PENERAPAN *RETRIEVAL-AUGMENTED GENERATION* (RAG) DAN MODEL INDOT5 UNTUK OTOMATISASI RINGKASAN LITERATUR BERBAHASA INDONESIA

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 :
AISYAH MUFIDAH
NIM 32602100001

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025

FINAL PROJECT
APPLICATION OF RETRIEVAL-AUGMENTED GENERATION (RAG) AND INDOT5 MODEL FOR AUTOMATICATION OF INDONESIAN LANGUAGE LITERATURE SUMMARY

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 :

AISYAH MUFIDAH

NIM 32602100001

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

LEMBAR PENGESAHAN
TUGAS AKHIR

PENERAPAN *RETRIEVAL-AUGMENTED GENERATION* (RAG) DAN MODEL INDOT5 UNTUK OTOMATISASI RINGKASAN LITERATUR BERBAHASA INDONESIA

AISYAH MUFIDAH
NIM 32602100001

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal 14 Juli 2025

TIM PENGUJI UJIAN SARJANA:

Bagus Satrio Waluyo Poetro, S.Kom., M.Cs NIDN. 1027118801 (Ketua Penguji)		06-08-2025
Moch. Taufik, S.T., M.IT NIDN. 0622037502 (Anggota Penguji)		06-08-2025
Badie'ah, S.T., M.Kom NIDN. 0619018701 (Pembimbing)		06-08-2025

Semarang, 06 Agustus 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T., M.IT
NIDN. 0622037502

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Aisyah Mufidah

NIM : 32602100001

Judul Tugas Akhir : PENERAPAN *RETRIEVAL - AUGMENTED GENERATION* (RAG) DAN MODEL INDOT5 UNTUK OTOMATISASI RINGKASAN LITERATUR BERBAHASA INDONESIA

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 siapa pun 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, 30 Juli 2025

Yang Menyatakan

Aisyah Mufidah

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama	: Aisyah Mufidah
NIM	: 32602100001
Program Studi	: Teknik Informatika
Fakultas	: Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : PENERAPAN *RETRIEVAL-AUGMENTED GENERATION* (RAG) DAN MODEL INDOT5 UNTUK OTOMATISASI RINGKASAN LITERATUR BERBAHASA INDONESIA

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/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 30 juli 2025

Yang Menyatakan

Aisyah Mufidah

KATA PENGANTAR

Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Penerapan *Retrieval-Augmented Generation* (RAG) dan Model IndoT5 untuk Otomatisasi Ringkasan Literatur Berbahasa Indonesia" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan 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, S.T., M.T.
3. Dosen pembimbing I penulis Badie'ah, S.T., M.Kom yang telah meluangkan waktu, memberi ilmu dan memberikan banyak nasehat dan saran.
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan rendah hati, penulis menyadari bahwa laporan masih memiliki banyak kekurangan dalam hal kuantitas, kualitas, dan ilmu pengetahuan. Oleh karena itu, penulis mengharapkan kritikan dan saran yang membangun untuk membantu laporan ini menjadi lebih baik di masa depan..

Semarang, 30 Juli 2025

Aisyah Mufidah

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

ABSTRAK

Peningkatan jumlah publikasi ilmiah di Indonesia menimbulkan tantangan dalam menyaring dan memahami literatur secara efisien. Proses kajian literatur manual memerlukan waktu yang panjang, rentan terhadap bias kognitif, dan sulit mengikuti perkembangan riset terkini. Untuk mengatasi permasalahan ini, penelitian ini mengembangkan sistem otomatisasi ringkasan literatur yang mengintegrasikan *Retrieval-Augmented Generation* (RAG) dengan model IndoT5 dan pendekatan struktur IMRAD (*Introduction, Methods, Results*, and *Discussion*). Sistem menggabungkan proses peringkasan menggunakan IndoT5, *indexing* berbasis FAISS, serta *embedding IndoBERT* untuk pencarian dokumen yang relevan secara semantik. Evaluasi sistem menggunakan metrik *BERTScore* menunjukkan kualitas ringkasan dengan skor *precision* 0.828, *recall* 0.881, dan *F1-score* 0.854. Penilaian menggunakan *LLM-as-a-Judge* dengan model *LLaMA-3-70B* menghasilkan skor rata-rata 4.47 dari skala 5 untuk aspek relevansi, kebenaran, dan kelengkapan respons. Hasil penelitian membuktikan bahwa sistem mampu menghasilkan ringkasan yang informatif dan kontekstual, serta mempercepat proses kajian literatur berbahasa Indonesia secara signifikan.
Kata Kunci: IndoT5, Literatur Akademik, Otomatisasi Ringkasan, *Retrieval-Augmented Generation*, *Text Summarization*

ABSTRACT

The increasing number of scientific publications in Indonesia poses challenges in efficiently filtering and understanding the literature. The manual literature review process is time-consuming, prone to cognitive bias, and makes it difficult to keep up with the latest research developments. To address these issues, this study developed an automated literature summary system that integrates Retrieval-Augmented Generation (RAG) with the IndoT5 model and the IMRAD (Introduction, Methods, Results, and Discussion) structure approach. The system combines summarization using IndoT5, FAISS-based indexing, and IndoBERT embedding for semantically relevant document retrieval. System evaluation using the BERTScore metric demonstrated summary quality with a precision score of 0.828, recall of 0.881, and F1-score of 0.854. Assessment using LLM-as-a-Judge with the LLaMA-3-70B model resulted in an average score of 4.47 out of 5 for the aspects of relevance, correctness, and completeness of responses. The results demonstrate that the system is capable of producing informative and contextual summaries and significantly accelerates the process of reviewing Indonesian-language literature.
Keywords: IndoT5, Academic Literature, Summarization Automation, Retrieval-Augmented Generation, Text Summarization

BAB I
PENDAHULUAN

1.1 Latar Belakang

Peningkatan pesat publikasi ilmiah dalam beberapa tahun terakhir telah membawa dampak signifikan bagi dunia riset. Di Indonesia, integrasi lembaga riset ke dalam BRIN mendorong percepatan output penelitian. Handoyo dkk., 2024 menunjukkan bahwa dalam periode 2015–2021, terdapat 12.209 dokumen dari institusi Indonesia di Scopus dengan pertumbuhan rata-rata 30 % per tahun. Bahkan, dalam dua tahun berikutnya (2022–2023), jumlahnya melonjak lagi sebesar 8.081 dokumen dengan laju 36 %. Namun, peningkatan ini menimbulkan tantangan baru: bagaimana menyaring dan memahami literatur dalam jumlah besar secara efisien.

Kajian literatur adalah fondasi krusial dalam proses ilmiah, berperan dalam merumuskan tujuan penelitian, menentukan metodologi, hingga menyusun kerangka teoritis. Sayangnya, proses ini kerap menyita waktu, memerlukan pembacaan intensif, dan rentan terhadap bias kognitif (Ridwan dkk., 2021). Dengan jumlah artikel yang terus bertambah, kemampuan peneliti untuk menganalisis literatur secara manual menjadi terbatas. Tantangan ini makin kompleks ketika literatur ditulis dalam bahasa lokal, seperti Bahasa Indonesia, yang belum sepenuhnya terakomodasi oleh sistem pemrosesan bahasa alami (Cahyawijaya dkk., 2021; Wilie dkk., 2020).

Kemajuan *Natural Language Proccessing* (NLP) telah menghasilkan berbagai model berbasis *Transformer* seperti *BERT* dan *GPT*. Salah satu pendekatan inovatif yang muncul adalah *Retrieval-Augmented Generation* (RAG), yang menggabungkan mekanisme pengambilan informasi (*retrieval*) dan pembangkitan teks (*generation*). Pendekatan ini terbukti mampu mengurangi fenomena *hallucination* dan menghasilkan ringkasan yang lebih relevan (Cheng dkk., 2025; Gupta & Ranjan, 2024). Namun, pengembangan RAG masih berfokus pada bahasa Inggris, dengan sumber data dan model yang tidak selalu sesuai dengan konteks lokal (Hahsler, 2023; Jaber & Gérard, 2025).

Bahasa Indonesia, dengan struktur morfologis kompleks dan variasi dialek yang beragam, sementara ketersediaan korpus berskala besar masih terbatas (Wilie dkk., 2020). Kondisi ini berdampak pada penurunan performa RAG ketika diterapkan pada dokumen ilmiah berbahasa Indonesia (Muhammad dkk., 2025). Selain itu, literatur Indonesia belum sepenuhnya mudah diakses, sementara mayoritas alat bantu literatur digital lebih mendukung konten berbahasa Inggris.

Di sisi lain, struktur artikel ilmiah yang umumnya mengikuti pola IMRAD (*Introduction, Methods, Results, and Discussion*) membuka peluang untuk menyusun sistem ringkasan otomatis yang lebih terarah. Sakti Wiradinata dkk., 2024 menyatakan bahwa *summarization* berbasis IMRAD mampu mengekstraksi elemen penting seperti variabel, metodologi, dan temuan utama. Dengan memanfaatkan segmentasi ini, sistem peringkasan dapat dibuat lebih sistematis dan informatif.

Sebagai solusi atas keterbatasan penerapan RAG pada dokumen ilmiah berbahasa Indonesia yang disebabkan oleh minimnya model bahasa lokal dan rendahnya ketersediaan korpus berkualitas serta untuk memanfaatkan potensi segmentasi berbasis IMRAD, integrasi pendekatan RAG dengan model bahasa lokal seperti IndoT5 menjadi pilihan yang menjanjikan. IndoT5 merupakan model transformer adaptif untuk Bahasa Indonesia yang unggul dalam memahami konteks lokal (Yani dkk., 2024). Penelitian ini mengusulkan sistem yang melakukan peringkasan otomatis berdasarkan struktur IMRAD, lalu mengindeks hasil ringkasan tersebut untuk menjawab pertanyaan pengguna secara kontekstual. Pendekatan ini mengombinasikan kekuatan IndoT5 dalam *summarization* dengan efisiensi *indexing* berbasis FAISS dan representasi teks dari *IndoBERT*.

Penelitian ini bertujuan untuk merancang dan mengevaluasi sistem RAG dengan model peringkas IndoT5 yang dapat mengotomatisasi pencarian dan peringkasan literatur ilmiah berbahasa Indonesia. Sistem ini diharapkan dapat mempercepat proses kajian literatur, meningkatkan akurasi pemahaman konten ilmiah, serta memperkuat pemanfaatan teknologi NLP lokal dalam ekosistem riset nasional.

1.2 Perumusan Masalah

1. Bagaimana penerapan metode *Retrieval-Augmented Generation* (RAG) untuk mendukung proses pencarian dan peringkasan literatur ilmiah berbahasa Indonesia secara otomatis?
2. Bagaimana pemanfaatan model IndoT5 dalam menghasilkan ringkasan literatur ilmiah yang relevan dan terstruktur berdasarkan pendekatan IMRAD?

1.3 Pembatasan Masalah

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

1. Lingkup dataset adalah jurnal di bidang *Artificial Intelligent* (AI) yang berfokus pada 3 topik yaitu *Supervised Learning*, *Unsupervised Learning* dan *Reinforcement Learning*.
2. Sistem hanya menangani dokumen ilmiah berbahasa Indonesia dalam format PDF yang di *download* manual dari platform google scholar.
3. Proses *preprocessing* dokumen hanya memproses teks, tidak mempertimbangkan konten visual seperti tabel atau gambar.
4. Model *summarization* yang digunakan adalah IndoT5 *pre-trained* tidak dilakukan *fine-tuning*.

1.4 Tujuan

Penelitian ini bertujuan untuk menerapkan metode *Retrieval-Augmented Generation* (RAG) dan model IndoT5 untuk mengotomatisasi ringkasan literatur ilmiah berbahasa Indonesia.

1.5 Manfaat

Manfaat dari penelitian ini adalah untuk memudahkan para peneliti dalam melakukan kajian literatur ilmiah berbahasa Indonesia melalui sistem ringkasan otomatis berbasis RAG dan IndoT5.

1.6 Sistematika Penulisan

Sistematika penulisan berikut akan digunakan oleh penulis saat membuat laporan tugas akhir:

Tabel 1. 1 Sistematika Penulisan

BAB I	:	PENDAHULUAN Menguraikan latar belakang masalah agar dapat dipilih sebagai judul penelitian, perumusan masalah untuk menguraikan masalah yang akan dipecahkan, batasan masalah agar ruang lingkup masalah tidak terlalu luas, tujuan yang ingin dicapai, manfaat dari pembuatan sistem, dan sistematika penulisan yang mencakup uraian dari penulisan laporan tugas akhir.
BAB II	:	TINJAUAN PUSTAKA DAN DASAR TEORI Sebagai acuan untuk menyusun Tugas Akhir, mempelajari tinjauan literatur dan dasar teori yang digunakan untuk mendukung analisis masalah.
BAB III	:	METODE PPENELITIAN Memuat tentang analisis proses sistem chatbot sebagai layanan tugas akhir menggunakan model BERT. Analisa ini mencakup perancangan sistem dan desain antarmuka.
BAB IV	:	HASIL DAN PEMBAHASAN Memuat hasil pengujian program, pembahasan tentang prosedur kerja program, dan tampilannya.
BAB V	:	KESIMPULAN DAN SARAN Memuat tentang kesimpulan dan saran dari penulis terhadap penelitian yang telah dilakukan.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka bertujuan untuk mengidentifikasi, menelaah, dan merangkum hasil penelitian sebelumnya yang relevan dengan topik tugas akhir ini. Fokus dari studi-studi terdahulu adalah pengembangan sistem *summarization* otomatis menggunakan model T5 dan pendekatan *Retrieval-Augmented Generation* (RAG), baik dalam konteks bahasa Indonesia maupun dalam sistem berbasis chatbot atau pencarian informasi.

