**PENERAPAN ALGORITMA TRANSFORMER DALAM APLIKASI PARAFRASE TEKS OTOMATIS**

**PROPOSAL JUDUL SKRIPSI**

**Oleh:**

**NAMA JENIME DARWIN**

**NIM. 2144041**



**JENJANG PENDIDIKAN STRATA-1**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**STMIK TIME**

**MEDAN**

**2024**

# DAFTAR ISI

**DAFTAR ISI i**

**DAFTAR TABEL ii**

**BAB I PENDAHULUAN** **1**

* 1. Latar Belakang Masalah 1
  2. Rumusan Masalah 3
  3. Batasan Masalah 3
  4. Tujuan dan Manfaat Penelitian 4
  5. Metodologi Penelitian 4

1.6. Analisis Sistem 6

1.7 Perancangan Sistem 6

1.8 Sistematika Penulisan 9

**BAB II TINJAUAN PUSTAKA 10**

* 1. Parafrase Teks 10
  2. Deep learning 11
  3. Transformer 12
  4. Natural Language Processing (NLP) 14
  5. GPT (Generative Pre-trained Transformer) 15
  6. Pycharm 16
  7. Penelitian Terdahulu 18

**BAB III METODE PENELITIAN 24**

* 1. Analisis Kondisi Saat Ini 28
  2. Analisis Metode 29
  3. Analisis Sistem Usulan 30
  4. Perancangan Sistem 30

**DAFTAR PUSTAKA 34**

**DAFTAR TABEL**

Tabel 1.1. Jadwal Penelitian 5

**DAFTAR GAMBAR**

Gambar 2.1. Interface Pycharm 18

Gambar 3.1. Arsitektur Transformer 25

Gambar 3.2. Flowchart Parafrase 31

Gambar 3.3. Activity Diagram 32

Gambar 3.4. Halaman Utama 33

# BAB I

# PENDAHULUAN

## Latar Belakang Masalah

Belakangan ini kemajuan teknologi dalam bidang pemrosesan bahasa alami *Natural Language Processing* (NLP) telah membuka peluang besar untuk mengembangkan berbagai aplikasi yang mampu memahami dan memanipulasi teks secara otomatis. Salah satu terobosan signifikan dalam NLP adalah pengenalan model *Transformer*. Model ini telah menjadi fondasi bagi berbagai tugas NLP yang kompleks, seperti penerjemahan mesin, pemahaman teks, dan parafrase teks. Arsitektur *Transformer*, dengan kemampuan pemrosesan paralelnya, telah memberikan dampak yang luar biasa dalam meningkatkan efisiensi dan akurasi dalam berbagai aplikasi NLP[1].

Parafrase teks otomatis adalah sebuah proses yang mengubah suatu teks menjadi teks lain dengan makna yang serupa namun dengan kata-kata atau struktur yang berbeda. Aplikasi ini mempunyai berbagai manfaat, mulai dari membantu penulis dalam menghasilkan konten baru tanpa plagiarisme, meningkatkan kualitas terjemahan mesin, hingga mendukung pengembangan *chatbot* dan asisten *virtual* yang lebih canggih[2]. Dalam era digital saat ini, kebutuhan akan konten yang orisinal dan beragam semakin meningkat, membuat parafrase teks menjadi salah satu solusi penting dalam manajemen konten.

Tantangan utama dalam parafrase teks adalah memastikan bahwa makna asli tetap terjaga sementara struktur dan kosa kata diubah. Dalam hal ini, model *Transformer* menyediakan solusi yang efektif. *Transformer*, dengan arsitektur berbasis perhatian (*attention mechanism*), mampu menangkap hubungan antar kata dalam sebuah kalimat dengan lebih baik dibandingkan model-model sebelumnya. Hal ini memungkinkan *Transformer* untuk menghasilkan teks parafrase yang lebih akurat dan natural atau alami. Mekanisme perhatian ini memungkinkan model untuk lebih memahami suatu konteks secara keseluruhan, sehingga dapat menghasilkan parafrase yang lebih relevan dan bermakna[3].

Selain itu, perkembangan dalam teknik pelatihan model seperti *transfer learning* dan *fine-tuning* pada model Pralatih seperti GPT (*Generative Pre-trained Transformer*) semakin meningkatkan kemampuan parafrase teks otomatis[4]. Model-model ini, yang telah dilatih pada sejumlah besar data teks, dapat diadaptasi untuk tugas parafrase dengan penyesuaian minimal, memberikan hasil yang memuaskan dalam berbagai konteks penggunaan. Dengan menggunakan model pralatih ini, proses pengembangan aplikasi parafrase teks menjadi lebih efisien dan efektif[5].

Penerapan algoritma *Transformer* dalam aplikasi parafrase teks otomatis menyuguhkan berbagai keunggulan. Selain akurasi yang akan menjadi lebih tinggi, aplikasi ini juga memiliki potensi untuk meningkatkan produktivitas dalam pembuatan konten, mengurangi risiko plagiarisme, dan memperkaya interaksi pengguna dengan teknologi digital. Keunggulan ini sangat relevan dalam berbagai industri, termasuk pendidikan, pemasaran, dan layanan pelanggan, di mana kualitas dan kecepatan dalam pengolahan teks sangat penting.

Dengan latar belakang ini, penelitian ini bertujuan untuk mengembangkan aplikasi parafrase teks otomatis menggunakan algoritma *Transformer* dengan mengambil judul **PENERAPAN ALGORITMA TRANSFORMER DALAM APLIKASI PARAFRASE TEKS OTOMATIS.**

## Rumusan Masalah

Berdasarkan latar belakang masalah tersebut maka perumusan masalah adalah sebagai berikut:

1. Bagaimana mengimplementasi algoritma *Transformer* dalam menghasilkan parafrase teks yang akurat.
2. Bagaimana mengukur performa model *Transformer* dalam memahami konteks dan makna teks.

## Batasan Masalah

Agar penelitian ini terarah dan tidak menyimpang dari permasalahan yang telah dirumuskan, aplikasi yang akan dibuat hanya sebatas pemikiran penulis yaitu:

1. Fokus pada parafrase teks dalam bahasa Indonesia.
2. Menggunakan model *Transformer* GPT 2.
3. Menggunakan teknik *fine-tuning* standar untuk model pralatih.

## Tujuan dan Manfaat Penelitian

Tujuan penelitian ini adalah:

1. Mengidentifikasi algoritma *Transformer* dalam menghasilkan parafrase teks yang akurat.
2. Mengevaluasi performa model *Transformer* dalam memproses berbagai jenis teks dan gaya bahasa dalam aplikasi parafrase otomatis.

Adapun manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Penelitian ini dapat menjelaskan bagaimana teknik transfer *learning* dan *fine-tuning* pada model pralatih seperti GPT mempengaruhi kualitas parafrase.
2. Aplikasi parafrase teks otomatis dapat membantu penulis, editor, dan profesional konten dalam menghasilkan berbagai versi teks dengan cepat, meningkatkan produktivitas dan efisiensi dalam pembuatan konten.
3. Meningkatkan parafrase yang dapat meningkatkan kualitas terjemahan mesin yang lebih alami dan sesuai konteks.

## Metodologi Penelitian

Penelitian ini dimulai dari November 2024 dan berakhir pada April 2025. Berikut ini tampilan terkait jadwal penelitian dan pembuatan sistem yang akan dilakukan pada penelitian ini sebagai berikut :

1. Jadwal Penelitian

**Tabel 1.1 Tabel Jadwal Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Waktu  Kegiatan | 2024 | | | | | | | | 2025 | | | | | | | | | | | |
| Nov | | | | Des | | | | Jan | | | | Feb | | | | Mar | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Perumusan Masalah |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Preprocessing Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pelatihan Model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Evaluasi Model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji Coba Model |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Merancang Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Bahan dan Alat

Adapun bahan dan alat yang digunakan yaitu berupa *hardware* dan *software* yang digunakan dalam proses penelitian. *Hardware* yang digunakan yaitu laptop dengan spesifikasi:

1. *Processor Intel* *core* i5- 12500H
2. *RAM* 8GB
3. *Nvidia Geforce* RTX 3050 RAM 8 GB DDR6.

*Software* yang digunakan, yaitu:

1. *Kaggle*
2. *Pycharm*
3. *Flask*
4. Me**t**ode Pengumpulan Data

Pengumpulan data yang dilakukan dalam merancang aplikasi ini melakukan metode yakni:

1. Metode Wawancara

Penulis melakukan wawancara kepada pengguna untuk mengetahui kendala ataupun permasalahan yang dihadapi saat menggunakan beberapa aplikasi parafrase online Indonesia.

