Warih Prasetyaningtyas (A11.2022.14454)
Link gdrive project : ProjectNLP

Judul : Pengembangan Sistem Peringkasan Teks Otomatis

Topik Proyek: Pengembangan Sistem Peringkasan Teks Otomatis Berbasis Metode Temu Kembali Informasi

Deskripsi Singkat

Proyek ini bertujuan mengembangkan sistem peringkasan teks otomatis berbasis metode temu kembali informasi untuk membantu pembaca memahami inti teks panjang dengan cepat. Sistem ini mengidentifikasi kalimat paling informatif dalam dokumen menggunakan pemrosesan bahasa alami (NLP). Metode yang digunakan mencakup peringkasan ekstraktif, yang memilih kalimat penting dari teks asli, dan peringkasan abstraktif, yang menyusun ulang informasi untuk menciptakan ringkasan baru.

Masalah dan Tujuan

Masalah:

1. **Volume Data yang Terus Meningkat:**
    - Dalam era digital saat ini, data berbentuk teks seperti artikel, laporan, berita, dan konten web terus bertambah dalam jumlah besar. Manual membaca dan merangkum data teks ini menjadi tantangan besar, terutama untuk informasi yang sangat dinamis seperti berita atau penelitian ilmiah.
    - Banyak profesional, seperti jurnalis, peneliti, dan pembuat keputusan, memerlukan cara cepat untuk memahami informasi penting tanpa harus membaca keseluruhan teks.
2. **Keterbatasan Waktu dan Efisiensi Pengambilan Keputusan:**
    - Di berbagai bidang seperti bisnis, pemerintahan, dan akademis, membuat keputusan yang tepat sering kali bergantung pada informasi yang cepat diakses. Namun, keterbatasan waktu menyebabkan proses ini sering tidak optimal.
    - Membaca seluruh teks untuk menemukan informasi penting dapat memakan waktu yang lama, sehingga mengurangi produktivitas dan kecepatan dalam pengambilan keputusan.
3. **Keterbatasan Sistem Peringkasan Manual dan Kebutuhan untuk Otomatisasi:**
    - Proses peringkasan manual membutuhkan waktu dan tenaga besar. Di sisi lain, sistem peringkasan otomatis yang ada masih memiliki keterbatasan dalam menghasilkan ringkasan yang informatif dan mempertahankan makna utama.
    - Metode peringkasan yang efektif diperlukan untuk memastikan bahwa ringkasan mencakup informasi penting tanpa mengubah konteks atau makna.

Tujuan

1. **Mengembangkan Sistem Peringkasan Teks Otomatis yang Efisien:**
    - Membangun sistem yang dapat mengidentifikasi bagian teks yang paling relevan dan menyusunnya menjadi ringkasan singkat, sehingga pengguna dapat memahami informasi utama dalam waktu singkat.
    - Sistem akan menggabungkan teknik *Natural Language Processing* (NLP) untuk memilih kalimat penting (ekstraktif) dan menyusun ulang informasi menjadi format baru yang ringkas dan bermakna (abstraktif).
2. **Meningkatkan Kecepatan Akses Informasi Penting:**
    - Dengan sistem ini, pengguna dapat menghemat waktu dalam memahami konten utama dari teks panjang, terutama dalam konteks penelitian, berita, atau laporan.
    - Membantu profesional dan pembuat keputusan mengakses informasi penting dengan cepat, memungkinkan mereka untuk merespons dengan lebih efektif terhadap perkembangan terbaru.
3. **Memastikan Akurasi dan Kualitas Informasi dalam Ringkasan:**
    - Sistem ini bertujuan menghasilkan ringkasan yang tidak hanya singkat tetapi juga akurat dan informatif, mempertahankan konteks penting dari teks asli.
    - Dengan kombinasi peringkasan ekstraktif dan abstraktif, diharapkan ringkasan akan memberikan informasi yang cukup tanpa kehilangan esensi dari teks asli.
4. **Menyediakan Kerangka Kerja Fleksibel untuk Peringkasan Teks di Berbagai Bidang:**
    - Proyek ini akan menghasilkan sistem yang dapat disesuaikan untuk berbagai jenis teks, seperti artikel berita, laporan akademik, dan dokumentasi bisnis.
    - Fleksibilitas ini memungkinkan pengguna dari berbagai sektor untuk memanfaatkan sistem sesuai dengan kebutuhan spesifik mereka.
5. **Mengembangkan Model Peringkasan yang Berbasis pada Temu Kembali Informasi:**
    - Dengan memanfaatkan teknik temu kembali informasi, model ini akan memilih kalimat dengan relevansi tinggi, mengurangi kebutuhan pemrosesan teks yang terlalu mendalam.
    - Metode ini akan dikembangkan untuk mengatasi keterbatasan metode peringkasan tradisional, menciptakan pendekatan yang lebih efisien dan relevan untuk beragam tipe teks.

