

**SISTEM RINGKASAN OTOMATIS UNTUK KAJIAN LITERATUR
MENGUNAKAN METODE *BIDIRECTIONAL AND AUTO-REGRESIVE
TRANSFORMERS (BART)***

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



Disusun Oleh :

RIKA AMELIA

NIM 32602100112

**PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT
AUTOMATIC SUMMARY SYSTEM FOR LITERATURE REVIEW
USING BIDIRECTIONAL AND AUTO-REGRESSIVE TRANSFORMERS
(BART) METHOD

Proposed to complete the requirement to obtain a bachelor's degree (S1) at
Informatics Engineering Departement of Industrial Technology Faculty Sultan
Agung Islamic University



Arranged By:

RIKA AMELIA

NIM 32602100112

PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

LEMBAR PENGESAHAN
TUGAS AKHIR

SISTEM RINGKASAN OTOMATIS UNTUK KAJIAN LITERATUR
MENGUNAKAN METODE BIDIRECTIONAL AND AUTO-
REGRESSIVE TRANSFORMER (BART)

RIKA AMELIA
NIM 32602100112

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 17 Februari 2025

TIM PENGUJI UJIAN SARJANA :

Bagus Satrio W.P, S.Kom. M.Cs
NIK. 210616051

(Ketua Penguji)



20-02-2025

Ghufron, ST, M.Kom
NIK. 210622056

(Anggota Penguji)



29-02-2025

Badie'ah, ST, M.Kom
NIK. 210615044

(Pembimbing)

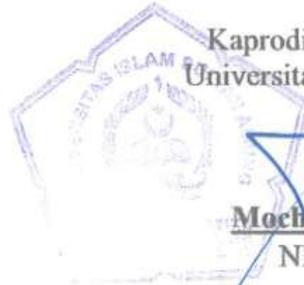


20-02-2025

Semarang, 24 Februari 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung



Moch Taufik, ST, MIT
NIK. 210604034



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Rika Amelia

NIM : 32602100112

Judul Tugas Akhir : SISTEM RINGKASAN OTOMATIS UNTUK KAJIAN LITERATUR MENGGUNAKAN METODE BIDIRECTIONAL AND AUTO-REGRESIVE TRANSFORMERS (BART)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 6 Maret 2024

Yang Menyatakan,



Rika Amelia

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Rika Amelia

NIM : 32602100112

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : SISTEM RINGKASAN OTOMATIS UNTUK KAJIAN LITERATUR MENGGUNAKAN METODE BIDIRECTIONAL AND AUTO-REGRESIVE TRANSFORMERS (BART)

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 6 Maret 2025

Yang menyatakan,



Rika Amelia

KATA PENGANTAR

Dengan penuh rasa syukur kepada Allah SWT atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir yang berjudul “Sistem Ringkasan Otomatis untuk Kajian Literatur Menggunakan Metode Bidirectional and Auto-Regressive Transformers (BART).” Tugas Akhir ini disusun sebagai salah satu syarat dalam menyelesaikan studi dan memperoleh gelar Sarjana (S-1) pada Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan diselesaikan dengan dukungan dari berbagai pihak, baik dalam bentuk materi maupun teknis. Oleh karena itu, saya sebagai penulis ingin menyampaikan rasa terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, ST.,M.T
3. Dosen pembimbing penulis Badie'ah, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan penuh kerendahan hati, penulis menyadari bahwa laporan ini masih memiliki berbagai kekurangan, baik dari segi kualitas, kuantitas, maupun aspek keilmuan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang konstruktif guna menyempurnakan laporan ini di masa mendatang.

Semarang, 14 Februari 2025



Rika Amelia

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
ABSTRAK	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Tinjauan Pustaka	4
2.2 Dasar Teori.....	5
2.2.1 <i>Automatic Text Summarization</i>	5
2.2.2 Natural Language Processing (NLP)	6
2.3 <i>Transformers</i>	6
2.3.1 Arsitektur <i>Transformers</i>	7

2.3.2 Bidirectional <i>and</i> Auto-Regressive Transformers (BART).....	11
2.3.3 IMRAD	12
2.3.4 PDF Extraction.....	12
2.3.5 BERTScore	12
BAB III METODE PENELITIAN	14
3.1 Metodologi Penelitian	14
3.1.1 Studi Literatur	14
3.1.2 Pengumpulan Data	14
3.1.3 Pemodelan sistem.....	14
3.2 Analisis Sistem.....	15
3.3 Evaluasi.....	16
3.4 Analisis Kebutuhan.....	17
3.5 Perancangan <i>User Interface</i>	19
3.5.1 Perancangan Halaman awal	19
3.5.2 Perancangan Halaman <i>Sidebar</i> Daftar Folder.....	19
3.5.3 Perancangan Halaman <i>Upload File</i>	20
3.5.4 Perancangan Halaman Hasil Ringkasan.....	21
BAB IV HASIL DAN ANALISIS PENELITIAN	22
4.1 Hasil Penelitian	22
4.1.1 Persiapan Data.....	22
4.1.2 Pemodelan Sistem	23
4.2 Hasil Evaluasi.....	29
4.3 Hasil Perancangan <i>User Interface</i>	30
4.4 Hasil Uji Coba Sistem.....	32
BAB V KESIMPULAN DAN SARAN	34

5.1 Kesimpulan	34
5.2 Saran.....	34
DAFTAR PUSTAKA	35



DAFTAR GAMBAR

Gambar 2. 1 Model Arsitektur <i>Transforemer</i> (Mohiuddin dkk., 2023).....	7
Gambar 2. 2 (Kiri) <i>Scaled Dot-Product Attention</i> . (Kanan) <i>Multi-Head Attention</i> (Mohiuddin dkk., 2023)	8
Gambar 3. 1 <i>Flowchart</i> Metode Penelitian	14
Gambar 3. 2 <i>Flowchart</i> Pemodelan Sistem.....	15
Gambar 3. 3 <i>Flowchart</i> Sistem Ringkasan	16
Gambar 3. 4 Perancangan Halaman Awal	19
Gambar 3. 5 Perancangan Halaman <i>Sidebar</i> Daftar <i>Folder</i>	20
Gambar 3. 6 Perancangan Halaman <i>Upload File</i>	20
Gambar 3. 7 Perancangan Halaman Hasil Ringkasan.....	21
Gambar 4. 1 Memeriksa Info <i>Dataset</i>	22
Gambar 4. 9 Halaman Awal.....	30
Gambar 4. 10 Halaman <i>Sidebar</i> Daftar <i>Folder</i>	30
Gambar 4. 11 Halaman <i>Upload File</i>	31
Gambar 4. 12 Halaman Hasil Ringkasan	32



DAFTAR TABEL

Tabel 4. 1 Hasil Evaluasi Model	24
Tabel 4. 2 Hasil Evaluasi Sistem.....	29
Tabel 4. 3 Hasil Pengujian Sistem	32



ABSTRAK

Pada dunia akademik dan riset, kegiatan kajian literatur merupakan salah satu tahapan yang sangat penting dalam proses penelitian. Melalui kajian literatur, peneliti dapat memahami perkembangan penelitian sebelumnya, menemukan celah penelitian (research gap), serta memperoleh landasan teoritis yang kuat untuk mendukung penelitian yang akan dilakukan. Dalam melakukan kajian literatur, peneliti sering kali harus membaca dan menganalisis banyak jurnal dari berbagai sumber yang berbeda. Setiap jurnal memiliki struktur penulisan yang berbeda-beda, tergantung pada gaya penulisan, metodologi penelitian, serta kebijakan dari penerbit jurnal tersebut. Banyaknya artikel yang harus dipahami membuat proses kajian literatur menjadi sangat kompleks dan memakan waktu yang cukup lama. Oleh karena itu, diperlukan suatu sistem yang dapat membantu mempermudah proses tersebut dengan lebih efisien dan efektif. Salah satu pendekatan yang dapat digunakan untuk mempercepat dan mempermudah proses kajian literatur adalah dengan menerapkan teknik peringkasan otomatis. Dalam penelitian ini, digunakan Bidirectional and Auto-Regressive Transformers (BART), yaitu model berbasis deep learning yang telah terbukti efektif dalam tugas-tugas pemrosesan bahasa alami (Natural Language Processing/NLP), termasuk peringkasan teks. BART mampu menangkap informasi penting dalam teks dan menyajikannya dalam bentuk yang lebih ringkas tanpa menghilangkan esensi utama dari isi jurnal yang diringkas. Dengan dikembangkannya sistem berbasis BART untuk peringkasan otomatis ini, diharapkan para peneliti dapat lebih mudah dalam melakukan kajian literatur, menghemat waktu, serta meningkatkan efisiensi dalam memahami banyak artikel yang relevan dengan penelitian mereka. Selain itu, sistem ini juga dapat digunakan dalam berbagai bidang penelitian, mulai dari ilmu sosial hingga sains dan teknologi, sehingga memberikan manfaat yang luas bagi komunitas akademik.

Keyword : Ringkasan otomatis, BART, IMRAD

ABSTRACT

In the academic and research world, literature review activities are one of the most important stages in the research process. Through a literature review, researchers can understand the development of previous studies, identify research gaps, and obtain a strong theoretical foundation to support the research to be conducted. In conducting a literature review, researchers often have to read and analyze many journals from various different sources. Each journal has a different writing structure, depending on the writing style, research methodology, and the policies of the journal's publisher. The large number of articles that need to be understood makes the literature review process very complex and time-consuming. Therefore, a system is needed that can help simplify the process more efficiently and effectively. One of the approaches that can be used to accelerate and simplify the literature review process is by applying automatic summarization techniques. In this study, Bidirectional and Auto-Regressive Transformers (BART) are used, which is a deep learning-based model that has proven effective in natural language processing (NLP) tasks, including text summarization. BART is capable of capturing important information in the text and presenting it in a more concise form without losing the main essence of the summarized journal content. With the development of the BART-based system for automatic summarization, it is hoped that researchers can more easily conduct literature reviews, save time, and increase efficiency in understanding many articles relevant to their research. In addition, this system can also be used in various fields of research, from social sciences to science and technology, thereby providing broad benefits to the academic community.

Keyword : Automatic Summarization, BART, IMRAD

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada dunia akademik dan riset, kegiatan kajian literatur adalah salah satu tahapan penting dalam proses penelitian. Secara umum, kajian literatur didefinisikan sebagai ringkasan yang didapatkan dari suatu sumber bacaan yang berkaitan dengan bahasan penelitian (Ridwan dkk., 2021). Seorang peneliti harus memahami dan merangkum berbagai sumber informasi seperti jurnal, artikel ilmiah, dan laporan penelitian untuk membuat landasan teori dan menambah wawasan terkait topik penelitian yang akan dikaji. Namun, jumlah literatur ilmiah yang tersedia semakin banyak seiring dengan pesatnya perkembangan teknologi dan pengetahuan. Hal ini membuat tantangan tersendiri bagi para peneliti, terutama terkait dengan waktu dan sumber daya yang dibutuhkan untuk membaca dan menganalisis dokumen-dokumen tersebut secara menyeluruh.

