**SKRIPSI**

**IMPLEMENTASI SELF-REFLECTIVE**

**RETRIEVAL-AUGMENTED GENERATION PADA**

**LARGE LANGUAGE MODEL BERBASIS**

**DEEPSEEK MoE MENGGUNAKAN FRAMEWORK LANGCHAIN**

Ditulis untuk memenuhi sebagian persyaratan akademik

guna memperoleh gelar Sarjana Komputer

**Oleh:**

**NAMA : Jovan Torio**

**NPM : 03082220026**

****

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS PELITA HARAPAN**

**MEDAN**

**2025**

# KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa sehingga peneliti dapat menyelesaikan Tugas Akhir ini tepat pada waktunya dan dapat dikerjakan sebaik mungkin. Tugas Akhir yang berjudul “IMPLEMENTASI SELF-REFLECTIVE RETRIEVAL-AUGMENTED GENERATION PADA LARGE LANGUAGE MODEL BERBASIS DEEPSEEKMOE MENGGUNAKAN FRAMEWORK LANGCHAIN”ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer Strata Satu Universitas Pelita Harapan, Medan.

Peneliti menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, Tugas Akhir ini dapat diselesaikan dengan baik dan tepat waktu. Oleh karena itu, peneliti menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses pengerjaan Tugas Akhir ini, terutama kepada:

1. Bapak Dr. (Hon.) Jonathan L. Parapak, M.Eng.Sc, selaku rektor Universitas Pelita Harapan.
2. Bapak Andree Emmanuel Widjaja, Ph.D. selaku Dekan Fakultas Teknologi Informasi.
3. Ibu Ferawaty, S.Kom, M.Kom., selaku Ketua Program Studi Informatika (Kampus Kota Medan).
4. Bapak Wenripin Chandra, S.Kom., MTI, selaku Dosen Pembimbing yang telah memberikan bimbingan dan masukan yang sangat berharga kepada peneliti.
5. Bapak Benz Edy Kusuma M.Si, selaku Penasehat Akademik peneliti.
6. Seluruh dosen di Universitas Pelita Harapan yang telah memberikan ilmu dan pengalaman berharga selama masa perkuliahan.
7. Kedua orang tua peneliti yang telah memberikan doa, dukungan, dan kasih sayang tiada henti kepada peneliti.
8. Teman-teman seperjuangan yang selalu memberikan semangat dan bantuan kepada peneliti selama perkuliahan.
9. Kepada teman-teman dekat dan seluruh pihak yang selalu mendukung dan membantu peneliti yang tidak dapat disebutkan satu per satu.

Akhir kata, peneliti menyadari bahwa masih ada kekurangan dalam penulisan Tugas Akhir ini sehingga peneliti menerima kritik dan saran dari pembaca yang bermanfaat bagi peneliti untuk penyempurnaan. Demikian Tugas Akhir ini dibuat, semoga Tugas Akhir ini dapat membawa manfaat bagi semua pihak yang membacanya.

Medan, 2 Mei 2025

Peneliti

# DAFTAR ISI

[KATA PENGANTAR vi](#_Toc203144331)

[DAFTAR ISI viii](#_Toc203144332)

[DAFTAR GAMBAR x](#_Toc203144333)

[DAFTAR TABEL xi](#_Toc203144334)

[BAB 1 PENDAHULUAN 1](#_Toc203144335)

[1.1 Latar Belakang Masalah 1](#_Toc203144336)

[1.2 Rumusan Masalah 3](#_Toc203144337)

[1.3 Tujuan Penelitian 4](#_Toc203144338)

[1.4 Batasan Masalah 4](#_Toc203144339)

[1.5 Metodologi Penelitian 5](#_Toc203144340)

[1.6 Manfaat Penelitian 5](#_Toc203144341)

[1.7 Sistematika Penulisan 5](#_Toc203144342)

[BAB 2 TINJAUAN PUSTAKA 7](#_Toc203144343)

[2.1 Landasan Teori 7](#_Toc203144344)

[2.1.1 Artificial Intelligence (AI) 7](#_Toc203144345)

[2.1.2 Natural Language Processing (NLP) 9](#_Toc203144346)

[2.1.3 Machine Learning (ML) 11](#_Toc203144347)

[2.1.4 Deep Learning (DL) 13](#_Toc203144348)

[2.1.5 Arsitektur Transformer 15](#_Toc203144349)

[2.1.6 Large Language Models (LLM) 17](#_Toc203144350)

[2.1.7 Arsitektur Mixture-of-Experts (MoE) 18](#_Toc203144351)

[2.1.8 DeepSeekMoE 20](#_Toc203144352)

[2.1.9 Retrieval-Augmented Generation (RAG) 23](#_Toc203144353)

[2.1.10 Self-Reflective Retrieval-Augmented Generation (Self-RAG) 25](#_Toc203144354)

[2.1.11 LangChain 27](#_Toc203144355)

[2.2 Penelitian Terdahulu 30](#_Toc203144356)

[BAB 3 METODOLOGI PENELITIAN 33](#_Toc203144357)

[3.1 Metode Pengumpulan Data 33](#_Toc203144358)

[3.2 Kerangka Pikir 34](#_Toc203144359)

[3.3 Tahapan Penelitian 38](#_Toc203144360)

[3.4 Metode Penyelesaian 41](#_Toc203144361)

[3.5 Metode Pengukuran Akurasi 44](#_Toc203144362)

[DAFTAR PUSTAKA 46](#_Toc203144363)

# DAFTAR GAMBAR

[Gambar 2.1 Jenis AI Sumber: Thareja, 2024 9](#_Toc203140538)

[Gambar 2.2 Klasifikasi NLP Sumber: Khurana et al., 2022 11](#_Toc203140539)

[Gambar 2.3 Kategori *Machine Learning* Sumber: Singh et al., 2025 13](#_Toc203140540)

[Gambar 2.4 Arsitektur Transformer Sumber: Atkinson-Abutridy, 2025 16](#_Toc203140541)

[Gambar 2.5 Arsitektur Mixture-of-Experts (MoE) Sumber: Kamath et al., 2025 20](#_Toc203140542)

[Gambar 2.6 Perbandingan DeepSeekMoE dengan LLM *open source* Sumber: Dai et al., (2024) 22](#_Toc203140543)

[Gambar 2.7 Perbandingan antara lapisan arsitektur MoE pada subfigur (a) dengan mekanisme yang digunakan DeepSeekMoE pada subfigur (b) dan (c). 23](#_Toc203140544)

[Gambar 2.8 Diagram konsep dasar cara kerja RAG. Sumber: Kamath et al., 2022 25](#_Toc203140545)

[Gambar 2.9 Diagram perbandingan proses RAG tradisional dan Self-Reflective Retrieval-Augmented Generation (Self-RAG) 27](#_Toc203140546)

[Gambar 2.10 Tabel yang menunjukkan jenis *reflection* tokenyang digunakan dalam Self-RAG. Token IsRel, IsSup, dan IsUse merupakan jenis *critique* token. 28](#_Toc203140547)

[Gambar 3.1 Kerangka Berpikir Penelitian 35](#_Toc203140548)

[Gambar 3.2 Tahapan Penelitian 38](#_Toc203140549)

[Gambar 3.3 Metode Penyelesaian 41](#_Toc203140550)

# DAFTAR TABEL

[Tabel 2.1 Penelitian terdahulu 30](#_Toc203140523)

# PENDAHULUAN

## Latar Belakang Masalah

Di era transformasi digital, pengambilan keputusan berbasis data semakin dibutuhkan, yang mendorong solusi berbasis *artificial intelligence* (AI) yang lebih canggih. Sebuah laporan menunjukkan bahwa ukuran pasar AI global bernilai USD 279,22 miliar pada tahun 2024 dan diproyeksikan tumbuh dengan *compound annual growth rate* (CAGR) sebesar 35,9% dari tahun 2025 hingga 2030 (Grand View Research, 2024). Salah satu teknologi AI adalah *large language models* (LLM), yang digunakan untuk menyelesaikan tugas bahasa kompleks seperti penerjemahan, peringkasan, dan tanya jawab otomatis. Sejak kemunculan ChatGPT, sebuah aplikasi berbasis AI yang merespons kueri pengguna secara otomatis, LLM menarik perhatian signifikan dari kalangan umum dan mendapatkan dorongan perkembangan (Kamath et al., 2025).