1. Metode Studi Pustaka

Metode Studi Pustaka dengan melakukan pengumpulan data melalui internet maupun jurnal yang sesuai dengan topik yang dibahas.

## Analisis Sistem

Pada tahap ini, perlu dilakukan identifikasi masalah yang sering dihadapi pengguna parafrase *online* Indonesia melalui wawancara. Setelah data hasil wawancara dikumpulkan, maka dilakukan analisis untuk mengidentifikasi penyebab masalah yang ada.

## Perancangan Sistem

Dari hasil analisis sistem, maka perlu ditetapkan langkah-langkah untuk merancang sistem yang sesuai dengan kebutuhan pengguna. Adapun tahapan dari perancangan sistem adalah sebagai berikut:

1. Pengumpulan Dataset

Dataset diambil dari link [*cahya/gpt2-large-indonesian-522M · Hugging Face*](https://huggingface.co/cahya/gpt2-large-indonesian-522M).

1. *Pre-processing* Data

Langkah-langkah *pre-processing* data umumnya mencakup:

1. Tokenisasi: Memecah teks menjadi unit-unit kecil (token) untuk diproses lebih lanjut.
2. *Cleaning*: Menghilangkan simbol-simbol atau karakter-karakter yang tidak dibutuhkan (seperti tanda baca berlebih, spasi ganda, dll.).
3. Normalisasi: Menyederhanakan teks dengan menyamakan bentuk penulisan kata-kata yang berbeda namun bermakna sama, seperti perubahan kapitalisasi atau singkatan.
4. *Padding*: Menyesuaikan panjang input teks dengan panjang yang konsisten agar dapat diproses dalam *batch* yang sama oleh model.
5. Pelatihan Model

Model yang digunakan adalah GPT-2 versi besar dengan basis 522M parameter dari *cahya/gpt2-large-indonesian-522M*. Untuk melatih model, langkah-langkah pelatihan melibatkan:

1. Pembagian data menjadi data latih dan data uji (*train-test split*).
2. Optimisasi menggunakan algoritma seperti AdamW untuk memperbarui bobot jaringan neural sesuai dengan fungsi *loss*.
3. Tuning *hyperparameter*, seperti *learning rate, batch size*, dan jumlah epoch untuk mencapai hasil pelatihan yang optimal.
4. Transfer learning juga mungkin dilakukan, di mana model pralatih GPT-2 ini dapat disempurnakan (*fine-tuned*) untuk tugas parafrase teks.
5. Evaluasi Model

Metode Evaluasi kinerja model dilakukan dengan menggunakan metrik seperti *BLEU* (*Bilingual Evaluation Understudy*) dan *ROUGE* (*Recall-Oriented Understudy for Gisting Evaluation*) untuk mengukur kesamaan antara hasil parafrase dan teks referensi. Selain itu, uji subjektif juga dilakukan dengan melibatkan pengguna yang mengevaluasi apakah hasil parafrase sesuai dan koheren. Dengan menggunakan pendekatan ini, diharapkan aplikasi parafrase teks otomatis dapat memberikan hasil yang akurat, mudah dipahami, dan membantu pengguna dalam menghasilkan teks yang orisinal.

1. Uji Coba Model

Hasil uji coba model melibatkan penggunaan dataset uji untuk menghasilkan parafrase dan membandingkannya dengan teks referensi. Skor BLEU dan ROUGE dihitung untuk menilai kualitas hasil parafrase secara kuantitatif. Selain itu, dilakukan juga uji subjektif di mana para pengguna mengevaluasi apakah hasil parafrase model tersebut dapat dipahami, alami, dan sesuai dengan konteks.

1. Merancang Aplikasi

Aplikasi web dibangun dengan menggunakan *framework* *Flask* untuk menyediakan antarmuka pengguna guna mengakses model parafrase secara otomatis.

## Sistematika Penulisan

Skripsi ini disusun dengan sistematika sebagai berikut:

**BAB I PENDAHULUAN**

Dalam bab ini, penulis membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, analisis sistem serta perancangan sistem.

**BAB II TINJAUAN PUSTAKA**

Dalam bab ini, penulis memaparkan dan menguraikan teori-teori yang berhubungan dengan judul penulisan skripsi.

**BAB III METODE PENELITIAN**

Dalam bab ini, penulis membahas mengenai analisis dari sistem yang akan dibangun dan rancangan tampilan atau *interface* dari media pembelajaran yang akan dibangun.

**BAB IV HASIL DAN PEMBAHASAN**

Dalam bab ini, penulis memberikan penjelasan mengenai hasil dari media pembelajaran yang sudah selesai dibangun oleh penulis serta penjelasan terhadap hasil dari tampilan media pembelajaran yang sudah selesai dibangun.

**BAB V KESIMPULAN DAN SARAN**

Dalam bab ini, penulis memberikan beberapa kesimpulan serta saran yang berhubungan dengan pembahasan dan perancangan yang telah dilakukan dengan tujuan untuk dapat dijadikan sebagai panduan pada pengembangan berikutnya.

# BAB 2

# TINJAUAN PUSTAKA

### Parafrase Teks

Parafrase teks adalah proses mengungkapkan kembali sebuah kalimat atau paragraf dengan menggunakan kata-kata dan struktur yang berbeda tanpa mengubah makna asli dari teks tersebut. Dalam dunia pemrosesan bahasa alami (NLP), parafrase digunakan untuk menghasilkan teks yang berbeda namun tetap menyampaikan ide yang sama, yang dapat digunakan untuk berbagai tujuan seperti menghindari plagiarisme, memperbaiki kualitas teks, atau menyederhanakan kalimat yang kompleks. Parafrase juga penting dalam aplikasi seperti *chatbots*, mesin pencari, dan sistem tanya-jawab untuk memvariasikan respon serta memberikan interpretasi yang lebih alami[6].

Pada dasarnya, parafrase melibatkan dua aspek utama: penggantian kata atau frasa dengan sinonim, dan perubahan struktur gramatikal tanpa mengubah informasi inti. Parafrase yang efektif harus menjaga konteks dan makna asli tetap konsisten, karena perubahan yang tidak tepat dapat menyebabkan pergeseran makna. Dalam implementasi teknologi, parafrase teks menjadi tantangan yang menarik, terutama ketika melibatkan bahasa yang kompleks atau idiomatis, yang membutuhkan pemahaman lebih dalam terhadap semantik dan sintaksis[7]

Dengan berkembangnya teknologi berbasis kecerdasan buatan, algoritma seperti *Transformer* kini memungkinkan sistem untuk melakukan parafrase secara otomatis dengan hasil yang lebih akurat dan alami. Model-model seperti GPT dan BERT yang menggunakan arsitektur *Transformer* dapat dilatih dengan dataset besar untuk mempelajari variasi bahasa dan menghasilkan teks parafrase secara mandiri. Hal ini memungkinkan berbagai aplikasi seperti alat penulisan otomatis, sistem pendeteksi plagiarisme, dan asisten virtual untuk bekerja lebih efektif dan adaptif terhadap kebutuhan pengguna[8].

### *Deep Learning*

*Deep Learning* adalah sub bidang dari *machine learning* yang menggunakan algoritma berbasis jaringan saraf tiruan (*neural networks*) untuk memodelkan dan memahami kompleksitas data. Konsep dasar dari deep learning adalah untuk meniru cara kerja otak manusia dalam mengolah informasi, yang dikenal sebagai *artificial neural networks*. Struktur dari *neural networks* ini terdiri dari lapisan-lapisan yang disusun berurutan, di mana setiap lapisan terdiri dari sejumlah neuron yang terhubung satu sama lain. Informasi diproses dalam jaringan ini melalui operasi matematis yang terjadi di setiap neuron, yang kemudian mengirimkan hasilnya ke lapisan berikutnya. Dengan cara ini, *deep learning* dapat mengekstrak fitur dan mempelajari pola dalam data secara otomatis dan efektif[9].

Kelebihan utama dari deep learning dibandingkan metode pembelajaran mesin tradisional adalah kemampuannya dalam mengolah volume data yang besar dan kompleks dengan akurasi yang tinggi. Hal ini membuat *deep learning* menjadi sangat efektif untuk aplikasi seperti pengenalan suara, pengolahan bahasa alami, dan pengenalan citra, di mana fitur-fitur penting dari data dapat ditemukan dan digunakan tanpa perlu dirancang dan dipilih oleh manusia. *Deep learning* secara otomatis menyesuaikan dan menyempurnakan fiturnya melalui proses yang disebut *backpropagation*, di mana jaringan menyesuaikan bobotnya berdasarkan kesalahan dalam prediksi yang dilakukan, sehingga meningkatkan kinerja model secara iteratif[10].