Disini peran teknologi komputasi dan kecerdasan buatan menjadi semakin dibutuhkan. Salah satu solusi yang dapat dilakukan adalah membuat sebuah sistem ringkasan dari banyak dokumen ilmiah dengan cepat dan efisien. Dalam dunia komputer peringkasan teks dikenal dengan *Automatic Summarization Text* atau peringkasan teks otomatis. Sistem *Text summarize* adalah teknologi yang digunakan untuk meringkas teks Panjang menjadi lebih ringkas dengan tetap mempertahankan informasi penting dari teks aslinya. Sistem ini diharapkan dapat meringankan peneliti dalam proses kajian literatur, memfasilitasi pengambilan keputusan, dan meningkatkan produktivitas penelitian (Magriyanti, 2021).

Dalam melakukan kajian literatur banyak jurnal dari berbagai sumber yang berbeda. Sumber yang berbeda menjadikan struktur dari masing-masing artikel jurnal juga berbeda. Namun, pada hakikatnya semua artikel menganut sistem penulisan menggunakan IMRAD sebagai acuannya, hanya saja dalam penulisannya menggunakan kata yang berbeda. IMRAD (*Introduction, Method, Result, and Discussion*) adalah struktur yang digunakan dalam penulisan ilmiah dan dianggap sebagai standar pengorganisasian teks akademik (Pangesti dkk., 2023).

IMRAD memiliki peran yang penting dalam melakukan kajian literatur. Hal ini dikarenakan IMRAD memberikan kerangka yang logis untuk membuat ringkasan dalam kajian literatur. Dengan mengenali bagian-bagian penting dari setiap elemen maka ringkasan yang dihasilkan akan lebih relevan, terorganisir, dan sesuai dengan kebutuhan (Özkan Çelik dan Al, 2024).

Ringkasan yang logis dan mudah dipahami biasanya menggunakan teks yang baru dan konseptual dalam arti memiliki pemahaman yang mendalam dan lebih ringkas dan jelas. Dalam hal ini diperlukan arsitektur yang dirancang untuk generasi teks, BART adalah salah satunya. BART adalah jenis *transforemer* yang memiliki *encoder* dua arah dan *decoder* Autoregresif. BART memiliki kelebihan dalam memahami instruksi, menghasilkan respon mirip manusia, dan beradaptasi terhadap NLP untuk membuat teks dan peringkasan (Yadav dkk., 2024).

Walaupun telah ada beberapa penelitian terkait ringkasan otomatis, Sebagian besar masih menghadapi keterbatasan dalam hal akurasi, relevansi, dan keterkaitan dengan konteks dokumen asli. Oleh sebab itu, penelitian ini bertujuan untuk mengimplementasikan metode BART dalam pengembangan sistem ringkasan otomatis untuk kajian literatur. Dengan sistem ini, diharapkan dapat membantu para peneliti dalam proses kajian literatur ilmiah.

1.2 Perumusan Masalah

Rumusan masalah pada penelitian ini adalah :

1. Bagaimana mengidentifikasi struktur artikel ilmiah terindeks garuda berformat IMRAD (*Introduction, Methods, Result, and Discussion*)
2. Bagaimana model *Transformers* BART dapat diaplikasikan untuk membuat ringkasan otomatis dari teks artikel ilmiah?

1.3 Pembatasan Masalah

Dalam penelitian ini ada beberapa batasan masalah yang diterapkan. Berikut adalah Batasan masalah dalam tugas akhir ini :

1. Ringkasan menggunakan artikel ilmiah terindeks garuda berbahasa inggris
2. Sistem hanya menerima *input file* berformat pdf

3. Sistem menampilkan *output* dalam bentuk tabel yang berisi metadata artikel dan ringkasan dari identifikasi IMRAD
4. Sistem ini hanya menggunakan *input* single dokumen

1.4 Tujuan

Penelitian ini bertujuan untuk membangun sistem ringkasan otomatis untuk membantu dalam melakukan kajian literatur.

1.5 Manfaat

Manfaat dari penelitian ini adalah untuk memudahkan para peneliti dalam melakukan kajian literatur.

1.6 Sistematika penulisan

BAB I PENDAHULUAN

Pada bab pendahuluan menyajikan permasalahan yang dibahas pada penelitian ini yaitu latar belakang, rumusan masalah, Batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini disajikan informasi mengenai tinjauan Pustaka dari penrelitian sebelumnya dan dasar teori yang digunakan.

BAB III METODE DAN PERANCANGAN SISTEM

Pada bab ini disajikan tentang metologi penelitian yang digunakan dan rancangan sistem yang akan dibuat.

BAB IV HASIL DAN ANALISA

Pada bab ini akan disajikan tentang hasil dan Analisa dari sistem yang telah dibuat.

BAB V KESIMPULAN DAN SARAN

Pada bab ini disajikan tentang kesimpulan dan saran dari seluruh rangkaian penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Kajian literatur merupakan proses penelusuran dan penelitian terhadap berbagai sumber kepustakaan, seperti buku, jurnal, dan publikasi lainnya yang relevan dengan topik penelitian. Tujuan dari kajian ini adalah untuk menghasilkan tulisan yang membahas suatu topik atau isu tertentu. Dalam suatu proyek penelitian, baik untuk penyusunan laporan bagi suatu lembaga, instansi, atau perusahaan, publikasi dalam jurnal, maupun sebagai syarat akademik dalam penyelesaian skripsi, tesis, atau disertasi, pemanfaatan berbagai literatur sebagai referensi atau bahan rujukan menjadi suatu keharusan (Abraham dan Supriyati, 2022) .

Berdasarkan penelitian (El-Kassas dkk., 2020), tentang kerangka kerja berbasis grafik untuk ringkasan otomatis, *Automatic Text Summarization* penting bagi pengguna karena akan menghemat banyak waktu dibandingkan melakukan peringkasan secara manual, akan membantu pengguna mendapatkan ringkasan untuk satu dokumen yang berkaitan dengan topik yang sama, dan ringkasan yang dihasilkan dapat memberikan gambaran umum ikhtisar tentang berbagai topik dalam dokumen masukan maka pengguna dapat Kembali ke dokumen asli jika dia membutuhkan lebih banyak detail.

Menurut penelitian (Fakhrezi dkk., 2021), *Automatic Text Summarization* merupakan metode peringkasan teks dari sebuah dokumen tetapi tidak mengubah hakikat teks aslinya. Secara umum, terdapat dua jenis peringkasan teks, yaitu : *extractive Summarization* yaitu membuat rangkuman dengan klaimat-kalimat yang ada dalam dokumen dan *abstractive Summarization* dengan menulis ulang topik utama dalam dokumen dengan kalimat-kalimat yang berbeda.

Menurut penelitian (Özkan Çelik dan Al, 2024), tentang analisis IMRAD struktur artikel dan pengaruhnya terhadap ringkasan ekstraktif, penting untuk memastikan teks sesuai pola IMRAD karena susunan artikel mempengaruhi distribusi bobot kalimat. Artikel tanpa metode ini tidak sesuai dengan penulisan

akademis, karena mereka tidak memberikan penjelasan lengkap tentang penelitian tersebut.

Menurut penelitian (Bao dkk., 2025), tentang meningkatkan peringkasan abstraktif karya ilmiah menggunakan struktur, peringkasan abstraktif karya ilmiah yang didasarkan pada pengenalan fungsi struktural otomatis menghasilkan ringkasan yang lebih komprehensif dan seimbang. Hal yang perlu dilakukan adalah merekonstruksi fungsi struktural, kemudian mengembangkan model klasifikasi teks panjang dengan menyempurnakan model terlatih dengan memanfaatkan informasi penting dari awal hingga akhir bab.

Menurut penelitian (Nan dkk., 2021), tentang konsistensi faktual tingkat entitas dari peringkasan teks abstraktif, arsitektur *Transformers* yang dikombinasikan dengan *pretraining* telah mencetak rekor baru untuk tugas pemrosesan bahasa alami. Secara khusus model BART telah menunjukkan kinerja yang unggul di banyak bidang tugas pembuatan teks termasuk peringkasan abstraktif.

Menurut penelitian (Ulker dan Ozer, 2024), tentang model peringkasan abstraktif untuk meringkas artikel ilmiah, BART menggunakan arsitektur *sequence-to sequence* lebih efektif bila disetel untuk tugas peringkasan abstraktif. Model BART disempurnakan dan dilatih untuk merangkum artikel ilmiah.

2.2 Dasar Teori

2.2.1 *Automatic Text Summarization*

Automatic Text Summarization (ATS) merupakan sistem yang secara otomatis merangkum dokumen tanpa menghilangkan informasi penting di dalamnya. ATS menjadi salah satu topik yang banyak diteliti dalam bidang *Natural Language Processing* (NLP). Beberapa contoh penerapan ATS antara lain adalah ringkasan yang muncul pada hasil pencarian mesin pencari, ringkasan produk berdasarkan ulasan di *platform e-commerce*, serta ringkasan artikel dalam media tertentu. (Alfhi Saputra, 2021).