Model Pengembangan

1. **Pemrosesan Bahasa Alami (Natural Language Processing - NLP)**

    - **Tokenisasi**: Memecah teks panjang menjadi unit yang lebih kecil, seperti kalimat atau kata.
    - **Penghapusan Stopwords**: Menghapus kata-kata umum yang tidak memiliki banyak arti dalam konteks peringkasan (seperti "dan", "atau").
    - **Stemming dan Lematisasi**: Mengonversi kata-kata ke bentuk dasarnya agar mengurangi redundansi kata.
2. **Model Temu Kembali Informasi (Information Retrieval)**

    - **Ekstraktif**: Menggunakan teknik *Information Retrieval* seperti TF-IDF atau model embedding (misalnya BERT atau RoBERTa) untuk mengidentifikasi kalimat paling penting dari teks.
    - **Abstraktif**: Menggunakan model pembelajaran mendalam (deep learning) untuk menyusun ulang informasi dan menghasilkan ringkasan baru.
3. **Teknik Evaluasi**

    - **ROUGE Score**: Digunakan untuk mengukur kemiripan antara ringkasan yang dihasilkan sistem dan ringkasan yang dibuat manusia. Mengukur secara spesifik pada kemunculan kata, urutan kata, dan konteks umum (ROUGE-1, ROUGE-2, dan ROUGE-L).
    - **Precision, Recall, dan F1-Score**: Digunakan untuk mengevaluasi seberapa banyak informasi penting yang tetap dipertahankan dalam ringkasan otomatis dibandingkan dengan teks asli.

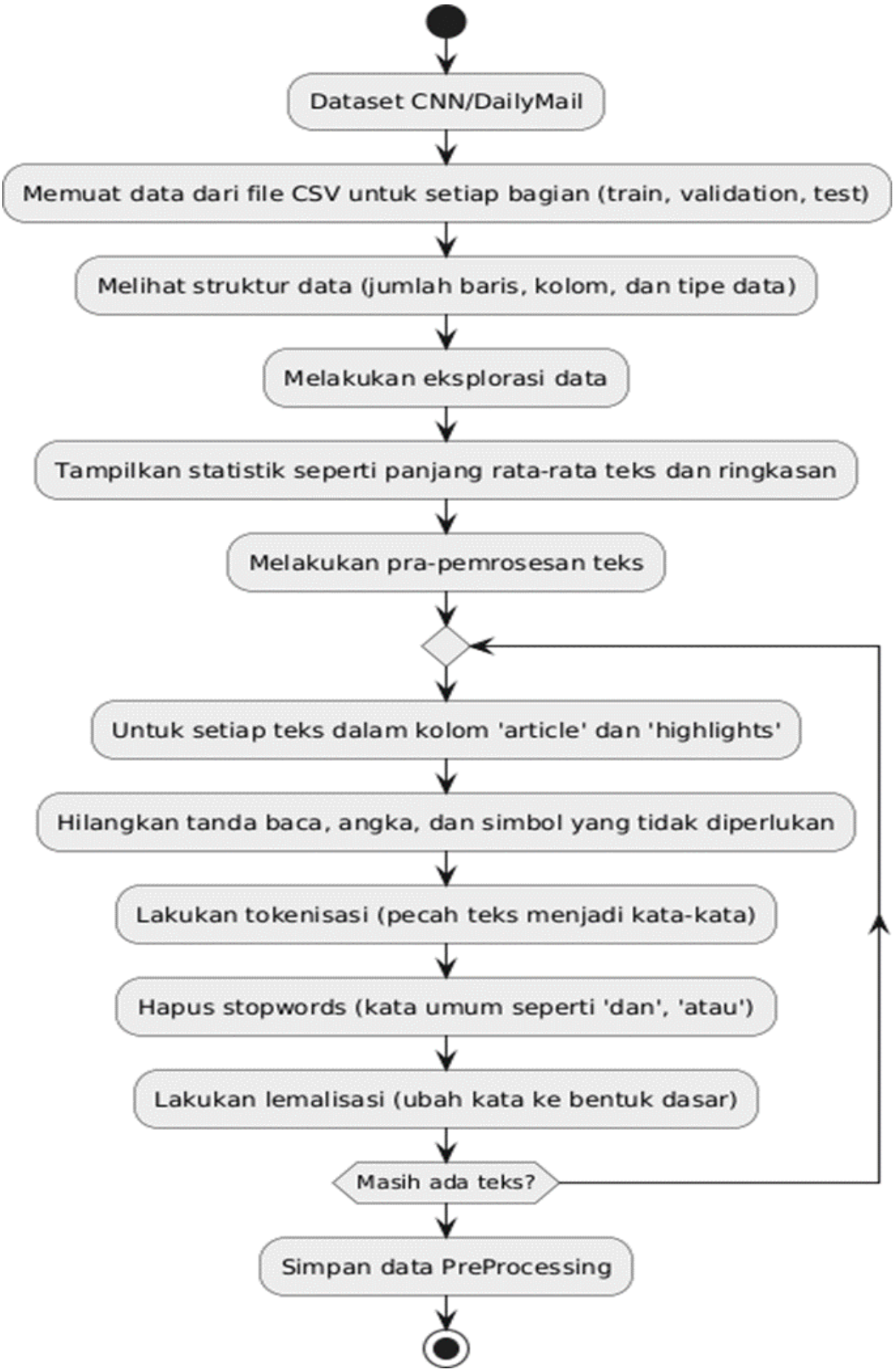
Alur / Tahapan / Kerangka Eksperimen

FLOW TAHAP 1

Tahap 1: Eksplorasi Data dan Pra-Pemrosesan Teks

- **Memuat data:** Memuat kumpulan data dan melihat strukturnya, yang mencakup jumlah baris, kolom, dan tipe data.