Meskipun LLM menunjukkan performa yang mendekati manusia dalam tugas-tugas tertentu (OpenAI et al., 2023), teknologi ini memiliki keterbatasan. Contohnya adalah halusinasi, yaitu ketika LLM menghasilkan informasi yang tidak benar (Ji et al., 2022), dan ukuran input terbatas, yang mengakibatkan LLM melupakan informasi dari input karena hanya dapat memproses sejumlah kata dalam satu waktu (Hosseini et al., 2024). *Retrieval-augmented generation* (RAG) diperkenalkan sebagai solusi untuk kedua keterbatasan ini, yang memungkinkan LLM untuk mengambil informasi dari dokumen eksternal dan menggunakannya untuk menghasilkan jawaban yang lebih kredibel.

RAG efektif untuk menjaga relevansi jawaban, tetapi menghadapi masalah seperti ketidakakuratan dalam proses pengambilan dokumen dan kesulitan dalam menjawab pertanyaanyang membutuhkan beberapa langkah penalaran. Berbagai teknik dikembangkan untuk menyelesaikan masalah ini, seperti *re-ranking* (Ma et al., 2023) dan Iterative RAG (Shao et al., 2023). Suatu pendekatan unik adalah Self-Reflective Retrieval-Augmented Generation (Self-RAG) yang memungkinkan LLM tidak hanya mengambil dokumen, tetapi juga melakukan evaluasi mandiri terhadap keandalan sumber sebelum menghasilkan jawaban (Asai et al., 2023).

Di sisi lain, seiring dengan perkembangan LLM, kebutuhan komputasinya juga semakin besar. Pada umumnya, perkembangan LLM secara tradisional meliputi peningkatan ukuran model. Akan tetapi, performa LLM tradisional tidak sebanding dengan potensi yang bisa dicapai dengan anggaran yang sama (Hoffmann et al., 2022). Solusinya adalah arsitektur Mixture-of-Experts (MoE) yang mengaktifkan jaringan saraf *(expert)* tertentu yang relevan untuk tugas yang diberikan (Kamath et al., 2025). Arsitektur MoE terbukti menggunakan data pelatihan dan sumber daya dengan lebih efisien (Wang et al., 2024).

Akan tetapi, arsitektur MoE juga memiliki kelemahan, yaitu hibriditas dan redundansi pengetahuan, yang menghambat spesialisasi *expert* dan menurunkan efektivitas model. Untuk mengatasi kendala ini, sebuah perusahaan AI bernama DeepSeek mengembangkan arsitektur DeepSeekMoE, yang bertujuan untuk meningkatkan efisiensi sekaligus kualitas representasi pengetahuan dalam LLM (Dai et al., 2024).

Meskipun Self-RAG telah terbukti efektif dalam meningkatkan akurasi jawaban pada LLM tradisional seperti Llama dan ChatGPT (Asai et al., 2023) dan LLM berbasis MoE (Islam et al., 2024), hingga saat ini belum ada penelitian yang mengeksplorasi pendekatan ini pada LLM berbasis DeepSeekMoE yang menawarkan spesialisasi *expert* yang lebih baik.

Dengan demikian, penelitian ini bertujuan untuk menjembatani kesenjangan tersebut dengan mengevaluasi performa Self-RAG pada LLM berbasis DeepSeekMoE.

## Rumusan Masalah

Dari latar belakang masalah yang ada, maka perumusan masalah sebagai berikut:

1. Belum ada penelitian yang mengeksplorasi pendekatan Self-Reflective Retrieval-Augmented Generation (Self-RAG) pada *large language model* (LLM) berbasis DeepSeekMoE.

Adapun pertanyaan penelitian sebagai berikut:

1. Bagaimana kinerja akurasi jawaban yang dihasilkan *large language model* (LLM) berbasis DeepSeekMoE saat menggunakan Self-Reflective Retrieval-Augmented Generation (Self-RAG)?
2. Apakah terdapat perbedaan performa antara *large language model* (LLM) berbasis DeepSeekMoE yang menggunakan Self-Reflective Retrieval-Augmented Generation (Self-RAG) dan yang tidak?

## Tujuan Penelitian

Setelah merumuskan dan mengidentifikasi masalah, peneliti mendeskripsikan tujuan penelitian yaitu mengimplementasi dan menguji Self-Reflective Retrieval-Augmented Generation (Self-RAG) pada *large language model* berbasis DeepSeekMoe.

## Batasan Masalah

Agar penelitian dapat dilakukan dalam ruang lingkup yang jelas, maka pada penelitian ini ditentukan beberapa batasan masalah, yaitu:

1. Arsitektur *large language model* (LLM) yang menjadi fokus penelitian adalah arsitektur DeekSeekMoE. LLM yang digunakan adalah deepseek-moe-16b-chat.
2. Penelitian ini membandingkan dua konfigurasi model, yaitu:
   1. LLM berbasis DeepSeekMoE yang menggunakan Self-Reflective Retrieval-Augmented Generation (Self-RAG).
   2. LLM berbasis DeepSeekMoE tanpa menggunakan Self-RAG.
3. Metrik evaluasi yang digunakan adalah akurasi.
4. Data evaluasi yang digunakan dalam penelitian bersumber dari *dataset* tolok ukur *(benchmark dataset)* yang telah digunakan dalam penelitian sebelumnya*,* yaitu TriviaQA-unfiltered, PopQA *long-tail subset*, dan ARC-Challenge.
5. Penelitian ini tidak mencakup pengembangan LLM berbasis DeepSeekMoE dari awal, melainkan memanfaatkan LLM *open source* yang telah tersedia.
6. Penelitian ini tidak melakukan *fine-tuning* terhadap LLM berbasis DeepSeekMoE dengan Self-RAG, melainkan menggunakan *framework* LangChain.
7. Penelitian ini menggunakan platform pengembangan Kaggle dan bahasa pemrograman Python.

## Metodologi Penelitian

Metodologi penelitian yang dilakukan adalah sebagai berikut:

1. Pengumpulan data

Pada tahap ini, peneliti mengumpulkan dan mengolah data berupa dokumen-dokumen sumber pengetahuan tekstual, yaitu dokumen berbasis teks yang digunakan dalam proses retrieval Self-RAG, seperti korpus Wikipedia yang telah diproses sebelumnya serta dataset pelatihan.

Selain itu, dikumpulkan juga dataset tolok ukur untuk mengevaluasi akurasi LLM. Dataset benchmark yang digunakan adalah TriviaQA-unfiltered, PopQA long-tailed Subset, dan ARC-Challenge.