Penggunaan *deep learning* telah mengubah banyak sektor industri dan penelitian, memungkinkan kemajuan signifikan dalam bidang kecerdasan buatan. Mesin dengan kemampuan *deep learning* dapat mengidentifikasi pola dan membuat keputusan dengan minim intervensi manusia, mendekati atau bahkan melebihi tingkat akurasi manusia dalam beberapa tugas. Meskipun memerlukan sumber daya komputasi yang besar, peningkatan terus-menerus dalam teknologi *hardware* dan pengembangan algoritma yang lebih efisien terus mendorong batas kemampuan *deep learning*, membuka peluang baru dalam inovasi dan penerapannya dalam berbagai bidang, dari otomotif hingga kesehatan[11].

### *Transformer*

*Transformer* adalah sebuah arsitektur model *deep learning* yang pertama kali diperkenalkan oleh Vaswani et al. pada tahun 2017 melalui makalah berjudul *Attention is All You Need*. *Transformer* dirancang untuk menangani berbagai tugas pemrosesan bahasa alami (NLP) dan telah menjadi dasar dari banyak model canggih seperti GPT dan BERT. Salah satu inovasi kunci dalam *Transformer* adalah penggunaan mekanisme *self-attention*, yang memungkinkan model untuk secara efektif menangkap hubungan antara kata-kata dalam sebuah kalimat, terlepas dari jarak antar kata. Berbeda dengan model RNN atau LSTM yang memproses data secara berurutan, *Transformer* dapat memproses data secara paralel, sehingga lebih efisien dan cocok untuk menangani dataset yang besar[12].

Arsitektur *Transformer* terdiri dari dua komponen utama: *encoder* dan *decoder*. Pada tugas-tugas seperti penerjemahan mesin, encoder bertugas untuk memetakan input (teks sumber) ke dalam representasi yang lebih dalam, sementara decoder menggunakan representasi ini untuk menghasilkan output (teks terjemahan). Setiap lapisan dalam encoder dan decoder terdiri dari dua mekanisme utama: mekanisme *self-attention* dan *feed-forward neural network*. Mekanisme *self-attention* memungkinkan model untuk menimbang hubungan antara setiap kata dalam input dan memahami konteks global dari teks, sedangkan *feed-forward network* bertugas untuk melakukan *transformasi non-linear* pada representasi tersebut[13].

Keunggulan utama *Transformer* dibandingkan dengan model-model sebelumnya terletak pada kemampuannya untuk memproses hubungan jarak jauh dalam teks dan efisiensi komputasinya. Model ini mampu mengatasi masalah vanishing gradient yang sering muncul pada model RNN, serta memaksimalkan penggunaan GPU untuk mempercepat pelatihan. Sejak diperkenalkan, *Transformer* telah diadopsi secara luas untuk berbagai aplikasi NLP seperti penerjemahan mesin, parafrase teks, summarization, dan bahkan pengenalan citra dengan adaptasi arsitektur yang lebih modern. Model-model berbasis *Transformer* telah mendorong batas performa dalam tugas-tugas NLP dan menjadi landasan bagi banyak perkembangan mutakhir dalam bidang kecerdasan buatan[14].

### *Natural Language Processing* (NLP)

*Natural Language Processing* (NLP) adalah cabang dari kecerdasan buatan (AI) yang berfokus pada interaksi antara komputer dan bahasa manusia. Tujuan utama NLP adalah memungkinkan komputer untuk memahami, menganalisis, dan menghasilkan bahasa yang digunakan oleh manusia secara alami. NLP mencakup berbagai teknik dan algoritma untuk memproses teks atau suara, seperti pengenalan suara, analisis sentimen, terjemahan mesin, dan pengenalan entitas bernama (*named entity recognition*). Dengan semakin berkembangnya teknologi AI, NLP telah menjadi komponen penting dalam berbagai aplikasi, termasuk asisten virtual seperti Siri dan Alexa, mesin pencari, serta sistem chatbot[15].

Salah satu tantangan utama dalam NLP adalah kerumitan bahasa manusia itu sendiri. Bahasa memiliki struktur yang fleksibel, ambiguitas, dan nuansa kontekstual yang sering kali sulit untuk dipahami oleh mesin. Oleh karena itu, NLP melibatkan beberapa tahapan pemrosesan seperti analisis morfologis (pembagian kata menjadi unit-unit lebih kecil), analisis sintaksis (pemahaman struktur kalimat), dan analisis semantik (pemahaman makna dari kata dan kalimat). Pendekatan NLP yang lebih tradisional menggunakan aturan berbasis linguistik dan statistik, namun perkembangan dalam *machine learning* dan *deep learning* telah membuka jalan untuk teknik yang lebih canggih dan akurat[16].

Dalam beberapa tahun terakhir, NLP mengalami kemajuan pesat berkat arsitektur deep learning seperti *Transformer*, yang digunakan dalam model-model terkemuka seperti *GPT* dan *BERT*. Model ini memungkinkan komputer untuk memahami konteks kata-kata dalam kalimat dengan lebih baik, bahkan pada teks yang panjang dan kompleks. Selain itu, NLP juga digunakan dalam berbagai aplikasi lain seperti analisis teks, parafrase otomatis, dan sistem tanya-jawab. Dengan perkembangan lebih lanjut, NLP terus membuka peluang untuk mengubah cara kita berinteraksi dengan teknologi dan memanfaatkan bahasa dalam aplikasi yang lebih cerdas dan intuitif.[17].

### GPT (*Generative Pre-trained Transformer*)

GPT (*Generative Pre-trained Transformer*) adalah model bahasa berbasis deep learning yang dikembangkan oleh OpenAI, menggunakan arsitektur *Transformer*. GPT dirancang untuk memahami dan menghasilkan teks yang menyerupai bahasa manusia melalui proses pelatihan yang mendalam pada sejumlah besar data teks. Salah satu fitur utama dari GPT adalah kemampuannya untuk dipra-latih (*pre-trained*) pada dataset besar, sehingga ia dapat memahami berbagai pola dan struktur bahasa. Setelah dilatih, GPT dapat di-*fine-tune* untuk tugas-tugas spesifik seperti terjemahan, penulisan konten, atau dialog *chatbot*. Model ini unggul dalam menghasilkan teks yang koheren dan kontekstual, memberikan solusi yang kuat untuk banyak masalah dalam pemrosesan bahasa alami (NLP)[18].

Keunggulan GPT dibandingkan model sebelumnya terletak pada penggunaan *unsupervised learning*, di mana model mempelajari pola-pola dari teks tanpa anotasi manusia. Ini memungkinkan GPT untuk menghasilkan teks berdasarkan prompt tertentu dengan fleksibilitas yang tinggi, mulai dari menyelesaikan kalimat, menjawab pertanyaan, hingga menghasilkan artikel panjang. Mekanisme *self-attention* dalam *Transformer* yang digunakan oleh GPT memungkinkannya untuk memperhitungkan konteks dari seluruh kalimat atau paragraf, sehingga hasil yang dihasilkan cenderung lebih alami dan relevan dengan input yang diberikan. Versi terbaru, seperti GPT-3, memiliki miliaran parameter yang meningkatkan kemampuannya dalam memahami konteks dan menghasilkan teks yang lebih kompleks[19].