Secara umum, ada dua cara untuk menampilkan teks ringkasan: Ekstraktif dan Abstraktif. Suatu metode dianggap ekstraktif jika kata-kata, frasa, dan kalimat

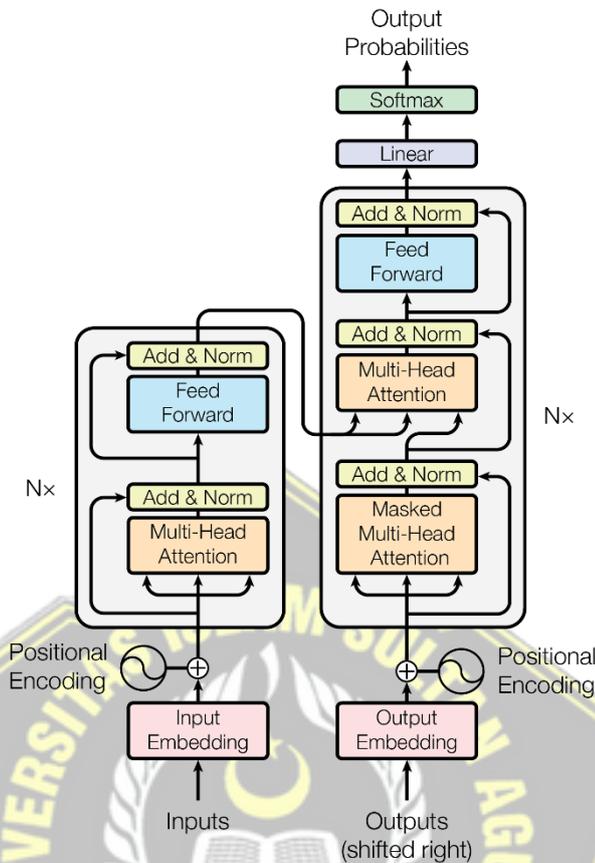
dalam ringkasan dipilih dari artikel sumber. Mereka relatif sederhana dan dapat menghasilkan kalimat yang benar secara tata bahasa. Ringkasan yang dihasilkan biasanya menyimpan informasi penting dari artikel sumber dan memiliki kinerja yang baik. Di sisi lain, teks abstraktif adalah ringkasan yang menghasilkan kata-kata baru dengan menggunakan model pembangkitan bahasa yang dikondisikan pada representasi dokumen sumber. Dengan demikian, mereka mempunyai potensi kuat untuk menghasilkan ringkasan berkualitas tinggi yang inovatif secara verbal dan dapat menggabungkan pengetahuan eksternal (Alfhi Saputra, 2021).

2.2.2 *Natural Language Processing (NLP)*

Natural Language Processing (NLP) adalah cabang kecerdasan buatan (AI) yang berfokus pada interaksi antara komputer dan bahasa manusia. Tujuan utama NLP adalah memungkinkan komputer untuk memahami, menafsirkan, dan memproses bahasa manusia secara efisien, baik dalam bentuk teks maupun suara. Secara umum, NLP dapat dianggap sebagai usaha untuk membuat komputer dapat “berpikir” dan “berkomunikasi” layaknya manusia. Teknologi dalam NLP mencakup berbagai aspek modern, seperti pemahaman teks, penerjemahan bahasa, dan pengenalan suara. (Maesaroh dan Yunianto, 2024).

2.3 *Transformers*

Transformers adalah arsitektur untuk menyelesaikan permasalahan sekuensial seperti kalimat, yang merupakan jaringan syaraf tiruan. *Transformers* bertugas menghubungkan *encoder* dan *decoder* ke teks secara bertahap *input* dan *output* yang saling berhubungan melalui vektor konteks. Proses NLP menggunakan banyak kata yang diproses melalui *Transformers* (Thoyyibah dkk., 2023).



Gambar 2. 1 Model Arsitektur *Transformer* (Mohiuddin dkk., 2023)

Pada gambar 2.1 diatas merupakan gambar dari arsitektur *Transformers* dan juga merupakan mekanisme yang mempelajari hubungan kontekstual antar kata. Pada gambar di atas menggambarkan proses *encoder* dan *decoder* melalui proses *input* dan menghasilkan *output* (Mohiuddin dkk., 2023)

2.3.1 Arsitektur *Transformers*

2.3.1.1 *Encoder dan Decoder*

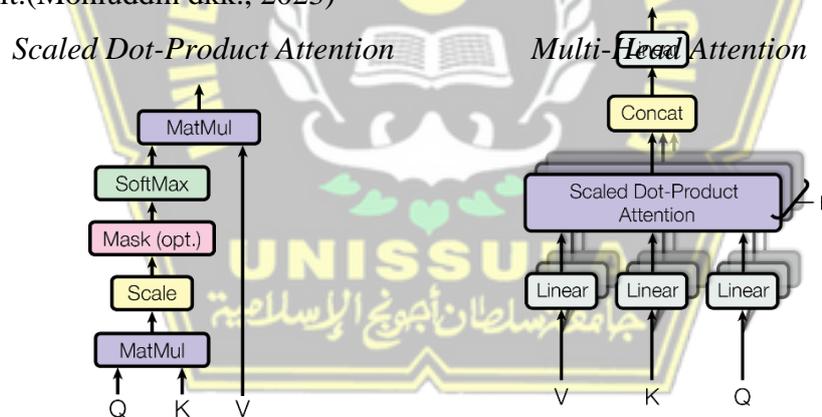
Encoder dan *decoder* merupakan salah satu arsitektur dalam *deep learning* yang umumnya digunakan dalam pembelajaran unsupervised, di mana data yang digunakan tidak memiliki label. Arsitektur ini terdiri dari dua bagian utama. Pertama, *encoder* berfungsi untuk memproses data masukan dan mengubahnya menjadi *state* atau vektor representasi yang merepresentasikan informasi dari data tersebut. Kedua, *decoder* bertugas mengambil *state* atau vektor representasi yang dihasilkan oleh *encoder*, kemudian mengolahnya menjadi keluaran yang sesuai dengan tujuan yang diinginkan. (Akbar, 2021).

Encoder memiliki tumpukan $N = 6$ lapisan yang identik, dengan dua lapisan di setiap lapisan *encoder*: yang pertama adalah mekanisme *self-attention multi-head*, dan yang kedua adalah jaringan *feed-forward* yang sederhana dan terhubung secara posisi.

Decoder juga terdiri dari tumpukan $N = 6$ lapisan yang identik, dengan dua sub-lapisan di setiap lapisan *encoder*. Selain itu, *decoder* menyisipkan sub-lapisan ketiga, yang melakukan *multi head attention* dari *output* tumpukan *encoder*. (Mohiuddin dkk., 2023)

2.3.1.2 Attention

Fungsi *attention* dapat dijelaskan sebagai proses pemetaan antara kueri dan sekumpulan pasangan nilai-kunci ke dalam sebuah keluaran, di mana kueri, kunci, nilai, dan keluaran semuanya direpresentasikan dalam bentuk vektor. *Output* dihasilkan melalui penjumlahan nilai yang telah diberi bobot, di mana bobot tersebut ditentukan berdasarkan tingkat kompatibilitas antara kueri dan kunci terkait. (Mohiuddin dkk., 2023)



Gambar 2. 2 (Kiri) *Scaled Dot-Product Attention*. (Kanan) *Multi-Head Attention* (Mohiuddin dkk., 2023)

Mekanisme *attention* adalah metode yang digunakan untuk menentukan tingkat perhatian yang dibutuhkan pada setiap bagian data, kemudian merepresentasikannya dalam bentuk vektor representasi yang selanjutnya diproses kembali menjadi *context vector*. Vektor representasi yang dihasilkan oleh *encoder* belum sepenuhnya mampu menangkap seluruh fitur data, sehingga *decoder* mengalami kesulitan dalam merekonstruksi keluaran secara optimal. Mekanisme

attention memungkinkan model untuk menentukan bagian mana dari data masukan yang relevan untuk setiap sekuens keluaran, sehingga informasi yang diproses menjadi lebih tepat sasaran dan meningkatkan kinerja *decoder*. (Akbar, 2021).

2.3.1.3 Scaled Dot-Product Attention

Vektor kueri, vektor kunci, dan vektor nilai adalah tiga vektor yang diterima *scaled dot-product attention*. Ini adalah ide dasar dari *self attention*. Perhitungan dilakukan antara vektor kueri dan vektor kunci untuk setiap pasangan kueri kunci menghasilkan berbagai skor perhatian yang menunjukkan seberapa banyak komponen dalam dua *input* berkontribusi pada komponen lain dalam konteks *self-attention*. Skalabilitas, atau akar kuadrat dari dimensi vektor kueri, dibagi oleh nilai skor agar tetap stabil saat dimasukkan ke dalam fungsi *softmax*. Untuk mengontrol banyaknya skor dan memastikan bahwa model dapat belajar perhatian yang halus, ini disebut “*scaling*”. Skor yang telah disesuaikan dengan skala digunakan sebagai bobot untuk menghitung vektor nilai rata-rata tertimbang, menunjukkan seberapa besar kontribusi setiap nilai terhadap vektor hasil akhir. Proses *scaled dot-product attention* menghasilkan vektor hasil. Mekanisme ini diulang untuk setiap pasangan kunci pertanyaan untuk memahami hubungan antara elemen-elemen tersebut. Hasilnya adalah matriks yang menunjukkan relevansi setiap elemen dalam data (Mohiuddin dkk., 2023).

2.3.1.4 Multi-Head Attention

Multi-head attention adalah mekanisme perhatian multi arah yang berjalan berkali-kali secara bersamaan. Sebuah linier transformasi dilakukan terhadap keluaran untuk memperoleh dimensi yang diharapkan sebesar menggabungkannya. Komponen utama *Transformers* adalah modul *multi-head self attention* yang dapat diakses menggunakan *remote*, yang menggabungkan sub-ruang representasi, dan model dapat menyimpulkan perhatian Bersama (Pandu, 2024).

Multi-Head Attention menerima tiga vektor *input* yaitu, vektor kueri, vektor kunci, dan vektor nilai. Sebelum melakukan perhitungan, setiap vektor diubah menjadi sejumlah sub-ruang melalui proyeksi linier. Proyeksi ini menggunakan matriks pembobotan yang berbeda untuk masing-masing kepala. Setiap kepala menghitung skor mereka sendiri dan menghasilkan sejumlah matriks yang berbeda.

Capaian ini didapat melalui perhitungan produk dot antara vektor kueri dan vektor kunci setelah proyeksi linier. Matriks akhir dibuat dengan menggabungkan hasil dari setiap kepala. Hasil akhir yang lebih halus dapat dicapai dengan menggunakan proyeksi linier tambahan. Seperti pada *scale dot-product attention*, matriks digunakan untuk mengambil rata-rata tertimbang dari vektor nilai. Ini menghasilkan vektor hasil yang menunjukkan interaksi antara komponen dalam data *input*. (Mohiuddin dkk., 2023)

2.3.1.5 Position-wise Feed-forward Networks

Masing-masing lapisan *encoder* dan *decoder* terdiri dari jaringan *feed-forward* yang terhubung sepenuhnya, yang diterapkan secara terpisah dan identik di setiap posisi. Ini terdiri dari dua transformasi linier yang masing-masing memiliki aktivasi ReLU.

Transformasi linier memiliki parameter yang berbeda disetiap lapisan, meskipun mereka identik pada posisi yang berbeda. Dua konvolusi dengan ukuran kernel 1 adalah cara lain untuk menggambarkan hal ini (Mohiuddin dkk., 2023).