Peneliti juga melakukan studi literatur dan analisis penelitian sebelumnya yang berkaitan dengan Self-RAG dan DeepSeekMoE. Studi ini dilakukan untuk mendapatkan pemahaman awal tentang metode yang digunakan.

1. Desain Program menggunakan LangChain

Peneliti mengimplementasikan Self-RAG pada LLM berbasis DeepSeekMoE menggunakan framework LangChain dengan tahapan sebagai berikut:

* + - * 1. Membangun komponen *retrieval* yang digunakan LLM untuk mencari dokumen dengan *embedding* yang mirip dengan *embedding* input.
        2. Integrasi DeepSeekMoE pada program agar dapat diinjeksikan dokumen hasil *retrieval* dan kueri *input*.
        3. Implementasi Self-RAG dengan memungkinkan *output* LLM untuk sekaligus mengeluarkan token *reflections*.
        4. Menguji dan menganalisis akurasi model terhadap dataset tolok ukur. LLM berbasis DeepSeekMoE tanpa Self-RAG juga diuji akurasinya pada dataset tolok ukur untuk perbandingan.

1. Pembuatan Laporan

Tahap terakhir adalah pembuatan laporan yang menjelaskan hasil pengujian implementasi Self-RAG terhadap LLM berbasis DeepSeekMoE.

## Manfaat Penelitian

Terdapat dua manfaat dari hasil penelitian ini, yaitu:

1. Bagi pengembangan teori ilmu komputer, penelitian ini memberikan pemahaman mengenai implementasi Self-Reflective Retrieval-Augmented Generation (Self-RAG) pada arsitektur Large Language Model (LLM) berbasis DeepSeekMoE.
2. Bagi para praktisi, hasil penelitian ini dapat menjadi referensi dalam menentukan strategi Retrieval-Augmented Generation (RAG) dan arsitektur LLM yang sesuai.

## Sistematika Penulisan

Pada penelitian ini, sistematika penulisan terdiri dari lima buah bab dengan uraian sebagai berikut:

1. BAB I PENDAHULUAN

Pada bab pertama, diuraikan dasar dilakukannya penelitian dari masalah yang diangkat, perumusan permasalahan, tujuan dari penelitian, batasan masalah, manfaat penelitian, dan penjelasan mengenai sistematika penulisan.

1. BAB II TINJAUAN PUSTAKA

Pada bab kedua, dijelaskan teori yang mendukung penelitian, mencakup teori tentang *artificial intelligence* (AI), *natural language processing* (NLP), *machine learning* (ML), *deep learning* (DL), arsitektur Transformer, *large language models* (LLM), arsitektur Mixture-of-Experts (MoE), arsitektur DeepSeekMoE, *retrieval-augmented generation* (RAG), Self-Reflective Retrieval-Augmented Generation (Self-RAG), LangChain, serta penelitian terdahulu.

1. BAB III METODOLOGI PENELITIAN

Pada bab ketiga, diuraikan metode yang digunakan dalam penelitian, kerangka pemikiran yang mendasari penelitian ini, mulai dari perancangan eksperimen, pemilihan *dataset*, pengukuran akurasi jawaban, serta desain program Self-RAG menggunakan *framework* LangChain.

1. BAB IV HASIL DAN PEMBAHASAN

Pada bab keempat, disajikan capaian hasil penelitian yang dilakukan serta dilakukan pembahasan terkait hasil penemuan.

1. BAB V KESIMPULAN DAN SARAN

Pada bab kelima, dirangkum apa yang telah dibahas di Bab IV (Hasil dan Pembahasan) dan menyampaikan saran bagi penelitian berikutnya.

# TINJAUAN PUSTAKA

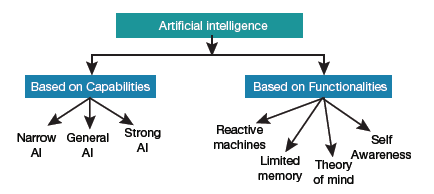
## Landasan Teori

### Artificial Intelligence (AI)

John McCarthy mendefinisikan Artificial Intelligence (AI) pada tahun 1955 sebagai ilmu pengetahuan dan teknik untuk membuat mesin, terutama program komputer yang cerdas (Khan, 2024). AI adalah sekumpulan algoritma yang menghasilkan *output* tanpa harus diinstruksikan secara eksplisit untuk melakukannya. Hal ini berbeda dengan program komputer pada umumnya, yang menghasilkan *output* sesuai dengan instruksi pengguna dalam bentuk kode pemrograman. Aplikasi AI melakukan tugas-tugas khusus dengan memproses data dalam jumlah besar dan mengenali pola di dalamnya (Thareja, 2024).

Tujuan AI adalah menciptakan teknologi yang memungkinkan komputer dan mesin untuk melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Dengan demikian, kemampuan kognitif manusia ditingkatkan, dan pada akhirnya hidup manusia lebih mudah, efisien, dan produktif. AI dapat melakukan tugas-tugas intelektual seperti memberikan solusi, menjawab pertanyaan, membuat prediksi, memecahkan masalah, pengambilan keputusan, menawarkan saran strategis, dan memahami komunikasi manusia (Khan, 2024).

Sistem AI lebih efektif ketika diberikan sejumlah besar data pelatihan berlabel. Data ini dianalisis secara menyeluruh untuk menemukan pola. Pola-pola ini digunakan untuk membuat prediksi tentang kondisi di masa depan. Sebagai contoh, sebuah bot obrolan yang diberi contoh obrolan teks dapat belajar untuk berkomunikasi dengan manusia dalam aplikasi dunia nyata (Thareja, 2024).



Gambar 2.1 Jenis AI  
Sumber: Thareja, 2024

Menurut Thareja (2024), sistem AI dapat dikategorikan berdasarkan kemampuannya, yaitu:

1. Weak AI

Jenis AI yang dirancang khusus untuk melakukan jenis tugas tertentu. Contohnya adalah Siri dan Alexa. AI ini dilatih dengan respons yang sesuai untuk mengklasifikasikan berbagai hal.

1. Strong AI

Jenis AI yang diusahakan untuk sepenuhnya menyerupai otak manusia. AI ini menggunakan keterampilan kognitif dan logika *fuzzy* untuk melakukan tugas-tugas yang belum pernah dilatih sebelumnya.

1. General AI

Jenis AI fiktif yang meniru kecerdasan manusia dan berpikir secara strategis untuk melakukan tugas-tugas yang kompleks.

Selain itu, Thareja (2024) juga berpendapat bahwa sistem AI juga dapat dikategorikan ke dalam empat kelompok berdasarkan fungsinya, yaitu:

1. Reactive Machines

Mesin yang sangat mendasar tanpa memori penyimpanan dan dengan demikian menggunakan pengalaman masa lalu untuk menentukan tindakan di masa depan. Jenis mesin ini hanya melihat dunia dan bereaksi terhadapnya.

1. Limited Memory Machines

Mesin yang menyimpan data tetapi hanya untuk waktu yang singkat dan hanya dapat menggunakannya dalam waktu yang terbatas. Jenis mesin ini tidak dapat terus menambahkan data secara permanen ke dalam perpustakaan pengalaman mereka.

1. Theory of Mind

Mesin yang berfokus pada peniruan otak manusia. Jenis mesin ini membentuk representasi tentang dunia, mulai dari pikiran, emosi, dan ingatan.