Berikut adalah ringkasan dari beberapa literatur yang menjadi rujukan dalam penelitian ini:

Tabel 1. 2 Hasil penelitian terdahulu

No.	Judul	Dataset /Metode	Hasil Penelitian
1.	Itsnaini et al. (2023)	Dataset berita Indonesia, model T5 *pre-trained* (220M), *fine-tuning* ATS	Penelitian menunjukkan model T5 efektif melakukan *abstractive summarization* pada teks berita. Evaluasi menggunakan ROUGE menghasilkan nilai ROUGE-1: 0.68, ROUGE-2: 0.61, dan ROUGE-L: 0.65. Namun, terdapat kelemahan berupa kesalahan dalam ringkasan referensi dan keterbatasan generalisasi ke domain akademik.
2.	Bahari & Dewi (2024)	Dataset IndoSum (~19.000 pasangan	Model T5 diuji pada teks berita dan menunjukkan kemampuan dalam menyusun ringkasan koheren. Skor ROUGE-1 mencapai 0.61 dan

No.	Judul	Dataset /Metode	Hasil Penelitian
		berita), T5 abstraktif	ROUGE-2 sebesar 0.51. Hal ini menegaskan potensi arsitektur Transformer dalam memproses teks Bahasa Indonesia untuk tugas *summarization*.
3.	Yani et al. (2025)	Multiformat input (teks, dokumen, web, gambar), model T5	Penelitian mengembangkan sistem peringkasan teks multiformat berbasis T5. Hasil evaluasi mencapai ROUGE rata-rata 0.87. Sistem mendukung berbagai format input dan output, menunjukkan fleksibilitas model dalam berbagai konteks penggunaan.
4.	Purnama & Utami (2023)	Dokumen akademik Indonesia, T5	Fokus penelitian pada adaptasi morfologis dan sintaksis Bahasa Indonesia. Model T5 dikustomisasi agar lebih mampu memahami struktur teks ilmiah. Meskipun tidak mencantumkan metrik numerik, penelitian ini menyoroti pentingnya pendekatan berbasis bahasa lokal untuk hasil yang lebih akurat.
5.	Bazzi et al. (2024)	Open-domain QA dengan RAG, GPT evaluator, metrik:	Penelitian ini menyajikan sistem RAG-end2end yang mampu mengurangi *hallucination* dan meningkatkan relevansi jawaban. Evaluasi dilakukan terhadap

No.	Judul	Dataset /Metode	Hasil Penelitian
		ROUGE, cosine similarity	kualitas jawaban pada sistem ODQA, dengan metode evaluasi campuran dan *real-time retrieval*. Hasil menunjukkan peningkatan signifikan pada kualitas jawaban dibanding model non-RAG.
6.	Han et al. (2024)	RAG+LLM untuk otomasi *Systematic Literature Review* (SLR)	Mengusulkan kerangka kerja berbasis RAG yang menangani empat tahap SLR: *retrieval, screening*, *data extraction*, *synthesis*. Sistem meningkatkan efisiensi dan akurasi SLR secara signifikan. Tantangan masih ada dalam pengolahan data multimodal dan adaptasi lintas bidang ilmu.
7.	Albert & Voutama (2025)	Local RAG dengan ChromaDB & Ollama, metode RAD	Sistem chatbot lokal dirancang untuk memproses PDF secara mandiri tanpa koneksi cloud. Model RAG digunakan untuk retrieval internal. Evaluasi ROUGE-L menunjukkan skor 0.85, menandakan relevansi jawaban tinggi. Kendala utama adalah keterbatasan kecepatan dan dukungan multi-bahasa.
8.	Vidivelli et al. (2024)	LangChain, RAG, LLMs	Penelitian mengembangkan chatbot efisien berbasis LLM terintegrasi dengan *web scraping*. Hasil

No.	Judul	Dataset /Metode	Hasil Penelitian
		+ LoRA, QLoRA	menunjukkan peningkatan akurasi respons dan pengalaman pengguna. Kombinasi teknologi efisien memungkinkan pemrosesan pertanyaan kompleks dengan waktu respons singkat.
9.	Samudra et al. (2025)	Bahasa Indonesia, model RAG dengan IndoT5 dan *embedding* lokal	Fokus penelitian pada pengembangan RAG untuk teks akademik Bahasa Indonesia. Tantangan utama meliputi keterbatasan korpus dan kompleksitas linguistik. Penelitian menunjukkan peningkatan akurasi sistem dengan pendekatan IMRAD dan model lokal.
10.	Cheng et al. (2025)	*Knowledge-intensive* NLP*, model* RAG *vs baseline, metrik: Accuracy, F1-score,* MRR	Penelitian menunjukkan bahwa model RAG, yang menggabungkan *retrieval* dan *generation*, memberikan hasil lebih akurat dalam tugas NLP berbasis pengetahuan dibanding model baseline. Evaluasi pada dataset QA menunjukkan akurasi mencapai 74.5%, F1-score meningkat signifikan, dan nilai MRR membuktikan kualitas urutan hasil yang baik.

Melalui kajian ini, dapat disimpulkan bahwa penggunaan model T5 untuk *summarization* teks berbahasa Indonesia telah terbukti efektif, terlebih jika disesuaikan dengan karakteristik linguistik lokal. Sementara itu, pendekatan RAG menunjukkan potensi besar dalam pengembangan sistem pencarian informasi berbasis teks dan chatbot, terutama jika diintegrasikan dengan LLM modern.

2.2 Dasar Teori

2.2.1 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah cabang *Artificial Intelligent* (AI) yang bertujuan untuk memungkinkan komputer memahami, menghasilkan, dan merespons bahasa manusia secara alami. NLP menggabungkan teknik pembelajaran mesin dengan linguistik untuk menganalisis struktur dan makna bahasa manusia. Jurafsky & Martin (2022) menjelaskan bahwa NLP mencakup berbagai tugas mulai dari analisis sentimen, peringkasan teks, hingga pengenalan entitas bernama, yang semuanya memungkinkan pemrosesan otomatis terhadap teks dalam skala besar.

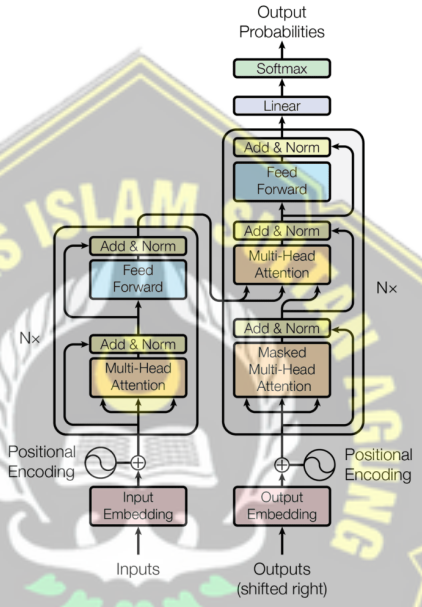
Aplikasi utama NLP Menurut (Cambria & White, 2014), NLP memiliki aplikasi yang luas, termasuk:

1. Penerjemahan Mesin: Mengkonversi teks dari satu bahasa ke bahasa lain, seperti pada *Google Translate.*
2. Sistem Tanya Jawab: Memberikan jawaban atas pertanyaan berdasarkan teks masukan, seperti dalam chatbot berbasis AI.
3. Peringkasan Teks: Menghasilkan ringkasan dari dokumen panjang untuk menyederhanakan informasi.
4. Analisis Sentimen: Menilai opini atau emosi dari teks, seperti ulasan produk atau posting media sosial.

Dalam penelitian ini, NLP menjadi fondasi utama untuk berbagai proses, mulai dari ekstraksi teks dari PDF, peringkasan otomatis menggunakan IndoT5, hingga *embedding* teks dengan IndoBERT. Dengan NLP, sistem dapat memahami dan mengolah dokumen berbahasa Indonesia secara otomatis.

2.2.2 *Transformers*

Transformers adalah arsitektur *deep learning* yang diperkenalkan oleh (Vaswani dkk., 2017), dirancang untuk menangani urutan data dengan efisiensi yang jauh lebih tinggi dibandingkan model sekuensial seperti RNN atau LSTM. Transformers menggunakan mekanisme perhatian (*attention mechanism*) yang memungkinkan model untuk menimbang elemen dalam urutan berdasarkan relevansinya, tanpa harus memproses data secara berurutan.



Gambar 2. 1 Model Arsitektur Transformers (Vaswani et. al, 2017)

Mekanisme inti dari *Transformers* adalah *self-attention*, yang memungkinkan model memahami hubungan antar kata dalam teks. Sebagai contoh, kata “bank” dalam kalimat “I went to the river bank” dapat dibedakan dari “bank” dalam “I deposited money in the bank” dengan konteks yang diberikan oleh mekanisme ini. Menurut Tay et al. (2021), *Transformer* telah menjadi dasar dari hampir semua model NLP modern, termasuk BERT, GPT, dan T5.

Dalam konteks penelitian, *Transformers* menjadi fondasi bagi IndoT5 untuk peringkasan teks dan IndoBERT untuk *embedding*, serta digunakan dalam integrasi *Retrieval-Augmented Generation* (RAG). Kombinasi ini memungkinkan pemrosesan teks akademik dalam bahasa Indonesia dengan efisiensi tinggi.

2.2.3 IndoBERT untuk Embedding

IndoBERT adalah model BERT (*Bidirectional Encoder Representations from Transformers*) yang diadaptasi untuk bahasa Indonesia. Model ini dilatih menggunakan korpus Indonesia besar, seperti Wikipedia dan berita lokal, untuk menghasilkan representasi semantik teks yang berkualitas tinggi. Menurut (Koto Jey Han Lau Timothy Baldwin, 2021.), IndoBERT menunjukkan keunggulan dalam tugas-tugas berbasis pemahaman teks, seperti klasifikasi dan pencarian informasi.

IndoBERT digunakan untuk menghasilkan *embedding* teks yang akan dimanfaatkan dalam proses pencarian dokumen pada RAG. *Embedding* ini memungkinkan sistem untuk menemukan dokumen yang relevan berdasarkan kesamaan semantik.

2.2.4 Ekstraksi PDF menggunakan pymupdf

Ekstraksi PDF adalah proses mengubah konten dalam dokumen PDF menjadi format yang dapat dibaca dan diolah oleh komputer, seperti teks atau data tabel. Hal ini penting karena PDF adalah format yang paling umum digunakan untuk mendistribusikan dokumen ilmiah dan akademik. Menurut (Hong dkk., 2021), PDF sering kali mengandung elemen kompleks seperti grafik, tabel, dan anotasi, sehingga memerlukan algoritma khusus untuk mengekstrak informasi yang relevan secara akurat.

2.2.5 Struktur IMRAD untuk *Summarization*

IMRAD adalah akronim dari *Introduction, Methods, Results, and Discussion* yang merupakan format standar dalam penulisan artikel ilmiah. Menurut (Sollaci & Pereira, 2021) format ini dirancang untuk menyederhanakan penyajian informasi ilmiah, sehingga pembaca dapat dengan mudah menemukan bagian yang relevan dengan kebutuhannya.

Struktur IMRAD digunakan sebagai dasar dalam proses peringkasan. Dokumen ilmiah dipecah menjadi bagian-bagian sesuai format ini, lalu diringkas secara terpisah menggunakan IndoT5 untuk mempertahankan esensi dari setiap bagian.

2.2.6 IndoT5 untuk *Summarization*

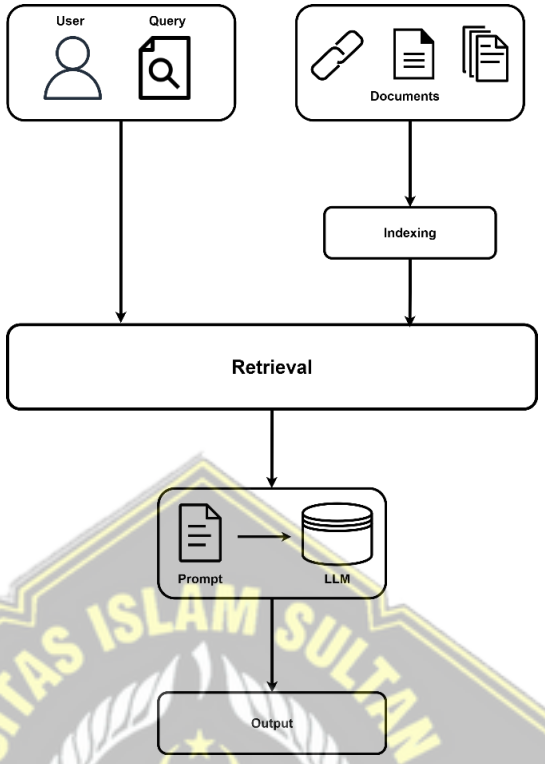
IndoT5 adalah varian lokal dari model T5 (*Text-to-Text Transfer Transformer*) yang dikembangkan khusus untuk bahasa Indonesia. T5 pada dasarnya mengubah semua tugas NLP, seperti klasifikasi, peringkasan, dan terjemahan, menjadi masalah generasi teks. Menurut (Raffel dkk., 2020), T5 menggunakan pendekatan berbasis *encoder-decoder* yang memungkinkan fleksibilitas dalam menangani berbagai tugas NLP.

IndoT5 dirancang untuk menangani tantangan unik bahasa Indonesia, seperti morfologi kompleks dan struktur sintaksis yang berbeda. Model ini dilatih menggunakan korpus teks Indonesia yang besar, sehingga dapat memahami konteks lokal dengan lebih baik. Setiawan et al. (2023) menunjukkan bahwa IndoT5 mencapai kinerja peringkasan yang unggul dibandingkan model generatif lainnya dalam tugas-tugas berbahasa Indonesia.

IndoT5 digunakan untuk melakukan peringkasan otomatis pada setiap bagian IMRAD dari dokumen ilmiah. Proses ini bertujuan untuk menyaring informasi penting dan relevan sebelum diintegrasikan ke dalam pipeline RAG.

2.2.7 *Retrieval Augmented Generation* (RAG)

Retrieval Augmented Generation (RAG) adalah metode untuk meningkatkan model bahasa besar (LLMs) melalui penggabungan data informasi tambahan dari sumber pengetahuan eksternal. Hal ini dapat mengurangi halusinasi dan memungkinkan respons LLM yang lebih tepat dan sadar konteks Metode ini sangat berguna dalam tugas-tugas seperti menjawab pertanyaan, meringkas dokumen, dan agen percakapan (Miao dkk., 2024).



Gambar 2. 2 Arsitektur RAG

Proses *Indexing* adalah proses yang dilakukan secara *offline*. *Indexing* dilakukan dengan membersihkan dan mengekstrak data awal terlebih dahulu, dan kemudian mengubah berbagai format file seperti PDF, HTML, dan Word menjadi teks sederhana. Untuk mengatasi kendala konteks model bahasa, teks ini dibagi menjadi bagian yang lebih kecil dan lebih mudah diatur, proses tersebut disebut juga dengan *chunking*. Potongan-potongan ini kemudian diubah menjadi representasi vektor melalui penggunaan model *embedding*. Setelah itu, indeks dibuat untuk menyimpan potongan teks ini dan menyematkan vektornya sebagai pasangan kunci nilai yang memungkinkan pencarian yang tepat (Miao dkk.,, 2024).