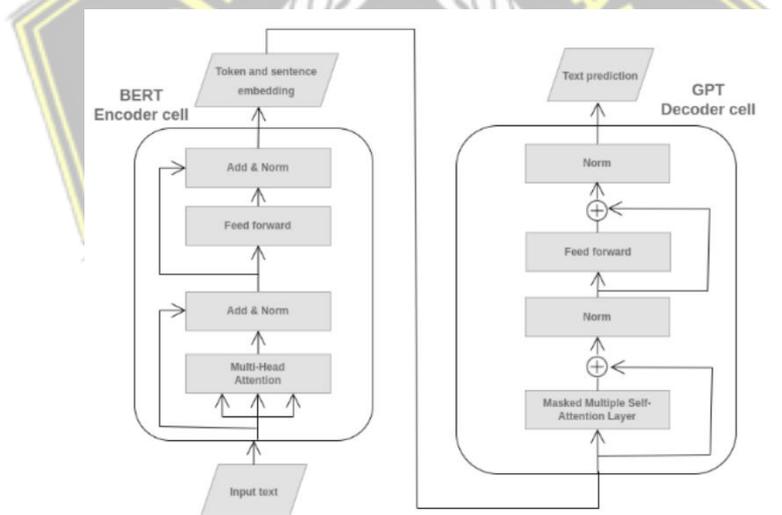
2.3.1.6 Word Embedding dan Softmax

Word embedding adalah teknik pembelajaran fitur yang memetakan setiap kata atau frasa dalam kosakata ke dalam vektor bilangan real dengan dimensi n . Metode ini berfokus pada penentuan vektor yang serupa untuk kata-kata yang memiliki makna semantik yang mirip. Sementara itu, *softmax* adalah versi lebih "halus" dari fungsi *argmax*, yang memungkinkan setiap kelas dalam output memiliki nilai probabilitas, di mana total dari seluruh probabilitas tersebut selalu berjumlah 1. (Putra dkk., 2022).

Dengan memanfaatkan embedding, model ini mengubah token *input* dan output menjadi vektor berdimensi. Selain itu, model ini menerapkan transformasi linier serta fungsi *softmax*, yang umum digunakan untuk mengonversi output *decoder* menjadi probabilitas token berikutnya. Model ini juga menggunakan matriks bobot yang sama antara dua lapisan *embedding* dan transformasi linier sebelum *softmax*. Pada lapisan *embedding*, bobot-bobot tersebut dikalikan dengan vektor-vektor berdimensi untuk menghasilkan representasi yang sesuai (Mohiuddin dkk., 2023).

2.3.2 Bidirectional and Auto-Regressive Transformers (BART)

Bidirectional and Auto-Regressive Transformers adalah model *transformer* inovatif yang menggabungkan dua arsitektur *transformer* terkemuka yaitu *Bidirectional Encoder Representations from Transformers* (BERT) *encoder* dan *decoder generative pretrained transformer* (GPT). Keunggulan BART terletak pada kemampuannya menghasilkan keluaran teks dua arah, menggabungkan konteks dari kiri ke kanan dan sebaliknya. Kapasitas ini menghasilkan teks yang lebih koheren dan bermakna dibandingkan model *transformer* tradisional. Model ini didasarkan pada arsitektur *transformer* yang telah terbukti keberhasilannya dalam berbagai tugas NLP, seperti peringkasan teks, terjemahan mesin, dan pembuatan bahasa alami. *Encoder* BERT memungkinkan BART lebih memahami konteks kalimat, sedangkan *decoder* GPT memungkinkan menghasilkan teks yang mengalir secara alami dan sesuai konteks (Hartawan dkk., 2024)



Gambar 2. 3 Arsitektur BART (Hartawan dkk., 2024)

Pada gambar 2. 3 diatas adalah arsitektur BART terdiri dari sel *decoder* GPT dan sel *encoder* BERT. Perhatikan multi-kepala di setiap sel pembuat encode memproses teks mentah yang diberi token; pra-pemrosesan tambahan (seperti huruf kecil, stemming, penghapusan stopword, dll.) dilakukan hanya berdasarkan tugas yang ada. Alih-alih mempelajari pengkodean dari sel mask BERT Encoder, sel *decoder* GPT menerima penyematan mask dari sel BERT Encoder. Ini meneruskannya ke beberapa blok perhatian independen yang terselubung. Blok-

blok ini mengikuti arsitektur yang sama dengan blok perhatian multi-kepala tetapi beroperasi secara berurutan, bukan secara bersamaan. Lapisan ini belajar mendekode penyematan bertopeng menjadi token yang koheren secara semantik dengan mempertimbangkan beberapa penyematan paralel dari teks masukan yang disamarkan pada tingkat berbeda (Hartawan dkk., 2024).

2.3.3 IMRAD

IMRAD (*Introduction, Method, Result, and Discussion*) adalah salah satu acuan yang digunakan untuk membuat artikel ilmiah. Metode IMRAD merupakan acuan pembuatan jurnal yang sesuai standar sinta, scopus, dan WOS. Berikut adalah bagian-bagian dari IMRAD:

1. *Introduction*: pada bagian ini berisi tentang latar belakang dan masalah yang diangkat
2. *Methods*: pada bagian ini berisi metodologi atau metode-metode yang digunakan untuk melakukan penelitian.
3. *Results*: bagian ini berisi tentang hasil-hasil yang telah didapat dari penelitian.
4. *Discussion* pada bagian ini berisi tentang kesimpulan dari penelitian yang telah dilakukan. (Rahmat dkk., 2023)

2.3.4 PDF Extraction

Portable document format (pdf) adalah jenis dokumen yang umum untuk dokumen akademik. Mengekstraksi dokumen pdf akademik sangat diperlukan untuk berbagai tugas pengindeksan, pengambilan, analisis, pencarian, rekomendasi, ringkasan dan berbagai pemrosesan teks lainnya. Pdf berfokus pada tata letak *visual* dokumen untuk memastikan tampilan yang konsisten dan mencakup struktur dan informasi semantik pada elemen dokumen (Meuschke dkk., 2023).

2.3.5 BERTScore

BERTScore adalah salah satu metrik evaluasi otomatis yang digunakan untuk menilai kualitas teks yang dihasilkan oleh sistem pemrosesan bahasa alami (NLP). Metrik ini digunakan sebagai alternatif dari metrik tradisional seperti ROUGE dan BLEU yang memiliki keterbatasan dalam menangkap makna semantik dari teks. BERTScore memanfaatkan representasi vektor dari model BERT (*Bidirectional*

Encoder Representations from Transformers) untuk menghitung kesamaan antara teks referensi dan teks yang dihasilkan (Zhang dkk., 2020).

Dalam penerapannya, BERTScore menghitung kesamaan vektor setiap kata dalam teks referensi dan teks prediksi dengan menggunakan *cosine similarity*. Proses ini dilakukan dengan mencocokkan token dalam kalimat kandidat dan kalimat referensi berdasarkan nilai *embedding* dari model BERT *pre-trained*. Setelah itu, nilai Precision, Recall, dan F1-Score dihitung untuk menilai kualitas teks yang dibandingkan (Hanna dan Bojar, 2021).

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \quad F_{\text{BERT}} = 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

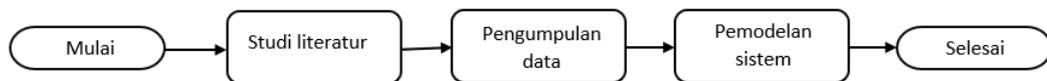
Gambar 2. 4 Rumus BERTScore

Pada gambar 2. 4 diatas adalah rumus dari BERTScore. Didalamnya terdapat tiga rumus yaitu *Recall*, *Precision*, dan F1. *Precision* Mengukur seberapa banyak token dalam teks kandidat yang memiliki pasangan serupa dalam teks referensi. *Recall* Mengukur seberapa banyak token dalam teks referensi yang memiliki pasangan serupa dalam teks kandidat dan F1 Merupakan rata-rata harmonis dari *Precision* dan *Recall* (Zhang dkk., 2020).

BAB III METODE PENELITIAN

3.1 Metodologi Penelitian

Penelitian ini menerapkan metode atau algoritma yang dikenal sebagai Bidirectional and Auto-Regressive Transformers (BART). Sistem ini menghasilkan ringkasan dari artikel ilmiah yang disajikan dalam format tabel.



Gambar 3.1 *Flowchart* Metode Penelitian

Pada gambar 3.1 diatas adalah flowchart metodologi penelitian. Gambar tersebut memperlihatkan Langkah kerja yang harus dilakukan dalam penelitian yaitu studi literatur, pengumpulan data, dan pemodelan sistem.

3.1.1 Studi Literatur

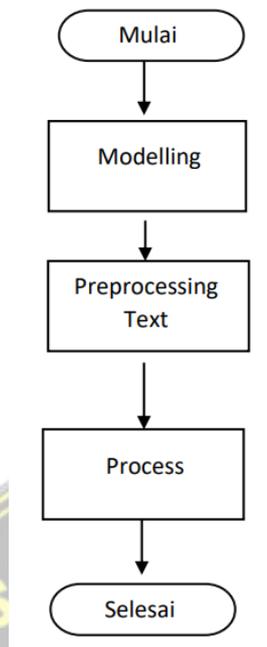
Dalam penelitian ini akan dilakukan peninjauan pada beberapa makalah, jurnal, tesis, dan skripsi dari penelitian terdahulu dan akan diulas. Tujuan utama dari kegiatan ini adalah untuk mempelajari teori dan konsep dari *Bidirectional And Auto-Regressive Transformers* (BART).

3.1.2 Pengumpulan Data

Pada tahap ini, langkah awal yang harus dilakukan adalah menentukan sumber data yang ditemukan. *Dataset* yang digunakan adalah *dataset* scisumm yang berisi dua kolom yaitu *Text* dan *Summary*. Pada penelitian ini sumber data yang akan digunakan diperoleh dari *website* Kaggle.

3.1.3 Pemodelan sistem

Berikut adalah desain pemodelan sistem yang akan dikembangkan:



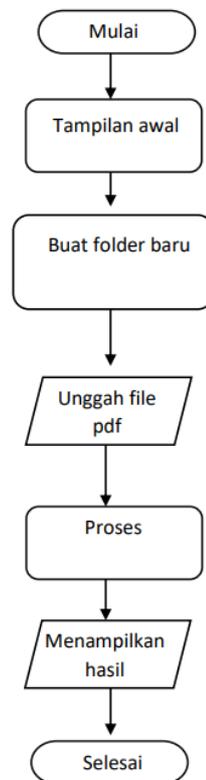
Gambar 3. 2 *Flowchart* Pemodelan Sistem

Pada gambar 3. 2 diatas adalah *Flowchart* pemodelan sistem,. Pada pemodelan sistem ini dibagi menjadi beberapa proses, berikut adalah uraiannya :

1. *Modeling*, pada tahap ini dilakukan proses memuat model BART dan membuat model. Proses ini dimulai dengan persiapan *dataset* dan melakukan *fine-tuning*. Persiapan *dataset* dilakukan dengan melakukan pembersihan dari data redundansi, membagi *dataset*, dan membuat data *loader*. *Fine-tuning* dilakukan dengan melakukan pelatihan model dan evaluasi.
2. *Preprocessing Text* , pada tahap ini dilakukan proses ekstrak teks dari dokumen pdf lalu melakukan identifikasi struktur penulisan berdasarkan IMRAD (*Introduction, Method, Results, and Discussion*).
3. *Process*, pada tahap ini dilakukan peringkasan pada setiap bagian yang telah diidentifikasi kemudian hasil ringkasan dari setiap bagian akan ditampilkan.

