berikut:

$$Recall = \frac{tp}{tp+fn} \quad (7)$$

Di mana:

fn (*false negative*) adalah kalimat yang terdapat dalam ringkasan manual tetapi tidak ada dalam hasil ringkasan aplikasi.

c. *F-measure*

F-measure menggabungkan nilai *precision* dan *recall* menjadi satu ukuran tunggal, yang dapat dinyatakan dalam persamaan (8):

$$F - measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (8)$$

Metrik-metrik ini sangat penting untuk mengevaluasi kualitas ringkasan otomatis dengan membandingkannya terhadap ringkasan manual yang dibuat oleh ahli. Nilai yang lebih tinggi pada *precision* dan *recall* mengindikasikan kinerja aplikasi yang lebih baik dalam menghasilkan ringkasan yang relevan dan informatif [20].

3. HASIL DAN PEMBAHASAN

3.1. *Text Preprocessing*

Proses *text preprocessing* merupakan tahap penting dalam penelitian ini untuk memastikan data yang digunakan bebas dari elemen-elemen tidak relevan yang dapat memengaruhi hasil peringkasan. Tahapan *preprocessing* mencakup konversi dokumen dari format PDF ke teks mentah, penghapusan *stopwords*, *stemming*, segmentasi kalimat, tokenisasi dan penghilangan elemen-elemen seperti header, footer, serta watermark. Salah satu inovasi yang diterapkan dalam penelitian ini adalah penggunaan *regular expression* (regex) untuk secara otomatis mendeteksi dan menghapus elemen-elemen tersebut tanpa intervensi manual. Pendekatan ini tidak hanya meningkatkan efisiensi tetapi juga mengurangi risiko inkonsistensi dalam pengolahan data.

a. *Parsing* Teks dari PDF

Proses parsing yang dilakukan bertujuan untuk mengekstrak teks dari dokumen PDF agar dapat diolah lebih lanjut. Pada tahap ini, teks mentah diambil dari file PDF menggunakan *library* Python yaitu PyPDF2. Hasil parsing berupa teks asli dari dokumen yang mencakup isi utama, seperti isi keputusan, tanda baca, baris kosong, serta informasi lainnya yang terdapat dalam dokumen. Teks hasil parsing ini masih dalam format mentah dan membutuhkan proses lanjutan, seperti pembersihan data untuk menghilangkan informasi tidak relevan, seperti disclaimer, atau elemen lain yang tidak diperlukan.

```
extracted_text = extract_text_from_pdf(pdf_path)
print("Hasil Parsing PDF:")
print(extracted_text)
```

```
7.Hakim Pengadilan Negeri Perpanjangan pertama oleh Ketua Pengadilan
Tinggi Palembang sejak tanggal 6 April 2020 sampai dengan 5 Mei 2020;
Terdakwa tidak mempergunakan haknya untuk didampingi oleh
Penasihat Hukum meskipun kepadanya telah diberitahukan tentang haknya untuk
didampingi penasihat hukum;
Halaman 1 dari 19 Halaman Putusan No : 5/Pid.Sus/2020/PN Kag
Disclaimer
Kepaniteraan Mahkamah Agung Republik Indonesia berusaha untuk selalu mencantumkan informasi paling kini dan al
pelaksanaan fungsi peradilan. Namun dalam hal-hal tertentu masih dimungkinkan terjadi permasalahan teknis ter
Dalam hal Anda menemukan inakurasi informasi yang termuat pada situs ini atau informasi yang seharusnya ada,
Email : kepaniteraan@mahkamahagung.go.id Telp : 021-384 3348 (ext.318) Halaman 1Mahkamah Agung Republik In
Mahkamah Agung Republik Indonesia
Mahkamah Agung Republik Indonesia
Mahkamah Agung Republik Indonesia
Mahkamah Agung Republik Indonesia
Direktori Putusan Mahkamah Agung Republik Indonesia
putusan.mahkamahagung.go.id
Pengadilan Negeri tersebut ;
Setelah membaca ;
-Penetapan Ketua Pengadilan Negeri Kayuagung Nomor :
5/Pid.Sus/2020/PN Kag tanggal 7 Januari 2020 tentang Penunjukan Majelis
Hakim;
```

Gambar 1. Hasil Proses *Parsing*

b. Pembersihan Teks

Setelah proses *parsing* dilakukan, langkah berikutnya adalah membersihkan teks hasil ekstraksi dari elemen-elemen yang tidak relevan atau tidak penting menggunakan *regular expression*. Tahapan ini bertujuan untuk menyederhanakan teks dengan menghilangkan informasi seperti *header*, *footer*, dan *watermark* yang tidak dibutuhkan dalam proses analisis lebih lanjut.

```
print(cleaned_text)
```

```
15 Desember 2019;
4.Penuntut Umum Perpanjangan Pertama Oleh Ketua P engadilan Negeri
sejak tanggal 13 Desember 2019 sampai dengan tanggal 11 Januari 2020 ;
5.Hakim Pengadilan Negeri sejak tanggal 7 Januari 2020 sampai dengan
tanggal 5 Februari 2020;
6.Hakim Pengadilan Negeri Perpanjangan oleh Ketua Pengadilan Negeri
Kayu Agung sejak tanggal 6 Februari 2020 sampai dengan tanggal 5 April
2020;
7.Hakim Pengadilan Negeri Perpanjangan pertama oleh Ketua Pengadilan
Tinggi Palembang sejak tanggal 6 April 2020 sampai dengan 5 Mei 2020;
Terdakwa tidak mempergunakan haknya untuk didampingi oleh
Penasihat Hukum meskipun kepadanya telah diberitahukan tentang haknya untuk
didampingi penasihat hukum;

Pengadilan Negeri tersebut ;
Setelah membaca ;
-Penetapan Ketua Pengadilan Negeri Kayuagung Nomor :
5/Pid.Sus/2020/PN Kag tanggal 7 Januari 2020 tentang Penunjukan Majelis
Hakim;
-Penetapan Majelis Hakim Nomor : 5/Pid.Sus/2020/PN Kag tanggal 7
Januari 2020 tentang Penetapan Hari Sidang;
-Berkas perkara dan surat-surat lain yang bersangkutan;
Setelah mendengar keterangan Saksi-saksi dan Terdakwa serta
memperhatikan barang bukti yang diajukan oleh Penuntut Umum di persidangan;
```