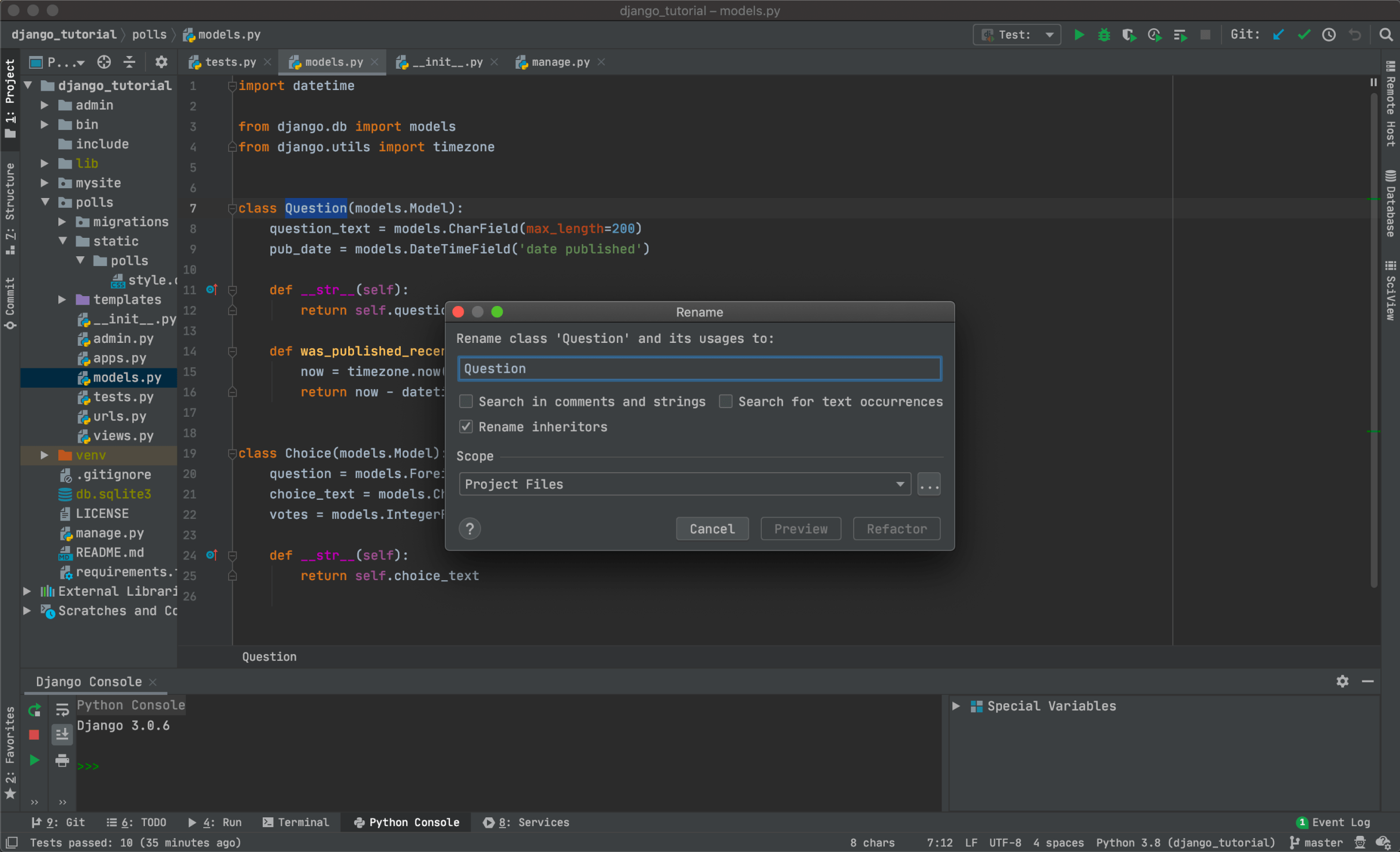
Model GPT telah diaplikasikan di berbagai bidang seperti penulisan otomatis, asistensi kreatif, tanya-jawab, hingga summarization. Selain itu, GPT juga digunakan dalam pembuatan chatbot yang dapat berinteraksi dengan pengguna secara lebih alami. Karena model ini dapat menghasilkan teks dengan kualitas tinggi tanpa pelatihan tambahan yang ekstensif, GPT telah menjadi landasan banyak aplikasi NLP modern. Kemampuannya yang luas, seperti menyelesaikan tugas penulisan teknis hingga kreatif, menjadikannya salah satu inovasi paling signifikan dalam teknologi bahasa di era AI[20].

### Pycharm

PyCharm adalah lingkungan pengembangan terintegrasi (IDE) yang khusus dirancang untuk bahasa pemrograman Python. Dikembangkan oleh JetBrains, PyCharm menawarkan berbagai fitur yang membantu pengembang untuk meningkatkan efisiensi dan produktivitas mereka dalam membuat kode Python. IDE ini menyediakan analisis kode, *debugger* yang canggih, dan dukungan untuk pengembangan web Python dengan *framework* seperti Django dan *Flask*. PyCharm juga mendukung pengembangan untuk beberapa bahasa lain seperti JavaScript, TypeScript, dan HTML, yang membuatnya menjadi pilihan yang ideal untuk pengembangan web dan aplikasi multi-bahasa.

PyCharm memudahkan pengelolaan proyek besar dan kompleks dengan menyediakan alat bantu seperti kontrol versi terintegrasi, penyelesaian kode otomatis, dan refaktorisasi kode. Penyelesaian kode cerdas IDE ini mengenali pola dalam kode Anda, menyediakan saran untuk melengkapi kode dengan cepat dan secara akurat. Fitur refaktorisasi yang canggih memungkinkan pengembang untuk mengorganisir kembali struktur kode mereka tanpa mengubah fungsionalitasnya. Selain itu, PyCharm mendukung pengembangan berbasis test, dengan alat bantu untuk menulis dan menjalankan tes otomatis, yang sangat penting untuk memastikan kualitas aplikasi.

Selain fitur-fitur utama tersebut, PyCharm menawarkan integrasi dengan berbagai tool pengembangan lainnya seperti database SQL, Docker, dan alat pengembangan *front-end*. Dengan adanya dukungan untuk *virtual environments*, manajemen dependensi, dan konfigurasi proyek yang dapat disesuaikan, pengembang dapat bekerja dalam sebuah lingkungan yang sepenuhnya dikontrol. PyCharm tersedia dalam dua edisi: *Community Edition* yang gratis dan *Professional Edition* yang berbayar, yang terakhir menawarkan lebih banyak fitur khusus untuk pengembangan profesional. Ini menjadikan PyCharm salah satu IDE yang paling populer dan dihormati di kalangan pengembang Python, dari pemula hingga profesional.

**Gambar 2.1 Interface PyCharm**

### Penelitian Terdahulu

Hal yang sangat penting dalam sebuah penelitian saat ini adalah dasar atau landasan teori-teori dari hasil penelitian dari peneliti terdahulu dan hasil dari penelitian sebelumnya tersebut dapat dipakai sebagai data pendukung. Data yang dipakai oleh peneliti sebagai data pendukung harus memiliki kaitan dengan masalah-masalah yang akan dibahas oleh peneliti saat ini. Penelitian ini didasarkan pada penelitian terdahulu dengan topik yang memiliki sebuah kesamaan dengan penelitian peneliti saat ini Adapun penelitian terdahulu untuk menjadi rujukan metodologi penelitian ini sebagai berikut.

Otomatisasi Pembentukan Class Diagram dengan Pendekatan Metode Pemrosesan Teks dan Algoritma CombineTF Penelitian ini telah menghasilkan sebuah model yang dapat mengotomatisasi pembentukan class diagram pada AREM. Proses pembentukan class diagram terdiri dari tiga tahapan yaitu pembentukan class diagram dari data kebutuhan pada AREM, dilanjutkan dengan penanganan duplikasi objek pada class diagram, dan proses refinement pada class diagram. Penanganan duplikasi objek dilakukan dengan menerapkan kombinasi algoritma CombineTF dengan Jaro-Winkler. Penanganan duplikasi objek mampu mengatasi 62,5% duplikasi objek dengan nilai precision 0,94 dan nilai akurasi 0,97 untuk nilai threshold algoritma ≥ 0,8. Hasil penanganan objek masih perlu untuk ditingkatkan namun jika dinilai dengan nilai akurasi algoritma diketahui bahwa algoritma telah bekerja dengan sangat baik. Untuk itu diperlukan pendekatan spesifik untuk meningkatkan hasil penanganan duplikasi objek. Metode yang dapat diimplementasikan pada penelitian selanjutnya adalah menggunakan pendekatan deteksi similaritas teks yang dilengkapi dengan kamus persamaan kata. Hal ini dibutuhkan karena sebagian besar kesamaan dua buah teks pada kasus yang diteliti merupakan kesamaan yang bersifat semantik. Untuk mendapatkan hasil yang lebih lengkap dibutuhkan uji coba model pada dataset kebutuhan yang lebih banyak dan perlu dilakukan uji coba penerapan algoritma yang berbeda untuk penanganan duplikasi objek pada class diagram dibutuhkan[4].

Pelatihan Teknik Parafrasa Dan Penggunaan Aplikasi Mendeley Untuk Sitasi Karya Akademik Hasil yang dicapai dalam pelaksanaan pelatihan teknik parafrase dan penggunaan aplikasi Mendeley untuk sitasi ini yaitu:

1. 29 dari 32 peserta menyatakan memahami dengan sangat baik materi pelatihan dan mampu mengaplikasikan teknik parafrase ke dalam sebuah tulisan, sementara sisanya memahami dengan cukup baik meskipun masih terbata-bata dalam pengaplikasian teknik parafrasa.
2. 30 dari 32 peserta mampu memahami dengan sangat baik dan mengaplikasikan teknik sitasi menggunakan aplikasi Mendeley sebagaimana materi yang telah diberikan.
3. Antusias peserta dalam pelatihan sangat tinggi. Hal tersebut terbukti dengan beragam pertanyaan yang diajukan peserta terkait proses-proses dan materi yang dijelaskan pada kegiatan pelatihan. Hal ini menunjukkan bahwa peserta menyimak dengan baik materi dan mampu mempraktikkannya secara langsung dan mandiri.