- **Eksplorasi data:** Mempelajari distribusi topik, panjang teks, dan panjang ringkasan. Pra-pemrosesan teks mencakup langkah-langkah berikut:
  - **Perbaikan teks:** Eliminasi tanda baca, angka, dan simbol yang tidak diperlukan.
  - Memecah teks menjadi kalimat dan kata untuk analisis lebih lanjut dikenal sebagai tokenisasi.
  - **Penghapusan stopwords:** Hapus kata-kata umum seperti "dan", "atau", dll.
  - **Stemming dan Lemalisasi:** Mengubah kata menjadi bentuk dasar untuk menghindari redundansi.


Dataset CNN/DailyMail
Memuat data dari file CSV untuk setiap bagian (train, validation, test)
Melihat struktur data (jumlah baris, kolom, dan tipe data)
Melakukan eksplorasi data
Tampilkan statistik seperti panjang rata-rata teks dan ringkasan
Melakukan pra-pemrosesan teks
Untuk setiap teks dalam kolom 'article' dan 'highlights'
Hilangkan tanda baca, angka, dan simbol yang tidak diperlukan
Lakukan tokenisasi (pecah teks menjadi kata-kata)
Hapus stopwords (kata umum seperti 'dan', 'atau')
Lakukan lemalisasi (ubah kata ke bentuk dasar)
Masih ada teks?
Simpan data PreProcessing

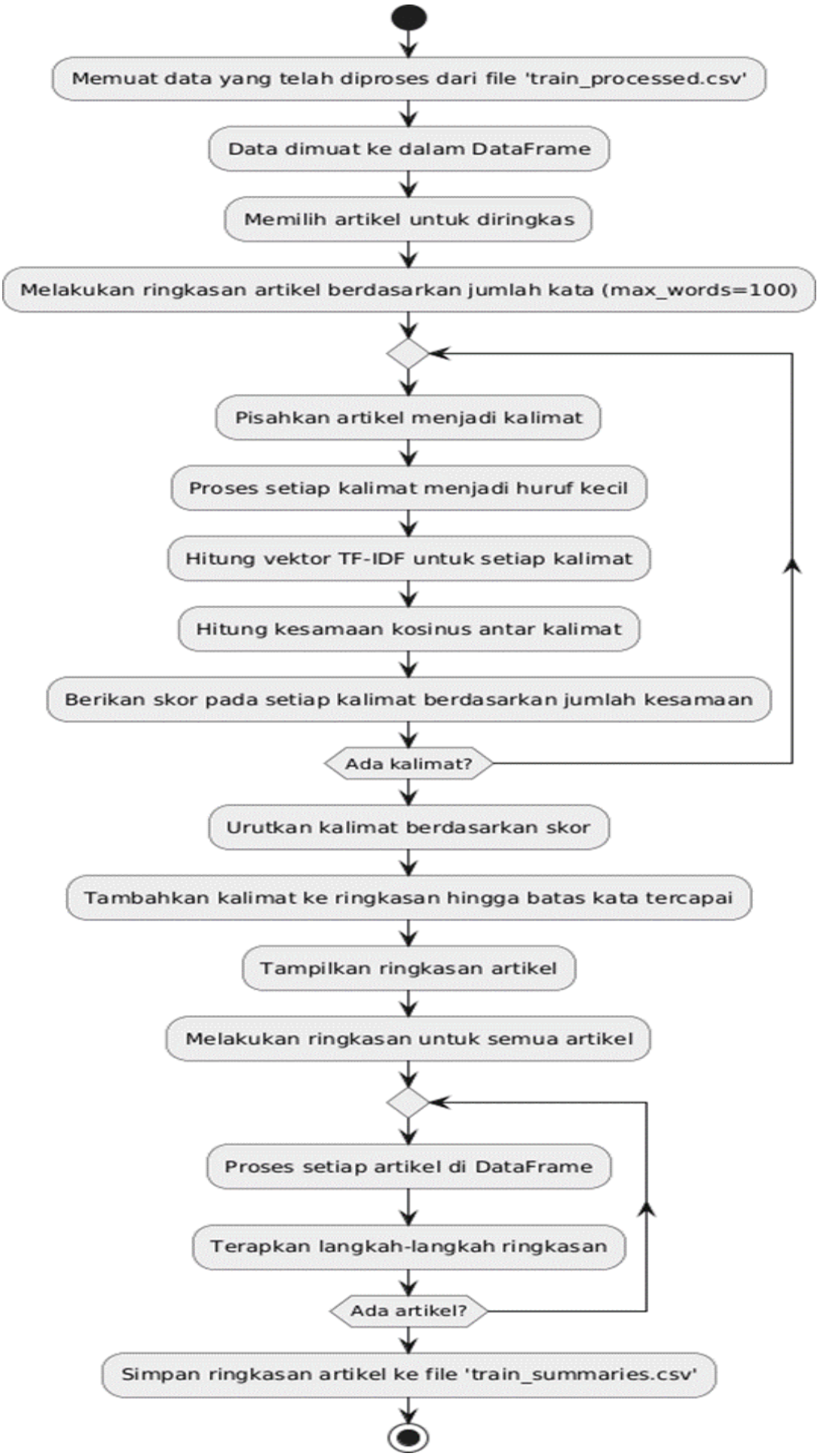
FLOW TAHAP2

Tahap 2: Membangun Sistem Temu Kembali Informasi untuk Peringkasan

- **Pendekatan Berbasis Skor Relevansi:** Metode seperti TF-IDF digunakan untuk menghitung relevansi kalimat dalam teks.