Gambar 2. Hasil Proses Pembersihan Teks dari Elemen Tidak Relevan

c. Normalisasi Singkatan

Setelah tahap penghapusan elemen tidak penting selesai, langkah berikutnya adalah menghilangkan titik pada singkatan yang memiliki titik di dalamnya. Hal ini dilakukan untuk mencegah sistem mengidentifikasi titik tersebut sebagai akhir dari sebuah kalimat. Singkatan seperti "Kec." atau "Kab." sering kali terdeteksi secara keliru sebagai pemisah kalimat, sehingga perlu dilakukan normalisasi untuk memastikan hasil yang lebih akurat dalam proses segmentasi kalimat.

```
# Membuat perbandingan
print("Teks sebelum penghapusan singkatan:")
print(cleaned_text)
print("\nTeks setelah penghapusan singkatan:")
print(processed_text)
```

```
Teks sebelum penghapusan singkatan:

-Bahwa pada hari Kamis tanggal 26 September 2019 sekira pukul 00.30 wib di sebuah orgen tunggal yang berada di Des
a Ulak Segelung Kec. Indralaya Kab. Ogan Ilir, saksi bersama anggota polisi lainnya menangkap terdakwa ka
rena menjual Narkotika berupa pil ekstasi.


Teks setelah penghapusan singkatan:
-Bahwa pada hari Kamis tanggal 26 September 2019 sekira pukul 00.30 wib di sebuah orgen tunggal yang berada di Des
a Ulak Segelung Kec Indralaya Kab Ogan Ilir, saksi bersama anggota polisi lainnya menangkap terdakwa karena menjua
l Narkotika berupa pil ekstasi.
```

Gambar 3. Hasil Proses Normalisasi Singkatan

d. Segmentasi Kalimat

Setelah proses normalisasi singkatan, langkah selanjutnya adalah segmentasi kalimat. Segmentasi kalimat bertujuan untuk memisahkan teks yang panjang menjadi potongan-potongan kalimat.

```
# Contoh penggunaan
sentence_segmentation = custom_tokenize(processed_text)
print("Hasil Segmentasi Kalimat:")
print(sentence_segmentation)
```

```
Hasil Segmentasi Kalimat:
['-Bahwa pada hari Kamis tanggal 26 September 2019 sekira pukul 00.30 wib di sebuah orgen tunggal yang berada di D
esa Ulak Segelung Kec Indralaya Kab Ogan Ilir, saksi bersama terdakwa ditangkap oleh polisi karena menjual Narkoti
ka berupa pil ekstasi.', '-Bahwa pada awalnya saksi bersama terdakwa sedang menjual pil ekstasi di sebuah acara or
gen tunggal di Desa Ulak Segelung namun pada saat kami sedang menjual pil ekstasi tersebut datanglah dua anggota P
olisi langsung menangkap kami dan melakukan penggeledahan terhadap terdakwa dan saksi yang sedang berdiri di belak
ang mobil yang terparkir di area orgen tunggal "golden star",', 'ketika dilakukan penggeledahan terhadap terdakwa
Polisi tidak menemukan pil ekstasi tersebut lalu kami dibawa ke Polres OI dan sesampainya disana lalu saksi digele
dah oleh seorang Polisi wanita dan ditemukan pil ekstasi berupa 6X (enam setengah) butir warna cream berbentuk tab
let segi empat panjang logo "gold" terbungkus plastic klip bening dan 4 (empat) butir warna cream berbentuk tablet
segi empat panjang logo "gold" terbungkus plastic warna hijau di dalam BH warna hitam sebelah kanan saksi .']
```

Gambar 4. Hasil Proses Segmentasi Kalimat

e. *Stemming*

Langkah berikutnya adalah melakukan stemming, yang bertujuan untuk mengubah setiap kata menjadi bentuk kata dasarnya. Proses ini memanfaatkan *library* Sastrawi sebagai referensi untuk menentukan kata yang akan diubah menjadi kata dasar.

```
# Contoh penggunaan
stemmed_sentences = stem_sentences(sentence_segmentation)
print("Hasil Stemming:")
print(stemmed_sentences)
```

```
Hasil Stemming:
['-bahwa pada hari kamis tanggal 26 september 2019 sekira pukul 00 30 wib di buah orgen tunggal yang ada di desa u
lak gelung kec indralaya kab ogan ilir saksi sama dakwa tangkap oleh polisi karena jual narkotika upa pil ekstas
i', '-bahwa pada awal saksi sama dakwa sedang jual pil ekstasi di buah acara orgen tunggal di desa ulak gelung nam
un pada saat kami sedang jual pil ekstasi sebut datang dua anggota polisi langsung tangkap kami dan laku geledah h
adap dakwa dan saksi yang sedang diri di belakang mobil yang parkir di area orgen tunggal golden star', 'ketika la
ku geledah hadap dakwa polisi tidak temu pil ekstasi sebut lalu kami bawa ke polres oi dan sampa sana lalu saksi g
eledah oleh orang polisi wanita dan temu pil ekstasi upa 6 enam tengah butir warna cream bentuk tablet segi empat
panjang logo gold bungkus plastic klip bening dan 4 empat butir warna cream bentuk tablet segi empat panjang logo
gold bungkus plastic warna hijau di dalam bh warna hitam belah kanan saksi']
```

Gambar 5. Hasil Proses *Stemming*

f. Stopwords Removal

Setelah proses *stemming* selesai, langkah berikutnya adalah melakukan penghapusan *stopwords*, yaitu dengan menghilangkan kata-kata yang dianggap tidak berkontribusi terhadap makna kalimat, seperti kata penghubung atau konjungsi.

```
result = preprocess_sentences(stemmed_sentences, stopwords)
print(result)
```

```
['-bahwa kamis tanggal 26 september 2019 sekira 00 30 wib buah orgen tunggal desa ulak gelung kec indralaya kab og
an ilir saksi dakwa tangkap polisi jual narkotika upa pil ekstasi', '-bahwa saksi dakwa jual pil ekstasi buah acar
a orgen tunggal desa ulak gelung jual pil ekstasi anggota polisi langsung tangkap laku geledah hadap dakwa saksi m
obil parkir area orgen tunggal golden star', 'laku geledah hadap dakwa polisi temu pil ekstasi bawa polres oi samp
a saksi geledah orang polisi wanita temu pil ekstasi upa 6 enam butir warna cream bentuk tablet segi logo gold bun
gkus plastic klip bening 4 butir warna cream bentuk tablet segi logo gold bungkus plastic warna hijau bh warna hit
am belah kanan saksi']
```

Gambar 6. Hasil Proses *Stopwords Removal*

g. Tokenisasi

Langkah berikutnya adalah melakukan tokenisasi, yang bertujuan memecah kalimat menjadi kata atau token, berdasarkan spasi.