Pelatihan dilakukan sebagai sebuah upaya yang tidak putus dari tim pengabdi untuk terus merawat kesadaran masyarakat terkait pentingnya apresiasi karya milik orang lain, termasuk karya akademik, serta mensosialisasikan dampak dan bahaya plagiarisme kepada masyarakat. Pelatihan juga diharapkan mampu meningkatkan keterampilan menulis dan sitasi bagi masyarakat dan berbagai pihak yang membutuhkan. Selain itu, pelatihan juga dilaksanakan untuk memberikan pelayanan publik yang prima oleh tim pengabdi. Pada sesi diskusi dan Google Formulir yang disediakan, peserta pelatihan juga menyampaikan bahwa kegiatan pelatihan ini sangat bermanfaat bagi para peserta. Para peserta menyatakan kegiatan pelatihan ini lebih memahamkan mereka terkait teknik parafrasa dan sitasi, terutama menggunakan aplikasi, sebab ternyata terdapat pula peserta pelatihan yang belum mengetahui terkait sitasi sebelum mengikuti pelatihan ini[8].

Pelatihan Penggunaan Alat Parafrase Online Quillbot Sebagai Upaya Peningkatan Kualitas Skripsi Pelatihan penggunaan alat parafrase online Quillbot telah memberikan kontribusi signifikan dalam peningkatan kualitas skripsi mahasiswa. Dengan memanfaatkan teknologi kecerdasan buatan, Quillbot membantu mahasiswa untuk menyusun ulang kalimat dan paragraf secara efektif tanpa mengubah makna aslinya. Hal ini tidak hanya mengurangi risiko plagiarisme tetapi juga meningkatkan kemampuan mahasiswa dalam menyajikan argumen dengan lebih baik. Selain itu, fitur-fitur tambahan seperti Grammar Checker dan Summarizer memberikan nilai tambah dalam meningkatkan keterampilan menulis secara keseluruhan. Pelatihan ini telah berhasil meningkatkan pemahaman mahasiswa tentang pentingnya orisinalitas dalam karya akademik serta mengoptimalkan hasil penulisan skripsi mereka.

Meskipun Quillbot dapat menjadi alat yang sangat bermanfaat, ada tantangan yang perlu diperhatikan, seperti potensi ketergantungan yang berlebihan pada teknologi tersebut. Mahasiswa perlu menggunakan alat ini secara bijak, dengan tetap mempertahankan keterampilan berpikir kritis dan kreatif dalam penulisan akademik. Dengan penggunaan yang tepat, Quillbot mampu membantu mahasiswa menghasilkan karya tulis yang lebih berkualitas dan profesional, namun tanpa mengorbankan integritas akademik dan kemampuan menulis secara mandiri [21].

EFEKTIVITAS WEB PARAFRASE ONLINE DALAM PENYUSUNAN KARYA ILMIAH/LAPORAN MAHASISWA JURUSAN TEKNOLOGI PANGAN Penelitian mengenai efektivitas web parafrase online dalam penyusunan karya ilmiah menunjukkan bahwa aplikasi seperti Quillbot mampu membantu mahasiswa meningkatkan kemampuan menulis akademik dan mengurangi plagiarisme. Web ini memungkinkan pengguna untuk menyusun ulang kalimat dari sumber asli secara efisien, sehingga memperkaya kosakata dan memberikan hasil parafrase yang lebih baik. Namun, meskipun aplikasi ini memudahkan dalam menulis, tetap diperlukan pemahaman mendalam terhadap sumber asli agar hasil parafrase tetap orisinal dan tidak menimbulkan makna yang salah atau rancu.

Penggunaan aplikasi parafrase online sebaiknya dilakukan dengan hati-hati, karena meskipun alat ini efektif dalam meningkatkan kualitas tulisan, mahasiswa tetap harus mengoreksi ulang hasil parafrase, terutama dalam hal tata bahasa dan kejelasan makna. Alat ini hanya menjadi pendukung, bukan pengganti keterampilan analisis dan pemahaman yang baik terhadap teks akademik. Mahasiswa juga disarankan untuk tidak hanya menyalin hasil parafrase mentah, tetapi meninjau kembali apakah hasilnya sesuai dengan standar akademik yang berlaku agar karya ilmiah mereka lebih berkualitas dan bebas dari plagiarisme[7].

Aplikasi pendeteksi plagiarisme menggunakan kombinasi metode TF-IDF dan algoritma Rabin-Karp Berdasarkan hasil yang didapat dari penelitian maka dapat disimpulkan sebagai berikut:

* 1. Kombinasi metode peringkasan teks otomatis TF-IDF dengan aloritam *Rabin-Karp* dapat meningkatkan waktu proses dibandingkan algoritma *Rabin-Karp*. Kecepatan proses dari kombinasi metode peringkasan otomatis TF-IDF dengan algoritama *Rabin-Karp* berbading lurus dengan banyaknya kalimat yang diringkas dengan metode TF-IDF.
  2. Hasil deteksi kemiripan dokumen yang dihasilkan dari kombinasi metode TF-IDF dengan algoritma *Rabin-Karp* mengalami perubahan hasil deteksi dari algoritma aslinya, perubahan hasil yang didapatkan dipengaruhi oleh proses peringkasan dengan metode TF-IDF.
  3. Dari hasil pendeteksian kemiripan dokumen, dokumen yang lebih dahulu diringkas sebanyak 40% memiliki nilai kemiripan yang terbaik dengan memakai situs pendeteksian plagiarisme online [www.tessy.id](http://www.tessy.id) sebagai acuan.

Agar diperoleh hasil yang maksimal, maka perlu dilakukan penelitian lebih lanjut apakah kombinasi metode peringkasan otomatis TF-IDF dengan aloritma *Rabin-Karp* dapat mempercepat waktu proses melebihi metode pendeteksi plagiarisme yang lainnya[22].