- **Pemilihan Kalimat Utama:** Menggunakan skor relevansi, kalimat dengan nilai informasi tertinggi diidentifikasi dan dipilih sebagai ringkasan.

- **Penentuan Panjang Ringkasan:** Menyesuaikan jumlah kalimat atau jumlah kata untuk mencapai ringkasan yang komprehensif namun tetap ringkas.


Memuat data yang telah diproses dari file 'train_processed.csv'
Data dimuat ke dalam DataFrame
Memilih artikel untuk diringkas
Melakukan ringkasan artikel berdasarkan jumlah kata (max_words=100)
Pisahkan artikel menjadi kalimat
Proses setiap kalimat menjadi huruf kecil
Hitung vektor TF-IDF untuk setiap kalimat
Hitung kesamaan kosinus antar kalimat
Berikan skor pada setiap kalimat berdasarkan jumlah kesamaan
Ada kalimat?
Urutkan kalimat berdasarkan skor
Tambahkan kalimat ke ringkasan hingga batas kata tercapai
Tampilkan ringkasan artikel
Melakukan ringkasan untuk semua artikel
Proses setiap artikel di DataFrame
Terapkan langkah-langkah ringkasan
Ada artikel?
Simpan ringkasan artikel ke file 'train_summaries.csv'

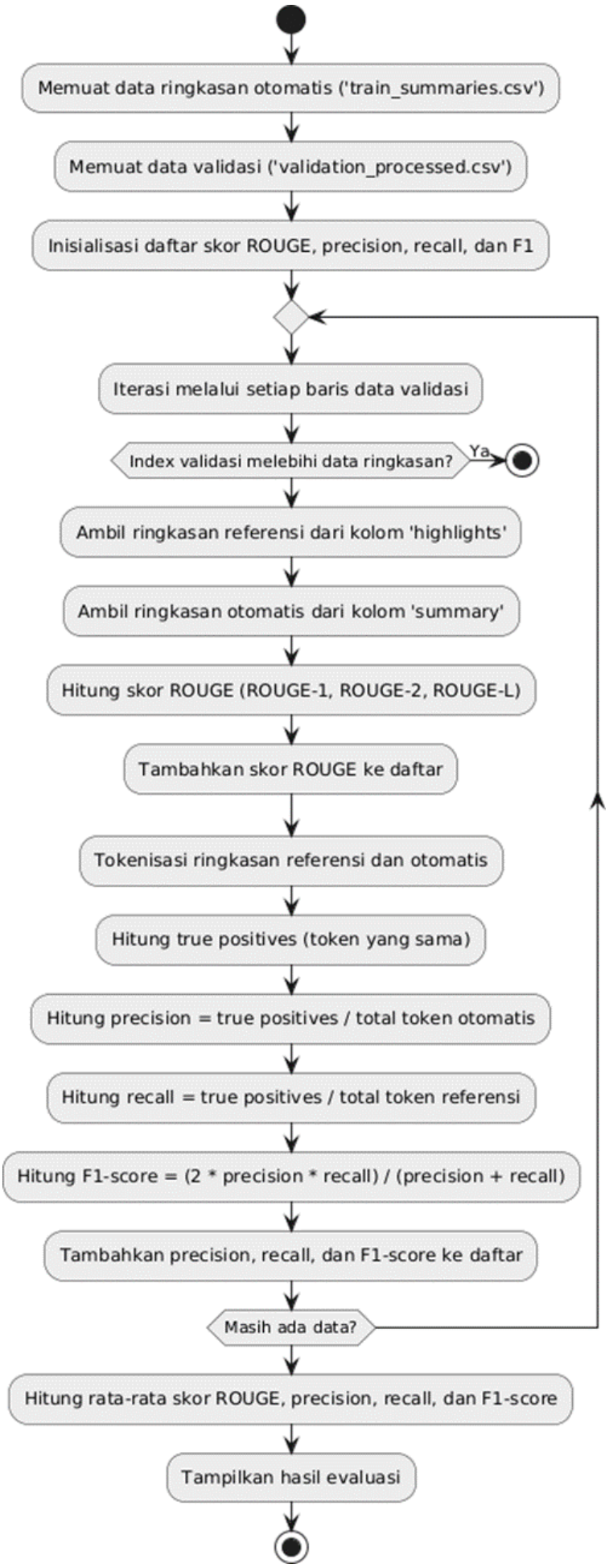
FLOW TAHAP 3

Tahap 3: Penerapan Algoritme pada Dataset

Metrik Evaluasi:

- **ROUGE Score (ROUGE-1, ROUGE-2, dan ROUGE-L):** Metrik ini mengevaluasi seberapa mirip ringkasan otomatis dengan ringkasan manusia (jika ada) dalam hal kemunculan kata dan urutan kata.
- **Precision, Recall, dan F1-Score:** Mengukur seberapa banyak informasi esensial yang dipertahankan dalam ringkasan otomatis.