3.2 Analisis Sistem

Pada penelitian ini penulis akan membuat sistem pencarian dan *input* pdf berbasis *website* yang bertujuan untuk membuat ringkasan berupa kajian literatur untuk artikel ilmiah . Berikut adalah gambaran sistem yang akan dibuat :



Gambar 3. 3 Flowchart Sistem Ringkasan

Pada gambar 3. 3 diatas adalah *flowchart* untuk menggambarkan proses dari sistem ringkasan otomatis. Berikut adalah uraian dari tahap-tahapnya :

1. Pertama *User* memasuki halaman awal dari *website* sistem ringkasan
2. Kedua *User* membuat *folder* untuk penelitiannya
3. Setelah itu *User* dapat meng-upload dokumen pdf yang diinginkan
4. Selanjutnya sistem akan memprosesnya untuk mendapatkan ringkasan dari setiap bagian
5. Setelah selesai diproses sistem akan menampilkan hasil pada tabel-tabel yang ditampilkan

3.3 Evaluasi

Pada penelitian ini metode evaluasi sistem yang digunakan adalah BERTScore. Evaluasi ini dilakukan dengan membandingkan teks referensi dan teks hasil prediksi. Evaluasi dilakukann terhadap hasil ringkasan dari sistem dan dibandingkan dengan ringkasan yang dibuat oleh QuillBot sebagai teks referensi.

QuillBot dipilih sebagai referensi karena hasil ringkasannya memiliki kualitas yang baik, dan mudah untuk diakses.

3.4 Analisis Kebutuhan

Pada bagian ini akan diuraikan mengenai kebutuhan apa saja yang dibutuhkan dalam pengembangan sistem ini. Berikut adalah hal-hal yang dibutuhkan dalam pengembangan sistem :

1. Phyton 3.10.5

Phyton adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991, Phyton juga merupakan bahasa yang sangat populer belakangan ini. Selain itu, Phyton merupakan bahasa pemrograman yang *multi* fungsi salah satunya pada bidang *Machine Learning* dan *Deep learning*. Penelitian ini menggunakan Phyton versi 3.10.5 karena bersifat *open source*, komunitasnya besar, dan banyak sumber daya *online*.

2. *Library Transformers*

Transformers memberikan akses model *Transformers* dan alat untuk pemrosesan bahasa alami (NLP) yang kuat. pada penelitian ini arsitektur *Transformers* yang digunakan adalah BART. Dengan algoritma ini ringkasan yang didapat akan lebih relevan, cepat dan efisien.

3. *Library PyTorch*

PyTorch adalah kerangka kerja berfitur lengkap untuk membangun model pembelajaran mendalam, yang merupakan jenis pembelajaran mesin yang umum digunakan dalam aplikasi seperti pengenalan gambar dan pemrosesan bahasa. Ditulis dalam Phyton, kerangka kerja ini relative mudah dipelajari dan digunakan oleh Sebagian besar pengembang pembelajaran mesin.

4. *Library Pandas*

Pandas adalah salah satu *Library* yang menyediakan struktur dan fungsi data tingkat tinggi yang dirancang agar berfungsi dengan data terstruktur atau tabular dengan cepat, mudah, dan ekspresif. Sejak kemunculannya pada tahun 2010, ini telah membantu kemungkinan Phyton menjadi lingkungan analisis data yang kuat dan produktif. Objek utama dalam *Pandas* adalah dataframe, struktur data

berbentuk tabel dan berorientasi kolom dengan label baris dan kolom, dan seri objek array berlabel satu dimensi.

5. *Library Numpy*

NumPy (*Numerical Python*), adalah *Library Python* yang menyediakan struktur dan fungsi dari struktur data, algoritma dan perekat perpustakaan yang dibutuhkan untuk Sebagian besar aplikasi ilmiah yang melibatkan data numerik dengan Python.

6. *Library PyMuPDFLLM4*

PyMuPDFLLM4 atau juga dikenal dengan *Fitz*, adalah pustaka python yang menyediakan perangkat lengkap untuk bekerja dengan *file PDF*. Dengan PyMuPDFLLM4, dapat digunakan untuk membuka teks PDF, mengekstrak teks dan gambar, dan sejenisnya.

7. *Library NLTK*

NLTK adalah *toolkit* yang dapat digunakan untuk pemrosesan bahasa alami. Toolkit ini tidak hanya memiliki fungsi pengambilan, tetapi juga memiliki fungsi seperti pembersihan teks, penggabungan bentuk kata, penandaan bagian ucapan, analisis tata bahasa, dan analisis semantik.

8. *Visual Studio Code*

Visual Studio Code adalah perangkat lunak *editor* kode sumber yang ringan namun sangat *powerful* dan dapat dijalankan di *desktop*. *Editor* ini digunakan untuk menulis dan mengembangkan kode program yang diperlukan dalam pembuatan aplikasi. *Visual Studio Code* mendukung berbagai bahasa pemrograman, seperti JavaScript, HTML, CSS, PHP, Python, C++, dan banyak lainnya. Selain itu, perangkat lunak ini kompatibel dengan berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Salah satu fitur unggulannya adalah *Live Share*, yang memungkinkan beberapa pengembang untuk bekerja secara bersamaan pada satu proyek dari lokasi yang berbeda.

9. *Streamlit*

Streamlit adalah salah satu *framework* yang mendukung *deployment* model ke dalam aplikasi berbasis web. *Framework* ini bersifat *open-source* dan dikembangkan menggunakan Python. *Streamlit* dirancang untuk mempermudah

developer dalam membangun aplikasi web interaktif di bidang *data science* dan *machine learning*. Salah satu keunggulan *Streamlit* adalah kemudahan penggunaannya, karena *developer* tidak perlu mengatur tampilan web menggunakan CSS, HTML, atau JavaScript, sebab *framework* ini sudah menyediakan berbagai fungsi bawaan untuk kebutuhan tersebut.

3.5 Perancangan *User Interface*

3.5.1 Perancangan Halaman awal

Seperti yang telah dijelaskan sebelumnya sistem ini digunakan untuk meringkas artikel ilmiah. Artikel ilmiah yang diringkas merupakan *file pdf*. Berikut adalah rancangan antarmuka untuk halaman awal.

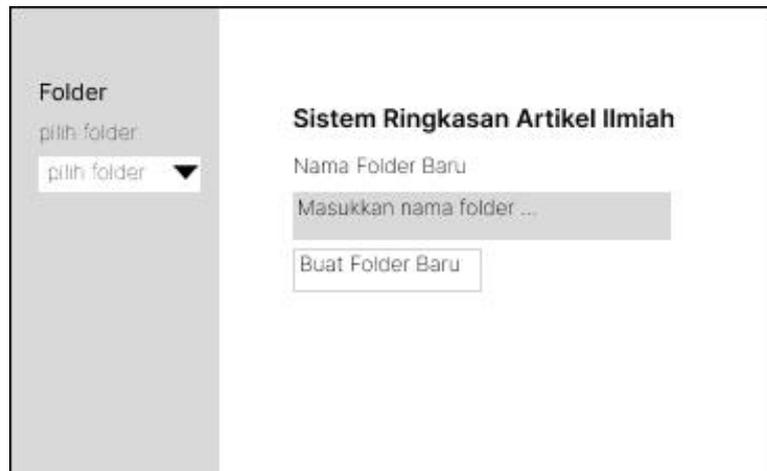


Gambar 3. 4 Perancangan Halaman Awal

Pada gambar 3.2 diatas adalah rancangan untuk halaman awal sistem. Halaman ini memuat judul, menu membuat *folder* baru, dan *expand icon*. *Expand icon* ini merupakan *sidebar* yang akan menampilkan daftar *folder* yang telah dibuat.

3.5.2 Perancangan Halaman *Sidebar* Daftar Folder

Seperti yang telah dijelaskan sebelumnya sistem ini digunakan untuk meringkas artikel ilmiah. Artikel ilmiah yang diringkas merupakan *file pdf*. Berikut adalah rancangan antarmuka untuk halaman *sidebar* daftar *folder*.



Gambar 3. 5 Perancangan Halaman *Sidebar* Daftar *Folder*

Pada gambar 3. 3 diatas adalah rancangan untuk halaman *sidebar* daftar *folder*. Daftar *folder* dibuat dengan *select box* yang ditampilkan secara *dropdown*. Setelah memilih *folder* maka pengguna akan diarahkan ke *folder* yang telah dipilih untuk masuk pada proses *upload file*.

3.5.3 Perancangan Halaman *Upload File*

Seperti yang telah dijelaskan sebelumnya sistem ini digunakan untuk meringkas artikel ilmiah. Artikel ilmiah yang diringkas merupakan *file pdf*. Berikut adalah rancangan antarmuka untuk halaman *sidebar* daftar *folder*.



Gambar 3. 6 Perancangan Halaman *Upload File*

Pada gambar 3. 4 diatas adalah perancangan untuk halaman *upload file*. Halaman ini memuat menu untuk mengunggah *file* dan menghapus *folder*. Untuk mengunggah *file* pengguna dapat mengklik tombol pilih *file* lalu sistem akan

mengarahkan pengguna ke penyimpanan internal komputer dan memilih *file* yang akan diringkas.

3.5.4 Perancangan Halaman Hasil Ringkasan

Seperti yang telah dijelaskan sebelumnya sistem ini digunakan untuk meringkas artikel ilmiah. Artikel ilmiah yang diringkas merupakan *file* pdf. Berikut adalah rancangan antarmuka untuk halaman *sidebar* daftar *folder*.



Gambar 3. 7 Perancangan Halaman Hasil Ringkasan

Pada gambar 3. 5 diatas adalah perancangan untuk halaman hasil ringkasan. Pada halaman ini menampilkan hasil ringkasan dan menu untuk mengunduh hasil ringkasan dalam format csv. Pada tabel hasil ringkasan menampilkan kolom no, metadata, dan hasil rangkuman perbagian IMRAD. Tabel ini dibuat dengan model *scroll table* sehingga untuk melihat keseluruhan kolom pengguna dapat menggeser tabel.