1. Self-Awareness

Mesin yang memiliki kesadaran diri sehingga memiliki kesadaran tingkat manusia, memahami kondisi dan keberadaannya sendiri di dunia, dan menggunakan informasi tersebut untuk menyimpulkan perasaan orang lain.

### Natural Language Processing (NLP)

*Natural language processing* (NLP) merupakan salah satu cabang AI yang berfokus untuk membuat mesin mampu memahami kata atau kalimat dalam bahasa manusia. Bahasa manusia dimaksud dengan serangkai aturan atau simbol yang digunakan untuk menyampaikan informasi (Khurana et al., 2022).

NLP memiliki penerapan yang luas di berbagai bidang. Dalam dunia bisnis, analisis sentimen digunakan untuk memperoleh opini dan tanggapan pelanggan dari data teks secara otomatis guna memahami preferensi pasar dan merumuskan strategi yang lebih efektif (Ahmed et al., 2022). Di bidang kesehatan, NLP dimanfaatkan untuk menganalisis data medis pasien (Clarido et al., 2023).

A diagram of a language processing

AI-generated content may be incorrect.

Gambar 2.2 Klasifikasi NLP  
Sumber: Khurana et al., 2022

Dalam dunia media, *named entity recognition* (NER) digunakan untuk mendeteksi entitas dalam berita guna mempermudah klasifikasi konten (Chavan & Patil, 2024). Mesin pencari seperti Google dan Bing juga menggunakan NLP untuk menerjemahkan isi halaman web ke dalam berbagai bahasa secara otomatis.

Berdasarkan tugas, NLP dibagi menjadi dua, Natural Language Understanding (ilmu bahasa) dan Natural Language Generation (NLG). NLU menganalisis arti bahasa manusia dan mencari konotasi dan implikasi dalam pernyataan manusia, yang mencakup emosi atau niat mereka (Karanikolas et al., 2023). Terdapat 5 aspek dalam NLU, yaitu *phonology* (suara bahasa), *morphology* (formasi kata), *syntax* (struktur kalimat), *semantics* (makna), dan *pragmatics* (bagaimana makna digunakan) (Khurana et al., 2022).

NLG berfokus pada teknologi yang mampu menghasilkan frasa, kalimat, atau paragraf yang seolah-olah dibuat manusia. NLG menggunakan NLU untuk mengotomatisasikan tugas-tugas menulis, mengurangi kesalahan dalam tulisan, dan menghasilkan informasi yang spesifik pada kebutuhan individu (Karanikolas et al., 2023).

### Machine Learning (ML)

*Machine learning* (ML) adalah suatu cabang AI yang berfokus untuk mengungkapkan pola di dalam data dan membangun model yang memecahkan masalah di dunia nyata. Teknologi ini membawa perkembangan di berbagai bidang, mulai dari otomatisasi tugas hingga menghasilkan temuan yang cerdas (Vasques, 2024).

Alur kerja ML dimulai dari pengelolaan data, yaitu proses pengumpulan dan pengubahan data mentah dari berbagai sumber menjadi format terstruktur melalui proses ETL (*Extract, Transform, Load*). Selanjutnya, data diproses dengan rekayasa fitur untuk membentuk atribut bermakna guna mendukung kinerja model. Atribut ini digunakan untuk melatih dan mengevaluasi model ML dalam sebuah proses iteratif. Melalui penyesuaian dan seleksi, dicari dan diaplikasikan model yang paling efektif untuk penggunaan nyata (Vasques, 2024).

A diagram of machine learning

AI-generated content may be incorrect.

Gambar 2.3 Kategori *Machine Learning*Sumber: Singh et al., 2025

Menurut Singh et al. (2025), berdasarkan strategi pembelajaran, terdapat 4 (empat) kategori ML, yaitu:

1. *Supervised learning*

Algoritma pembelajaran dari data berlabel yang *output*nya diketahui. Tujuannya mempelajari hubungan antara input dan *output* tersebut, sehingga *output* dari data baru dapat diprediksi seakurat mungkin. Contoh penerapannya adalah prediksi harga rumah.

1. *Unsupervised learning*

Algoritma yang mencari pola atau struktur tersembunyi dari data yang tidak berlabel. Contohnya adalah *clustering* yang mengelompokkan data yang berkaitan. Contoh penerapannya adalah segmentasi pembeli dan analisis pasar.

1. *Semi-supervised learning*

Perantara *supervised* dan *unsupervised learning*. Algoritma ini berguna ketika data yang berlabel susah atau mahal untuk dicari. Contoh penerapan metode ini adalah klasifikasi teks dan pencarian rute berdasarkan data GPS.

1. *Reinforcement learning*

Algoritma yang belajar membuat keputusan berdasarkan hasil berinteraksi dengan lingkungannya. Tergantung aktivitas algoritma tersebut, hadiah atau hukuman diberikan. Seiring waktu, algoritma mampu mengidentifikasi aksi optimal untuk memaksimalkan total hadiah. Contoh penerapannya adalah pengendaraan mobil otomatis.

### Deep Learning (DL)

*Deep learning* (DL) adalah cabang ML yang memanfaatkan lapisan jaringan saraf *(neural network)* untuk menangkap pola kompleks dalam data secara efektif (Kamath et al., 2025).

DL digunakan untuk diagnosis medis dengan mengubah input menjadi *output*, misalnya dalam klasifikasi gambar kanker kulit. Selain itu, model DL juga dapat mempelajari gambar untuk membuat gambar baru sejenis, misalnya foto manusia yang tidak nyata.DL memungkinkan model untuk menghasilkan gambar berdasarkan teks input (*prompt*) yang dimasukkan, sehingga konten gambar mencerminkan input tersebut. Model DL yang menghasilkan *output* berupa gambar, video, audio, teks, dan lain-lain dikenal juga sebagai *generative* AI (Bishop & Bishop, 2024).

Menurut Kamath et al. (2025), terdapat beberapa faktor yang memengaruhi desain dan efektivitas model DL, yaitu:

1. Fungsi aktivasi *(activation functions)*

Fungsi aktivasi memperkenalkan non-linieritas ke dalam *neural network* agar dapat mempelajari pola-pola yang kompleks. Contohnya adalah ReLU, sigmoid, dan tanh.

1. Fungsi kerugian *(loss functions)*

Fungsi kerugian mengukur seberapa baik prediksi jaringan sesuai dengan data aktual. Misalnya *mean squared error* dan *cross-entropy*.

1. Metode optimasi *(optimization methods)*

Metode optimasi memperbarui bobot jaringan guna meminimalkan fungsi kerugian. Contoh metode optimasi adalah *stochastic gradient descent* (SGD), Adam, dan RMSprop.

1. Teknik regularisasi *(regularization techniques)*

Teknik regularisasi mencegah *overfitting*, yaitu kondisi saat model terlalu menyesuaikan diri dengan data pelatihan, sehingga berkinerja buruk saat diuji pada data baru. Teknik yang umum digunakan adalah *dropout*, regularisasi L2, dan normalisasi *batch*.

### Arsitektur Transformer

Arsitektur Transformer adalah arsitektur *deep learning* yang umumnya digunakan untuk tugas-tugas NLP seperti penerjemahan mesin, yaitu proses yang melibatkan urutan kata diterjemahkan dari satu bahasa ke bahasa lain.

A diagram of a decoder

AI-generated content may be incorrect.