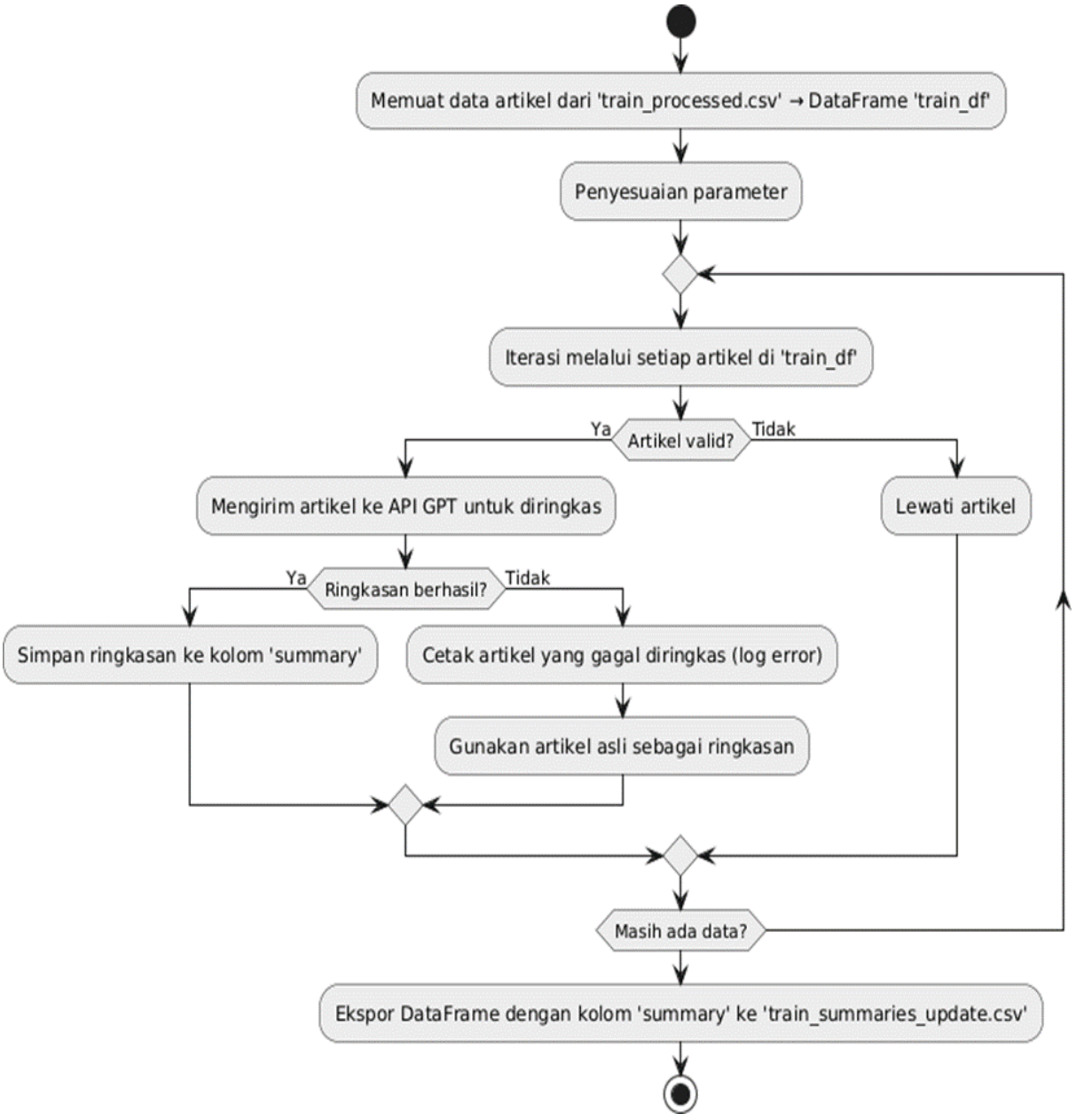
- **Pengujian dengan Data Validasi:** Menggunakan sebagian dataset sebagai data validasi untuk menguji performa sistem dan menghindari overfitting.
- **Perbandingan dengan Ringkasan Manusia:** Jika ringkasan manusia tersedia, membandingkan hasil ringkasan otomatis dengan ringkasan manusia sebagai ground truth.