Proses *Retrieval* adalah proses mengambil konten atau informasi tambahan yang relevan dari sumber pengetahuan eksternal menggunakan kueri pengguna. Untuk melakukan proses ini, model pengkodean akan digunakan untuk memproses kueri pengguna yang menghasilkan penyematan semantik. Selanjutnya, pencarian kesamaan dilakukan pada *database* vektor untuk menemukan objek terdekat (Miao dkk.,, 2024).

Proses Generasi adalah proses menggabungkan perintah masukan dengan dokumen atau informasi yang diambil dari sumber pengetahuan lainnya. Dalam proses generasi bahasa diperkaya dilakukan dengan menggunakan informasi yang ditemukan dalam tahap *retrieval*. Model generasi bahasa seperti GPT (*Generative Pre-trained Transformer*) digunakan untuk membuat jawaban atau konten baru berdasarkan informasi yang ditemukan dalam tahap *retrieval*. Pendekatan RAG telah terbukti berhasil dalam berbagai tugas proses pengolahan bahasa, seperti menjawab pertanyaan, membuat teks informatif, dan mendukung pengambilan keputusan berbasis informasi. Dengan integrasi kedua tahap ini, sistem dapat menghasilkan jawaban yang lebih akurat, informatif, dan relevan terhadap permintaan pengguna, terutama dalam kasus di mana informasi yang mendalam atau spesifik diperlukan dari kumpulan dokumen yang sangat besar (Miao dkk.,, 2024).

Dalam pemrosesan bahasa alami, "*chunking*" mengacu pada pembagian teks menjadi "potongan" kecil, ringkas, dan bermakna. Sistem RAG dapat menemukan konteks yang relevan dalam potongan teks yang lebih kecil lebih cepat dan akurat daripada dalam dokumen yang lebih besar. Potongan yang lebih kecil menangkap lebih sedikit konteks, tetapi mungkin tidak sepenuhnya menangkap konteks yang diperlukan, meskipun potongan yang lebih besar dapat menangkap lebih banyak konteks dan membutuhkan lebih banyak waktu dan biaya komputasi untuk diproses. Salah satu cara untuk menyeimbangkan kedua kendala ini adalah dengan menggunakan potongan yang tumpang tindih. Metode ini memungkinkan kueri untuk mengumpulkan data yang relevan di banyak vektor untuk menghasilkan tanggapan kontekstual yang tepat.

2.2.8 Indexing & FAISS

FAISS (*Facebook AI Similarity Search*) adalah pustaka perangkat lunak yang dirancang untuk pencarian kesamaan berbasis vektor pada data besar. FAISS menggunakan algoritma pengindeksan canggih untuk mengurangi waktu pencarian, terutama dalam basis data besar. (Johnson et al., 2021) mencatat bahwa FAISS mampu menangani jutaan vektor dengan efisiensi tinggi, menjadikannya pilihan utama untuk sistem berbasis embedding.

FAISS digunakan untuk mengindeks *embedding* teks yang dihasilkan oleh IndoBERT. Proses ini memungkinkan sistem untuk menemukan dokumen dengan relevansi semantik tinggi dalam waktu singkat.

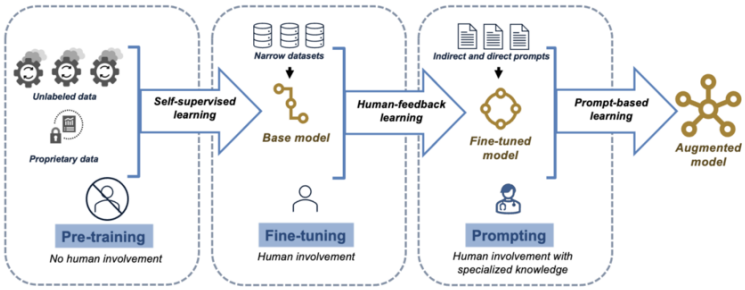
2.2.9 Large Language Models(LLM)

Large Language Models (LLM) adalah model pembelajaran mesin berskala besar yang dilatih pada kumpulan data masif untuk memahami, memprediksi, dan menghasilkan teks. Model ini menggunakan arsitektur *Transformer*, yang dirancang untuk menangani kompleksitas pola bahasa alami. (Brown dkk., 2020) memperkenalkan GPT-3 sebagai salah satu LLM dengan 175 miliar parameter, yang mampu melakukan tugas-tugas kompleks melalui pendekatan *few-shot learning*.

Dalam proses pelatihan model LLM, Langkah pertama dikenal sebagai pra-pelatihan yang merupakan sebuah pendekatan pengawasan mandiri yang melibatkan pelatihan pada kumpulan besar data tidak berlabel, seperti teks internet, kode Github, postingan media sosial, Wikipedia, dan Books Corpus. Tujuan dari pelatihan ini adalah untuk memprediksi kata berikutnya dari sebuah kalimat, dan dari proses ini membutuhkan sumber daya yang banyak. Sebelum dimasukkan ke dalam model, diperlukan konversi teks menjadi token, sehingga dari langkah ini akan menghasilkan model dasar yang hanya sebagai model penghasil Bahasa umum, dan tidak memiliki kapasitas untuk tugas-tugas yang berbeda.

Langkah kedua dikenal sebagai *fine-tuning*, dimana model akan dilatih lebih lanjut terhadap kumpulan data yang lebih sempit seperti transkip medis untuk aplikasi layanan kesehatan atau ringkasan hukum untuk bot asisten hukum. Proses *fine-tuning* dapat ditingkatkan dengan pendekatan Al Konstutusional, selain itu *fine-tuning* dapat ditingkatkan dengan *reward training*, dimana manusia dapat menilai kualitas keluaran beberapa model, serta pendekatan pembelajaran penguatan dari umpan balik manusia. Teknik lain yang menjanjikan adalah SLAM (*Self-tuning Language Model*), yaitu metode penyetelan *prompt* (*prompt tuning*) yang secara otomatis mengoptimalkan *prompt* untuk tugas tertentu tanpa memerlukan penyesuaian parameter besar. Pendekatan ini lebih hemat biaya dan terbukti dapat meningkatkan kinerja model dalam tugas-tugas spesifik seperti

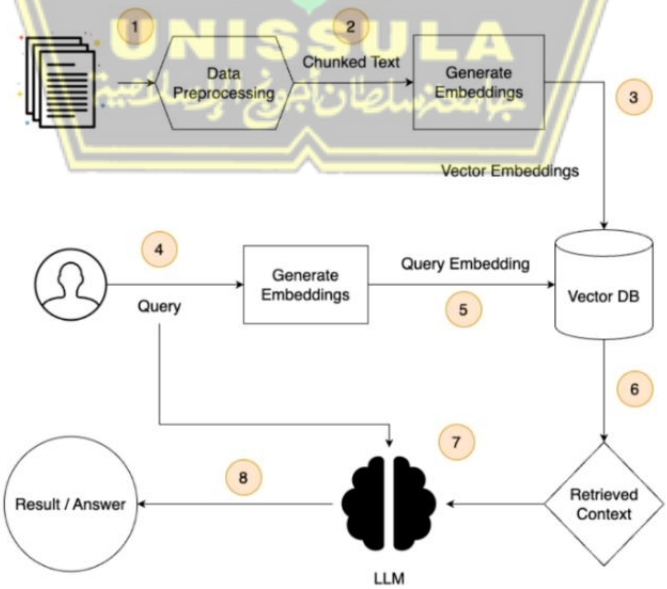
domain medis (Omiye dkk., 2023). Gambar 2 1 berikut menunjukkan serangkaian proses sebelum terbentuknya LLM.



Gambar 2. 3 Alur Pelatihan LLM

LLM belajar dari setiap masukan yang lebih terfokus pada tahap proses pelatihan. Pra- pelatihan, dimana LLM akan dilatih tentang campuran data tidak berlabel serta data kepemilikan tanpa pengawasan manusia. *Fine-tuning*, dimana terdiri atas kumpulan data yang lebih sempit serta umpan balik manusia dimasukkan sebagai masukan ke model dasar. Sehingga dengan model yang telah disempurnakan dapat memasuki tahap tambahan, dimana manusia dengan pengetahuan khusus menerapkan teknik dorongan yang dapat mengubah LLM menjadi sebuah model yang diperbesar untuk melakukan tugas-tugas khusus.

Model yang sudah di *fine-tuning* dari fase kedua ini adalah model yang diterapkan dalam aplikasi fleksibel seperti *chatbot*.



Gambar 2. 4 LLM dalam sistem RAG

Gambar 2.4 menunjukkan implementasi model LLM dalam suatu sistem RAG melibatkan beberapa proses penting di dalamnya, seperti *prepocessing* sebuah teks atau data agar lebih bisa dikenali, kemudian vektorisasi dari teks menjadi vektor agar bisa terbaca. Proses ini berlaku dalam tahap pengumpulan data yang akhirnya disimpan dalam *vectorbase* dan tahap *input* dari *users*. Dilakukan uji kecocokan atau *similarity* antara pertanyaan *users* dan data yang ada di dalam *vectorbase*, kemudian hasil similarity akan digenerate oleh LLM berdasarkan model LLM *Pre-Trained*. Setelah semua tahap dilalui, maka output berupa jawaban yang paling sesuai akan ditampilkan lagi kepada users.

LLM menjadi dasar dalam banyak aplikasi NLP modern karena fleksibilitasnya dalam menangani berbagai tugas hanya dengan sedikit contoh. (Touvron dkk., 2023a) menyatakan bahwa model seperti *Llama-4-scout-17b-16e-instruct* dirancang lebih efisien untuk bahasa-bahasa yang memiliki sumber daya terbatas, seperti Bahasa Indonesia.

LLM digunakan dalam sistem *Retrieval-Augmented Generation* (RAG) untuk menghasilkan teks berbasis informasi yang relevan. *Llama-4-scout-17b-16e-instruct* dipilih karena efisiensinya dalam menangani tugas berbasis bahasa Indonesia.

2.2.9.1 LLaMA

Definisi LLaMA (*Large Language Model Meta AI*) LLaMA adalah keluarga model LLM (*Large Language Models*) yang dirancang untuk tugas generasi teks berbasis pemahaman kontekstual. Menurut (Touvron dkk., 2023b), LLaMA mengunggulkan efisiensi model dengan parameter yang lebih kecil, namun tetap memberikan performa setara atau lebih baik dibandingkan model-model besar seperti GPT-3.

Llama-4-scout-17b-16e-instruct merupakan varian LLaMA yang dilatih secara khusus untuk tugas-tugas generatif berbasis instruksi. Varian ini menonjol dalam memproses teks panjang dan memahami konteks mendalam, menjadikannya pilihan ideal untuk sistem berbasis RAG, terutama untuk tugas yang memerlukan pemahaman literatur akademik.

Keunggulan *LLaMA-4-scout-17b-16e-instruct*:

1. Kompresi Parameter: Dengan parameter 17 miliar.
2. Pelatihan Khusus: Model dilatih menggunakan instruksi berbasis dialog, yang meningkatkan relevansi respons terhadap *query* spesifik.
3. Generasi Teks Akurat: Kemampuan generasi teks LLaMA unggul dalam menjaga kesesuaian fakta, yang sangat penting untuk tugas seperti analisis literatur ilmiah.

2.2.10 Evaluasi Sistem

2.2.10.1 Evaluasi Peringkasan Teks dengan *BERTScore*

BERTScore adalah metrik evaluasi teks yang menggunakan representasi *embedding* dari model bahasa berbasis *Transformer*, seperti *BERT*, untuk membandingkan keluaran teks dengan referensi. Berbeda dengan metrik tradisional seperti *ROUGE* yang hanya mengandalkan kecocokan token secara langsung (n-gram), *BERTScore* menghitung kesamaan semantik pada level vektor, sehingga lebih robust terhadap sinonim dan variasi struktur kalimat (Zhang dkk., 2019).

BERTScore sangat relevan untuk tugas peringkasan teks, di mana model evaluasi harus mampu mengenali kesamaan semantik antara ringkasan otomatis dengan ringkasan referensi, meskipun menggunakan kata atau frasa yang berbeda. Cara kerja *BERTScore* menggunakan representasi *embedding* dari setiap token dalam teks yang dianalisis. Representasi ini diperoleh dengan melewatkan teks input dan referensi ke dalam model *pre-trained*, seperti *BERT*, *IndoBERT*, atau IndoT5. Setelah itu, kesamaan antar-token dihitung menggunakan *cosine similarity*. Proses utama dalam *BERTScore*:

1. *Token Embedding*: Setiap token dari teks hasil ringkasan (prediksi) dan ringkasan referensi (*ground truth*) diubah menjadi vektor *embedding* menggunakan model bahasa pra-latih (*pre-trained language model*).
2. *Cosine Similarity*: Kesamaan antara token input dan referensi dihitung menggunakan formula:

$$\textit{Cosine Similarity}(u,v) = \frac{u.v}{||u||\,||v||} \quad (1)$$

Di mana:

- u = vektor *embedding* untuk token ke-i dari teks prediksi
- v = vektor *embedding* untuk token ke-j dari teks referensi

- u.v = hasil perkalian dot product antara vektor u dan v
- ||u||, ||v|| = panjang (norma) dari vektor u dan v

3. *Matching Tokens*: Untuk setiap token u_i pada teks prediksi, dihitung kemiripannya dengan seluruh token v_j dalam teks referensi. Nilai cosine similarity tertinggi yang ditemukan dari semua pasangan u_i dan v_j dianggap sebagai skor kemiripan token u_i terhadap referensi.
4. *Aggregate Score*: *BERTScore* dihitung sebagai rata-rata dari semua skor kemiripan token prediksi terhadap token referensi, dengan rumus:

$$BERTscore = \frac{1}{n} \sum_{i=1}^{n} \max_{j \in Reference} Cosine\ Similarity(u_i, u_j) \qquad (2)$$

Di mana:

- n = jumlah token dalam teks prediksi
- u_i = token ke-i dari teks prediksi
- v_j = token ke-j dari teks referensi
- $max\ j \in reference$ = memilih nilai *cosine similarity* maksimum dari token u_i terhadap semua token v_j dalam referensi

BERTScore dapat dihitung untuk tiga jenis nilai:

- ***Precision***: seberapa baik token prediksi cocok ke referensi
- ***Recall***: seberapa baik token referensi terwakili di prediksi
- ***F1-score***: rata-rata harmonis dari *precision* dan *recall*, biasanya dipakai sebagai nilai utama.