Gambar 2.4 Arsitektur Transformer  
Sumber: Atkinson-Abutridy, 2025

Arsitektur ini terdiri dari dua komponen, yaitu *encoder* dan *decoder.* *Encoder* mengubah urutan input token menjadi rangkaian vektor *embedding* yang disebut juga sebagai konteks. Sementara itu, *decoder* menggunakan konteks *encoder* untuk secara berulang menghasilkan urutan *output* token (Tunstall et al., 2022).

Proses arsitektur Transformer dimulai saat teks input diubah menjadi token *embedding*. Token *embeddings* memiliki posisi relatif yang disisipkan, yang disebut dengan *positional embeddings*, sebagai cara agar sebuah mekanisme yang disebut dengan mekanisme *attention* tetap mengetahui posisi relatif token. Hal ini memungkinkan *decoder* untuk memberikan jumlah bobot atau “perhatian” yang berbeda untuk masing-masing konteks *encoder*. *Output* dari *encoder* diberikan ke setiap lapisan *decoder*, yang kemudian menghasilkan prediksi berupa token yang paling memungkinkan untuk rangkaian token tersebut. *Output* dari langkah ini kemudian diberikan kembali ke dalam *decoder* untuk menghasilkan token berikutnya, dan seterusnya hingga token *end-of-sequence* (EOS) yang menandakan akhir prediksi dihasilkan (Tunstall et al., 2022).

Melatih model menggunakan arsitektur Transformer lebih cepat daripada model *deep learning* sebelumnya (seperti *recurrent neural networks*) dan berkinerja lebih baik dalam tugas-tugas NLP. Selain itu, keuntungan lainnya adalah kemampuannya untuk berfokus pada bagian yang menjadi perhatian *neural network*, terutama saat memproses atau menerjemahkan kata tertentu. Oleh karena itu, model ini dapat memahami bagaimana informasi ditransmisikan melalui *neural network* (Atkinson‑Abutridy, 2025).

Model berbasis Transformer paling awal seperti GPT-1 dan BERT dapat dianggap sebagai generasi pertama model bahasa dengan parameter yang banyak. Kategori model bahasa berparameter banyak ini disebut juga dengan *large language models* (LLM) (Kamath et al., 2025).

### Large Language Models (LLM)

*Large language models* (LLM) adalah model matematika generatif yang terbuat dari distribusi statistik token (kata, bagian dari kata, karakter individu, dan simbol) dalam *dataset* teks buatan manusia publik yang luas (Shanahan, 2022). LLMmerupakan kategori model bahasa berparameter banyak yang umumnya berbasis Transformer, sebuah arsitektur *deep learning* yang terdiri dari *encoder* dan *decoder.* *Encoder* mengubah urutan input token menjadi rangkaian vektor *embedding* yang disebut juga sebagai konteks. Sementara itu, *decoder* menggunakan konteks *encoder* untuk secara berulang menghasilkan urutan *output* token (Tunstall et al., 2022).

LLM berbasis Transformer seperti GPT-1 dan BERT dianggap sebagai generasi pertama LLM (Kamath et al., 2025). Popularitas LLM meningkat ketika OpenAI mengumumkan ChatGPT, sebuah program *chatting* dengan AI *(chatbot)* yang memiliki kemampuan untuk menjawab pertanyaan, mengakui kesalahannya, menantang premis yang salah, dan menolak permintaan yang tidak sesuai (OpenAI, 2022). Popularitas teknologi ini memicu ledakan investasi di perusahaan-perusahaan AI.

LLM memiliki beberapa keuntungan seperti kemampuan untuk menghasilkan teks seperti manusia, mendukung berbagai bahasa, dan memproses data berskala besar secara efisien. LLM menawarkan konfigurasi yang fleksibel dan respons yang cepat sesuai dengan kebutuhan yang dapat meningkatkan pengalaman pengguna di seluruh domain (Karanikolas et al., 2023).

Namun, LLM juga memiliki keterbatasan. Teknologi ini sering berhalusinasi (Ji et al., 2022) dan rentan menghasilkan konten yang bias, tidak sesuai fakta, atau berbahaya karena kekurangan dalam data pelatihan. Kelemahan ini juga berasal dari keterbatasan dalam ukuran input (Hosseini et al., 2024). Selain itu, LLM juga menggunakan informasi yang sudah ketinggalan zaman, sambil membutuhkan sumber daya komputasi yang besar (Karanikolas et al., 2023). Oleh karena itu, dibutuhkan solusi untuk masalah jawaban yang kurang relevan dan juga kebutuhan sumber daya komputasi yang terus meningkat.

### Arsitektur Mixture-of-Experts (MoE)

Arsitektur Mixture-of-Experts (MoE) merupakan suatu varian dari arsitektur Transformer yang menggunakan beberapa lapisan MoE. Setiap lapisan MoE mengandung beberapa unit paralel yang disebut dengan expert, masing-masing dengan parameternya sendiri. Sebuah pengarah (router) mengarahkan token input ke expert tertentu berdasarkan kemampuan mereka. Pemilihan expert ini ditentukan oleh mekanisme routing seperti top-k gating. Pemilihan ini memungkinkan hanya beberapa expert yang diaktifkan berdasarkan input token. Kemudian, para expert memproses token secara independen (Kamath et al., 2025).

Istilah “Mixture-of-Experts” berawal dari suatu penemuan Jacobs et al. (1991) yang mengimplementasikan suatu prosedur *supervised learning* untuk sistem yang terdiri dari berbagai *neural network* terpisah yang dinamakan *expert*. Masing-masing *neural network* ini mempelajari bagaimana menyelesaikan suatu tugas, yang jika digabung bersama dengan *neural network* lain memberikan sistem tersebut kemampuan untuk menyelesaikan suatu masalah kompleks yang utuh.

A diagram of a router

AI-generated content may be incorrect.

Gambar 2.5 Arsitektur Mixture-of-Experts (MoE)  
Sumber: Kamath et al., 2025

Tidak seperti LLM berbasis Transformer, yang disebut juga sebagai LLM tradisional, yang mengalami peningkatan biaya komputasi yang lebih tinggi seiring dengan peningkatan ukuran (Shen et al., 2023), LLM berbasis MoE hanya mengaktifkan sejumlah *expert* tertentu secara bersamaan. Hal ini memungkinkan ukuran model yang lebih besar tanpa peningkatan kebutuhan komputasi secara proporsional (Kamath et al., 2025).

Tentu saja, arsitektur MoE tidak sempurna. LLM berbasis arsitektur ini mengalami kelemahan seperti hibriditas dan redundansi pengetahuan. Hibriditas pengetahuan terjadi karena arsitektur MoE sering kali menggunakan sejumlah *expert* yang terbatas, sehingga token yang ditugaskan ke *expert* tertentu cenderung mencakup pengetahuan yang beragam. Akibatnya, *expert* yang ditunjuk mengumpulkan jenis pengetahuan yang sangat berbeda, yang sulit untuk digunakan secara bersamaan. Redundansi pengetahuan terjadi karena token yang ditugaskan ke *expert* yang berbeda mungkin memerlukan pengetahuan yang sama. Sehingga, beberapa *expert* dapat berkumpul untuk memperoleh pengetahuan bersama, yang menyebabkan redundansi dalam parameter *expert* tersebut. Masalah-masalah ini menghambat adanya spesialisasi *expert* dalam arsitektur MoE, dan mencegah LLM berbasis MoE untuk mencapai kinerja yang lebih maksimal (Dai et al., 2024).