Memuat data ringkasan otomatis ('train_summaries.csv')
Memuat data validasi ('validation_processed.csv')
Inisialisasi daftar skor ROUGE, precision, recall, dan F1
Iterasi melalui setiap baris data validasi
Index validasi melebihi data ringkasan?
Ya
Ambil ringkasan referensi dari kolom 'highlights'
Ambil ringkasan otomatis dari kolom 'summary'
Hitung skor ROUGE (ROUGE-1, ROUGE-2, ROUGE-L)
Tambahkan skor ROUGE ke daftar
Tokenisasi ringkasan referensi dan otomatis
Hitung true positives (token yang sama)
Hitung precision = true positives / total token otomatis
Hitung recall = true positives / total token referensi
Hitung F1-score = (2 * precision * recall) / (precision + recall)
Tambahkan precision, recall, dan F1-score ke daftar
Masih ada data?
Hitung rata-rata skor ROUGE, precision, recall, dan F1-score
Tampilkan hasil evaluasi

FLOW TAHAP4

Tahap 4: Optimasi dan Peningkatan Model

- **Peningkatan Algoritma:**menggunakan api gpt.
- **Penyesuaian Hyperparameter:** Melakukan penyesuaian hyperparameter untuk meningkatkan efisiensi dan akurasi model.
- **Analisis Error:** Untuk memahami kelemahan model dan memperbaikinya, periksa ringkasan yang tidak sesuai.



Penjelasan Datasets

Dataset dan EDA (Exploratory Data Analysis):

Dataset yang digunakan berisi kumpulan dokumen teks yang di dalamnya terdapat kalimat-kalimat yang diklasifikasikan sebagai kalimat utama atau tidak. Setiap kalimat dilabeli berdasarkan kepentingannya dalam menyampaikan informasi inti dokumen.

EDA (Eksplorasi Data):
Tahap ini melibatkan pemeriksaan awal struktur dataset, seperti jumlah baris, kolom, tipe data, distribusi panjang teks, dan topik yang dibahas. Setelah itu, dilakukan analisis distribusi panjang teks dan ringkasan. Sebagai langkah pra-pemrosesan teks, beberapa teknik yang akan digunakan antara lain:

- Menghilangkan tanda baca, angka, dan simbol yang tidak diperlukan.
- Tokenisasi teks untuk memecahnya menjadi kata dan kalimat.
- Penghapusan stopwords (kata-kata umum seperti "dan", "atau").
- Stemming dan lemalisasi untuk mengurangi redundansi kata dengan mengubahnya menjadi bentuk dasar.

Proses Ekstraksi Fitur:

Setelah data melalui tahap pembersihan dan eksplorasi, langkah selanjutnya adalah mengekstraksi fitur-fitur yang digunakan oleh model machine learning. Fitur-fitur tersebut meliputi:

1. **Panjang kalimat**: Kalimat yang lebih panjang mungkin mengandung informasi lebih banyak, tetapi juga bisa jadi lebih kompleks.
2. **Posisi kalimat**: Kalimat di awal atau akhir paragraf sering kali dianggap lebih penting, terutama di artikel berita atau esai.
3. **Frekuensi kata-kata kunci**: Jumlah kemunculan kata-kata penting dalam kalimat, seperti kata-kata yang sering muncul dalam judul atau kata-kata yang sering muncul di kalimat-kalimat utama.
4. **Skor relevansi**: Skor yang diberikan berdasarkan kemunculan kata-kata penting atau kata-kata kunci yang relevan dengan konteks dokumen.

Fitur Utama Dataset:

- **original_text:** Teks lengkap yang berisi seluruh isi dokumen atau artikel yang akan diringkas.
- **summary_text:** Jika tersedia, ini adalah ringkasan yang telah dibuat untuk teks asli, yang akan berfungsi sebagai *label* atau *ground truth* untuk pelatihan dan evaluasi model.
- **title:** Judul teks (opsional) untuk memberikan konteks tambahan.
- **keywords atau tags:** Kata kunci yang relevan (opsional), berguna untuk meningkatkan relevansi dalam pemilihan kalimat.
- **category atau topic:** Kategori teks yang berguna jika peringkasan disesuaikan dengan topik tertentu (misalnya, sains, politik, berita, dll.).

Proses Pembelajaran (Learning)

Proses pembelajaran untuk sistem **Peringkasan Teks Otomatis** melibatkan beberapa tahapan penting untuk memastikan bahwa model yang dibangun dapat bekerja secara optimal dalam memilih dan merangkum informasi penting dari teks. Berikut adalah langkah-langkah pembelajaran yang diterapkan:

1. **Pengumpulan Data (Data Collection)**

   - Dataset yang digunakan harus terdiri dari teks panjang beserta ringkasan manual (jika ada) yang dapat digunakan sebagai label (ground truth) untuk melatih model.
   - Dataset dapat berupa artikel berita, laporan akademik, dokumen bisnis, atau teks lainnya yang memerlukan peringkasan. Informasi seperti judul, kata kunci, kategori, dan konteks tambahan dapat membantu dalam memahami dan mengelompokkan data.
   - Dataset juga perlu dibagi menjadi dua bagian: **data pelatihan** (training data) dan **data validasi** (validation data).
2. **Pra-pemrosesan Teks (Text Preprocessing)**