2.2.10.2 Evaluasi Sistem RAG menggunakan *LLM-as-a-Judge*

Large Language Models (LLM) tidak hanya digunakan sebagai sistem generatif, tetapi juga berkembang sebagai alat evaluasi otomatis terhadap keluaran model berbasis teks. Salah satu pendekatan yang berkembang pesat adalah *LLM-as-a-Judge*, yaitu pemanfaatan LLM sebagai evaluator semantik untuk menilai kualitas keluaran sistem, seperti ringkasan, jawaban, atau hasil dialog, berdasarkan dimensi kualitas seperti relevansi, kesesuaian isi (*faithfulness*), dan kelengkapan informasi (*completeness*) (Gu dkk., 2024). Pendekatan ini muncul sebagai solusi terhadap keterbatasan metrik evaluasi otomatis tradisional seperti *ROUGE, BLEU*,

dan *METEOR*, yang dinilai kurang mampu menangkap kualitas semantik secara utuh, terutama pada tugas-tugas generatif terbuka (Li dkk., 2024).

Dalam praktiknya, evaluasi berbasis *LLM-as-a-Judge* umumnya menggunakan skala Likert 1–5 untuk tiap aspek penilaian. Misalnya, relevansi menilai kesesuaian isi jawaban terhadap pertanyaan atau konteks; *faithfulness* mengukur apakah informasi dalam jawaban benar dan tidak menyesatkan; sedangkan kelengkapan mengevaluasi sejauh mana informasi penting telah tercakup secara menyeluruh (Li dkk., 2024). Penilaian dilakukan dengan menggunakan *prompt* evaluasi eksplisit, sering kali diperkuat dengan teknik *chain-of-thought prompting*, yang meminta LLM untuk menjelaskan alasan sebelum memberikan skor. Teknik ini terbukti dapat meningkatkan konsistensi dan interpretabilitas hasil evaluasi (Gu dkk., 2024). Untuk memperoleh nilai akhir dari setiap jawaban, skor dari ketiga aspek tersebut dirata-ratakan dengan rumus sebagai berikut:

$$TotalScore_i = \frac{S_{relevansi}+S_{faithfulness}+S_{kelengkapan}}{3} \quad (3)$$

Keterangan:

- TotalScore : Nilai evaluasi akhir untuk jawaban ke-i
- $S_{relevansi}$: Skor relevansi jawaban ke-i
- $S_{faithfulness}$: Skor kesesuaian isi (*faithfulness*) jawaban ke-i
- $S_{kelengkapan}$: Skor kelengkapan jawaban ke-i
- Semua skor berada dalam rentang 1 (sangat buruk) sampai 5 (sangat baik)

Berbeda dengan metode evaluasi konvensional yang membutuhkan jawaban referensi (*ground truth*), pendekatan *LLM-as-a-Judge* sangat cocok untuk digunakan dalam konteks di mana jawaban benar tidak tersedia atau bersifat terbuka, seperti pada sistem *Retrieval-Augmented Generation* (RAG) yang dirancang dalam penelitian ini. Oleh karena itu, dalam penelitian ini, penilaian hasil ringkasan dilakukan dengan meminta LLM untuk memberikan skor 1–5 pada ketiga dimensi kualitas tersebut. Skor akhir dihitung sebagai rata-rata dari ketiga dimensi, sehingga diperoleh satu nilai agregat untuk setiap hasil. Pendekatan ini dipilih karena telah terbukti dapat memberikan evaluasi yang praktis, skalabel, dan

mendalam secara semantik terhadap keluaran model generatif, khususnya dalam studi-studi terbaru yang meneliti akurasi dan reliabilitas LLM sebagai evaluator (Gu dkk., 2024; Li dkk., 2024).

Dalam penelitian ini, sistem evaluasi otomatis dirancang menggunakan model LLaMA 3 70B dari Meta AI, salah satu model *open-source* yang digunakan dalam eksperimen evaluasi LLM pada studi (Gu dkk., 2024). Model ini diintegrasikan melalui API dari Groq, yang dikenal menawarkan inferensi berkecepatan tinggi dan latensi rendah. Meskipun jurnal tersebut mencatat bahwa performa evaluasi LLaMA 3 masih berada di bawah GPT-4 dan Qwen2.5, penggunaan LLaMA 3 dipilih karena bersifat *open-source*, lebih hemat biaya, dan dapat diakses secara luas. Model ini memungkinkan integrasi *prompt* evaluasi eksplisit dengan output terstruktur yang sesuai untuk penilaian ringkasan secara semantic (Groq Inc., 2024; Meta AI., 2024).

BAB III
METODOLOGI PENELITIAN

3.1 Deskripsi Sistem

Proyek ini berfokus pada merancang sistem untuk membantu proses kajian literatur dengan memanfaatkan teknologi *Retrieval-Augmented Generation* (RAG). Metode RAG digunakan karena kemampuannya menggabungkan teknik pencarian relevan (*retrieval*) dengan generasi teks (*generation*), yang memungkinkan sistem untuk menghasilkan ringkasan yang informatif dan kontekstual. Inovasi utama penelitian terletak pada penambahan tahap *summarization* menggunakan model IndoT5 sebelum proses RAG, yang bertujuan untuk mengoptimalkan kualitas dan efisiensi pencarian informasi. Sistem yang dikembangkan mengintegrasikan tiga model bahasa yang berbeda: LLaMA sebagai *Large Language Model* utama, IndoT5 untuk proses peringkasan teks, dan IndoBERT untuk pembentukan *embedding*. Pendekatan ini diharapkan dapat meningkatkan akurasi dan relevansi dalam proses kajian literatur berbahasa Indonesia, sekaligus mengatasi keterbatasan umum dalam pemrosesan teks bahasa Indonesia.

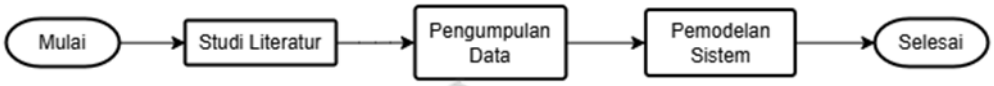
Pada sistem, admin dapat mengunggah dataset berupa jurnal-jurnal dan artikel ilmiah dalam format PDF ke sistem pada bagian sisi admin, kemudian dataset diproses oleh sistem untuk diekstrak teks, dibersihkan, dilakukan pembagian IMRAD, dilakukan peringkasan menggunakan model IndoT5, lalu dilakukan *embedding* menggunakan IndoBERT. Setelah teks diembed, sistem membangun indeks menggunakan FAISS untuk memungkinkan pencarian paragraf yang relevan berdasarkan *query* yang dimasukkan *user*.

Kemudian pada sisi *user*, *user* bisa memasukkan *query* atau pertanyaan sesuai topik yang ada melalui form *input* yang tersedia. Sistem akan mencari paragraf yang relevan dari sumber-sumber eksternal yang sudah dimasukkan tadi untuk menghasilkan jawaban yang relevan sesuai dengan pertanyaan *user* melalui model *Llama*.

3.2 Metode Penelitian

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 *Retrieval-Augmented Generation* (RAG).

Tahapan penelitian yang akan dilakukan seperti pada gambar 3.1.



Gambar 3. 1 *Flowchart* Metode Penelitian

Pada gambar *Flowchart* Metode penelitian memperlihatkan langkah – langkah yang harus dilakukan dalam penelitian yaitu Studi Literatur, pengumpulan data, dan pemodelan sistem.

3.2.1 Studi Literatur

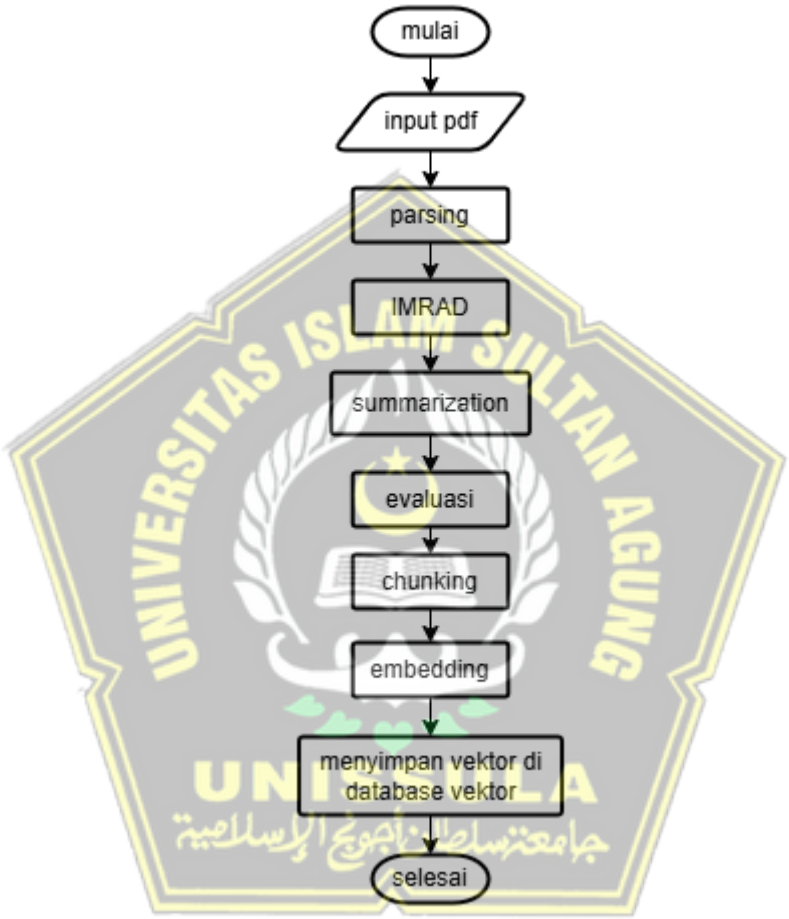
Studi literatur adalah langkah awal yang penting dalam penelitian ini, yang bertujuan untuk memahami dasar teori dan metodologi terkait dengan sistem ringkasan otomatis, khususnya yang menggunakan *Retrieval-Augmented Generation* (RAG). Proses ini melibatkan pengkajian mendalam terhadap berbagai sumber ilmiah yang relevan, seperti jurnal akademik, buku teks, dan laporan teknis, yang membahas penggunaan RAG dalam pemrosesan teks untuk membuat ringkasan otomatis. Fokus utama studi literatur adalah memahami *state-of-the-art* dalam penerapan model RAG untuk meringkas informasi, serta pendekatan-pendekatan terbaru dalam *text summarization*. Hasil dari studi literatur ini akan membentuk landasan teoritis dan metodologis yang mendukung keseluruhan penelitian ini.

3.2.2 Pengumpulan Data

Pada tahap ini Langkah awal yang harus dilakukan adalah menentukan sumber data yang akan digunakan. Pada penelitian ini sumber data ini yang akan digunakan diperoleh dari situs Google Scholar. Dokumen dari Google Scholar tidak diambil langsung dari Google, melainkan diarahkan ke repositori resmi, sehingga memastikan keaslian dan kualitas artikel ilmiah tetap terjaga. Artikel jurnal yang

dipilih harus memiliki kriteria memiliki format PDF dan berbahasa Indonesia. Teknik pengumpulan data akan dilakukan secara manual *download*, yaitu mengunduh *file* PDF secara langsung dari situs sumber jurnal.

Gambar 3.2 dibawah ini menunjukkan penjelasan tahapan pengumpulan data yang ditunjukkan menggunakan *flowchart*.



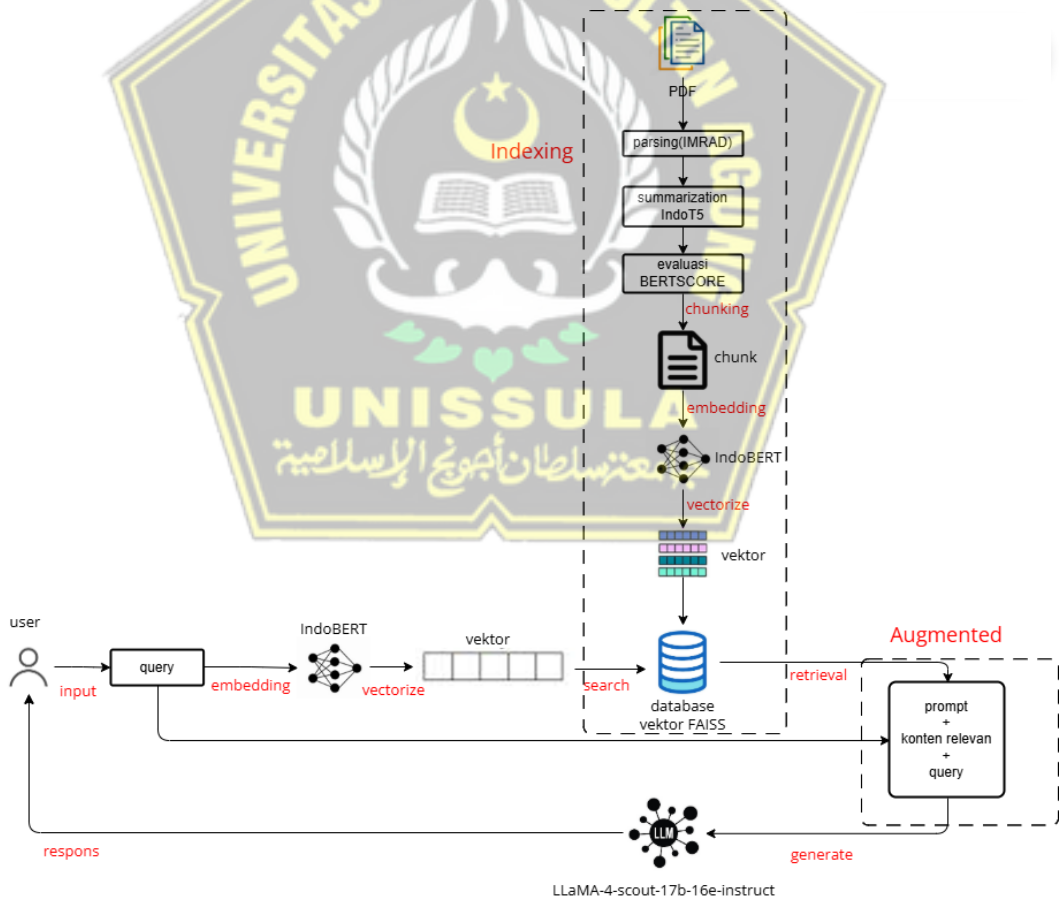
Gambar 3. 2 *Flowchart* pengumpulan data ke dalam *database* vektor

Gambar diatas menunjukkan tahapan pengumpulan data yang digambarkan dalam alur *flowchart* diatas, terbagi dalam beberapa proses. Pertama, pengumpulan data secara manual yaitu berupa artikel mengenai 3 topik, yaitu *Supervised Learning, Unsupervised Learning* dan *Reinforcement Learning* yang digunakan sebagai sumber dalam membuat kajian literatur. Kedua, mengekstrak file menjadi teks menggunakan *library* pymupdf. Ketiga, melakukan pembagian teks sesuai kaidah IMRAD(*introduction, methods, result and discussion*) untuk mengambil bagian penting dari jurnal. Keempat, dilakukan peringkasan teks pada setiap bagian

IMRAD, peringkasan dibuat menggunakan model IndoT5. Kelima, setelah dilakukan peringkasan pada setiap bagian IMRAD, dilakukan penggabungan hasil lalu di evaluasi menggunakan *BERTScore*, hasil dibandingkan dengan abstrak jurnal itu sendiri. Keenam, setelah hasil dievaluasi, lalu dilakukan *chunking* atau biasa disebut *split* data, pada setiap *chunk* langsung di *embedding* menggunakan model *IndoBERT* kemudian yang terakhir, hasil *embedding* disimpan ke dalam *database* vektor FAISS(*Facebook AI Similarity Search*) yang digunakan untuk menyimpan teks yang sudah dikonversi menjadi vektor.