### DeepSeekMoE

DeepSeekMoE adalah sebuah arsitektur berbasis Mixture-of-Experts (MoE). Arsitektur MoE merupakan varian dari Transformer yang menggunakan beberapa lapisan unit paralel *(expert)* berparameter. Sebuah pengarah (*router*) mengarahkan token input ke *expert* tertentu berdasarkan kemampuan mereka (Kamath et al., 2025).

DeepSeekMoE dirancang untuk mendorong pemilihan *expert* yang lebih baik dari MoE. Arsitektur ini dikembangkan oleh sebuah perusahaan AI bernama DeepSeek yang didirikan pada tahun 2023 oleh Liang Wenfeng (DeepSeek AI, 2025).

Berikut Gambar 2.6, yang menunjukkan LLM berbasis DeepSeekMoE memiliki performa yang lebih tinggi daripada LLM lain yang memiliki parameter yang sebanding. LLM berbasis DeepSeekMoE juga memiliki jumlah parameter aktif yang lebih sedikit, namun memiliki performa yang setara dengan LLaMA2 7B.

A graph with numbers and letters

AI-generated content may be incorrect.

Gambar 2.6 Perbandingan DeepSeekMoE dengan LLM *open source*Sumber: Dai et al., (2024)

DeepSeek menarik perhatian kalangan publik karena telah mengembangkan LLM *open source* yang mampu menyaingi LLM berukuran serupa dengan biaya lebih rendah. LLM mereka, seperti DeepSeek-R1 (DeepSeek-AI et al., 2025) dan DeepSeekV3 (DeepSeek-AI et al., 2024), dibangun dengan menggunakan DeepSeekMoE sebagai fondasinya.

A diagram of a network

AI-generated content may be incorrect.

Gambar 2.7 Perbandingan antara lapisan arsitektur MoE pada subfigur (a) dengan mekanisme yang digunakan DeepSeekMoE pada subfigur (b) dan (c).

Sumber: Dai et al., (2024)

Menurut Dai et al. (2024), DeepSeekMoE menggabungkan dua strategi, yaitu:

1. *Fine-grained expert segmentation*

Sambil mempertahankan jumlah parameter dan kebutuhan daya komputasi, para *expert* disegmentasi menjadi butiran yang lebih halus dengan memisahkan salah satu lapisan tersembunyi dan mengaktifkan lebih banyak *expert* halus (*fine-grained expert)*. Hal ini memungkinkan pengetahuan yang beragam untuk diuraikan lebih halus dan dipelajari lebih tepat oleh *fine-grained expert* yang berbeda.

1. *Shared expert isolation*

*Expert* tertentu diisolasi dan dimanfaatkan sebagai *expert* bersama *(shared expert)* yang selalu diaktifkan. *Shared expert* bertugas untuk menangkap pengetahuan umum di berbagai konteks, sehingga mengatasi redundansi di antara para *expert* yang diarahkan. Hal ini memastikan bahwa setiap *expert* yang dirutekan tetap memiliki spesialisasi.

### Retrieval-Augmented Generation (RAG)

*Retrieval-augmented generation* (RAG) adalah teknik yang memungkinkan sebuah LLM untuk menerima kueri pengguna dan mengembalikan respons dengan merujuk pada basis pengetahuan statis. Basis pengetahuan statis berupa kumpulan dokumen, tabel data, dan sumber informasi lainnya yang telah diubah menjadi representasi vektornya *(vector embeddings)* (Kamath et al., 2025).

RAG pertama kali diperkenalkan oleh Lewis et al. (2020) sebagai solusi atas kelemahan LLM yang sering menghasilkan jawaban yang tidak akurat dan halusinatif.

Dalam prosesnya, RAG menggunakan model bahasa kecil yang disebut dengan model *embedding* yang mengubah kueri pengguna menjadi representasi vektor. Representasi vektor ini dibandingkan dengan vektor dokumen untuk menemukan segmen teks yang serupa. Sejumlah *k* dokumen yang paling mirip diambil. Selanjutnya, kueri asli dan dokumen hasil pencarian diteruskan ke LLM untuk menghasilkan jawaban yang lebih relevan (Kamath et al., 2025).

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 2.8 Diagram konsep dasar cara kerja RAG.  
Sumber: Kamath et al., 2022

Sejak meningkatnya popularitas penggunaan LLM seperti ChatGPT, penelitian terkait RAG pun berkembang pesat. Berbagai studi dan pendekatan baru terus ditemukan untuk meningkatkan efektivitas teknik ini. RAG memiliki beberapa kelemahan utama, seperti:

1. Ketidakakuratan dalam proses pengambilan dokumen karena segmen teks yang paling relevan terhadap kueri pengguna tidak selalu muncul di peringkat atas hasil pencarian. Hal ini disebabkan oleh keterbatasan ukuran model *embedding,* serta fenomena *lost in the middle*, yaitu kecenderungan LLM untuk mengabaikan informasi yang terletak di tengah konteks (Kamath et al., 2025). Beberapa strategi telah dikembang, seperti *re-ranking* yang menggunakan model *embedding* untuk pengambilan awal dokumen, yang kemudian dokumen-dokumen teratas diurutkan kembali berdasarkan relevansinya menggunakan model *embedding* yang lebih besar (Ma et al., 2023).
2. Kesulitan dalam menanggapi pertanyaan *multi-hop*, yaitu pertanyaan yang membutuhkan beberapa langkah penalaran. Solusi yang dikembangkan untuk kelemahan ini adalah *iterative retrieval* yang menjalankan proses pengambilan dokumen dan penghasilan jawaban secara berulang (Shao et al., 2023).
3. RAG dikembangkan untuk mengatasi keterbatasan konteks pada LLM dengan mengambil informasi eksternal, tetapi RAG sendiri menghadapi masalah tersebut. Konteks atau ukuran input yang dapat dimasukkan ke LLM terbatas, sehingga jika hasil RAG dan kueri pengguna terlalu panjang, sebagian informasi yang diteruskan ke LLM terpotong atau terabaikan. Hal ini mengakibatkan respons yang dihasilkan tidak lengkap atau tidak akurat (Hosseini et al., 2024). Solusi yang dikembangkan untuk mencegah masalah ini adalah *context consolidation*, yaitu menggunakan LLM untuk merangkum fakta utama dari dokumen yang diambil, baik secara individual maupun kolektif. Hal ini membuat informasi yang disajikan ke LLM penghasil jawaban menjadi lebih padat, ringkas, dan bermakna (Kamath et al., 2025).

### Self-Reflective Retrieval-Augmented Generation (Self-RAG)

Self-Reflective Retrieval-Augmented Generation (**Self-RAG**) merupakan variasi dari teknik *retrieval-augmented generation* (RAG).RAG adalah teknik yang memungkinkan LLM untuk menerima input pengguna dan mengembalikan respons dengan merujuk pada basis pengetahuan statis seperti kumpulan dokumen, tabel data, dan sumber informasi lainnya yang telah diubah menjadi representasi vektornya *(vector embeddings)* (Kamath et al., 2025).

SELF-RAG dikembangkan untuk meningkatkan kualitas *output* yang dihasilkan LLM dalam hal akurasi faktual tanpa mengorbankan fleksibilitasnya. Tidak seperti RAG yang mengandalkan pengambilan *(retrieval)* dokumen secara satu arah, Self-RAG menggabungkan proses *retrieval* dan *self-reflection.* (Asai et al., 2023).

A screenshot of a computer

AI-generated content may be incorrect.