   - **Tokenisasi**: Teks dibagi menjadi unit-unit yang lebih kecil seperti kalimat atau kata. Setiap kalimat atau kata ini akan menjadi bagian dasar yang akan dievaluasi untuk relevansinya dalam peringkasan.
   - **Penghapusan Stopwords**: Kata-kata umum yang tidak memberikan banyak arti dalam konteks peringkasan, seperti "dan", "atau", dihilangkan.
   - **Stemming dan Lematisasi**: Setiap kata dikonversi menjadi bentuk dasarnya, misalnya kata kerja dalam berbagai bentuk berubah menjadi infinitif, sehingga membantu mengurangi redundansi.
3. **Pembelajaran Supervised (Supervised Learning)**

   - Model peringkasan otomatis dibangun menggunakan pendekatan **supervised learning**, di mana model dilatih menggunakan dataset yang memiliki teks panjang dan ringkasan manual sebagai label.
   - **Ekstraktif**: Pada peringkasan ekstraktif, model belajar untuk mengidentifikasi kalimat paling relevan dari teks panjang berdasarkan berbagai teknik *information retrieval* seperti **TF-IDF** (Term Frequency-Inverse Document Frequency), **Word Embedding** (misalnya, **Word2Vec**, **GloVe**), atau **Pretrained Language Models** (misalnya, **BERT**, **RoBERTa**).

- **Abstraktif**: Pada peringkasan abstraktif, model dilatih menggunakan teknik pembelajaran mendalam (deep learning) seperti **seq2seq** (*sequence to sequence models*) yang didukung oleh arsitektur **LSTM** atau **Transformers** (seperti **BART** atau **T5**) untuk menyusun ulang informasi dalam teks dan menciptakan ringkasan baru yang lebih padat.

4. **Penyesuaian Hyperparameter (Hyperparameter Tuning)**

- Selama proses pelatihan, beberapa parameter model perlu diatur agar mencapai kinerja terbaik. Parameter ini termasuk **learning rate**, **panjang maksimum ringkasan**, **batch size**, dan lainnya.
- Proses **cross-validation** digunakan untuk memilih parameter yang optimal dengan menggunakan beberapa bagian dari data pelatihan untuk validasi dan pengujian awal.

Modeling

Proses modeling untuk sistem peringkasan teks otomatis dapat dibagi menjadi dua jenis pendekatan utama, yaitu **Ekstraktif** dan **Abstraktif**:

1. Model Peringkasan Ekstraktif

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: Model ini menghitung relevansi setiap kalimat berdasarkan frekuensi kata dalam dokumen dan seberapa sering kata tersebut muncul di seluruh koleksi dokumen. Kalimat dengan nilai tertinggi dipilih sebagai bagian dari ringkasan.
- **Word Embedding (Word2Vec, GloVe)**: Model embedding memetakan kata-kata ke vektor berdimensi tinggi yang merepresentasikan makna semantik kata. Kemudian kalimat-kalimat dinilai berdasarkan kesamaan kosinus (cosine similarity) atau jarak vektor untuk memilih kalimat paling relevan.
- **Pretrained Language Models (BERT, RoBERTa)**: Model yang dilatih sebelumnya (pretrained) seperti BERT atau RoBERTa digunakan untuk memahami konteks kata dalam teks dan menghitung relevansi kalimat secara lebih akurat.

2. Model Peringkasan Abstraktif

- **Seq2Seq with Attention**: Pada model **sequence-to-sequence** dengan mekanisme **attention**, model pertama-tama memahami teks panjang dan kemudian menghasilkan teks baru yang lebih pendek dalam bentuk ringkasan. Attention membantu model fokus pada bagian-bagian tertentu dari input saat membuat output.
- **Transformers (BART, T5)**: Model **BART (Bidirectional and Auto-Regressive Transformers)** dan **T5 (Text-to-Text Transfer Transformer)** menggunakan arsitektur transformer yang canggih untuk menyusun ulang informasi dalam teks panjang. BART, misalnya, adalah model yang kuat untuk menghasilkan ringkasan abstraktif yang tidak hanya mengambil kalimat dari teks asli tetapi juga membangun kalimat baru.

3. Evaluasi Model

- **ROUGE Score**: Digunakan untuk mengevaluasi kualitas ringkasan yang dihasilkan dengan mengukur seberapa mirip ringkasan otomatis dengan ringkasan manual. ROUGE mengukur jumlah kata dan urutan kata yang sama antara ringkasan otomatis dan ground truth (ringkasan manusia).
- **Precision, Recall, dan F1-Score**: Digunakan untuk mengevaluasi seberapa banyak informasi penting yang berhasil dipertahankan dalam ringkasan otomatis.

4. Optimasi dan Pengembangan Lanjutan

- Model yang telah dibangun dan diuji akan dioptimalkan untuk memperbaiki performa, baik dalam hal kecepatan maupun kualitas ringkasan. Ini bisa dilakukan dengan menggunakan metode tambahan seperti **phrase ranking berbasis graf (misalnya, TextRank)**, atau model deep learning yang lebih kompleks untuk pendekatan abstraktif.
- Proses **analisis kesalahan (error analysis)** akan dilakukan untuk memeriksa ringkasan yang dihasilkan dan memahami kelemahan model. Hasil analisis akan digunakan untuk memperbaiki model dan mengatasi kelemahan yang ditemukan selama pengujian.