3.2.3 Pemodelan Sistem

Pada tahap ini, penulis akan menentukan rancangan alur kerja sistem melalui diagram alur kerja, yang nantinya akan memberikan gambaran alur kerja sistem, seperti yang ditunjukkan pada gambar 3. 3.



Gambar 3. 3 *Workflow* perancangan sistem

Gambar diatas menunjukkan alur kerja sistem RAG yang terdiri dari beberapa tahapan. Tahap yang pertama adalah tahap *indexing* yaitu proses penyimpanan data yang memungkinkan sistem mencari data secara efisien. Tahapan ini bisa disebut tahapan pengumpulan data, terbagi dalam beberapa proses. Pertama, pengumpulan data secara manual yaitu berupa artikel mengenai 3 topik, yaitu *Supervised Learning, Unsupervised Learning* dan *Reinforcement Learning* yang digunakan sebagai sumber dalam membuat kajian literatur. Kedua, mengekstrak file menjadi teks menggunakan *library* pymupdf. Ketiga, melakukan pembagian teks sesuai kaidah IMRAD(*introduction, methods, result and discussion*) untuk mengambil bagian penting dari jurnal. Keempat, dilakukan peringkasan teks pada setiap bagian IMRAD, peringkasan dibuat menggunakan model IndoT5. Kelima, setelah dilakukan peringkasan pada setiap bagian IMRAD, dilakukan penggabungan hasil lalu di evaluasi menggunakan *BERTScore*, hasil dibandingkan dengan abstrak jurnal itu sendiri. Keenam, setelah hasil dievaluasi, lalu dilakukan *chunking* atau biasa disebut *split* data, pada setiap *chunk* langsung di *embedding* menggunakan model *IndoBERT* kemudian yang terakhir, hasil *embedding* disimpan ke dalam *database* vektor FAISS(*Facebook AI Similarity Search*) yang digunakan untuk menyimpan teks yang sudah dikonversi menjadi vektor.

Tahapan selanjutnya adalah tahapan pemodelan sistem yang digambarkan dengan alur *flowchart* di atas, dimulai dari *user* menginput *query*, selanjutnya dilakukan *embeddings query* dari *user* menggunakan model yang sama dengan model *embeddings* pada saat pengumpulan data yaitu *indoBERT*, kemudian dilakukan *similarity search* menggunakan FAISS dari *database* vektor. *Retrieval* atau pengambilan konteks yang cocok antara data *input* dengan data yang ada di *database* vektor dengan menggunakan *library create_retrieval_chain*. Dari beberapa konteks relevan yang sudah diambil, selanjutnya digabungkan dengan *query* dan *prompt* baru ini adalah proses *augmented*, selanjutnya hasil gabungan teks di *generate* menggunakan *Llama-4-scout-17b-16e-instruct* dan menampilkan responnya ke pengguna.

3.3 Analisis Kebutuhan

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

1. *Python* 3.12.6

   Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991, *Python* juga merupakan bahasa yang sangat popular belakangan ini. Selain itu, *Python* merupakan bahasa pemrograman yang multi fungsi salah satunya pada bidang *Machine Learning* dan *Deep Learning*. Penelitian ini menggunakan *Python* versi 3.12.6 karena bersifat *opensource*, 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 *IndoBERT* untuk *embedding* nya dan indoT5 untuk model peringkasan nya. Dengan algoritma ini ringkasan yang didapat akan lebih relevan, cepat dan efisien.

3. *Langchain*

   Langchain pada pembuatan sistem ini tidak kalah penting dengan keberadaan *python*, kerangka kerja atau *framework* tersebut berperan sebagai rantai untuk mempermudah dalam mengimport *library* yang akan digunakan. *Langchain* juga didesain untuk bahasa pemograman alami.

4. *Library FAISS*

   FAISS atau yang dikenal juga sebagai *Facebook AI Similarity Search*, adalah lembaga pencarian dan *clustering* vektor yang besar yang memungkinkan pencarian vektor yang efektif dalam ruang dimensi besar. Dalam penelitian ini digunakan untuk mengindeks dan mencari vektor *embedding* dalam teks buku. FAISS membantu sistem menemukan kalimat yang relevan berdasarkan *query*.

5. *Library NumPy*

   NumPy (*Numerical Phyton*), 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 PyMuPDF*

   PyMuPDF adalah pustaka Python untuk mengekstrak teks, tabel, dan metadata dari file PDF. Digunakan untuk membaca dan memproses dokumen ilmiah dalam format PDF, membagi teks ke dalam struktur seperti IMRAD (*Introduction, Methods, Results, Discussion*).

7. *Library BeautifulSoup*

   BeautifulSoup adalah *library Python* yang digunakan untuk *parsing* HTML dan XML dan digunakan untuk mengekstrak data dari *file* HTML. Dalam penelitian ini, *library* ini digunakan untuk memparsing dan mengekstrak teks dari halaman *web* yang diambil menggunakan *Requests*, dan teks ini kemudian digunakan dalam proses pengumpulan data.

8. *Library llama-index*

   Llama-index adalah sebuah *library* yang dirancang untuk mempermudah proses pembuatan dan penggunaan indeks dalam aplikasi yang menggunakan *Large Language Models,* khususnya model-model seperti LLaMA. Library ini dirancang untuk bekerja dengan berbagai jenis data dan memudahkan pengelolaan informasi dalam konteks pemrosesan bahasa alami. Pada penelitian ini *library llama-index* digunakan untuk generate teks dengan sumber tambahan dari jurnal dan *query* dari pengguna menggunakan fungsi *query_engine* untuk memungkinkan menjawab pertanyaan berdasarkan isi dokumen, isi dokumen tersebut adalah hasil dari gabungan *query* dan hasil *retrieve*.

9. *Visual Studio Code*

   Visual studio code adalah *software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. *Visual studio code* digunakan untuk pembuatan kode-kode program yang dibutuhkan untuk sebuah aplikasi.

Visual studio code dapat digunakan untuk berbagai bahasa pemrograman seperti Javascript, HTML, CSS, PHP, Python, C++, dan masih banyak lagi. *Visual studio code* bekerja pada berbagai sistem operasi seperti windows, macOs, dan Linux. Selain itu *visual studio code* menyediakan fitur *live share* sehingga memungkinkan beberapa pengembang bekerja pada satu proyek yang sama secara bersamaan dari lokasi yang berbeda.

10. *Streamlit*

Streamlit adalah salah satu *framework* yang mendukung untuk melakukan *deployment* model kedalam program berbasis web. *Streamlit* merupakan *framework* berbasis *Python* yang bersifat open source. *Framework* ini dibuat untuk memudahkan developer dalam membangun program berbasis web dibidang data *science* dan *machine learning* yang interaktif. Salah satu kelebihan dari *streamlit* adalah developer tidak perlu mengatur tampilan *website* dengan CSS, HTML, dan javascript karena *framework streamlit* telah menyediakannya melalui fungsi-fungsi yang terdapat pada *framework* tersebut.

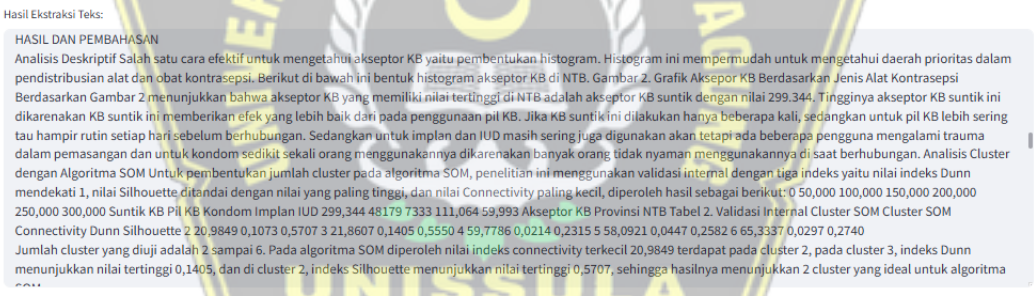
BAB IV
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Penelitian

Sistem yang dikembangkan merupakan aplikasi berbasis web yang mengotomatisasi proses kajian literatur ilmiah berbahasa Indonesia menggunakan pendekatan *Retrieval-Augmented Generation* (RAG) dengan model IndoT5 sebagai *summarizer* dan IndoBERT untuk pembentukan *embedding*. Sistem terbagi menjadi dua sisi pengguna, yaitu:

1. Sisi Admin, bertugas mengunggah dan memproses dokumen jurnal ilmiah PDF.
2. Sisi *User*, yang dapat mengajukan pertanyaan dan menerima jawaban berdasarkan literatur yang telah diproses.

4.1.1 Hasil ekstraksi PDF


Hasil Ekstraksi Teks:

HASIL DAN PEMBAHASAN
Analisis Deskriptif Salah satu cara efektif untuk mengetahui akseptor KB yaitu pembentukan histogram. Histogram ini mempermudah untuk mengetahui daerah prioritas dalam pendistribusian alat dan obat kontrasepsi. Berikut di bawah ini bentuk histogram akseptor KB di NTB. Gambar 2. Grafik Aksepor KB Berdasarkan Jenis Alat Kontrasepsi Berdasarkan Gambar 2 menunjukkan bahwa akseptor KB yang memiliki nilai tertinggi di NTB adalah akseptor KB suntik dengan nilai 299.344. Tingginya akseptor KB suntik ini dikarenakan KB suntik ini memberikan efek yang lebih baik dari pada penggunaan pil KB. Jika KB suntik ini dilakukan hanya beberapa kali, sedangkan untuk pil KB lebih sering tau hampir rutin setiap hari sebelum berhubungan. Sedangkan untuk implan dan IUD masih sering juga digunakan akan tetapi ada beberapa pengguna mengalami trauma dalam pemasangan dan untuk kondom sedikit sekali orang menggunakannya dikarenakan banyak orang tidak nyaman menggunakannya di saat berhubungan. Analisis Cluster dengan Algoritma SOM Untuk pembentukan jumlah cluster pada algoritma SOM, penelitian ini menggunakan validasi internal dengan tiga indeks yaitu nilai indeks Dunn mendekati 1, nilai Silhouette ditandai dengan nilai yang paling tinggi, dan nilai Connectivity paling kecil, diperoleh hasil sebagai berikut: 0 50,000 100,000 150,000 200,000 250,000 300,000 Suntik KB Pil KB Kondom Implan IUD 299,344 48179 7333 111,064 59,993 Akseptor KB Provinsi NTB Tabel 2. Validasi Internal Cluster SOM Cluster SOM Connectivity Dunn Silhouette 2 20,9849 0,1073 0,5707 3 21,8607 0,1405 0,5550 4 59,7786 0,0214 0,2315 5 58,0921 0,0447 0,2582 6 65,3337 0,0297 0,2740
Jumlah cluster yang diuji adalah 2 sampai 6. Pada algoritma SOM diperoleh nilai indeks connectivity terkecil 20,9849 terdapat pada cluster 2, pada cluster 3, indeks Dunn menunjukkan nilai tertinggi 0,1405, dan di cluster 2, indeks Silhouette menunjukkan nilai tertinggi 0,5707, sehingga hasilnya menunjukkan 2 cluster yang ideal untuk algoritma

Gambar 4. 1 Hasil ekstraksi atau *load* pdf

Pada Gambar 4.1 menampilkan hasil dari ekstraksi pdf jurnal-jurnal yang digunakan untuk dataset sistem. Ekstraksi menggunakan *library pymupdf4llm* yang mengubah pdf menjadi bentuk teks dalam format *markdown*, lalu sistem mengambil hasil teks tersebut dan mengidentifikasi judul setiap bab. Sistem menggunakan *library pymupdf4llm* dikarenakan memudahkan dalam mengklasifikasikan dalam format IMRAD(*Introduction, Methods, Result and Discussion*), meningkatkan akurasi dan kelengkapan hasil klasifikasi.

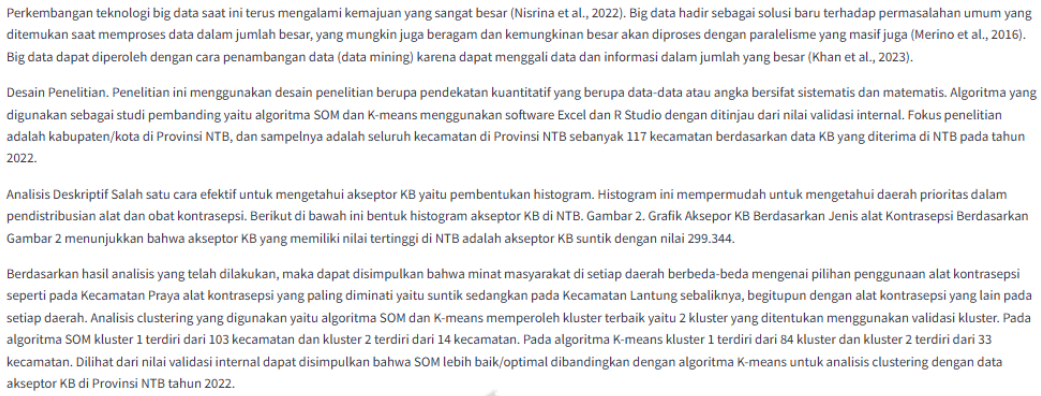
4.1.2 Hasil Pengelompokkan Teks menjadi Struktur IMRAD

Gambar 4. 2 Hasil pengelompokkan IMRAD

Gambar 4. 3 Hasil pengelompokkan IMRAD

Pada Gambar 4.2 dan 4.3 menunjukkan hasil dari pengelompokkan IMRAD(*Introduction, Methods, Result and Discussion*). Pengelompokkan ini bertujuan untuk mengambil hal hal penting dari jurnal-jurnal yang akan digunakan sebagai dataset sistem. Pengelompokkan ini menggunakan regex untuk mengidentifikasi tiap bagian.