BAB IV

HASIL DAN ANALISIS PENELITIAN

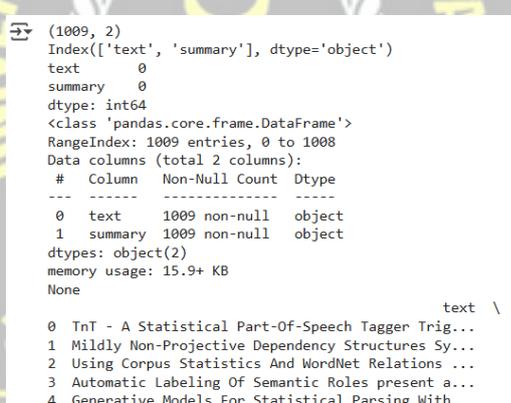
4.1 Hasil Penelitian

4.1.1 Persiapan Data

Pada penelitian ini data yang digunakan untuk melakukan *pretraining* bersumber dari *dataset* dari scisumm. *Dataset* ini terdiri dari 1000 data yang berisi teks dan ringkasan dari artikel ilmiah. Pada penelitian ini data yang digunakan untuk uji coba sistem berupa artikel ilmiah yang bersumber dari *website* garuda. Ketentuan data yang digunakan adalah *file* harus berformat pdf, topik penelitian tentang teknologi, dan menggunakan bahasa inggris.

4.1.1.1 Load Dataset

Pada bagian ini *dataset* akan diperiksa jumlah, struktur, menampilkan lima data teratas dan menampilkan tipe data.



```
(1009, 2)
Index(['text', 'summary'], dtype='object')
text      0
summary   0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1009 entries, 0 to 1008
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  ---      -
0   text    1009 non-null  object
1   summary 1009 non-null  object
dtypes: object(2)
memory usage: 15.9+ KB
None
```

	text
0	InT - A Statistical Part-Of-Speech Tagger Trig...
1	Mildly Non-Projective Dependency Structures Sy...
2	Using Corpus Statistics And WordNet Relations ...
3	Automatic Labeling Of Semantic Roles present a...
4	Generative Models For Statistical Parsing With...

Gambar 4. 1 Memeriksa Info *Dataset*

Pada gambar 4.1 diatas menampilkan informasi dari *dataset* yaitu jumlah data, nama kolom, tipe data, nilai yang kosong, dan menampilkan beberapa data teratas. Dari gambar diatas dapat diketahui bahwa jumlah data pada *dataset* berjumlah 10009 data dengan dua kolom yaitu *Text* dan *Summary*. *Dataset* memiliki tipe data int64 dan tidak memiliki data yang kosong.

4.1.1.2 Preprocessing Data

Pada *Preprocessing* data hal yang akan dilakukan adalah membuat tokenizer, membuat fungsi *Preprocessing*, mengkonversi *dataset*, dan membagi *dataset*. Berikut adalah uraiannya:

1. Langkah pertama dalam proses preprocessing data adalah memuat *tokenizer*. *Tokenizer* ini digunakan untuk mengkonversi teks menjadi token yang dapat dimengerti oleh model BART. *Tokenizer* yang digunakan merupakan varian dari 'facebook/bart-large' yang biasa digunakan untuk *Summarization*.
2. Langkah kedua adalah fungsi *Preprocessing*, fungsi ini digunakan untuk menyiapkan data sebelum dimasukkan ke model BART. Didalamnya terdapat tiga fungsi utama yaitu tokenisasi teks *input*, tokenisasi teks target, dan menambahkan label. Dalam tokenisasi teks *input*, tokenizer mengubah teks menjadi token numerik, lalu mengisi token hingga panjang maksimum 1024 token, dan memotong teks jika panjang melebihi batas. Pada tokenisasi teks target ringkasan diproses dan panjang maksimalnya dibatasi 200 token. Selanjutnya pada penambahan label diperlukan dalam pelatihan model untuk memahami hubungan antara teks asli dan ringkasan.
3. Langkah ketiga adalah proses mengkonversi *dataset*. Proses ini digunakan untuk menkonversi dataframe ke *dataset hugging face*. Hal ini dilakukan agar proses lebih optimal untuk NLP. Didalamnya dilakukan tokenisasi untuk seluruh *dataset* dan hasil akhirnya adalah *dataset* yang siap untuk dilatih.
4. Langkah terakhir adalah melakukan pembagian *dataset*. *Dataset* dibagi menjadi tiga yaitu data *train*, data *validation*, dan data *test*. Masing-masing jumlahnya yaitu data *train* sebanyak 807 data, data *validation* 101 data, dan data *test* sebanyak 101 data.

4.1.2 Pemodelan Sistem

4.1.2.1 Modelling

Pada *Modeling* dilakukan proses *fine tuning* dan evaluasi model. *Fine tuning* adalah proses melatih model dengan *dataset* baru agar sesuai dengan tugas yang spesifik. Pada tahap *fine-tuning*, terdapat beberapa langkah penting yang dilakukan untuk memastikan model dapat bekerja secara optimal. Pertama, dilakukan inisiasi

model, di mana model BART diinisialisasi dengan bobot awal yang telah dipelajari sebelumnya dari *pretraining*. Selanjutnya, dilakukan *preprocessing dataset*, yaitu menyiapkan data yang akan digunakan untuk pelatihan, termasuk proses tokenisasi, normalisasi, serta pembagian *dataset* menjadi data pelatihan, validasi, dan pengujian. Setelah itu, model menjalani proses pelatihan menggunakan teknik *gradient descent* untuk mengoptimalkan parameter model berdasarkan *dataset* yang telah dipersiapkan. Setelah model dilatih, dilakukan prediksi, di mana model menghasilkan ringkasan otomatis berdasarkan pola yang telah dipelajari. Untuk mengevaluasi sejauh mana hasil prediksi model sesuai dengan label yang diharapkan, dilakukan perhitungan *loss function*, yang digunakan sebagai indikator dalam mengoptimalkan performa model. Evaluasi juga dilakukan per *epoch* guna memantau perkembangan akurasi serta menghindari *overfitting* atau *underfitting*. Setelah proses pelatihan selesai, model yang telah di-*fine-tune* akan disimpan agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang. Terakhir model yang telah disimpan akan di lakukan evaluasi model.

Tabel 4. 1 Hasil Evaluasi Model

Metrik	Nilai
Precision	0.9200
Recall	0.8651
F1-Score	0.8902

Pada table 4. 1 diatas adalah hasil dari evaluasi model. Metode BERTScore digunakan untuk mengukur kualitas ringkasan model ini. Evaluasi ini dilakukan dengan *dataset* uji. Model membuat ringkasan dari teks *input*, lalu dibandingkan dengan ringkasan referensi. Hasil evaluasinya menunjukkan bahwa nilai F1-score adalah 0.8902 atau sekitar 89.02%.

4.1.2.2 Preprocessing Text

Pada bagian ini terdapat dua proses yaitu ekstraksi teks pdf dan pembagian teks sesuai dengan format IMRAD, rincian kegiatannya akan dijelaskan sebagai berikut:

1. Ekstraksi teks

Pada proses ini tidak hanya dilakukan ekstraksi teks tetapi juga dilakukan proses *cleaning* data meliputi penghapusan daftar pustaka, daftar gambar, daftar tabel, abstrak, *header*, catatan kaki, menghapus spasi yang berlebih, dan menghapus karakter-karakter lain yang tidak diperlukan. Hasil dari ekstraksi teks dapat dilihat pada hasil dibawah ini.

Teks Markdown yang telah dibersihkan:

p g j **COVERAGE, DIVERSITY, AND COHERENCE OPTIMIZATION FOR MULTI-DOCUMENT SUMMARIZATION** **Khoirul Umam, Fidi Wincoko Putro, Gulpi Qorik Oktagalu Pratamasunu, Agus Zainal Arifin, and Diana Purwitasari** Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo, Surabaya, 60111, Indonesia E-mail: khoirul.umam35@gmail.com, agusza@cs.its.ac.id **Abstract** A great *Summarization* on *multi-document* with similar topics can help *Users* to get useful information. A good *Summary* must have an extensive coverage, minimum redundancy (high diversity), and smooth connection among sentences (high coherence). Therefore, *multi-document Summarization* that considers the coverage, diversity, and coherence of *Summary* is needed. In this paper we propose a novel *Method* on *multi-document Summarization* that optimizes the coverage, diversity, and coherence among the *Summary's* sentences simultaneously.....

Berdasarkan hasil pada hasil teks ekstraksi diatas terlihat bahwa *file* pdf telah sukses di ekstrak menjadi teks. *Library* yang digunakan untuk ekstraksi ini adalah *pymupdf4llm*. Teks diekstrak dengan markdown, yang bertujuan agar lebih mudah dikenali bagiannya. Dalam proses ini setelah teks diekstraksi kemudian dilakukan *cleaning Text*. *Cleaning Text* ini meliputi penghapusan daftar pustaka, spasi dan tab yang berlebihan, menghapus baris yang kosong, dan menghapus sitasi gambar dan tabel.

2. Pembagian IMRAD

Pada bagian ini adalah pemrosesan teks yang telah diekstraksi menjadi bagian-bagian IMRAD. Pembagian ini dilakukan dengan mengidentifikasi pola dari bagian IMRAD. Hasil identifikasi IMRAD dapat dilihat pada lampiran 2.

== *Introduction* ==

****Introduction**** The contents of a document can be long. It presents several information with specified topic. Current technological developments makes people can find related documents with similar topic easier than before.....

== *Methods* ==

****Methods**** versity for *Summarization* using Self-adaptive Differential Evolution (OCDsum-SaDE) *Method* that ****Summary's Quality Factors**** proposed in [4]. In the *Method*, an optimal *Summary* is searched by considering the coverage *and* In this section, we describe three factors of sumdiversity of *Summary's* sentences.....

== *Results* ==

****Results and Analysis**** In this paper we use *Text Analysis Conference (TAC) 2008 dataset* from National Institute of Standards *and* Technology (NIST)[3] to *test* our *CoDiCo Method*. This *dataset* provides articles that classified into some topics *and* coherent summaries which created manually by human for each topic.....

== *Discussion* ==

****Discussion**** Series of experiment has been conducted to evaluate our proposed *Method* (i.e. *CoDiCo-A and CoDiCo-B*) in comparison with compared *Method* (OCDsum-SaDE). Based on the evaluation *Results* as shown in Table II, we know that the *CoDiCo-A Method* using *Tsim* = 0.7 has higher ROUGE score on ROUGE-1, ROUGE-L, *and* ROUGE-SU than the other *Methods*.....