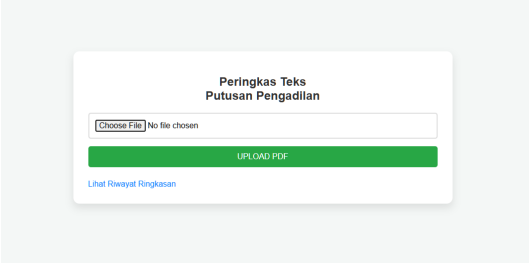
```
print(tokenized_word)
```

```
[['-bahwa', 'kamis', 'tanggal', '26', 'september', '2019', 'sekira', '00', '30', 'wib', 'buah', 'orgen', 'tungga
l', 'desa', 'ulak', 'gelung', 'kec', 'indralaya', 'kab', 'ogan', 'ilir', 'saksi', 'dakwa', 'tangkap', 'polisi', 'j
ual', 'narkotika', 'upa', 'pil', 'ekstasi'], ['-bahwa', 'saksi', 'dakwa', 'jual', 'pil', 'ekstasi', 'buah', 'acar
a', 'orgen', 'tunggal', 'desa', 'ulak', 'gelung', 'jual', 'pil', 'ekstasi', 'anggota', 'polisi', 'langsung', 'tang
kap', 'laku', 'geledah', 'hadap', 'dakwa', 'saksi', 'mobil', 'parkir', 'area', 'orgen', 'tunggal', 'golden', 'sta
r'], ['laku', 'geledah', 'hadap', 'dakwa', 'polisi', 'temu', 'pil', 'ekstasi', 'bawa', 'polres', 'oi', 'sampa', 's
aksi', 'geledah', 'orang', 'polisi', 'wanita', 'temu', 'pil', 'ekstasi', 'upa', '6', 'enam', 'butir', 'warna', 'cr
eam', 'bentuk', 'tablet', 'segi', 'logo', 'gold', 'bungkus', 'plastic', 'klip', 'bening', '4', 'butir', 'warna',
'cream', 'bentuk', 'tablet', 'segi', 'logo', 'gold', 'bungkus', 'plastic', 'warna', 'hijau', 'bh', 'warna', 'hita
m', 'belah', 'kanan', 'saksi']]
```

Gambar 7. Hasil Proses Tokenisasi

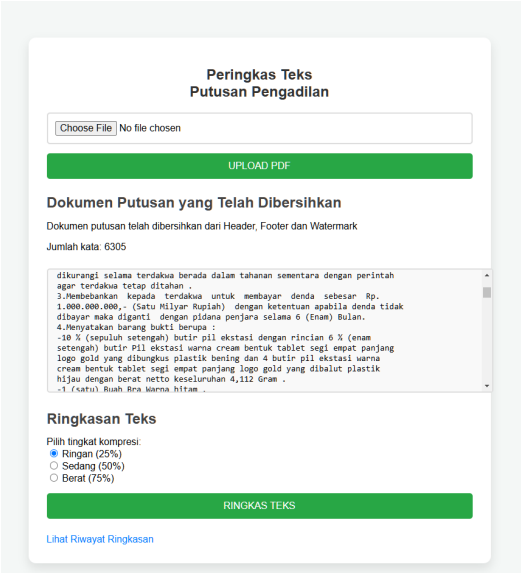
3.2. Hasil Implementasi Aplikasi

Aplikasi yang dikembangkan menggunakan Python dengan *framework* Flask dan memiliki antarmuka serta fitur sebagaimana yang terlihat pada antarmuka Gambar 3 di bawah ini:



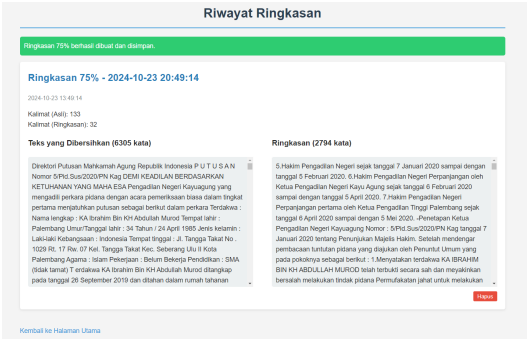
Gambar 3. Unggah Dokumen

Setelah pengguna mengunggah dokumen dan menekan tombol "UPLOAD PDF", aplikasi akan melakukan parsing dan pembersihan teks otomatis, yang mencakup ekstraksi teks dan penghilangan elemen tidak relevan seperti*watermark*, *header*, dan *footer*.



Gambar 4. Tampilan Hasil Pembersihan Teks

Gambar 4 menunjukkan antarmuka aplikasi setelah dokumen diunggah, di mana pengguna dapat melihat hasil *parsing* dan pembersihan teks. Pengguna juga dapat mengatur tingkat kompresi dokumen melalui radio button. Setelah memilih tingkat kompresi dan menekan tombol "RINGKAS TEKS", hasil ringkasan akan muncul secara otomatis. Basis data SQLite dengan pustaka SQLAlchemy digunakan untuk menyimpan informasi dan hasil peringkasan.



Gambar 5. Tampilan Hasil Ringkasan

Gambar 5 menunjukkan antarmuka aplikasi setelah dokumen diringkas, menampilkan hasil parsing, pembersihan teks, hasil ringkasan, jumlah kata sebelum dan sesudah, serta waktu pemrosesan. Hasil ringkasan ditampilkan dengan jelas untuk referensi lebih lanjut.

Untuk menunjukkan cara kerja aplikasi peringkasan, disajikan cuplikan teks asli dari dokumen putusan pengadilan beserta hasil ringkasannya. Teks asli diambil dari dokumen lengkap, sedangkan hasil ringkasan dihasilkan oleh aplikasi menggunakan algoritma TextRank.