4.1.3 Hasil *Summarization* IndoT5


Perkembangan teknologi big data saat ini terus mengalami kemajuan yang sangat besar (Nisrina et al., 2022). Big data hadir sebagai solusi baru terhadap permasalahan umum yang ditemukan saat memproses data dalam jumlah besar, yang mungkin juga beragam dan kemungkinan besar akan diproses dengan paralelisme yang masif juga (Merino et al., 2016). Big data dapat diperoleh dengan cara penambangan data (data mining) karena dapat menggali data dan informasi dalam jumlah yang besar (Khan et al., 2023).

Desain Penelitian. Penelitian ini menggunakan desain penelitian berupa pendekatan kuantitatif yang berupa data-data atau angka bersifat sistematis dan matematis. Algoritma yang digunakan sebagai studi pembanding yaitu algoritma SOM dan K-means menggunakan software Excel dan R Studio dengan ditinjau dari nilai validasi internal. Fokus penelitian adalah kabupaten/kota di Provinsi NTB, dan sampelnya adalah seluruh kecamatan di Provinsi NTB sebanyak 117 kecamatan berdasarkan data KB yang diterima di NTB pada tahun 2022.

Analisis Deskriptif Salah satu cara efektif untuk mengetahui akseptor KB yaitu pembentukan histogram. Histogram ini mempermudah untuk mengetahui daerah prioritas dalam pendistribusian alat dan obat kontrasepsi. Berikut di bawah ini bentuk histogram akseptor KB di NTB. Gambar 2. Grafik Aksepor KB Berdasarkan Jenis alat Kontrasepsi Berdasarkan Gambar 2 menunjukkan bahwa akseptor KB yang memiliki nilai tertinggi di NTB adalah akseptor KB suntik dengan nilai 299.344.

Berdasarkan hasil analisis yang telah dilakukan, maka dapat disimpulkan bahwa minat masyarakat di setiap daerah berbeda-beda mengenai pilihan penggunaan alat kontrasepsi seperti pada Kecamatan Praya alat kontrasepsi yang paling diminati yaitu suntik sedangkan pada Kecamatan Lantung sebaliknya, begitupun dengan alat kontrasepsi yang lain pada setiap daerah. Analisis clustering yang digunakan yaitu algoritma SOM dan K-means memperoleh kluster terbaik yaitu 2 kluster yang ditentukan menggunakan validasi kluster. Pada algoritma SOM kluster 1 terdiri dari 103 kecamatan dan kluster 2 terdiri dari 14 kecamatan. Pada algoritma K-means kluster 1 terdiri dari 84 kluster dan kluster 2 terdiri dari 33 kecamatan. Dilihat dari nilai validasi internal dapat disimpulkan bahwa SOM lebih baik/optimal dibandingkan dengan algoritma K-means untuk analisis clustering dengan data akseptor KB di Provinsi NTB tahun 2022.

Gambar 4. 4 Hasil *summarization* IndoT5

Pada Gambar 4.4 menunjukkan hasil summarisasi menggunakan model IndoT5 *pretrained* yang diambil dari platform huggingface, hasil merupakan ringkasan dari tiap bagian IMRAD yang sudah dilakukan pembagian sebelumnya, tiap paragraf menunjukkan satu *section* dari IMRAD.

4.1.4 Hasil Evaluasi Summarization menggunakan *BERTScore*

Hasil ringkasan dari setiap bagian IMRAD dievaluasi dengan membandingkannya terhadap abstrak jurnal menggunakan *BERTScore*. Aspek yang diukur meliputi kesamaan semantik, relevansi, dan representasi makna.

Gambar 4. 5 Hasil Evaluasi *Summarization*

Tabel 4. 1 Hasil Evaluasi *Summarization* IndoT5

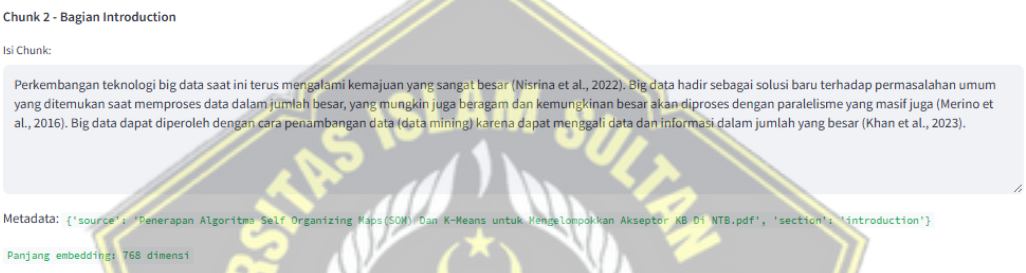
No	Dokumen	Precision	Recall	F1-Score
1.	Pemanfaatan Citra Penginderaan Jauh untuk Pemetaan Klasifikasi Tutupan Lahan Menggunakan Metode Unsupervised K-Means Berbasis Web GIS.pdf	0.8155	0.873	0.8433

2.	Pendekatan Supervised Learning untuk Diagnosa Kehamilan.pdf	0.841	0.9012	0.8701
3.	Penerapan Clustering DBSCAN untuk Pertanian Padi.pdf	0.834	0.916	0.8731
4.	Penerapan Algoritma Self Organizing Maps(SOM) Dan K-Means untuk Mengelompokkan Akseptor KB Di NTB.pdf	0.8253	0.8642	0.8443
5.	Penerapan Model Pembelajaran dengan Metode Reinforcement Learning Menggunakan Simulator Carla.pdf	0.8005	0.8705	0.834
6.	Penerapan Pemodelan Topik menggunakan Metode Latent Dirichlet.pdf	0.8376	0.8793	0.8579
7.	Penerapan Principal Component Analysis (PCA) Untuk Reduksi Dimensi Pada Proses Clustering Data Produksi Pertanian Di Kabupaten Bojonegoro.pdf	0.8202	0.8707	0.8447
8.	Penerapan support vector mechine (SVM) untuk pengkategorian Penelitian.pdf	0.8258	0.867	0.8459
9.	Pengambilan Keputusan Pada Trafik Management dengan Menggunakan Reinforcement Learning.pdf	0.8157	0.8718	0.8428
10.	Pengaplikasian Deep Reinforcement Q-Learning untuk Prediksi Perdagangan Valas Otomatis.pdf	0.8266	0.868	0.8468

Pada Gambar 4.5 dan Tabel 4.1 menunjukkan hasil dari evaluasi ringkasan, evaluasi dilakukan dengan membandingkan hasil ringkasan dengan abstrak jurnal itu sendiri, hal ini dilakukan karena dalam hal peringkasan banyak dokumen tidak

bisa dilakukan pembandingan manual yang jelas akan memakan banyak sekali waktu, sedangkan abstrak dari jurnal itu sendiri merupakan hasil ringkasan dari penulis sendiri yang jelas didalamnya merupakan ringkasan yang terdapat semua hal penting, meskipun tidak mencakup hal hal mendetail seperti penjelasan metode dan hal teknis mendetail, namun masih bisa menjadi pembanding ringkasan. Nilai *BERTScore* rerata menunjukkan tingkat kesamaan tinggi terhadap ringkasan referensi, yang menunjukkan efektivitas *summarizer* IndoT5 dalam konteks akademik berbahasa Indonesia.

4.1.5 Hasil *Chunking* Teks



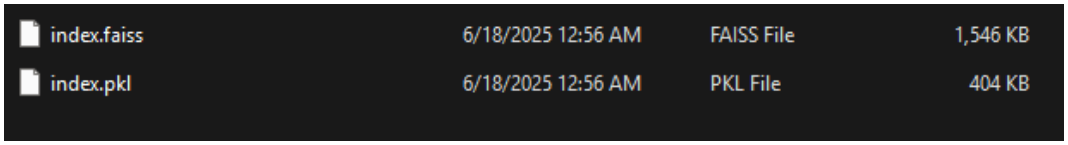
Gambar 4. 6 Hasil *Chunking* Teks

Pada Gambar 4.6 merupakan hasil *chunking* atau biasa disebut *split* teks, hal ini dilakukan untuk mempermudah melakukan *embedding* dan memasukkannya ke dalam *database* vektor, tiap dokumen memiliki 4 *chunk* dan tiap *chunk* nya berisi tiap *section* imrad, hal ini dilakukan untuk mencegah tercampurnya konteks.

4.1.6 Hasil Embedding dan Penyimpanan Vektor

Gambar 4. 7 Hasil *Embedding* Teks menggunakan Indobert

Pada Gambar 4.7 menunjukkan hasil *embedding* dari salah satu *chunk* yang akan dimasukkan ke dalam *database* vektor. *Embedding* menggunakan model *Indobert* atau BERT yang sudah dilatih untuk bahasa Indonesia, model diambil dari huggingface. *Embedding* memiliki *maks token* 512, maka dari itu, ini berhubungan dengan *chunk* yang menggunakan *max token* 500.



Gambar 4. 8 *Database Vektor*

Pada gambar diatas menampilkan hasil file faiss dan pkl, index FAISS atau hasil *embedding* yang sudah dibagun kemudian disimpan ke dalam file index faiss. Kemudian penyimpanan metadata merupakan informasi tambahan yang memberikan dan juga menjelaskan konteks pada data utama dengan menggunakan pickle. Dengan adanya metadata dapat mempertahankan hubungan antara dokumen asli dan *embedding* dan mempercepat dalam proses pencarian.

4.1.7 Hasil *Retrieval* dan *Generate* Jawaban

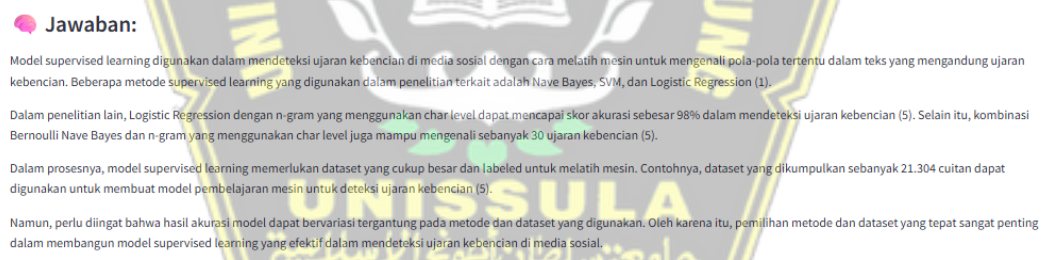
Gambar 4. 9 Hasil *Retrieval* 1

Gambar 4. 10 Hasil *Retrieval* 2

Pada Gambar 4.9 dan 4.10 menunjukkan hasil *retrieval* yang dilakukan oleh sistem, sistem mengambil 5 dokumen terelevan untuk diambil sebagai konteks yang akan dimasukkan ke dalam *prompt* untuk di *generate* oleh LLM.

Gambar 4. 11 *Prompt* sebelum masuk LLM

Pada Gambar 4.11 diatas menunjukkan *prompt* yang akan dimasukkan sebagai input LLM. *Prompt* ini terdiri dari *query user*, *prompt* tambahan, serta konteks relevan hasil *retrieve*. *Prompting* ini dilakukan supaya LLM mendapatkan konteks valid yang cukup untuk *generate* jawaban yang relevan untuk pengguna.



Gambar 4. 12 Hasil *Generate* LLM

Ringkasan IMRAD + Link Unduh PDF

No	Judul	Introduction	Methods	Results	Discussion
1	Penerapan Pemodelan Topik menggunakan Metode Latent Dirichlet.pdf	Di Indonesia terdapat sistem yang dianut, yaitu sistem demokrasi. Salah satu elemen mendasar dalam sistem demokrasi adalah proses mengumpulkan sudut pandang publik secara berkala melalui pemilihan umum (Putri, Laia and Laia, 2023). Sistem demokrasi juga dipraktikkan melalui proses pemilihan yang digunakan untuk memilih perwakilan dan berbagai pejabat publik (Dalimunthe, 2024). Pada 14 Februari 2024 telah dilaksanakan Pemilu, dimana sebelum Pemilu terdapat masa kampanye yang menjadi perbincangan hangat di media sosial. Banyak masyarakat yang memberikan pandangannya terhadap visi dan misi para calon presiden dan Fakultas Ilmu Komputer, Universitas Brawijaya wakil presiden saat kampanye	Rekayasa metode yang akan digunakan dalam penelitian ini terdapat pada gambar 3. 1 yang melakukan pemodelan topik pada Pemilu 2024 menggunakan metode Latent Dirichlet Allocation (LDA). Perancangan metode yang akan digunakan dalam penelitian ini terdapat pada gambar 3. 1 yang melakukan pemodelan topik pada Pemilu 2024 menggunakan metode Latent Dirichlet Allocation (LDA). Gambar 1. Diagram Alir Penelitian 2. 1. Pengumpulan Data Twitter Data yang digunakan pada penelitian ini, diambil dari Twitter menggunakan teknik crawling data Twitter. Proses crawling Fakultas Ilmu Komputer, Universitas Brawijaya dilakukan menggunakan alat berbasis baris perintah yaitu, Tweet-harvest.	Integrasi bigram dengan metode Latent Dirichlet Allocation (LDA) berpangaruh terhadap kualitas topik yang dihasilkan, yang diukur dengan metrik perplexity dan coherence score. Untuk mengengevaluasi dampak integrasi bigram dengan metode LDA, pengujian dilakukan pada model unigram dan bigram. Tabel 1 menyajikan metrik perplexity dan coherence score pada jumlah topik yang berbeda disetiap model. Tabel 1 menyajikan metrik perplexity dan coherence score pada jumlah topik yang berbeda disetiap model. Tabel 1.	Penggunaan metode Latent Dirichlet Allocation (LDA) dengan bigram untuk analisis Pemilu Indonesia 2024 di Twitter meningkatkan akurasi, diukur dengan metrik perplexity yang semakin rendah dan coherence score yang semakin tinggi. Pengujian menemukan bahwa 7 atau 8 topik merupakan jumlah topik yang optimal, dengan nilai perplexity masing-masing -12,6649 dan -13,0147, serta coherence score masing-masing 0,5826 dan 0,5579. Pendekatan ini menghasilkan keseimbangan antara perplexity yang rendah dan coherence score yang semakin tinggi, meningkatkan kualitas dan Fakultas Ilmu Komputer, Universitas Brawijaya keinformatifan hasil pemodelan topik.