Berdasarkan hasil pada lampiran 2 terlihat bahwa sistem berhasil mengidentifikasi struktur Imrad. Teks ini dibagi menjadi empat bagian yaitu *Introduction*, *Methods*, *Result*, dan *Discussion*. Masing-masing bagian ini dikenali dengan pola penamaan yang biasa digunakan dalam artikel ilmiah. Kemudian pola ini akan dicocokkan dengan hasil ekstraksi teks.

4.1.2.3 Process

Pada bagian ini proses yang dilakukan adalah melakukan peringkasan pada teks yang telah didapat pada proses sebelumnya. Proses ini mencakup *load model*, dan meringkas dengan BART, dan meringkasnya pada bagian IMRAD masing-masing. Berikut adalah hasil ringkasannya:

==== Introduction =====

The massive quantity of data available in the Internet today has reached such a huge volume. It becomes humanly unfeasible to get efficiently useful information from [1]. Thus, Automatic Methods are needed Information documents efficiently. Document Summarization is one these Methods. presents several with specified topic. creates compressed versions that provides covers all original relevantly. Various kinds algorithms proposed for document Summarization, including ontology-based, clustering, level coverage, diversity, and coherences, heuristic approach, optimization. Single-document Processes between Summary's sentences may help Users understand absorb summaries more than Summary easily. This paper proposes novel Method exact meaning have ambiguous multi-document considers word based on Yago ontology scores. inspired by self-adaptive similarity evolution (SaDE) algorithm [4] [6]. can be used generate an extensive minimum redundancy, high coherence among summaries. also smoothing connections documents. Optimization Coverage Coherence (OCC), which uses optimization find solution optimal coverage coherence. approach able present without any convoluted single-document multi-documents.

==== Methods =====

This paper proposes a Method for summarizing documents using Self-adaptive Differential Evolution (OCDsum-SaDE) to optimize Summary quality by

considering coverage, diversity, and coherence. The considers variety of distinct terms from documents, including chronological, probabilistic, topical, semantic, Text-stinct term approaches. coverage factor helps Users get information easily. similarity between sentences can be calculated frequency in factors. coherence is the Summary. An optimal expected meet three key factors: extensive high diversity or minimum redundancy, coherences among Summary's sentences. Process performed coherence, sentence composition. analyzed variety, composition sentences, with being separated ordering Process. Results show that good has an higher than poor one, reflecting similarity, redundancy. Coherence measures within Summary, their precedence, succession, semantic Text structure, syntactic structure. study aims multi-document Summarization focusing on complexity, diversity. carried out after document Summarization. determined calculating terms.

==== Results =====

This paper uses the Text Analysis Conference (TAC) 2008 dataset from National Institute of Standards and Technology (NIST) to test our CoDiCo Method. The provides articles classified into some topics coherent summaries which created manually by human for each topic. experiments are performed using Matlab R2013a run on Microsoft Windows platform. We both proposed sentences ordering algorithm type A B CoDico use three threshold values, i.e. 0.7, 0.8, 0.9. In order compare Summarization Results Co DiCo Method with another multi-document that considers coverage diversity factors only, we OCDsum-SaDE [4]. also sentence involving a *Tsim* in similarity value evaluate impact between Summary's toward optimal solution. testing evaluated Re-Oriented Understudy Gisting (ROUGE) ROUGE-1 RouGE-2 variants ROUG-N consider *_n*-gram Recall reference *n* assigned 1 2. show coherence is most important quality problem. score calculated comparing two sentences.

==== Discussion =====

The study evaluates CoDiCo-A and CodiCo-B, a multi-document Summarization Method that considers coverage, diversity, coherence simultaneously, producing better Summary than the OCDsum-SaDE Method. It achieves higher RO-UGE

scores on ROUGE-1, ROUG-L, ROUGERGE-SU other Methods. value is average values of OCDsumSaDE, indicating quality smoother connectivity among sentences. should be noted CoDico Method, which involves ordering steps in optimization Process, produces with coherences or connectivity. CoDCo also coverage diversity factors simultaneously to find optimal solution, different compared only coverage-and-diversity factors. comparison two proposed sentences algorithm same threshold using their ROUGET shows CoDco-B CoDi-A. CoDCO-B without others, achieving score 0.7. more similar created manually by human. This focuses between sentence step, an step Process. CoD Co-Co ordering-step technique simultaneously.

Teks diatas merupakan hasil dari proses peringkasan yang dilakukan oleh sistem. Dalam teks diatas dapat terlihat bahwa hasil ringkasan yang ditampilkan menghasilkan kesesuaian yang baik dengan teks asli.

4.2 Hasil Evaluasi

Pada penelitian ini evaluasi dilakukan dengan metode BERTScore. Berikut adalah hasil dari evaluasinya:

Tabel 4. 2 Hasil Evaluasi Sistem

Metrik	Score
Precision	0.8472
Recall	0.8672
F1-Score	0.8569

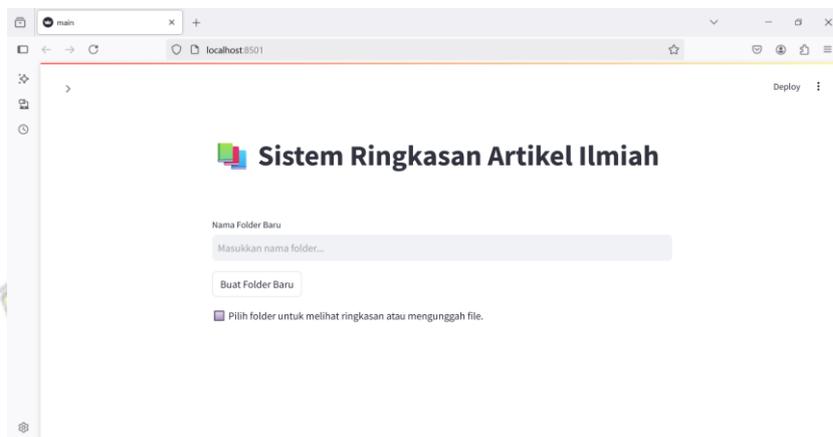
Pada table 4.2 diatas adalah hasil dari evaluasi sistem menggunakan metode BERTScore. Dalam penelitian ini evaluasi dilakukan terhadap 30 file yang telah diringkaskan oleh sistem. Ringkasan ini kemudian dibandingkan dengan ringkasan referensi yang didapatkan dari QuillBot. Pada gambar terlihat bahwa nilai Precision mencapai 0.8472 menunjukkan bahwa sebanyak 84.72% kata dalam teks prediksi cocok dengan teks referensi dan nilai Recall 0.8672 menunjukkan bahwa sebanyak 86.72% kata dalam teks referensi ditemukan dalam teks prediksi. F1-Score

sebanyak 0.8569 menunjukkan skor keseimbangan antara *Precision* dan *Recall* memiliki kualitas keseluruhan prediksi yang cukup tinggi.

4.3 Hasil Perancangan *User Interface*

Setelah menyelesaikan tahap pemodelan sistem, langkah berikutnya adalah mengimplementasikannya pada website. Berikut ini adalah tampilan antarmuka yang telah dikembangkan.

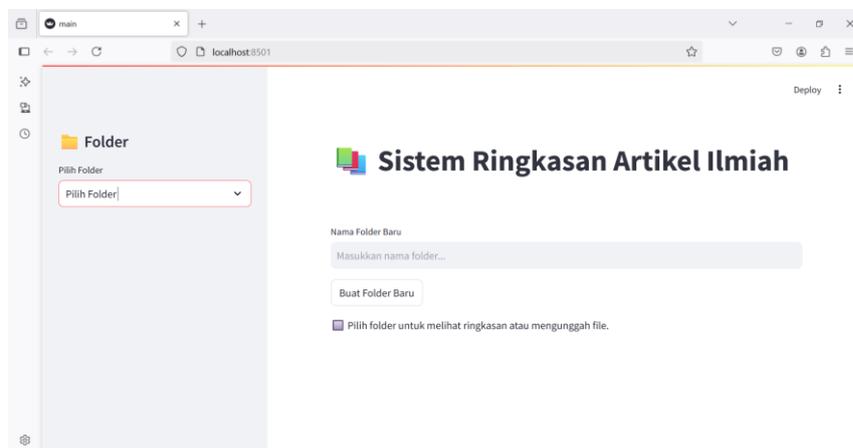
1. Halaman awal



Gambar 4. 2 Halaman Awal

Gambar 4. 2 diatas adalah hasil dari implementasi rancangan halaman awal. Halaman ini menampilkan judul sistem dan menu untuk membuat *folder* ringkasan. Pada sudut kiri atas terdapat *expand icon* yang merupakan *sidebar* yang berisi daftar *folder* yang telah dibuat sebelumnya.

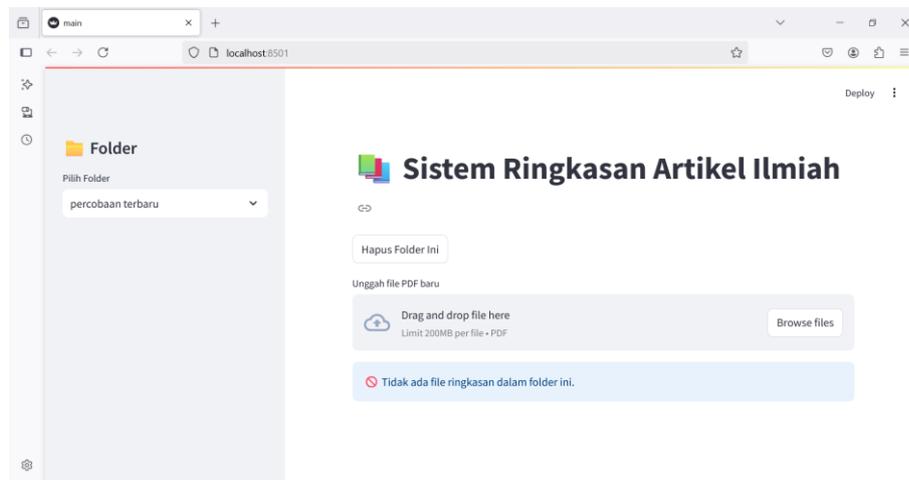
2. Halaman *sidebar* daftar *folder*



Gambar 4. 3 Halaman *Sidebar* Daftar *Folder*

Pada gambar 4. 3 diatas adalah hasil dari implementasi rancangan halaman *sidebar* untuk menampilkan daftar *folder*. Daftar *folder* dibuat dengan *select box* yang ditampilkan secara *dropdown*. Pilihan teratas yaitu “Pilih Folder” akan menuju pada halaman awal untuk membuat *folder* baru dan pilihan yang lain akan masuk pada *folder* yang dipilih dan akan masuk pada halaman untuk *upload file* dan ringkasan.