Cuplikan Teks

Kemudian pada hari Kamis tanggal 26 September 2019 Sekira pukul 00.30 wib saksi Eksa Mahyudi dan saksi Benny Wiryadi melakukan penggeledahan dan pemeriksaan terhadap terdakwa dan saksi Laila sedang berdiri di belakang mobil yang terparkir di area orgen tunggal "golden star". Ketika dilakukan penggeledahan terhadap terdakwa tidak ditemukan barang bukti narkotika dan Saksi Laila tidak dilakukan penggeledahan oleh saksi Eksa Mahyudi dan saksi Benny Wiryadi. Selanjutnya terdakwa dan saksi Laila dibawa ke Polres Ogan Ilir sempai di Polres Ogan Ilir sekira pukul 01.10 wib saksi Laila digeledah oleh Polisi Wanita yaitu saksi Mella Putriana dan ditemukan pil ekstasi berupa 6½ (enam setengah) butir warna cream berbentuk tablet segi empat panjang logo "gold" terbungkus plastic klip bening dan 4 (empat) butir warna cream berbentuk tablet segi empat panjang logo "gold" terbungkus plastic warna hijau di dalam BH warna hitam sebelah kanan dan barang bukti yang ditemukan yaitu 1 (satu) unit handphone warna merah merk oppo beserta simcard 3 0895604331314 milik saksi Laila yang dipegang tangan kanan saksi Laila, uang tunai Rp2.927.000 (dua juta Sembilan ratus dua puluh tujuh ribu rupiah) hasil dari terdakwa dan saksi Laila menjual pil ekstasi di dalam kantong kanan celana terdakwa. Bahwa berdasarkan berita acara pemeriksaan laboratorium forensic nomo 2710/NNF/2019 tanggal 1 Oktober 2019 yang ditandatangani Kepala Bidang Laboratorium forensic Polda Sumsel Haris Aksara, SH pemeriksa 1 I Made Swetra, S.Si.,M.Si, pemeriksa 2 Edhi Suryanto, S.Si.,Apt., MM., MT, pemeriksa 3 Aliyus Saputra, S. Kom dengan kesimpulan bahwa BB 1, BB 2, BB 3, BB 4 adalah positif mengandung MDMA yang terdaftar sebagai golongan I (satu) nomor urut 37 lampiran peraturan Menteri Kesehatan RI No 50 tahun 2018. Bahwa perbuatan terdakwa dalam menawarkan untuk dijual, menjual, membeli, menerima, menjadi perantara dalam jual beli, menukar, atau menyerahkan Narkotika golongan I (satu) bukan tanaman tersebut, adalah tanpa seizin dari pihak yang berwenang. Perbuatan terdakwa sebagaimana diatur dan diancam Pidana dalam Pasal 112 ayat (1) Jo Pasal 132 ayat (1) UU RI No. 35 Tahun 2009 Tentang Narkotika.

Hasil Ringkasan

Kemudian pada hari Kamis tanggal 26 September 2019 Sekira pukul 00.30 wib saksi Eksa Mahyudi dan saksi Benny Wiryadi melakukan penggeledahan dan pemeriksaan terhadap terdakwa dan saksi Laila sedang berdiri di belakang mobil yang terparkir di area orgen tunggal "golden star". Selanjutnya terdakwa dan saksi Laila dibawa ke Polres Ogan Ilir sempai di Polres Ogan Ilir sekira pukul 01.10 wib saksi Laila digeledah oleh Polisi Wanita yaitu saksi Mella Putriana dan ditemukan pil ekstasi berupa 6½ (enam setengah) butir warna cream berbentuk tablet segi empat panjang logo "gold" terbungkus plastic klip bening dan 4 (empat) butir warna cream berbentuk tablet segi empat panjang logo "gold" terbungkus plastic warna hijau di dalam BH warna hitam sebelah kanan dan barang bukti yang ditemukan yaitu 1 (satu) unit handphone warna merah merk oppo beserta simcard 3 0895604331314 milik saksi Laila yang dipegang tangan kanan saksi Laila, uang tunai Rp2.927.000 (dua juta Sembilan ratus dua puluh tujuh ribu rupiah) hasil dari terdakwa dan saksi Laila menjual pil ekstasi di dalam kantong kanan celana terdakwa. Perbuatan terdakwa sebagaimana diatur dan diancam Pidana dalam Pasal 114 ayat (1) Jo Pasal 132 ayat (1) UU RI No 35 Tahun 2009 Tentang Narkotika

Gambar 6. Contoh Peringkasan *Textrank* pada Cuplikan Teks

3.3. Deskripsi Data

Dalam penelitian ini, data yang digunakan untuk pengujian merupakan hasil dari proses peringkasan otomatis menggunakan aplikasi berbasis algoritma *TextRank*. Dataset terdiri dari 50 dokumen putusan pengadilan dalam format PDF. Setiap dokumen telah melalui *text preprocessing*, seperti parsing teks, pembersihan elemen tidak relevan, segmentasi kalimat, *stemming, stopwords removal* dan tokenisasi.

Pengujian dalam penelitian ini dilakukan dengan membandingkan panjang kalimat sebelum dan sesudah proses peringkasan, serta dengan ringkasan yang dibuat oleh pakar. Selain itu, evaluasi kualitas ringkasan juga diukur menggunakan metrik seperti *precision*, *recall*, dan *F-measure*. Pertama-tama, analisis difokuskan pada perbandingan panjang kalimat antara teks asli dan hasil ringkasan untuk memahami dampak algoritma *TextRank* terhadap struktur kalimat.

Tabel 1: Perbandingan Jumlah Kalimat Sebelum dan Sesudah Peringkasan dengan *TextRank* dan Ringkasan Pakar

Nomor Dok.	Sebelum	Ringkasan *TextRank*			Ringkasan Pakar
		75%	50%	25%	
Dok. 1	131	32	65	98	60
Dok. 11	448	111	223	335	137
Dok. 21	203	50	101	152	77
Dok. 31	254	63	126	189	92
Dok. 41	226	56	113	169	80
Dok. 50	107	25	52	80	41

Berdasarkan Tabel 1, perbandingan jumlah kalimat hasil peringkasan menggunakan algoritma TextRank dengan tiga *compression rate* (75%, 50%, dan 25%) menunjukkan *bahwa compression rate* 50% menghasilkan jumlah kalimat yang paling mendekati ringkasan pakar. Sebagai contoh, pada Dokumen 1 yang memiliki 131 kalimat awal, hasil peringkasan dengan *compression rate* 75% menghasilkan 32 kalimat, 50% menghasilkan 65 kalimat, dan 25% menghasilkan 98 kalimat, sedangkan pakar menghasilkan 60 kalimat. Pola serupa ditemukan pada Dokumen 11, di mana dari 448 kalimat, *compression rate* 50% menghasilkan 223 kalimat, lebih mendekati 137 kalimat dari ringkasan pakar dibandingkan dengan 111 kalimat pada *compression rate* 75% dan 335 kalimat pada 25%.

Setelah menganalisis hasil peringkasan berdasarkan jumlah kalimat, evaluasi kualitas ringkasan oleh algoritma *TextRank* dilakukan menggunakan metrik *precision*, *recall*, dan *F-measure*. Metrik ini memberikan gambaran tentang efektivitas ringkasan dalam menangkap informasi relevan dibandingkan dengan ringkasan pakar.