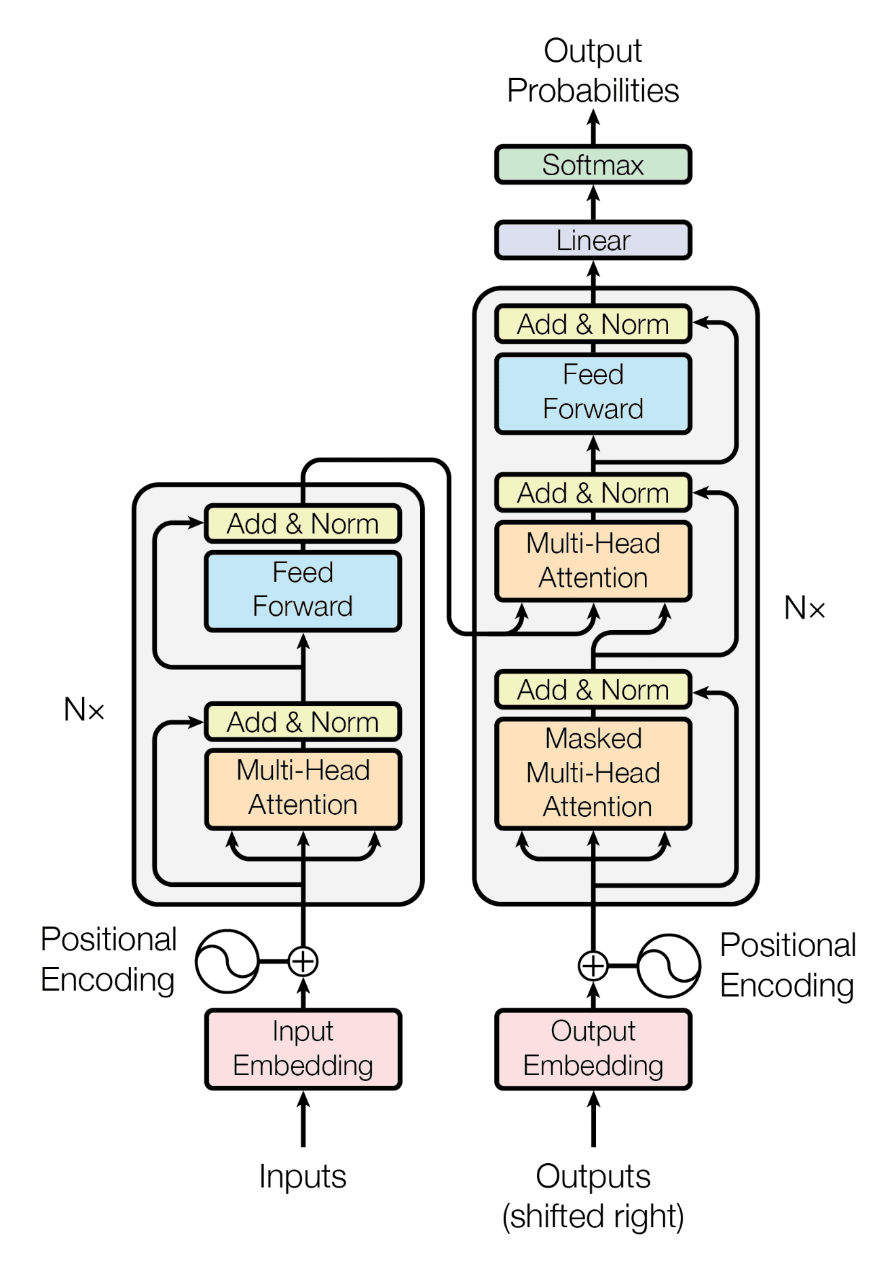
# BAB III

# METODE PENELITIAN

Pada metode penelitian ini, penulis melakukan investigasi mengenai penerapan algoritma *Transformer* dalam pengembangan aplikasi parafrase teks otomatis. *Transformer*, yang diperkenalkan oleh Vaswani et al. pada tahun 2017, telah menjadi model unggulan dalam pemrosesan bahasa alami (*Natural Language Processing* atau *NLP*). Algoritma ini sangat efektif dalam menangani tugas-tugas sekuensial, termasuk parafrase teks, karena kemampuannya untuk memahami konteks dalam skala yang lebih luas dan menghasilkan teks baru yang lebih koheren [12].

Langkah pertama dalam penelitian ini adalah pemilihan dan pemrosesan dataset. Dataset yang digunakan untuk melatih model parafrase harus mencakup berbagai variasi kalimat dan frasa agar model mampu memparafrase teks dengan baik. Dataset yang digunakan adalah korpus paralel yang terdiri dari pasangan kalimat asli dan parafrase dalam bahasa yang sama, seperti dari dataset *Quora Question Pairs* atau *Parafrase Database (PPDB)*. Pemilihan dataset yang representatif sangat penting untuk memastikan bahwa model *Transformer* dapat menggeneralisasi pada berbagai variasi teks dan menghasilkan parafrase yang berkualitas.

Selanjutnya, penulis melakukan desain model dengan menggunakan arsitektur *Transformer*, yang terdiri dari *encoder*-*decoder*. *Encoder* bertugas untuk memahami makna dari teks input, sementara *decoder* menghasilkan teks baru mempertahankan makna asli namun dengan struktur yang berbeda. Proses pelatihan dilakukan menggunakan framework *TensorFlow*. Ilustrasi struktur algoritma *transformer* dapat dilihat pada gambar 3.1 dibawah



**Gambar 3.1 Arsitektur *Transformer***

1. *Encoder*

Bagian kiri gambar adalah *encoder*, yang menerima input berupa urutan kata atau token. Langkah-langkah utama dalam *encoder* adalah sebagai berikut:

1. *Input Embedding*: Token input pertama-tama diubah menjadi representasi vektor (*embedding*) yang dapat diproses oleh model.
2. *Positional Encoding*: Karena *Transformer* tidak memiliki urutan dalam struktur dasar seperti RNN, *positional encoding* digunakan untuk menambahkan informasi posisi token dalam urutan.
3. *Multi-Head Attention*: *Layer* ini memungkinkan model untuk memperhatikan informasi dari berbagai posisi dalam urutan input secara bersamaan, yang penting untuk menangkap hubungan antar-token.
4. *Add & Norm*: Setelah melakukan operasi *multi-head* *attention*, hasilnya dijumlahkan (*Add*) dengan input awal dan dinormalisasi (*Norm*). Ini membantu menjaga kestabilan jaringan selama proses pembelajaran.
5. *Feed Forward*: Setelah *layer attention, layer feed forward* digunakan untuk mengolah informasi lebih lanjut.
6. *Add & Norm* (ulang): Hasil dari *layer feed forward* dijumlahkan dengan *input* awal *layer* tersebut, lalu dinormalisasi lagi.
7. Langkah-langkah ini diulang sebanyak N kali, sesuai jumlah *layer encoder* yang ditetapkan.

2. *Decoder*

Bagian kanan gambar adalah *decoder*, yang menerima hasil dari *encoder* dan menghasilkan *output* berupa urutan kata atau token yang diinginkan (misalnya, dalam tugas penerjemahan, ini adalah urutan dalam bahasa target). *Decoder* memiliki struktur yang mirip dengan *encoder* tetapi dengan beberapa perbedaan penting:

1. *Output Embedding*: Sama seperti pada *encoder*, setiap token pada output juga diubah menjadi *embedding*.
2. *Positional Encoding*: Informasi posisi token dalam urutan *output* juga ditambahkan.
3. *Masked Multi-Head Attention*: Layer ini mirip dengan *multi-head attention* pada *encoder*, tetapi dengan *masking*. *Masking* ini mencegah model melihat token-token berikutnya dalam urutan *output*, memastikan bahwa prediksi token tertentu hanya dipengaruhi oleh token sebelumnya, bukan token berikutnya.
4. *Multi-Head Attention*: Layer ini menerima *output* dari *encoder* sebagai konteks tambahan, yang membantu *decoder* mempelajari hubungan antara *input* dan *output*.
5. *Add & Norm* dan *Feed Forward*: Langkah-langkah ini mirip dengan yang ada di *encoder*.
6. Langkah-langkah ini juga diulang sebanyak N kali, sesuai jumlah *layer* *decoder*.

3. *Output*

Setelah melalui semua layer di *decoder*, hasil akhir dikirim ke *layer* linier yang mengubah output menjadi skor probabilitas untuk setiap kata di dalam kosakata. *Softmax* digunakan untuk mengubah skor ini menjadi probabilitas yang dapat ditafsirkan sebagai prediksi kata berikutnya dalam urutan output. Secara keseluruhan, *Transformer* menggunakan perhatian multi-kepala (*multi-head attention*) untuk menangkap hubungan kompleks antar-token dalam urutan *input* dan *output*, dengan efisiensi yang tinggi. Arsitektur ini telah menjadi dasar banyak model bahasa canggih, seperti BERT dan GPT.

Evaluasi kinerja model dilakukan dengan menggunakan metrik seperti *BLEU* (*Bilingual Evaluation Understudy*) dan *ROUGE* (*Recall-Oriented Understudy for Gisting Evaluation*) untuk mengukur kesamaan antara hasil parafrase dan teks referensi. Selain itu, uji subjektif juga dilakukan dengan melibatkan pengguna yang mengevaluasi apakah hasil parafrase sesuai dan koheren. Dengan menggunakan pendekatan ini, diharapkan aplikasi parafrase teks otomatis dapat memberikan hasil yang akurat, mudah dipahami, dan membantu pengguna dalam menghasilkan teks yang orisinal.

#### Analisis Kondisi Saat Ini

Dalam beberapa tahun terakhir, teknologi pemrosesan bahasa alami *Natural Language Processing* (NLP) telah berkembang pesat, khususnya melalui penggunaan model-model berbasis *Transformer*. Namun, aplikasi parafrase teks otomatis dalam Bahasa Indonesia masih memerlukan peningkatan akurasi dan kemudahan penggunaan. Model-model sebelumnya, seperti RNN dan LSTM, memiliki keterbatasan dalam memahami konteks teks yang lebih panjang, sehingga menghasilkan parafrase yang kurang natural. Hal ini mendorong kebutuhan akan metode yang lebih efektif, seperti model *Transformer* yang dikembangkan pada penelitian ini.

#### Analisis Metode

Pada saat ini, penggunaan aplikasi berbasis web untuk memproses teks secara otomatis telah menjadi pilihan utama karena aksesibilitasnya yang tinggi dan kemudahan penggunaannya. Namun, aplikasi parafrase teks otomatis khususnya dalam Bahasa Indonesia masih terbatas dan sering kali memerlukan peningkatan dalam hal akurasi dan pemahaman konteks. Penggunaan arsitektur *Transformer* seperti *IndoT5-base-parafrase* merupakan solusi potensial untuk menghasilkan parafrase teks yang lebih akurat dan natural dalam platform berbasis web.