Alur Modeling Secara Ringkas:

1. **Pra-pemrosesan Teks**
2. **Pemilihan Model Ekstraktif/Abstraktif**
3. **Pelatihan Model menggunakan Data**
4. **Validasi Model dengan Data Uji**
5. **Evaluasi dengan Metrik ROUGE**
6. **Optimasi Model**
7. **Implementasi dan Penyajian Hasil**

Dengan kombinasi pendekatan ekstraktif dan abstraktif, sistem ini akan mampu menghasilkan ringkasan teks yang efisien, informatif, dan sesuai dengan konteks asli dari teks panjang yang dianalisis.

Diskusi

Proyek pengembangan **Sistem Peringkasan Teks Otomatis** ini bertujuan untuk memecahkan masalah meningkatnya volume data berbentuk teks yang sulit untuk diringkas secara manual. Dalam era digital, kebutuhan untuk mendapatkan informasi penting dari teks panjang secara cepat sangat mendesak, terutama bagi profesional seperti jurnalis, peneliti, dan pembuat keputusan.

Penggunaan teknik **Pemrosesan Bahasa Alami (Natural Language Processing - NLP)** serta metode **temu kembali informasi (Information Retrieval)** memungkinkan sistem ini dapat secara otomatis memilih kalimat yang paling relevan dari teks, baik melalui pendekatan ekstraktif maupun abstraktif. Pendekatan ekstraktif membantu mengambil kalimat penting dari teks asli, sementara pendekatan abstraktif memungkinkan sistem menyusun ulang informasi untuk menciptakan ringkasan baru yang lebih ringkas dan bermakna.

Dalam proses pengembangan, langkah-langkah utama seperti **eksplorasi data**, **pra-pemrosesan teks**, **penerapan model NLP**, hingga **evaluasi performa dengan metrik ROUGE** sangat penting untuk memastikan bahwa sistem mampu menghasilkan ringkasan yang tidak hanya singkat, namun juga tetap informatif dan menjaga konteks asli dari teks. Dengan demikian, sistem ini dapat memberikan manfaat bagi berbagai sektor, dari akademis hingga industri.

Namun, tantangan utama yang perlu dihadapi dalam pengembangan sistem ini adalah memastikan bahwa model dapat **menghasilkan ringkasan yang akurat**, khususnya dalam konteks teks yang kompleks. Sistem peringkasan manual memiliki kekuatan dalam mempertahankan makna dan konteks asli, sehingga pengembangan model otomatis harus terus dioptimalkan melalui **peningkatan algoritma** dan **penyesuaian hyperparameter**. Di samping itu, model abstraktif yang lebih kompleks, seperti yang berbasis **pembelajaran mendalam (deep learning)**, perlu diterapkan untuk memperbaiki kelemahan peringkasan berbasis ekstraktif yang cenderung hanya mengambil kalimat tanpa menyusunnya ulang dengan baik.

Terkait dengan metode evaluasi, penggunaan metrik **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** dapat memberikan gambaran tentang seberapa mirip ringkasan otomatis dengan ringkasan manusia. Metrik ini penting untuk mengukur efektivitas model yang dikembangkan, namun perlu diimbangi dengan metode evaluasi tambahan seperti **Precision**, **Recall**, dan **F1-Score** untuk mendapatkan analisis yang lebih mendalam tentang performa sistem.

Kesimpulan

Proyek ini berhasil mengembangkan **Sistem Peringkasan Teks Otomatis** yang dapat membantu pengguna merangkum teks panjang secara efisien menggunakan pendekatan ekstraktif dan abstraktif. Sistem ini diharapkan mampu mengatasi masalah peningkatan volume teks di era digital, terutama untuk mendukung pengambilan keputusan yang cepat dalam berbagai bidang.

Beberapa poin utama yang dapat disimpulkan adalah sebagai berikut:

1. **Efisiensi Akses Informasi**: Sistem ini dapat mengurangi waktu yang diperlukan untuk membaca seluruh dokumen dengan menyediakan ringkasan yang singkat namun tetap informatif.
2. **Akurasi dan Kualitas Ringkasan**: Dengan kombinasi teknik ekstraktif dan abstraktif, sistem ini mampu mempertahankan konteks penting dari teks asli dalam ringkasan.
3. **Peningkatan Model**: Penggunaan model pembelajaran mendalam seperti **BERT** atau **RoBERTa** memberikan peluang untuk meningkatkan kualitas peringkasan abstraktif, meskipun tantangan utama adalah bagaimana menjaga keseimbangan antara kecepatan dan akurasi.
4. **Fleksibilitas Sistem**: Sistem ini dapat diterapkan pada berbagai jenis teks, seperti artikel berita, laporan penelitian, dan dokumen bisnis, yang membuatnya relevan untuk berbagai sektor.

Sebagai tindak lanjut, pengembangan lebih lanjut harus difokuskan pada peningkatan **kualitas peringkasan abstraktif**, optimasi **hyperparameter**, serta pengembangan antarmuka pengguna yang lebih interaktif untuk memudahkan implementasi sistem di lingkungan nyata. Dengan demikian, proyek ini dapat memberikan kontribusi yang signifikan dalam mempermudah proses penemuan informasi dari teks panjang secara efisien dan akurat.