Pada tahap pengujian ini, kami akan menekankan kembali metrik evaluasi yang digunakan untuk menilai performa aplikasi peringkasan teks, sebagaimana telah dijelaskan di bagian 2.7. *Precision* mengukur akurasi aplikasi dalam memilih kalimat penting, yaitu rasio antara kalimat relevan yang ditemukan dan total kalimat yang dipilih. *Recall* mengukur kemampuan aplikasi dalam menemukan semua kalimat relevan, yaitu rasio kalimat relevan yang ditemukan dengan total kalimat penting. *F-measure* menggabungkan precision dan recall untuk menilai keseimbangan antara akurasi dan cakupan. Ketiga metrik ini digunakan untuk mengevaluasi kualitas ringkasan otomatis dengan membandingkannya dengan ringkasan manual. Nilai *precision* dan *recall* yang lebih tinggi menunjukkan kinerja aplikasi yang lebih baik dalam menghasilkan ringkasan yang relevan [21][22]. Berikut adalah perhitungan metrik evaluasi untuk sebuah dokumen:

Sebagai contoh, untuk Dokumen 1:

- *True Positive* (TP) = 290 (kalimat yang muncul di ringkasan aplikasi dan ringkasan pakar)

- *False Positive* (FP) = 140 (kalimat yang hanya muncul di ringkasan aplikasi)
- *False Negative* (FN) = 116 (kalimat yang hanya muncul di ringkasan pakar)

Maka perhitungan:

$$Precision = \frac{tp}{tp + fp} = \frac{290}{290 + 140} = 0.67$$

$$Recall = \frac{tp}{tp + fn} = \frac{290}{290 + 116} = 0.71$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$F - measure = \frac{2 \times 0.67 \times 0.71}{0.67 + 0.71} = 0.69$$

Berikut ini disajikan hasil evaluasi detail untuk dokumen 1-10 yang merupakan sampel dari Kelompok 1, untuk menunjukkan bagaimana nilai *True Positive* (TP), *False Positive* (FP), serta *False Negative* (FN) digunakan untuk menghitung nilai *Precision* (P), *Recall* (R), dan *F-measure* (F-1). Tabel 2, 3 dan 4 menyajikan hasil evaluasi untuk setiap *compression rate* (75%, 50%, dan 25%).

Tabel 2: Evaluasi metrik dokumen pada *compression rate* 75%

Dok	TP	FP	FN	P	R	F-1
1	290	140	116	0.67	0.71	0.69
2	354	182	84	0.66	0.81	0.73
3	386	251	92	0.61	0.81	0.69
4	322	295	230	0.52	0.58	0.55
5	330	201	125	0.62	0.73	0.67
6	299	220	61	0.58	0.83	0.68
7	305	236	130	0.56	0.70	0.63
8	362	384	110	0.49	0.77	0.59
9	362	368	113	0.50	0.76	0.60
10	398	222	140	0.64	0.74	0.69
Rata-rata				**0.58**	**0.74**	**0.65**

Tabel 3: Evaluasi metrik dokumen pada *compression rate* 50%

Dok	TP	FP	FN	P	R	F-1
1	341	233	65	0.59	0.84	0.70
2	403	389	35	0.51	0.92	0.66
3	411	334	67	0.55	0.86	0.67
4	399	543	153	0.42	0.72	0.53
5	398	282	57	0.59	0.87	0.70
6	312	271	48	0.54	0.87	0.66
7	345	305	90	0.53	0.79	0.64
8	400	542	72	0.42	0.85	0.57
9	447	598	28	0.43	0.94	0.59
10	414	308	124	0.57	0.77	0.66
Rata-rata				**0.52**	**0.84**	**0.64**

Tabel 4: Evaluasi metrik dokumen pada *compression rate* 25%

Dok	TP	FP	FN	P	R	F-1
1	357	328	49	0.52	0.88	0.65
2	408	390	30	0.51	0.93	0.66
3	444	419	34	0.51	0.93	0.66
4	458	750	94	0.38	0.83	0.52
5	416	432	39	0.49	0.91	0.64
6	323	353	37	0.48	0.90	0.62
7	399	429	36	0.48	0.92	0.63
8	428	672	44	0.39	0.91	0.54
9	453	763	22	0.37	0.95	0.54
10	473	370	65	0.56	0.88	0.69
Rata-rata				**0.47**	**0.90**	**0.62**

Berdasarkan perhitungan detail yang telah ditunjukkan sebelumnya, berikut disajikan rangkuman hasil evaluasi untuk seluruh kelompok dokumen pada masing-masing compression rate.

Tabel 5: Hasil pengujian dengan *compression rate* 75%

Kelompok Dokumen	Precision	Recall	F-measure
Kelompok 1 (Dokumen 1-10)	0.58	0.74	0.65
Kelompok 2 (Dokumen 11-20)	0.62	0.77	0.68
Kelompok 3 (Dokumen 21-30)	0.63	0.69	0.65
Kelompok 4 (Dokumen 31-40)	0.67	0.67	0.67
Kelompok 5 (Dokumen 41-50)	0.61	0.71	0.65
Rata-rata	**0.62**	**0.71**	**0.66**

Tabel 6: Hasil pengujian dengan *compression rate* 50%

Kelompok Dokumen	Precision	Recall	F-measure
Kelompok 1 (Dokumen 1-10)	0.52	0.84	0.64
Kelompok 2 (Dokumen 11-20)	0.56	0.84	0.67
Kelompok 3 (Dokumen 21-30)	0.57	0.83	0.67
Kelompok 4 (Dokumen 31-40)	0.61	0.82	0.69
Kelompok 5 (Dokumen 41-50)	0.56	0.82	0.66
Rata-rata	**0.56**	**0.83**	**0.67**

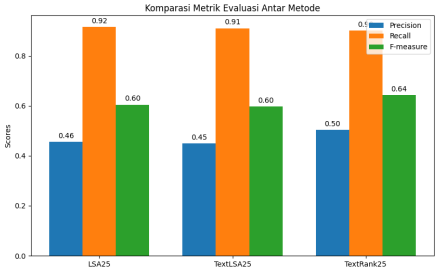
Tabel 7: Hasil pengujian dengan *compression rate* 25%

Kelompok Dokumen	Precision	Recall	F-measure
Kelompok 1 (Dokumen 1-10)	0.47	0.90	0.62
Kelompok 2 (Dokumen 11-20)	0.50	0.90	0.64
Kelompok 3 (Dokumen 21-30)	0.51	0.90	0.65
Kelompok 4 (Dokumen 31-40)	0.54	0.91	0.68
Kelompok 5 (Dokumen 41-50)	0.50	0.91	0.64
Rata-rata	**0.50**	**0.90**	**0.64**