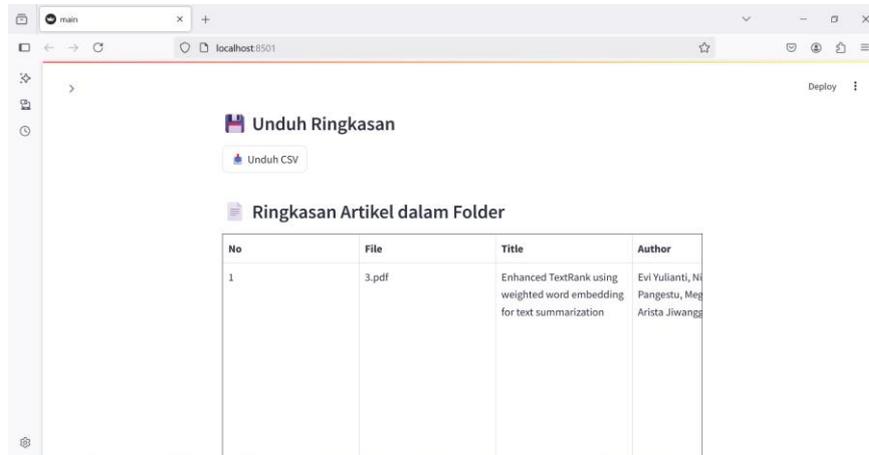
3. Halaman *upload file*



Gambar 4. 4 Halaman *Upload File*

Pada gambar 4. 4 diatas adalah halaman hasil implementasi rancangan *upload file*. Halaman ini menampilkan menu untuk meng-*upload file* dan menghapus *folder*. Untuk melakukan *upload file* pengguna dapat mengklik tombol *browse file* lalu sistem akan mengarahkan ke penyimpanan internal dalam komputer pengguna lalu pengguna dapat memilih *file pdf* yang ingin diringkaskan. Setelah itu sistem akan memproses *file* untuk dijadikan ringkasan.

4. Halaman hasil ringkasan



Gambar 4. 5 Halaman Hasil Ringkasan

Pada gambar 4. 5 di atas adalah hasil dari implementasi rancangan halaman hasil ringkasan. Halaman ini menampilkan tabel ringkasan artikel dalam *folder* dan menu untuk mengunduh ringkasan. Tabel ringkasan berisi nomor, metadata, serta ringkasan untuk setiap bagian sesuai struktur IMRAD. Menu unduh ringkasan digunakan untuk mengunduh kumpulan ringkasan yang telah dibuat dalam format csv.

4.4 Hasil Uji Coba Sistem

Pengujian sistem dilakukan untuk memastikan bahwa setiap komponen dalam sistem berfungsi dengan baik. Dalam penelitian ini pengujian sistem dilakukan untuk memastikan bahwa sistem dapat merangkum setiap struktur IMRAD (*Introduction, Method, Result, Discussion*) dan menyusun ringkasan dalam format tabel sesuai dengan struktur tersebut. Berikut adalah hasil pengujian sistem :

Tabel 4. 3 Hasil Pengujian Sistem

<i>Input</i>	Hasil yang diharapkan	<i>Output</i>	Kesimpulan
<i>File uji 1</i>	Sistem dapat Menampilkan ringkasan sesuai dengan struktur	Menampilkan ringkasan sesuai dengan struktur	Sistem berjalan dengan baik

<i>File uji 2</i>	Sistem dapat Menampilkan ringkasan sesuai dengan struktur	Menampilkan ringkasan sesuai dengan struktur	Sistem berjalan dengan baik
<i>File uji 3</i>	Sistem dapat Menampilkan ringkasan sesuai dengan struktur	Menampilkan ringkasan sesuai dengan struktur	Sistem berjalan dengan baik

Pada table 4.3 diatas menampilkan hasil dari pengujian sistem. Sistem diuji dengan tiga *file* yang hasilnya terlihat bahwa hasil yang diharapkan sesuai dengan *output*. Hasil uji menunjukkan bahwa sistem telah berjalan dengan baik.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian ini, dapat diambil kesimpulan tentang Sistem Ringkasan Otomatis untuk Kajian Literatur Menggunakan Metode *Bidirectional and Auto-Regresive Transformers* (BART), sistem yang dihasilkan mampu bekerja dengan baik dalam melakukan peringkasan. Hasil evaluasi menunjukkan bahwa BART dengan *Precision* 84.72%, *Recall* 86.72%, dan *F1-Score* 85.69% mampu meringkas dengan baik.

5.2 Saran

Saran untuk pengembangan sistem lebih lanjut adalah untuk meningkatkan kemampuan sistem dalam mendeteksi struktur IMRAD yang lebih akurat, melakukan peningkatan kualitas ringkasan, dan melakukan evaluasi sistem dengan berbagai metrik evaluasi. Deteksi struktur pada sistem ini mengandalkan kata kunci yang ada dalam program sehingga penggunaanya terbatas dan membutuhkan pengembangan lebih lanjut. Kualitas ringkasan juga masih memerlukan peningkatan dan dapat menggunakan evaluasi yang lebih bervariasi agar mendapatkan hasil yang lebih baik.

DAFTAR PUSTAKA

- Abraham, I., dan Supriyati, Y. (2022). Desain Kuasi Eksperimen Dalam Pendidikan: Literatur Review. *Jurnal Ilmiah Mandala Education*, 8(3), 2476–2482. <https://doi.org/10.58258/jime.v8i3.3800>
- Akbar, G. (2021). *Automatic Text Summarization Berita Bahasindonesia Menggunakan Metode Attentional Encoder Decoder*. <https://dspace.uui.ac.id/handle/123456789/37691%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/37691/14523091.pdf?sequence=1>
- Alfhi Saputra, M. (2021). Peringkat Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory. *e-Proceeding of Engineering : Vol.8, No.2 April 2021* |, 8(2), 3474–3488.
- Bao, T., Zhang, H., dan Zhang, C. (2025). Enhancing abstractive summarization of scientific papers using structure information. *Expert Systems with Applications*, 261(September 2024), 125529. <https://doi.org/10.1016/j.eswa.2024.125529>
- El-Kassas, W. S., Salama, C. R., Rafea, A. A., dan Mohamed, H. K. (2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing and Management*, 57(6). <https://doi.org/10.1016/j.ipm.2020.102264>
- Fakhrezi, M. F., Bijaksana, M. A., dan Huda, A. F. (2021). Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopedia. *Procedia Computer Science*, 179(2020), 391–398. <https://doi.org/10.1016/j.procs.2021.01.021>
- Hanna, M., dan Bojar, O. (2021). A Fine-Grained Analysis of BERTScore. *WMT 2021 - 6th Conference on Machine Translation, Proceedings*, 507–517.
- Hartawan, G., Maylawati, D. S., dan Uriawan, W. (2024). Bidirectional and Auto-Regressive Transformer (BART) for Indonesian Abstractive Text Summarization. *Jurnal Informatika Polinema*, 10(4), 535–542. <https://doi.org/10.33795/jip.v10i4.5242>
- Maesaroh, S., dan Yuniyanto, I. (2024). *Pembelajaran Mesin dan Kecerdasan*

Buatan : Teori dan Aplikasi Praktis (Nomor October).

- Magriyanti, A. A. (2021). Maximum Marginal Relevance Berbasis Boolean Model Pada Peringkasan Artikel Berita Pendek. *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, 1(3), 2827–7945. <http://journal.sinov.id/index.php/JUITIK>
- Meuschke, N., Jagdale, A., Spinde, T., Mitrović, J., dan Gipp, B. (2023). A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13972 LNCS, 383–405. https://doi.org/10.1007/978-3-031-28032-0_31
- Mohiuddin, K., Welke, P., Alam, M. A., Martin, M., Alam, M. M., Lehmann, J., dan Vahdati, S. (2023). Retention Is All You Need. *International Conference on Information and Knowledge Management, Proceedings, Nips*, 4752–4758. <https://doi.org/10.1145/3583780.3615497>
- Nan, F., Nallapati, R., Wang, Z., dos Santos, C. N., Zhu, H., Zhang, D., McKeown, K., dan Xiang, B. (2021). Entity-level factual consistency of abstractive text summarization. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 2727–2733. <https://doi.org/10.18653/v1/2021.eacl-main.235>
- Özkan Çelik, A. E., dan Al, U. (2024). Structured abstract generator (SAG) model: analysis of IMRAD structure of articles and its effect on extractive summarization. *International Journal on Digital Libraries*, 787–801. <https://doi.org/10.1007/s00799-024-00402-8>
- Pandu, J. (2024). *CSWin Transformer-CNN Encoder and Multi-Head Self-Segmentation*. 5(1), 57–69.
- Pangesti, A., Ulfah, B., dan Hartono, R. (2023). Investigating the Implementation of IMRAD Structure in Abstracts of Undergraduate Students' Theses. *Pedagogy: Journal of English Language Teaching*, 11(2), 109. <https://doi.org/10.32332/joelt.v11i2.7144>
- Putra, T. I. Z. M., Suprpto, S., dan Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding

- dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 1–28.
<https://doi.org/10.35912/jisted.v1i1.1509>
- Rahmat, A., Ntelu, A., dan Mirnawati, M. (2023). Teknik Keterampilan Menulis Artikel Model IMRAD dengan standar Sinta, Scopus dan WOS. *AKSARA: Jurnal Ilmu Pendidikan Nonformal* 1383, 9(02), 1383–1390.
<http://ejurnal.pps.ung.ac.id/index.php/Aksara>
- Ridwan, M., AM, S., Ulum, B., dan Muhammad, F. (2021). Pentingnya Penerapan Literature Review pada Penelitian Ilmiah. *Jurnal Masohi*, 2(1), 42.
<https://doi.org/10.36339/jmas.v2i1.427>
- Thoyyibah, T., Haryono, W., Zailani, A. U., Djaksana, Y. M., Rosmawarni, N., dan Arianti, N. D. (2023). Transformers in Machine Learning: Literature Review. *Jurnal Penelitian Pendidikan IPA*, 9(9), 604–610.
<https://doi.org/10.29303/jppipa.v9i9.5040>
- Ulker, M., dan Ozer, A. B. (2024). Abstractive Summarization Model for Summarizing Scientific Article. *IEEE Access*, 12(July), 91252–91262.
<https://doi.org/10.1109/ACCESS.2024.3420163>
- Yadav, M., Sahu, N. K., Chaturvedi, M., Gupta, S., dan Lone, H. R. (2024). *Fine-tuning Large Language Models for Automated Diagnostic Screening Summaries*. <http://arxiv.org/abs/2403.20145>
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., dan Artzi, Y. (2020). Bertscore: Evaluating Text Generation With Bert. *8th International Conference on Learning Representations, ICLR 2020*, 1–43.