Gambar 4. 13 Menampilkan Ringkasan 5 Dokumen Relevan

Pada Gambar 4.12 menampilkan respon sistem ke pengguna, jawaban sesuai dengan konteks yang relevan serta pada gambar 4.13 ditampilkan ringkasan dari 5 dokumen relevan.

4.1.8 Hasil Evaluasi Sistem RAG menggunakan *LLM-as-a-Judge*

Evaluasi dilakukan untuk menilai kualitas jawaban sistem RAG terhadap pertanyaan *user*. Model LLaMA 3 digunakan sebagai evaluator dengan pendekatan *LLM-as-a-Judge*. Aspek evaluasi meliputi:

- Relevansi: kecocokan jawaban dengan pertanyaan.
- *Faithfulness*: kebenaran informasi berdasarkan sumber.
- Kelengkapan: seberapa utuh informasi yang diberikan.

Masing-masing aspek dinilai dalam skala 1–5, lalu dirata-rata.

Tabel 4. 2 Hasil Evaluasi Sistem menggunakan *LLM-as-a-Judge*

NO	Pertanyaan Pengguna	Jawaban Sistem
1.	Jelaskan pendekatan simulasi RL untuk menyelesaikan masalah Vehicle Routing Problem	Dalam konteks *Heterogeneous Vehicle Routing Problem with Soft Time Windows* (HVRPSTW), pendekatan *Reinforcement Learning* (RL) dapat digunakan untuk menyelesaikan masalah penentuan rute secara adaptif dan efisien. RL

	dengan soft time windows.	merupakan metode pembelajaran yang memungkinkan agen (sistem) untuk belajar dari pengalaman melalui interaksi berulang dengan lingkungan guna memaksimalkan *reward* atau imbalan kumulatif. Dalam HVRPSTW, tujuan utama adalah menentukan rute optimal untuk armada kendaraan yang beragam, dengan mempertimbangkan batasan waktu pengiriman yang fleksibel (*soft time windows*). Untuk itu, RL dapat digunakan untuk mempelajari kebijakan yang optimal dalam menentukan rute kendaraan. Berikut adalah langkah-langkah pendekatan simulasi RL untuk menyelesaikan HVRPSTW: 1. **Definisikan lingkungan**: Lingkungan dalam HVRPSTW dapat didefinisikan sebagai sistem transportasi yang terdiri dari kendaraan, pelanggan, dan depot. 2. **Tentukan state dan action**: State dapat didefinisikan sebagai informasi tentang posisi kendaraan, waktu pengiriman, dan status pelanggan. Action dapat didefinisikan sebagai pilihan rute yang tersedia untuk kendaraan. 3. **Tentukan fungsi reward**: Fungsi reward dapat didefinisikan sebagai ukuran kinerja yang ingin dioptimalkan, seperti biaya transportasi, waktu pengiriman, atau kualitas layanan.

		4. **Implementasikan algoritma RL**: Algoritma RL seperti Q-learning, SARSA, atau Deep Q-Networks (DQN) dapat digunakan untuk mempelajari kebijakan yang optimal dalam menentukan rute kendaraan. 5. **Simulasikan lingkungan**: Lingkungan dapat disimulasikan untuk mengumpulkan data dan pengalaman yang diperlukan untuk melatih algoritma RL. 6. **Evaluasi kinerja**: Kinerja algoritma RL dapat dievaluasi dengan menggunakan metrik seperti biaya transportasi, waktu pengiriman, atau kualitas layanan. Dengan menggunakan pendekatan simulasi RL, perusahaan dapat menentukan rute optimal untuk armada kendaraan yang beragam, dengan mempertimbangkan batasan waktu pengiriman yang fleksibel. Hal ini dapat membantu meningkatkan efisiensi sistem distribusi dan mengurangi biaya transportasi.
2.	Apa insight yang diperoleh dari studi simulasi RL dalam konteks exergame untuk penurunan berat badan?	Insight yang diperoleh dari studi simulasi Reinforcement Learning (RL) dalam konteks exergame untuk penurunan berat badan adalah: 1. **Algoritma RL dapat efektif**: Algoritma PPO (Proximal Policy Optimization) mampu mencapai kinerja yang sebanding dengan metode kendali konvensional dalam mengendalikan sistem simulasi.

		2. **Kinerja dapat ditingkatkan**: Dengan teknik reward shaping, kinerja algoritma RL dapat ditingkatkan untuk memasukkan spesifikasi kinerja ke dalam fungsi reward. 3. **Penerapan RL pada exergame**: RL dapat digunakan untuk menciptakan sistem simulasi perhitungan yang dapat membantu pengguna menurunkan berat badan dengan menentukan aktivitas fisik yang tepat. 4. **Keterbatasan saat ini**: Saat ini, proses RL masih menggunakan brute force untuk mencari variasi gerakan, yang membutuhkan waktu lama (hingga 200 iterasi) untuk menemukan solusi. Dengan demikian, studi simulasi RL dalam konteks exergame untuk penurunan berat badan menunjukkan potensi besar dalam menciptakan sistem yang dapat membantu pengguna menurunkan berat badan dengan lebih efektif dan menyenangkan.
3.	Apa saja algoritma supervised learning yang digunakan dalam jurnal-jurnal tersebut untuk klasifikasi teks atau citra, dan bagaimana perbandingan performanya?	Algoritma supervised learning yang digunakan dalam jurnal-jurnal tersebut untuk klasifikasi teks atau citra antara lain: 1. Support Vector Machine (SVM) dengan kernel Radial Basis Function (RBF) untuk klasifikasi teks, yang digunakan dalam penelitian klasifikasi berita Indonesia [1]. 2. Convolutional Neural Network (CNN) untuk pengenalan citra, yang digunakan dalam penelitian dengan metode Deep Learning [2].

		3. You Only Look Once (YOLO) untuk deteksi objek manusia pada citra grayscale, yang digunakan dalam penelitian literatur [3][4]. Namun, tidak ada perbandingan performa secara langsung antara algoritma-algoritma tersebut dalam konteks yang diberikan. Tetapi, beberapa informasi tentang performa dapat diperoleh: * SVM dapat melakukan klasifikasi serta prediksi sebuah data terhadap beberapa kelas sekaligus dengan baik [1]. * CNN memiliki hasil paling signifikan dalam pengenalan citra [2]. * YOLO merupakan algoritma yang efektif untuk deteksi objek manusia pada citra grayscale [3][4]. Selain itu, penelitian lain juga membandingkan algoritma pengklasteran dokumen teks, yaitu algoritma berbasis centroid dan algoritma DBSCAN, serta pengaruh reduksi dimensi oleh SVD [5].
4.	Dalam konteks klasifikasi judul skripsi, algoritma mana yang lebih unggul antara SVM dan K-NN menurut jurnal yang digunakan?	Tidak ada informasi yang cukup dalam konteks yang disediakan untuk menentukan algoritma mana yang lebih unggul antara SVM dan K-NN, karena konteks hanya membahas tentang SVM dalam memprediksi masa studi dan IPK mahasiswa, tetapi tidak membahas tentang K-NN.

		Namun, berdasarkan informasi yang tersedia, dapat diketahui bahwa SVM memiliki nilai akurasi sebesar 0,70 dalam memprediksi variabel masa studi. Tetapi, tidak ada perbandingan langsung dengan K-NN dalam konteks yang disediakan. Jadi, jawaban ringkasnya adalah: Tidak dapat ditentukan karena tidak ada informasi yang cukup tentang K-NN dalam konteks yang disediakan.
5.	Bagaimana metode klasifikasi pada "Deteksi Malware Android" dibandingkan dengan metode pada "Prediksi Penyakit Hepatitis"?	Metode klasifikasi pada "Deteksi Malware Android" menggunakan algoritma machine learning dengan metode analisis statis dan Support Vector Machine (SVM) yang mencapai akurasi 96,94% (1). Sementara itu, metode klasifikasi pada "Prediksi Penyakit Hepatitis" menggunakan metode supervised learning yang menerapkan algoritma Naive Bayes dan K-Nearest Neighbor, yang menghasilkan nilai akurasi sebesar 91,67% (4). Jadi, metode klasifikasi pada "Deteksi Malware Android" memiliki akurasi yang lebih tinggi dibandingkan dengan metode pada "Prediksi Penyakit Hepatitis". Selain itu, "Deteksi Malware Android" menggunakan algoritma SVM, sedangkan "Prediksi Penyakit Hepatitis" menggunakan algoritma Naive Bayes dan K-Nearest Neighbor.

6.	Bagaimana model *supervised learning* digunakan dalam mendeteksi ujaran kebencian di media sosial menurut jurnal terkait?	Model *supervised learning* dapat digunakan dalam mendeteksi ujaran kebencian di media sosial dengan beberapa metode, seperti *Logistic Regression, Nave Bayes*, dan *SVM* (1). Logistic Regression dengan n-gram yang menggunakan char level dapat mencapai skor akurasi sebesar 98% dalam mendeteksi ujaran kebencian (5). Selain itu, kombinasi Bernoulli Nave Bayes dan n-gram yang menggunakan char level juga mampu mengenali ujaran kebencian dengan baik (5). Metode Nave Bayes juga dapat digunakan dalam klasifikasi ujaran kebencian, seperti yang digunakan dalam penelitian yang menggunakan dataset UCI Machine Learning Repository untuk memprediksi penyakit hepatitis (2). Oleh karena itu, model supervised learning dapat menjadi salah satu solusi dalam mendeteksi ujaran kebencian di media sosial dengan menggunakan metode yang tepat (1).

Tabel 4. 3 Hasil Evaluasi Sistem RAG

Pertanyaan	**Relevansi**	**Faithfullnes**	**Kelengkapan**	**Rata - rata**
1.	5	5	4	4,67
2.	5	5	4	4,67
3.	5	4	4	4,33
4.	5	5	5	5
5.	5	5	4	4,67
6.	5	4	4	4,33
Rata- rata				4,6

Keterangan:

1. Nilai 5 = Sangat Baik
2. Nilai 4 = Baik
3. Nilai 3 = Cukup
4. Nilai 2 = Kurang
5. Nilai 1 = Sangat Kurang

Berikut adalah pembahasan dari hasil evaluasi yang telah dilakukan terhadap materi yang dihasilkan oleh program:

1. Relevansi Jawaban

   Pada hampir semua pertanyaan, relevansi jawaban terhadap topik pertanyaan dinilai sangat baik. Hal ini menunjukkan bahwa llm sebagian besar berhasil *generate* jawaban sesuai dengan konteks yang diberikan serta menjawab pertanyaan dengan baik. Relevansi jawaban mencakup aspek relevansi terhadap topik yang sesuai dengan dokumen-dokumen relevan pada dataset.

2. Kebenaran Jawaban

   Salah satu keunggulan utama hasil program ini adalah kebenaran jawaban atau *faithfulness*. Sebagian besar pertanyaan dinilai memiliki kebenaran yang baik, hal ini menunjukkan sistem mampu menjawab dengan baik dan meminimalisir halusinasi.

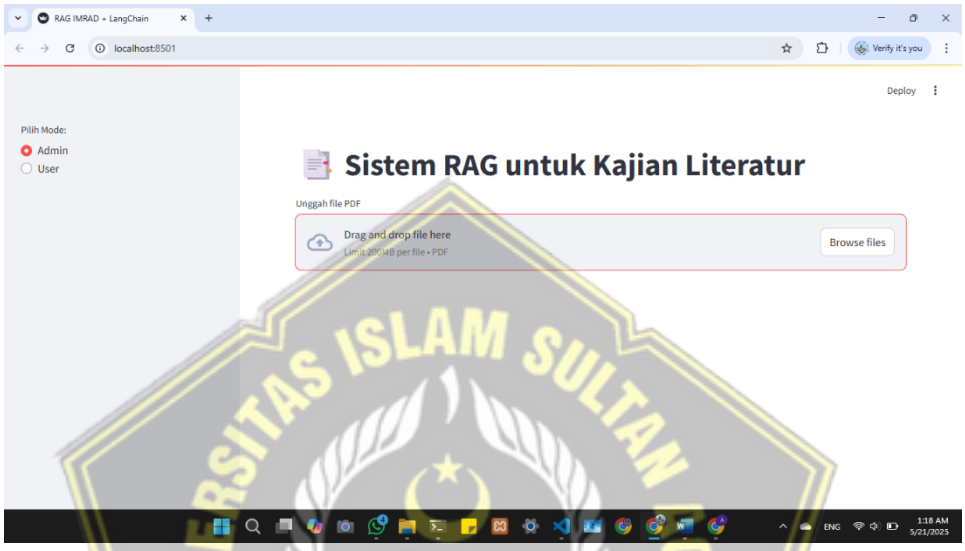
3. Kelengkapan Jawaban

   Berdasarkan hasil evaluasi kelengkapan jawaban sebagian besar dinilai sangat lengkap atau lengkap. Hal ini menunjukkan bahwa jawaban yang dibuat oleh program cukup luas dan mencakup berbagai aspek. Namun, beberapa pertemuan seperti pada pertemuan 4 dinilai baik, yang berarti bahwa konten mungkin perlu ditambahkan untuk memberikan evaluasi yang lebih mendalam.

Pada tabel 4.3 Menunjukkan hasil dan evaluasi *LLM-as-a-Judge* untuk sistem RAG ini. Nilai yang digunakan merupakan skala 1-5 *likert* dengan evaluasi akurasi/relevansi, kebenaran/*faithfulness*, kelengkapan/*comprehensive*. Evaluasi menggunakan konteks relevan, pertanyaan serta jawaban sistem dan dimasukkan

ke dalam *prompt* dan menjalankannya menggunakan llama-3 yang dipanggil dari api groq. Dari 10 soal yang diuji menunjukkan bahwa rata-rata hasil relevansi berada di angka 4,6 , sehingga hasil jawaban dari sistem RAG ini bisa dikatakan akurat.

4.1.9 Hasil Perancangan *User Interface*



Gambar 4. 14 Tampilan Sistem Sisi Admin

Gambar 4. 15 Tampilan Sistem Sisi *User*

Gambar 4.14 adalah *interface* sisi admin yang dirancang untuk membantu admin memasukkan dataset secara otomatis. Halaman ini mengandung beberapa komponen penting yang mendukung proses *input* dokumen sampai masuk ke database vektor. Komponen yang ada di halaman sisi admin ini adalah Unggah PDF,

admin dapat mengunggah satu *file* PDF sumber referensi yang terkait dengan topik. Teks dari *file* PDF ini akan diekstraksi dan diproses hingga dimasukkan ke dalam database vektor.