Gambar 2.9 Diagram perbandingan proses RAG dan Self-RAG

Sumber: Asai et al., (2023)

Dalam pendekatan ini, LLM dilatih untuk secara mandiri mengevaluasi proses penghasilan *output* sendiri terhadap sebuah input. Caranya adalah menghasilkan *output* tugas sekaligus token refleksi (*reflection tokens*). Token ini dibagi menjadi dua jenis, yaitu token *retrieval* yang memicu proses pencarian informasi eksternal jika dianggap dibutuhkan, dan token *critique* yang digunakan untuk mengevaluasi kualitas dan faktualitas dari *output* yang dihasilkan. Setelah *retrieval* dijalankan, Self-RAG memproses beberapa dokumen yang diperoleh secara paralel, menilai relevansinya, lalu menghasilkan jawaban berdasarkan masing-masing konteks. Di tahap akhir, LLM melakukan kritik mandiri terhadap semua *output* dan memilih hasil terbaik berdasarkan kualitas informasi dan konsistensinya dengan fakta. Dengan demikian, proses penalaran dalam LLM jauh lebih adaptif, reflektif, dan akurat dibandingkan pendekatan RAG (Asai et al., 2023).

A black and white text

AI-generated content may be incorrect.

Gambar 2.10 Tabel yang menunjukkan jenis *reflection* tokenyang digunakan dalam Self-RAG. Token IsRel, IsSup, dan IsUse merupakan jenis *critique* token.

Sumber: Asai et al., (2023)

Eksperimen yang dilakukan Asai et al. (2023) menunjukkan bahwa Self-RAG memiliki kinerja yang unggul dibandingkan LLM yang dilatih baik secara *supervised* maupun RAG dalam berbagai *dataset* tolok ukur *(benchmark datasets)* seperti PubHealth, PopQA, ASQA, dan *biography generation*. Selain itu, dengan metrik presisi faktual, Self-RAG yang berukuran 7B terkadang mengungguli versi 13B, karena model yang lebih kecil cenderung menghasilkan jawaban yang lebih terfokus (Asai et al., 2023).

### LangChain

LangChain adalah sebuah *framework* yang dirancang untuk mengembangkan aplikasi yang didukung oleh LLM. *Framework* ini dapat menyederhanakan tahapan dalam siklus pengembangan aplikasi berbasis LLM, mulai dari tahap pengembangan *(development)*, produksi *(production)*, dan penerapan *(deployment)* (LangChain, 2024a).

Pada tahap pengembangan, LangChain menyediakan berbagai komponen *open source* danintegrasi dengan layanan pihak ketiga untuk memudahkan pengembang membangun aplikasi. Pada tahap produksi, LangChain menyediakan LangSmith, sebuah alat untuk menguji, memantau, dan mengevaluasi performa aplikasi, agar optimasi dan iterasi dapat dilakukan. Untuk tahap penerapan, aplikasi yang dibangun dapat diubah menjadi API melalui LangGraph Platform. (LangChain, 2024b).

Berdasarkan dokumentasi LangChain, *framework* ini menyediakan berbagai komponen yang memungkinkan implementasi mekanisme seperti token *retrieval* dan *critique* yang digunakan dalam Self-RAG, tanpa perlu menggunakan sumber daya komputasi yang besar untuk melakukan *fine-tuning.* Komponen LangChain yang relevan dengan penelitian ini adalah *retrieval*, yaitu kategori komponen yang berinteraksi dengan data eksternal. Alat yang tersedia dalam kategori komponen ini adalah *vectorstore* untuk menyimpan *embedding* sumber teks, *retrievers* untuk mengambil dokumen berdasarkan kueri pengguna, dan model *embedding* yang berfungsi untuk membuat representasi vektor dari teks (LangChain, 2024b).

## Penelitian Terdahulu

Berikut tabel 2.1, rangkuman hasil penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan.

Tabel 2.1 Penelitian terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Peneliti | Judul Penelitian | Hasil Penelitian |
| 1. | (Islam et al., 2024) | OPEN-RAG: Enhanced Retrieval-Augmented Reasoning with Open-Source Large Language Models | Hasil penelitian menunjukkan bahwa OPEN-RAG melampaui performa berbagai model LLM dan pendekatan RAG lainnya. OPEN-RAG berhasil mencapai skor yang baik pada beberapa *dataset* pengujian seperti HotpotQA (*exact match* 63.3%) dan PubQA (akurasi 75,9%). |
| Persamaan: Penelitian yang dilakukan Islam et al. (2024) membahas konsep yang sama dengan penelitian ini, yaitu Self-RAG dan arsitektur varian Transformer. | | |
| Perbedaan: Penelitian yang dilakukan Islam et al. (2024) menggunakan arsitektur MoE, sementara penelitian ini menggunakan arsitektur DeepSeekMoE, yang merupakan implementasi lanjutan arsitektur MoE. | | |
| 2. | (Zhou et al., 2024) | Unveiling and Consulting Core Experts in Retrieval-Augmented MoE-based LLMs | Hasil penelitian ini memperkenalkan Expert-RAG yang mampu mencapai performa yang kompetitif dibandingkan dengan RAG tradisional, meskipun memanfaatkan jumlah token yang lebih sedikit. Bahkan, Expert-RAG unggul pada beberapa *dataset* pengujian, seperti RGBqa (akurasi 58,50%), PubHealth (akurasi 58,05%), dan BalanceQA (akurasi 58,75%). |
| Persamaan: Penelitian yang dilakukan Zhou et al. (2024) memiliki dasar yang serupa dengan penelitian ini, yaitu menggabungkan pendekatan RAG dan arsitektur MoE. | | |
| Perbedaan: Penelitian yang dilakukan Zhou et al. (2024) mengeksplorasi mekanisme internal arsitektur MoE terhadap RAG, sementara penelitian ini mengeksplorasi akurasi dan performa implementasi Self-RAG, yang merupakan bentuk lanjutan RAG, pada arsitektur DeepSeekMoE, yang merupakan pengembangan dari MoE. | | |