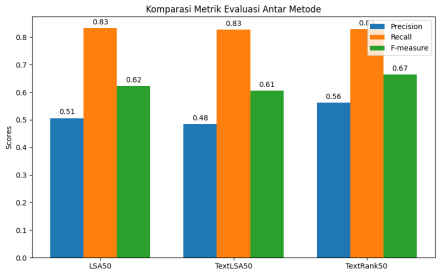
Tabel 5, 6, dan 7 menampilkan hasil pengujian peringkasan otomatis yang dibandingkan dengan peringkasan manual oleh pakar, dengan tingkat *compression rate* masing-masing sebesar 75%, 50%, dan 25%. Hasil ini mencakup metrik evaluasi seperti precision, recall, dan F-measure, yang menunjukkan efektivitas algoritma *TextRank* dalam menghasilkan ringkasan yang informatif dan relevan.

Untuk memperluas evaluasi, dilakukan pengujian tambahan menggunakan algoritma LSA dan kombinasi LSA dan *TextRank*. Pengujian ini bertujuan untuk membandingkan efektivitas algoritma TextRank dengan pendekatan lain dalam menghasilkan ringkasan yang informatif dan relevan. LSA digunakan untuk menangkap hubungan semantik antar kalimat, sementara kombinasi LSA + *TextRank* menggabungkan kekuatan analisis semantik dari LSA dengan pemeringkatan berbasis graf dari TextRank.

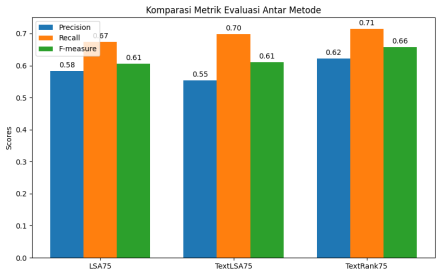
Gambar berikut menyajikan komparasi metrik evaluasi (Precision, Recall, dan F-measure) antar metode LSA, *TextRank,* dan kombinasi LSA + *TextRank* pada berbagai tingkat *compression rate* (25%, 50%, dan 75%).



Gambar . Komparasi Metrik Evaluasi Antar Metode pada Compression Rate 25%



Gambar . Komparasi Metrik Evaluasi Antar Metode pada Compression Rate 50%



Gambar . Komparasi Metrik Evaluasi Antar Metode pada Compression Rate 75%

Hasil evaluasi menunjukkan bahwa LSA memiliki recall tertinggi pada *compression rate* 25% dan 50%, mencerminkan kemampuannya dalam mempertahankan lebih banyak informasi dari dokumen asli. Namun, pada *compression rate* 75%, *TextRank* mengungguli LSA dalam recall, precision, dan F-measure. TextRank juga menunjukkan *precision* tertinggi pada semua tingkat *compression rate*, menunjukkan keunggulannya dalam menghasilkan ringkasan yang lebih relevan. Sementara itu, kombinasi LSA + TextRank menghasilkan keseimbangan yang baik antara *precision* dan *recall*, tetapi tidak selalu unggul dalam metrik individu.

3.4. Pembahasan

Secara umum, *compression rate* 75% cenderung menghasilkan ringkasan yang terlalu singkat, sementara 25% menghasilkan ringkasan yang terlalu panjang. Selain itu, algoritma *TextRank* menunjukkan konsistensi dengan mempertahankan proporsi jumlah kalimat dalam hasil ringkasan berdasarkan jumlah kalimat awal dokumen. Contohnya pada Tabel 1, Dokumen 11 dengan 448 kalimat menghasilkan ringkasan yang lebih panjang dibandingkan Dokumen 50 dengan 107 kalimat. Hal ini mengindikasikan bahwa *compression rate* 50% memberikan keseimbangan yang lebih baik dalam menghasilkan ringkasan yang mendekati penilaian pakar. Walaupun terdapat

gap yang signifikan antara ringkasan sistem dengan ringkasan pakar pada compression rate 75% di Tabel 1, hasil analisis menunjukkan rata-rata precision sebesar 0.62, recall 0.71, dan F-measure 0.66. Meskipun demikian, rata-rata ini membuktikan bahwa precision tertinggi berada pada compression rate 75%, yang artinya sistem peringkasan otomatis mampu menghasilkan ringkasan yang lebih relevan dan akurat pada tingkat kompresi tersebut. Hal ini menunjukkan bahwa meskipun ada perbedaan dengan ringkasan pakar, sistem ini masih dapat memberikan hasil yang cukup baik dalam hal kualitas informasi yang disajikan, tanpa menghilangkan informasi penting.

Compression rate (tingkat kompresi) dalam konteks ini mengacu pada persentase pengurangan teks dari dokumen asli ke versi ringkasannya. Semakin tinggi *compression rate*, semakin banyak teks yang dihilangkan dan semakin pendek ringkasannya. Semakin tinggi *compression rate* (ringkasan makin pendek), *Precision* meningkat namun *Recall* menurun, dan sebaliknya*. F-measure*, sebagai keseimbangan antara *Precision* dan *Recall*, tetap stabil pada kisaran 0.64-0.67, menunjukkan konsistensi kinerja aplikasi di berbagai *compression rate*.

Dalam penelitian ini, kami berfokus pada penerapan *TextRank* untuk menghasilkan ringkasan dokumen putusan pengadilan. Hasil pengujian terhadap 50 dokumen yang dibagi menjadi 5 kelompok menunjukkan bahwa *compression rate* memiliki pengaruh signifikan kepada nilai *precision*, *recall*, dan *F-measure*.

Saat *compression rate* 75%, aplikasi menunjukkan rata-rata *precision* tertinggi sebesar 0.62, dengan nilai tertinggi mencapai 0.67 pada Kelompok 4 (Dokumen 31-40), yang mengindikasikan bahwa aplikasi berhasil memilih kalimat-kalimat yang lebih akurat dan relevan dengan ringkasan manual. Namun, seiring dengan peningkatan *precision*, *recall* cenderung menurun, dengan rata-rata hanya mencapai 0.71.

Sebaliknya, pada *compression rate* yang lebih rendah seperti 25%, *recall* meningkat signifikan hingga rata-rata 0.90, dengan beberapa kelompok dokumen mencapai 0.91. Ini menunjukkan bahwa aplikasi dapat menangkap lebih banyak kalimat yang relevan dibandingkan dengan ringkasan manual. Namun, *precision* justru menurun hingga rata-rata 0.50, yang berarti bahwa banyak dari kalimat yang dipilih oleh aplikasi tidak semuanya relevan atau sesuai dengan ringkasan manual.