Gambar 4.15 adalah *interface* sisi *user* yang dirancang untuk membantu *user* input pertanyaan. Halaman ini mengandung beberapa komponen penting yang mendukung proses *input* pertanyaan hingga sistem *generate* jawaban. Komponen-komponen yang ada pada halaman sisi *user* diuraikan di bawah ini:

1. *Input* Pertanyaan/*Prompt*

   User diminta untuk memasukkan satu atau lebih *prompt* atau pertanyaan dalam *text area*. Pertanyaan ini akan digunakan untuk membuat jawaban yang relevan. *User* diusahakan membuat pertanyaan/*prompt* yang lebih spesifik mengenai topik yang ingin dihasilkan agar sesuai dengan kebutuhan *user*.

2. Tombol *Generate*

   Setelah semua informasi dimasukkan, *user* dapat menekan tombol "Dapatkan Jawaban" untuk memulai proses pembuatan jawaban. Model RAG akan digunakan oleh sistem untuk mengumpulkan informasi yang relevan dari jurnal-jurnal relevan yang telah di-*embed*. Selanjutnya, menggunakan *llm-4-scout-17b-instruct* untuk membuat jawaban berdasarkan pertanyaan yang diberikan.

4.2 Analisis Hasil Penelitian

4.2.1 Analisa Kelebihan Sistem

1. Otomatisasi Ringkasan Literatur Berbahasa Indonesia

Sistem mampu secara otomatis menghasilkan ringkasan dari berbagai jurnal atau dokumen ilmiah dalam bahasa Indonesia. Hal ini sangat membantu pengguna dalam memahami inti sari dari beberapa artikel tanpa harus membaca keseluruhannya. Penggunaan model IndoT5 yang telah *dituning* untuk tugas *summarization* memungkinkan sistem menghasilkan ringkasan yang cukup informatif, ringkas, dan sesuai konteks isi dokumen.

2. **Penerapan Arsitektur *Retrieval-Augmented Generation* (RAG)**

   Dengan menggabungkan dua proses utama—*retrieval* dan *generation*—sistem mampu menghasilkan jawaban berbasis konteks yang lebih relevan. Proses *retrieval* menggunakan FAISS dan *embedding* menggunakan IndoBERT memungkinkan sistem menemukan dokumen relevan dari corpus, lalu bagian pentingnya diringkas menggunakan model IndoT5. Ini menciptakan keseimbangan antara kecerdasan informasi dari sumber nyata (*retrieval*) dan kekuatan *natural language generation* dari LLM.

3. **Evaluasi Jawaban Menggunakan *LLM-as-a-Judge***

   Sistem memiliki fitur evaluasi otomatis menggunakan pendekatan *LLM-as-a-Judge* untuk menilai kualitas jawaban berdasarkan tiga aspek: relevansi, kebenaran (*faithfulness*), dan kelengkapan (*completeness*). Ini menjadikan evaluasi tidak hanya objektif tetapi juga konsisten, karena tidak lagi bergantung pada penilaian manual. Skor skala 1–5 untuk tiap dimensi memberikan granularitas dalam melihat kekuatan atau kelemahan jawaban sistem.

4. **Fleksibilitas Pertanyaan dan Input Dinamis**

   Pengguna dapat memberikan pertanyaan terbuka terkait isi dokumen, dan sistem akan merespons berdasarkan konteks ter-retrieved. Hal ini memungkinkan sistem digunakan tidak hanya untuk ringkasan tetapi juga untuk Q&A literatur akademik, memperluas kegunaan sistem ke arah asisten penelitian.

5. **Integrasi dengan UI Streamlit yang Interaktif**

   Sistem dikembangkan menggunakan antarmuka Streamlit, yang memudahkan pengguna dalam mengunggah dokumen, mengajukan pertanyaan, dan melihat hasil evaluasi dalam bentuk tabel dan visual. Ini menjadikan sistem ramah pengguna dan siap digunakan sebagai prototipe aplikasi web berbasis penelitian akademik.

4.2.2 Analisa Kekurangan Sistem

Meskipun sistem menunjukkan performa yang menjanjikan, terdapat beberapa kelemahan yang memengaruhi kualitas hasil dan fleksibilitas sistem:

1. **Ketergantungan pada Kualitas Embedding dan Dokumen Awal**

   Apabila *embedding IndoBERT* tidak merepresentasikan konteks pertanyaan secara akurat, dokumen yang diretrieval bisa kurang relevan. Hal ini berdampak langsung pada ringkasan IndoT5 yang bergantung pada *input* tersebut. Untuk mengatasinya, diperlukan peningkatan pada model *embedding* atau eksplorasi *embedding* alternatif seperti *SBERT* atau *XLM-RoBERTa* versi Indonesia.

2. **Batasan Panjang Konteks pada Model IndoT5**

   Model IndoT5 memiliki batasan *input* maksimal (sekitar 512–1024 token), sehingga informasi penting dari dokumen panjang mungkin terpotong dan tidak diproses. Hal ini menurunkan kelengkapan ringkasan. Solusi potensialnya adalah dengan menerapkan segmentasi IMRAD lebih halus dan eksplisit atau menggunakan model *long-context* seperti LongT5.

3. **Evaluasi Belum Menggunakan Dataset *Benchmark***

   Evaluasi sistem masih bersifat eksperimental dan belum divalidasi dengan dataset standar seperti IndoSum atau *IDN Summarization*. Akibatnya, klaim performa belum dapat dibandingkan secara objektif dengan studi lain. Langkah perbaikan ke depan adalah melakukan evaluasi tambahan menggunakan dataset *benchmark* untuk mengukur generalisasi sistem.

4. **Belum Mendukung *Multi-Modal Input***

   Sistem hanya menerima teks sebagai input. Padahal banyak literatur ilmiah menyajikan data penting dalam bentuk grafik atau tabel. Implikasinya, sistem gagal menangkap informasi visual. Perlu dilakukan pengembangan ke arah model *multi-modal* atau integrasi OCR dan ekstraksi tabel.

5. **Ketergantungan terhadap API Eksternal untuk Evaluasi**

   Fitur evaluasi berbasis *LLM-as-a-Judge* menggunakan model *LLaMA-3* via API eksternal (Groq/OpenAI). Ini menimbulkan keterbatasan pada ketersediaan, privasi data, dan biaya. Untuk mengurangi ketergantungan, dapat dikembangkan evaluator lokal berbasis model bahasa *open-source* yang dijalankan secara offline.

6. **Potensi *Hallucination* pada Model Generatif**

   Model IndoT5 kadang menghasilkan kalimat yang tampak koheren namun tidak sesuai isi dokumen. Fenomena ini disebut *hallucination*. Untuk mengurangi efek ini, sistem perlu dilengkapi dengan verifikasi atau *reranking post-generation* yang membandingkan *output* dengan isi dokumen sumber.

7. **Terbatasnya Jumlah Dokumen pada Dataset per Topik**

   Dataset terdiri dari 30 dokumen per kategori (*supervised, unsupervised*, dan *reinforcement learning*). Jumlah ini terbatas mengingat luasnya topik dan variasi gaya penulisan. Hal ini berdampak pada cakupan sistem dan mempersempit kemampuan generalisasi. Pengumpulan dokumen tambahan dan integrasi dengan sumber jurnal Indonesia lainnya sangat direkomendasikan.

   Secara umum, kelemahan-kelemahan tersebut menunjukkan bahwa sistem masih memiliki ruang pengembangan. Penulis menyadari keterbatasan tersebut dan telah menyusun rekomendasi saran pengembangan lebih lanjut sebagaimana dijelaskan pada bagian 5.2.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan evaluasi, dapat disimpulkan bahwa:

1. Sistem otomatisasi kajian literatur berbasis *Retrieval-Augmented Generation* (RAG) dengan *summarization* IndoT5 mampu merangkum literatur akademik berbahasa Indonesia secara otomatis dan responsif terhadap pertanyaan pengguna.
2. Model IndoT5 efektif dalam merangkum bagian struktur IMRAD, sehingga mendukung proses *indexing* dan *retrieval* dengan hasil yang lebih terstruktur.
3. Proses evaluasi menggunakan *BERTScore* menunjukkan hasil ringkasan yang mendekati kualitas abstrak asli, dan evaluasi respon sistem menggunakan *LLM-as-a-Judge* memberikan skor rata-rata di atas 4.0 untuk relevansi, *faithfulness*, dan kelengkapan.
4. Sistem mampu menjawab kebutuhan peneliti terhadap pencarian dan pemahaman literatur secara efisien, kontekstual, dan interaktif.
5. Meskipun sistem menunjukkan potensi besar, masih terdapat beberapa kelemahan, seperti keterbatasan panjang *input*, potensi *hallucination*, dan ketergantungan pada API eksternal. Namun, kekurangan ini telah dianalisis dan menjadi dasar untuk pengembangan lebih lanjut.

5.2 Saran

Untuk pengembangan selanjutnya, disarankan:

1. Melakukan *fine-tuning* model IndoT5 menggunakan korpus jurnal akademik Indonesia dari berbagai disiplin ilmu untuk meningkatkan akurasi dan cakupan ringkasan.
2. Menambahkan fitur verifikasi otomatis untuk meminimalkan risiko *hallucination*, seperti *reranking* atau perbandingan *output* terhadap isi dokumen.

3. Mengintegrasikan *pipeline* otomatis dari unggah dokumen hingga proses *indexing* untuk efisiensi sistem secara menyeluruh.
4. Menambahkan dukungan *input multi-modal*, seperti grafik dan tabel, melalui integrasi OCR atau ekstraksi tabel otomatis.
5. Menggunakan *evaluator* lokal berbasis *open-source* agar sistem dapat digunakan secara *offline* tanpa ketergantungan API eksternal.
6. Memperluas jumlah dan ragam dokumen dalam dataset serta menguji sistem terhadap domain non-AI untuk mengukur generalisasi.
7. Melakukan evaluasi lanjutan menggunakan dataset *benchmark* agar performa sistem dapat dibandingkan secara objektif dalam skala akademik.

DAFTAR PUSTAKA

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners*. http://arxiv.org/abs/2005.14165

Cahyawijaya, S., Winata, G. I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., Ruder, S., Lim, Z. Y., Bahar, S., Khodra, M. L., Purwarianti, A., & Fung, P. (2021). *IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation*. http://arxiv.org/abs/2104.08200

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. Dalam *IEEE Computational Intelligence Magazine* (Vol. 9, Nomor 2, hlm. 48–57). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MCI.2014.2307227

Cheng, M., Luo, Y., Ouyang, J., Liu, Q., Liu, H., Li, L., Yu, S., Zhang, B., Cao, J., Ma, J., Wang, D., & Chen, E. (2025). *A Survey on Knowledge-Oriented Retrieval-Augmented Generation*. http://arxiv.org/abs/2503.10677

Groq Inc. (2024). *Groq LLaMA3 Inference Platform*.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., & Guo, J. (2024). *A Survey on LLM-as-a-Judge*. http://arxiv.org/abs/2411.15594

Gupta, S., & Ranjan, R. (2024). *A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions*.

Hahsler, M. (2023). *ARULESPY: Exploring Association Rules and Frequent Itemsets in Python*. http://arxiv.org/abs/2305.15263

Handoyo, S., Prastiti, P. I. D., & Stiaji, I. R. (2024). Bibliometric analysis of publications trends in Indonesian research institutions: A comparison of pre-integration (2015–2021) and post-integration (2022–2023) periods. *European Science Editing*, *50*. https://doi.org/10.3897/ese.2024.e118015

Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and Advances in Information Extraction from Scientific Literature: a Review. Dalam *JOM* (Vol. 73, Nomor 11, hlm. 3383–3400). Springer. https://doi.org/10.1007/s11837-021-04902-9

Jaber, E. A., & Gérard, L.-A. (2025). *Signature volatility models: pricing and hedging with Fourier*. https://doi.org/10.1137/24M1636952

Johnson, J., Douze, M., & Jégou, H. (2017). *Billion-scale similarity search with GPUs*. http://arxiv.org/abs/1702.08734

Koto, F., Lau, J. H., & Baldwin, T. (2021). *INDOBERTWEET: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization*. https://huggingface.co/huseinzol05/

Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., Shu, K., Cheng, L., & Liu, H. (2024). *From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge*. http://arxiv.org/abs/2411.16594

Meta AI. (2024). *Meta LLaMA 3 Open Weights*.

Miao, J., Thongprayoon, C., Suppadungsuk, S., Garcia Valencia, O. A., & Cheungpasitporn, W. (2024). Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications. *Medicina (Lithuania)*, *60*(3), 1–15. https://doi.org/10.3390/medicina60030445

Muhammad, T., Rahardiansyah, R., Setya Perdana, R., & Fatyanosa, T. N. (2025). *Analisis Teknik Embedding Model NV-Embed pada Large Language Models Berbasis Retrieval Augmented Generation* (Vol. 9, Nomor 2). http://j-ptiik.ub.ac.id

Omiye, J. A., Gui, H., Rezaei, S. J., Zou, J., & Daneshjou, R. (2023). *Large language models in medicine: the potentials and pitfalls*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Dalam *Journal of Machine Learning Research* (Vol. 21). http://jmlr.org/papers/v21/20-074.html.

Ridwan, M., Ulum, B., Muhammad, F., Indragiri, I., & Sulthan Thaha Saifuddin Jambi, U. (2021). *Pentingnya Penerapan Literature Review pada Penelitian Ilmiah (The Importance Of Application Of Literature Review In Scientific Research)*. http://journal.fdi.or.id/index.php/jmas/article/view/356

Sakti Wiradinata, A., Viny,), & Mawardi, C. (2024). *Jurnal Ilmu Komputer dan Sistem Informasi Abstractive Text Summarization Berita Bahasa Indonesia Menggunakan Retrieval-Augmented Generation*. https://www.cnbcindonesia.com/indeks

Sollaci, L. B., & Pereira, M. G. (2004). The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. Dalam *J Med Libr Assoc* (Vol. 92, Nomor 3).

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023a). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023b). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need*.

Wilie, B., Vincentio, K., Winata, G. I., Cahyawijaya, S., Li, X., Lim, Z. Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., & Purwarianti, A. (2020). *IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding*. http://arxiv.org/abs/2009.05387

Yani, M., Siti Khodijah, N., & Mustamiin, M. (2024). *Aplikasi Peringkas Teks Bahasa Indonesia Menggunakan Model Text-to-Text Transfer Transformer (T5)*. https://doi.org/10.37817/ikraith-informatika.v9i2

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). *BERTScore: Evaluating Text Generation with BERT*. http://arxiv.org/abs/1904.09675