Model yang digunakan adalah *IndoT5-base-parafrase*, yang tersedia di *platform Hugging Face*. Model ini didasarkan pada arsitektur *Transformer* dengan kemampuan untuk memahami dan menghasilkan teks parafrase yang koheren. Dataset yang digunakan terdiri dari korpus paralel yang memuat pasangan kalimat asli dan parafrase dalam Bahasa Indonesia. Dataset ini dipilih untuk melatih model agar mampu menghasilkan parafrase yang bermakna dan beragam. Persiapan data melibatkan tokenisasi teks input menggunakan *AutoTokenizer* agar sesuai dengan format yang dibutuhkan oleh model *IndoT5-base-parafrase*.

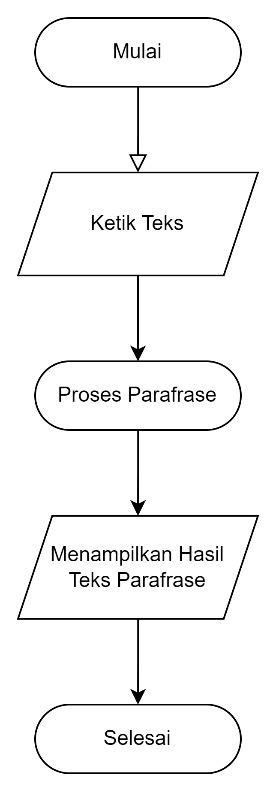
#### Analisis Sistem Usulan

Sistem usulan ini memungkinkan pengguna untuk melakukan parafrase teks secara otomatis melalui website. Pengguna cukup memasukkan teks ke dalam antarmuka web, dan sistem akan memproses teks tersebut menggunakan model *IndoT5-base-parafrase* untuk menghasilkan beberapa versi parafrase yang berbeda. Sistem ini menyediakan solusi yang lebih praktis bagi pengguna dibandingkan dengan aplikasi desktop atau perangkat lunak lainnya, karena dapat diakses dari perangkat apa pun dengan koneksi internet.

#### Perancangan Sistem

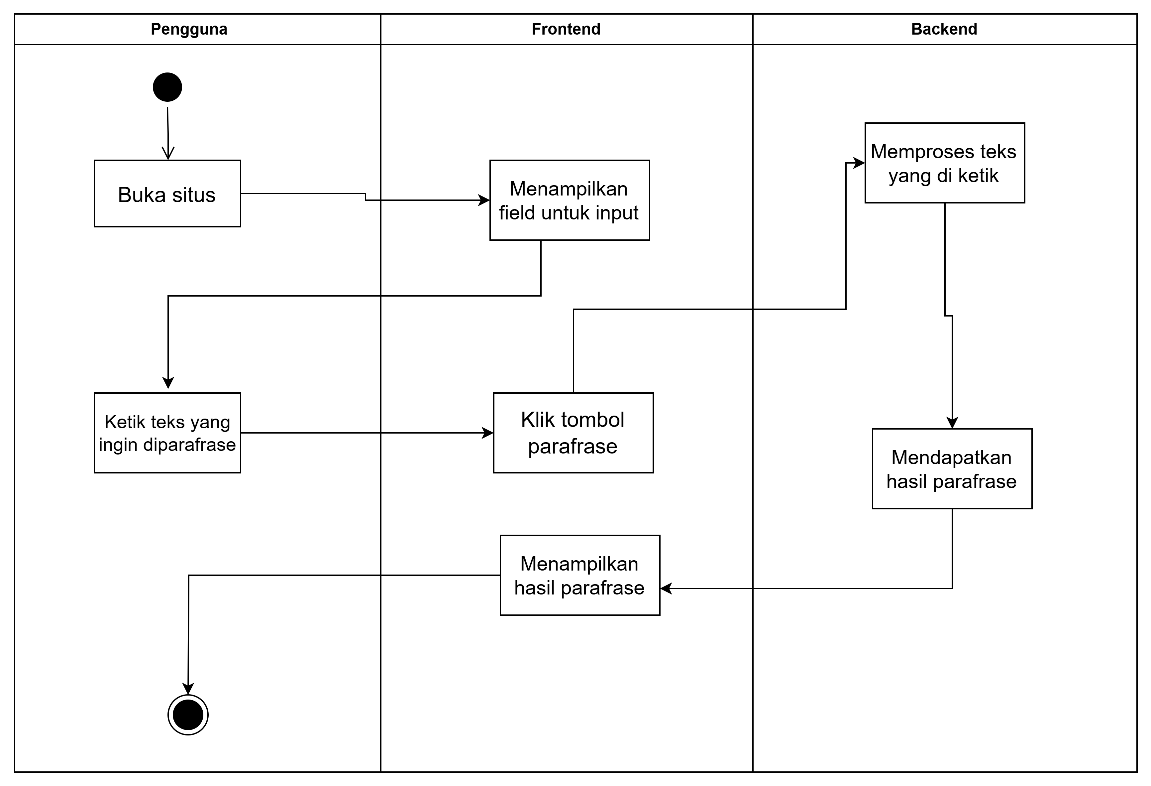
1. **Arsitektur Sistem**:
   1. Aplikasi web terdiri dari dua bagian utama: *frontend* (antarmuka pengguna) dan *backend* (server aplikasi).
   2. *Frontend* menerima *input* dari pengguna dan menampilkan hasil parafrase.
   3. *Backend* memproses input menggunakan model *IndoT5-base-parafrase* dan mengirim hasilnya kembali ke *frontend*.
2. **Komponen Utama Sistem**:
   1. *Frontend*: Dibuat dengan *HTML*, *CSS*, dan *JavaScript*.
   2. *Backend*: Dikembangkan dengan menggunakan *Flask*. Mengintegrasikan model *IndoT5-base-parafrase* untuk memproses teks yang diterima.
   3. *Model Processing*: Menggunakan pustaka *Transformer* dari *Hugging Face* untuk memproses teks input dan menghasilkan parafrase.
3. **Desain Alur Kerja**

a. *Flowchart*

*******Flowchart* ini menggambarkan proses dari input pengguna hingga keluarnya hasil parafrase:

**Gambar 3.2 Flowchart Parafrase**

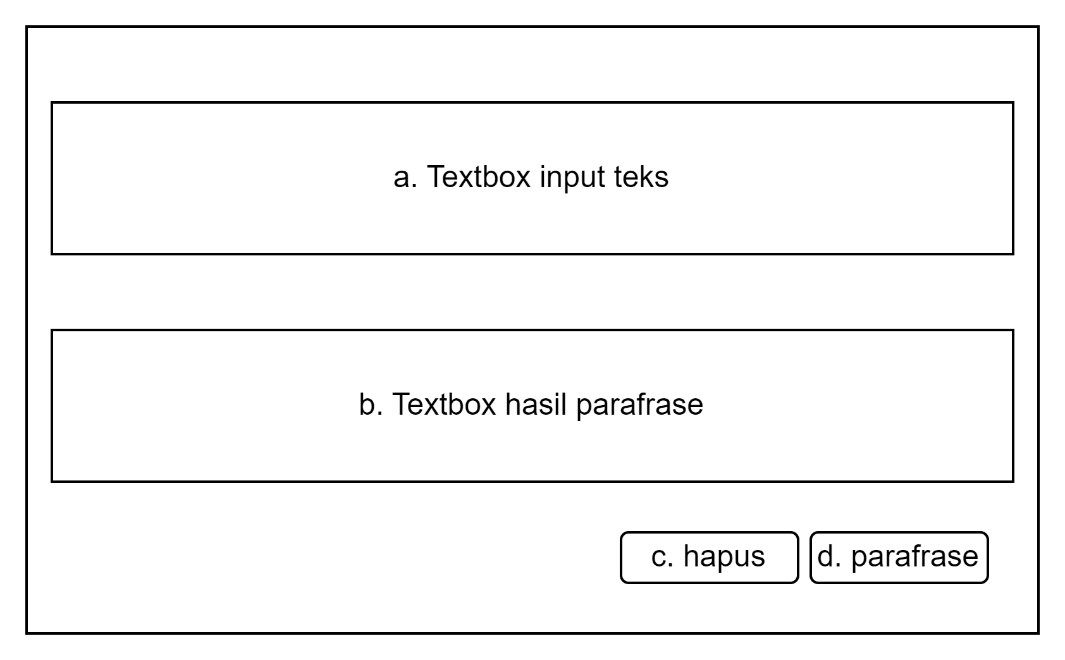
1. *Activity Diagram*

*Activity diagram* ini menggambarkan interaksi antara pengguna, frontend, dan backend:

**Gambar 3.3 *Activity Diagram***

* 1. Pengguna memasukkan teks yang akan diparafrase melalui *form* di halaman web.
  2. Teks dikirim ke *backend* melalui *API*.
  3. *Backend* melakukan tokenisasi dan pemrosesan teks menggunakan model *IndoT5-base-parafrase* dan menghasilkan hasil parafrase.
  4. Hasil parafrase dikirim kembali ke *frontend* dan ditampilkan kepada pengguna.