Fenomena ini menunjukkan adanya *trade-off* antara *precision* dan *recall* ketika *compression rate* teks diubah. Semakin tinggi *compression rate*, semakin pendek ringkasannya, namun semakin selektif aplikasi dalam memilih kalimat, yang berkontribusi pada *precision* yang lebih tinggi. Di sisi lain, *compression rate* yang lebih rendah memberikan ringkasan yang lebih panjang, sehingga *recall* meningkat, tetapi *precision* ringkasan berkurang. Menariknya, nilai *F-measure* tetap stabil pada kisaran 0.64-0.67 di berbagai *compression rate*, menunjukkan konsistensi kinerja aplikasi. Hasil ini menunjukkan bahwa *TextRank* tetap relevan dalam menghasilkan ringkasan yang informatif untuk dokumen putusan pengadilan. Keunggulannya adalah tidak memerlukan pelatihan atau pengetahuan domain spesifik, sehingga cocok untuk aplikasi peringkasan dokumen yang memerlukan kecepatan dan efisiensi.

Selain menggunakan TextRank, penelitian ini juga membandingkan hasil ringkasan dengan algoritma Latent Semantic Analysis (LSA) untuk mengevaluasi efektivitas pendekatan berbasis graf dan pendekatan semantik dalam menghasilkan ringkasan dokumen hukum. Hasil analisis menunjukkan bahwa pada compression rate rendah (25%), LSA memiliki keunggulan signifikan dalam recall (0.92), mengindikasikan kemampuannya dalam menangkap lebih banyak informasi dari dokumen asli. Namun, precision LSA hanya sebesar 0.46, yang berarti sebagian besar informasi yang disertakan tidak relevan dengan ringkasan pakar. Sebaliknya, TextRank menunjukkan keseimbangan yang lebih baik dengan precision 0.50, recall 0.90, dan F-measure tertinggi pada 0.64, menjadikannya metode yang lebih efektif untuk menghasilkan ringkasan yang relevan pada compression rate ini.

Pada compression rate menengah (50%), TextRank kembali menunjukkan performa terbaik dengan precision sebesar 0.56 dan F-measure tertinggi 0.67. Sementara itu, LSA dan kombinasi TextLSA memiliki recall yang sama (0.83), tetapi LSA sedikit lebih unggul dalam F-

measure (0.62) dibandingkan TextLSA (0.61). Hal ini menunjukkan bahwa TextRank lebih baik dalam menjaga relevansi ringkasan dibandingkan kedua metode lainnya pada tingkat kompresi ini.

Pada compression rate tinggi (75%), TextRank tetap menunjukkan performa unggul dengan precision 0.62, recall 0.71, dan F-measure tertinggi sebesar 0.66. LSA pada tingkat kompresi ini memiliki precision sebesar 0.58 dan F-measure sebesar 0.61, sedangkan TextLSA mencatatkan precision lebih rendah (0.55) namun dengan recall yang sama dengan LSA (0.70). Hal ini menunjukkan bahwa TextRank lebih efektif untuk menghasilkan ringkasan pendek yang relevan dibandingkan kedua metode lainnya.

Pendekatan gabungan TextRank dan LSA (TextLSA) memberikan hasil yang cukup stabil di berbagai compression rate, tetapi tidak selalu melampaui performa masing-masing metode secara individu. Efektivitas TextLSA sangat bergantung pada struktur dokumen dan tingkat kompresi yang digunakan, sehingga metode ini memberikan fleksibilitas tanpa menjadi yang terbaik dalam kondisi tertentu.

Secara keseluruhan, penelitian ini menunjukkan bahwa algoritma TextRank efektif dalam menghasilkan ringkasan dokumen putusan pengadilan dengan mempertimbangkan berbagai tingkat *compression rate*. Hasil evaluasi menunjukkan bahwa *compression rate* 50% memberikan keseimbangan terbaik antara precision dan recall, menghasilkan ringkasan yang paling mendekati penilaian pakar. Meskipun *compression rate* 75% menghasilkan precision yang lebih tinggi, hal ini diimbangi dengan penurunan recall, sedangkan pada *compression rate* 25%, *recall* meningkat tetapi *precision* menurun. Fenomena ini menegaskan adanya *trade-off* yang signifikan antara kedua metrik tersebut. Hasil evaluasi menunjukkan bahwa aplikasi ini mampu menghasilkan ringkasan yang informatif, dengan nilai *precision* tertinggi sebesar 0.62 pada compression rate 75%.

Hasil penelitian ini memberikan wawasan yang berpotensi mendukung pengembangan aplikasi peringkasan teks otomatis, khususnya dalam konteks dokumen hukum, serta membuka peluang untuk penelitian lebih lanjut dalam meningkatkan algoritma peringkasan yang ada.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi peringkasan otomatis untuk putusan pengadilan dengan menggunakan algoritma TextRank. Hasil evaluasi menunjukkan bahwa aplikasi ini mampu menghasilkan ringkasan yang informatif, dengan precision tertinggi sebesar 0.62 pada compression rate 75%, yang menandakan kemampuan TextRank untuk menangkap informasi penting meskipun terdapat gap dengan ringkasan manual pakar. Selain itu, analisis menunjukkan bahwa compression rate yang lebih rendah, seperti 25%, memberikan recall yang lebih tinggi, meskipun dengan penurunan precision.

Hasil penelitian ini memperkuat relevansi algoritma TextRank dalam konteks dokumen hukum, terutama karena kemampuannya untuk menangani elemen tidak relevan seperti header, footer, dan watermark secara otomatis. Penelitian ini juga menunjukkan bahwa penggabungan teknik, seperti kombinasi TextRank dan LSA, memiliki potensi untuk meningkatkan keseimbangan antara precision dan recall pada tingkat kompresi tertentu. Namun, performa TextRank secara individual tetap kompetitif di berbagai tingkat compression rate.

Sebagai langkah lanjutan, penelitian ini membuka peluang untuk eksplorasi kombinasi algoritma berbasis graf dengan pendekatan deep learning, seperti **BERT**, untuk menghasilkan ringkasan yang lebih akurat dan relevan. Potensi penerapan hasil penelitian ini juga cukup luas, mulai dari sistem informasi hukum untuk meringkas dokumen putusan pengadilan hingga aplikasi lain dalam bidang pemrosesan bahasa alami.