1. **Desain Tampilan Antarmuka**

Tampilan utama adalah tampilan yang pertama kali muncul ketika situs diakses, pada tampilan utama terdapat beberapa tampilan seperti terlihat pada Gambar 3.4 dibawah ini.

**Gambar 3.4 Tampilan Utama**

Keterangan:

a. *Textbox* yang disediakan untuk mengisi teks yang ingin di parafrase.

b. Hasil parafrase akan muncul di *textbox* ini.

c. Menghapus seluruh teks yang telah diketik.

d. Tombol yang berfungsi untuk konfirmasi parafrase dari *texbox* a.

# DAFTAR PUSTAKA

[1] I. N. Purnama and N. N. Widya Utami, “Implementasi Peringkas Dokumen Berbahasa Indonesia Menggunakan Metode Text To Text Transfer Transformer (T5),” *J. Teknol. Inf. dan Komput.*, vol. 9, p. 4, 2023.

[2] A. Wahyudi, A. Pamuji, and R. Martuti, “Pelatihan Peningkatan Kemampuan Parafrasa Mahasiswa Program Studi Ilmu Keperawatan STIK Bina Husada Palembang Dengan Menggunakan Sinonim Dan Mengubah Struktur Teks,” *CARE J. Pengabdi. Multi Disiplin*, vol. 1, no. 2, pp. 30–36, 2023, doi: 10.31004/care.v1i2.15402.

[3] K. Nisa, E. Syafitri, S. R. D. Saragih, Y. Aryni, and E. Rahmadani, “Penggunaan Aplikasi Parafrasa untuk Menurunkan Plagiarisme pada Mahasiswa FKIP Universitas Asahan dalam Menyelesaikan Skripsi,” *J. War. Pengabdi. Andalas*, vol. 29, no. 1, pp. 55–60, 2022, doi: 10.25077/jwa.29.1.55-60.2022.

[4] R. Delima, A. Rachmat, and C. #2, “Otomatisasi Pembentukan Class Diagram dengan Pendekatan Metode Pemrosesan Teks dan Algoritma CombineTF,” *J. Edukasi dan Penelit. Infomatika ( JEPIN )*, vol. 10, no. 1, pp. 120–127, 2024.

[5] R. Mas, R. W. Panca, K. Atmaja1, and W. Yustanti2, “Analisis Sentimen Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from Transformers),” *Jeisbi*, vol. 02, no. 3, pp. 55–62, 2021.

[6] H. Mardiana, Y. Chandra, and A. Arnelli, “Pemberdayaan Penelitian: Pentingnya Parafrase Dalam Penyusunan Artikel Jurnal Ilmiah,” *J. Pengabdi. Kolaborasi dan Inov. IPTEKS*, vol. 2, no. 4, pp. 1109–1118, 2024, doi: 10.59407/jpki2.v2i4.996.

[7] C. A. RIZQY, V. R. H. P. , NADIYA R. ALI, A. A. Salsabila, and N. D. A. , Dwi A. Putri, “EFEKTIVITAS WEB PARAFRASE ONLINE DALAM PENYUSUNAN KARYA ILMIAH/LAPORAN MAHASISWA JURUSAN TEKNOLOGI PANGAN,” *Argopuro J. Multidisiplin Ilmu Bhs.*, vol. 1, no. 4, pp. 1–14, 2023, [Online]. Available: https://ejournal.warunayama.org/index.php/argopuro/article/view/1107.

[8] S. R. Wulandari, D. W. Hamka, R. Kurniawan, and ..., “Pelatihan Teknik Parafrasa Dan Penggunaan Aplikasi Mendeley Untuk Sitasi Karya Akademik,” *J. Pengabdi. …*, vol. 2, no. 2, pp. 47–51, 2024, [Online]. Available: https://mand-ycmm.org/index.php/jpmm/article/view/693%0Ahttps://mand-ycmm.org/index.php/jpmm/article/download/693/749.

[9] C. Angga Marcelio *et al.*, “Aplikasi Analisis Wajah, Klasifikasi Gender dan Prediksi Usia Menggunakan Deep Learning pada Dataset Citra Wajah Manusia P-Issn,” *J. Media Infotama*, vol. 20, no. 1, pp. 2–6, 2024.

[10] V. Karenina, M. F. Erinsyah, and D. S. Wibowo, “Klasifikasi Rentang Usia Dan Gender Dengan Deteksi Suara Menggunakan Metode Deep Learning Algoritma Cnn (Convolutional Neural Network),” *Komputika J. Sist. Komput.*, vol. 12, no. 2, pp. 75–82, 2023, doi: 10.34010/komputika.v12i2.10516.

[11] M. Ali, A. Diwan, and D. Kumar, “Attendance System Optimization through Deep Learning Face Recognition,” *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1527–1540, 2024, doi: 10.12785/ijcds/1501108.

[12] I. Ashish Vaswani, Ariful *et al.*, “Attention Is All You Need,” *Int. Conf. Inf. Knowl. Manag. Proc.*, no. Nips, pp. 4752–4758, 2023, doi: 10.1145/3583780.3615497.

[13] A. D. Rendragraha, M. A. Bijaksana, and A. Romadhony, “Pendekatan Metode Transformers untuk Deteksi Bahasa Kasar dalam Komentar Berita Online Indonesia,” *e-Proceeding Eng.*, vol. 8, no. 2, pp. 3385–3395, 2021.

[14] J. U. S. Lazuardi and A. Juarna, “Analisis Sentimen Ulasan Pengguna Aplikasi Joox Pada Android Menggunakan Metode Bidirectional Encoder Representation From Transformer (Bert),” *J. Ilm. Inform. Komput.*, vol. 28, no. 3, pp. 251–260, 2023, doi: 10.35760/ik.2023.v28i3.10090.

[15] A. Mukti, A. D. Hadiyanti, A. Nurlaela, and J. Panjaitan, “Sistem Analisa Sentiment Bakal Calon Presiden 2024 Menggunakan Metode NLP Berbasis Web,” *Jurikom*, vol. 6, no. 1, p. p-ISSN, 2023.

[16] M. Amien, “SEJARAH DAN PERKEMBANGAN TEKNIK NATURAL LANGUAGE PROCESSING (NLP) BAHASA INDONESIA: TINJAUAN TENTANG SEJARAH, PERKEMBANGAN TEKNOLOGI, DAN APLIKASI NLP DALAM BAHASA INDONESIA,” pp. 1–7, 2022.

[17] R. Bangun, C. Informasi, and L. Pekerjaan, “Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP ( Natural Language Processing ),” *BRILIANT J. Ris. dan Konseptual*, vol. 5, no. 1, pp. 619–626, 2020.

[18] I. Pendidikan, “PERAN GURU DAN SISWA DALAM MENGGUNAAN CHAT GPT (GENERATIVE PRE-TRAINING TRANSFORMER) DALAM IMPLEMENTASI PENDIDIKAN,” *KRIDA CENDEKIA*, vol. 3, no. 1, 2024.

[19] A. Aruna, E. P. Surya, A. R. Prasetyo, G. A. Wijaya, and U. N. Malang, “Integrasi Text-to-Image dan Generative Pre-Trained Transformer dalam Pengembangan Wisata Edukasi Ramah Anak Usia Dini Alby,” *VISA J. VISIONS IDEAS*, vol. 4, no. 3, pp. 2186–2198, 2024.

[20] D. Setiawan, “Peran Chat Gpt ( Generative Pre-Training Transformer ) Dalam Implementasi Ditinjau Dari Dataset,” *Innov. J. Soc. Sci. Res.*, vol. 3, no. 3, pp. 9527–9539, 2023.

[21] M. Yusnan and A. Rezki, “Pelatihan Penggunaan Alat Parafrase Online Quillbot Sebagai Upaya Peningkatan Kualitas Skripsi,” *Termasyhur J. Pengabdi. …*, vol. 2, no. 2, pp. 35–43, 2024, [Online]. Available: http://www.jurnal-umbuton.ac.id/index.php/termasyhur/article/view/5978.

[22] A. Dalam, R.- Karp, and M. Tf-idf, “Aplikasi pendeteksi plagiarisme menggunakan kombinasi metode tf-idf dan algoritma rabin-karp,” pp. 1–10, 2020.