Secara keseluruhan, penelitian ini diharapkan memberikan kontribusi signifikan terhadap pengembangan aplikasi peringkasan teks otomatis, khususnya dalam konteks dokumen hukum, sekaligus menjadi pijakan untuk penelitian-penelitian di masa depan dalam domain ini.

DAFTAR PUSTAKA

[1] G. Z. Maulidya, S. N. Rahmawati, V. Rahmawati, and A. F. Mardany, "Ratio Decidendi Putusan, Jenis-Jenis Putusan dan Upaya Hukum Terhadap Putusan yang Telah Memiliki Kekuatan Hukum Tetap Ditinjau dari Perspektif Hukum Acara Pidana di Indonesia," *HUKMY Jurnal Hukum*, vol. 3, no. 1, pp. 211–230, May 2023, doi: 10.35316/hukmy.2023.v3i1.211-230.

[2] S. Shidarta, "Putusan Pengadilan sebagai Objek Penulisan Artikel Ilmiah," *Undang Jurnal Hukum*, vol. 5, no. 1, pp. 105–142, Jul. 2022, doi: 10.22437/ujh.5.1.105-142.

[3] D. A. Wicaksana, *Penelitian format putusan pengadilan Indonesia: studi empat lingkungan peradilan di bawah Mahkamah Agung*. 2020.

[4] S. Thange, J. Dange, V. Karjule, and J. Sase, "A survey on text summarization Techniques," *International Journal of Scientific and Research Publications*, vol. 13, no. 11, pp. 528–535, Nov. 2023, doi: 10.29322/ijsrp.13.11.2023.p14355.

[5] S. Alhojely and J. Kalita, "Recent Progress on Text Summarization," *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1503–1509, Dec. 2020, doi: 10.1109/csci51800.2020.00278.

[6] G. W. Wicaksono, S. F. A. Asqalani, Y. Azhar, N. P. Hidayah, and A. Andreawana, "Automatic Summarization of Court Decision Documents over Narcotic Cases Using BERT," *JOIV International Journal on Informatics Visualization*, vol. 7, no. 2, p. 416, May 2023, doi: 10.30630/joiv.7.2.1811.

[7] M. Rusbandi, I. F. Rozi, and K. S. Batubulan, "Otomatisasi peringkasan teks pada dokumen hukum menggunakan metode latent semantic analysis," *Jurnal Informatika Polinema*, vol. 7, no. 3, pp. 9–16, Jun. 2021, doi: 10.33795/jip.v7i3.515.

[8] M. S. Zulvi, "Systematic Literature Review Penerapan metodologi agile dalam berbagai bidang," *Jurnal Komputer Terapan*, vol. 7, no. 2, pp. 300–313, Dec. 2021, doi: 10.35143/jkt.v7i2.5116.

[9] F. Paramudita and M. I. Zulfa, "Aplikasi Android pendeteksi kualitas beras berbasis machine learning menggunakan metode Convolutional Neural Network," *Jurnal Pendidikan Dan Teknologi Indonesia*, vol. 3, no. 7, pp. 297–305, Aug. 2023, doi: 10.52436/1.jpti.310.

[10] Sheilafitria, "Court Decision Document in Narcotic Cases Summarization Using BERT.," *GitHub*. https://github.com/sheilafitria/Court-Decision-Document-in-Narcotic-Cases-Summarization-Using-BERT (accessed Sep. 20, 2024).

[11] J. Petrus, E. Ermatita, S. Sukemi, and E. Erwin, "An adaptable sentence segmentation based on Indonesian rules," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 3, p. 1491, Mar. 2023, doi: 10.11591/ijai.v12.i3.pp1491-1499.

[12] D. Mustikasari, I. Widaningrum, R. Arifin, and W. H. E. Putri, "Comparison of effectiveness of stemming algorithms in Indonesian documents," *Advances in Engineering Research/Advances in Engineering Research*, Jan. 2021, doi: 10.2991/aer.k.210810.025.

[13] U. Rani and K. Bidhan, "Comparative assessment of extractive summarization: TextRank, TF-IDF and LDA," *Journal of Scientific Research*, vol. 65, no. 01, pp. 304–311, Jan. 2021, doi: 10.37398/jsr.2021.650140.

[14] G. Salton, *Automatic text processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley Publishing Company, 1989.

[15] T. Tokunaga and M. Iwayama, "Text Categorization based on Weighted Inverse Document Frequency," *IPSJ SIG Notes*, vol. 1994, no. 28, pp. 33–40, Mar. 1994, [Online]. Available: https://ci.nii.ac.jp/naid/110002934824

[16] N. Y. Januzaj and N. A. Luma, "Cosine Similarity – a computing approach to match similarity between higher education programs and job market demands based on maximum number of common words," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, no. 12, pp. 258–268, Jun. 2022, doi: 10.3991/ijet.v17i12.30375.

[17] G. Tapas and N. Mehala, "Latent semantic analysis in automatic text summarisation: a state-of-the-art analysis," *International Journal of Intelligence and Sustainable*

Computing, vol. 1, no. 2, p. 128, Jan. 2021, doi: 10.1504/ijisc.2021.113294.

[18] J. Steinberger and K. Jezek, "Using Latent Semantic Analysis in Text Summarization and Summary Evaluation," in *Proceedings of the International Conference on Information Systems Implementation and Modeling (ISIM)*, 2004, pp. 93-100.

[19] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Text," *Empirical Methods in Natural Language Processing*, pp. 404–411, Jul. 2004, [Online]. Available: https://digital.library.unt.edu/ark:/67531/metadc30962/m2/1/high_res_d/Mihalcea-2004-TextRank-Bringing_Order_into_Texts.pdf

[20] I. M. S. Putra, Y. Adiwinata, D. P. S. Putri, and N. P. Sutramiani, "Extractive text summarization of student essay assignment using sentence weight features and fuzzy C-Means," *International Journal of Artificial Intelligence Research*, vol. 5, no. 1, Jan. 2021, doi: 10.29099/ijair.v5i1.187.

[21] P. Christen, D. J. Hand, and N. Kirielle, "A review of the F-Measure: its history, properties, criticism, and alternatives," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–24, Jun. 2023, doi: 10.1145/3606367.

[22] A. F. Sadeli and I. I. Lawanda, "Recall, precision, and F-Measure for evaluating information retrieval system in Electronic Document Management Systems (EDMS)," *Khizanah al-Hikmah Jurnal Ilmu Perpustakaan Informasi Dan Kearsipan*, vol. 11, no. 2, pp. 231–241, Nov. 2023, doi: 10.24252/kah.v11i2a8.