# METODOLOGI PENELITIAN

## Metode Pengumpulan Data

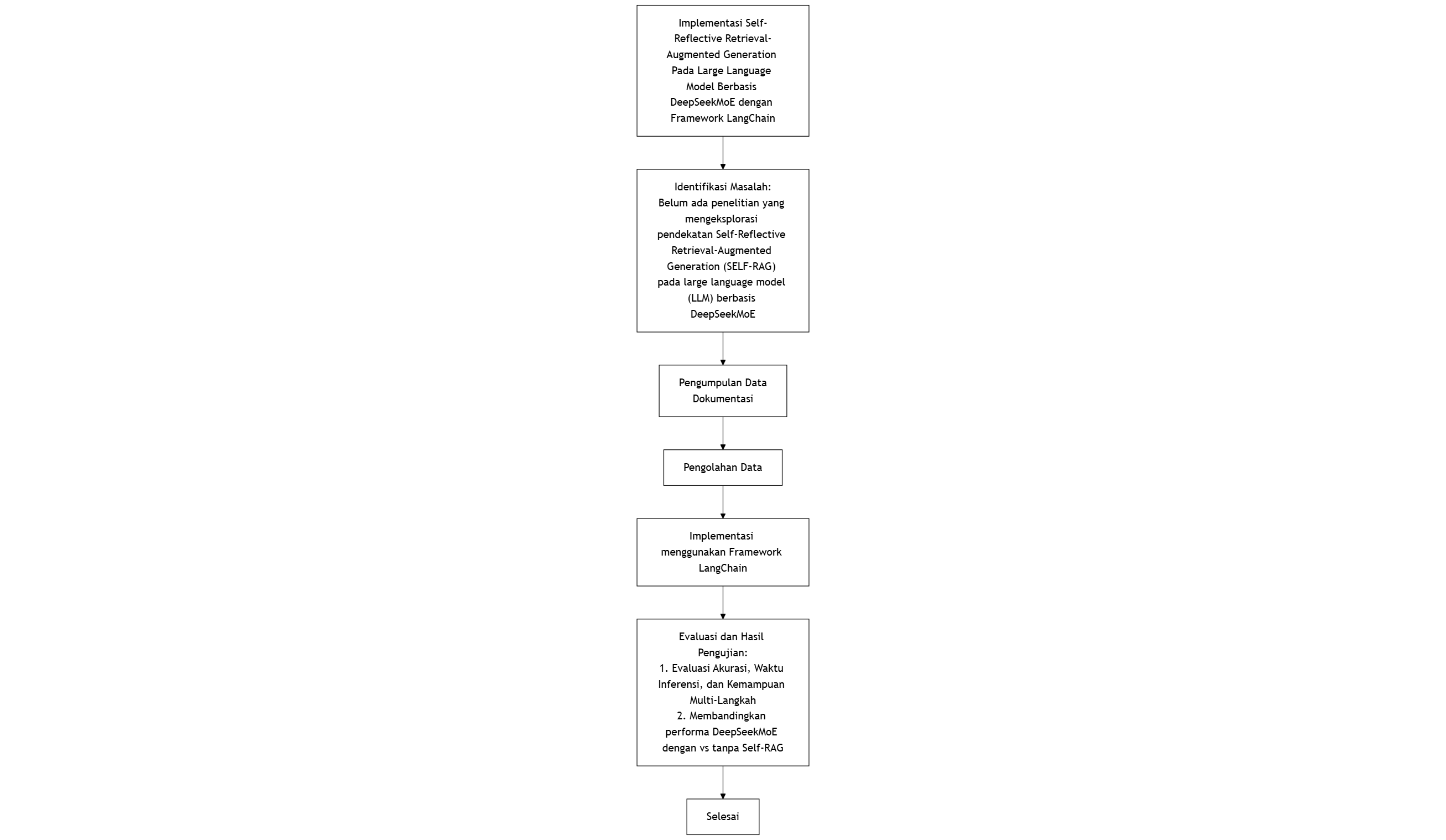
Agar tujuan penelitian tercapai, pengumpulan data diperlukan untuk memperoleh informasi yang dibutuhkan dalam penelitian ini. Metode pengumpulan data mengacu pada garis besar jenis data yang dikumpulkan, sumber data, dan instrumen pengumpulan data (Ferawaty et al., 2021). Metode pengumpulan data yang digunakan adalah metode dokumentasi. Metode dokumentasi adalah metode yang pengambilan data yang diperoleh melalui dokumen-dokumen (Nur Hikmatul Auliya et al., 2020). Metode ini dipilih untuk memperoleh data yang tersedia dari berbagai sumber terpercaya. Terdapat dua jenis data yang dikumpulkan, yaitu:

1. Dokumen-dokumen sumber pengetahuan tekstual, yaitu dokumen berbasis teks yang digunakan dalam proses *retrieval* Self-RAG. Proses pengumpulan dilakukan dengan metode dokumentasi melalui penelurusan *internet* pada sumber data yang digunakan dalam penelitian Self-RAG sebelumnya, yaitu korpus Wikipedia yang telah diproses sebelumnya serta *dataset* pelatihan asli dari Self-RAG.
2. *Dataset* tolok ukur *(benchmark dataset)* untuk mengevaluasi akurasi LLM*.* Proses pengumpulan dilakukan dengan metode dokumentasi melalui penelurusan *internet. Dataset* tolok ukur yang digunakan dalam penelitian ini telah digunakan oleh penelitian sebelumnya. Adapun *dataset* tolok ukuryang digunakan dalam penelitian ini, yaitu TriviaQA-unfiltered, PopQA *long-tailed subset,* dan ARC-Challenge. *Dataset benchmark* diolah sesuai dengan keperluan input dan *output* programLangChain dalam implementasi Self-RAG pada LLM berbasis DeepSeekMoE.

## Kerangka Pikir

Kerangka pikir penelitian adalah pola pikir kegiatan penelitian secara keseluruhan (Ferawaty et al., 2021). Kerangka pikir ini disusun untuk menggambarkan alur pemikiran dan strategi penyelesaian dalam mengatasi keterbatasan LLM, khususnya dalam hal akurasi jawaban dan kemampuan menyelesaikan tugas yang membutuhkan penalaran multi-langkah. Permasalahan utama adalah implementasi Self-RAG, yang terbukti efektif dalam meningkatkan akurasi respons, masih terbatas pada LLM tradisional dan LLM berbasis MoE. Arsitektur baru seperti DeepSeekMoE berpotensi untuk memiliki akurasi dan efisiensi yang lebih baik, tetapi belum ada kajian yang mengevaluasi sinergi antara Self-RAG dan DeepSeekMoE.

Berdasarkan latar belakang dan rumusan masalah yang sudah dijelaskan pada bab sebelumnya, berikut gambar 3.1, kerangka berpikir yang digunakan peneliti pada penelitian ini:



Gambar 3.1 Kerangka Berpikir Penelitian

Diagram tersebut terdiri dari beberapa bagian utama, yaitu:

1. Identifikasi Masalah

Tahap ini menjelaskan masalah yang dihadapi, yaitu belum adanya penelitian yang mengeksplorasi pendekatan ini pada LLM berbasis DeepSeekMoE.

1. Pengumpulan Data

Pada tahap ini, dikumpulkan data berupa dokumen-dokumen sumber pengetahuan tekstual yang digunakan dalam proses *retrieval* Self-RAG dan juga *dataset* tolok ukuruntuk mengevaluasi akurasi LLM.

1. Pengolahan Data

Pada tahap ini, data diolah agar dapat digunakan dalam programLangChain.

1. Implementasi menggunakan *Framework* LangChain

Pada tahap ini, *framework* LangChain digunakan untuk mendesain program Self-RAG pada LLM berbasis DeepSeekMoE.

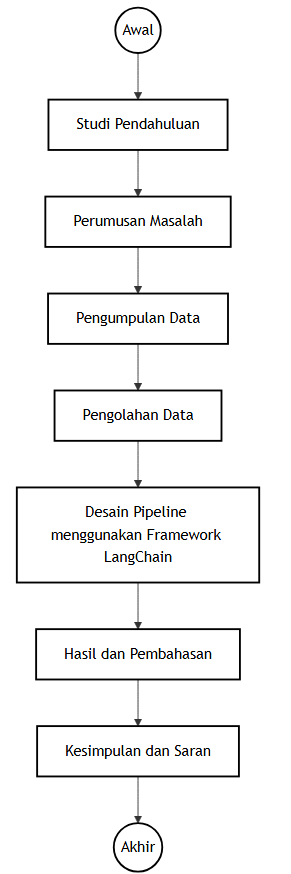
1. Evaluasi dan Hasil Pengujian

Impementasi Self-RAG pada LLM berbasis DeepSeekMoE menggunakan *framework* LangChain diuji menggunakan pertanyaan-pertanyaan yang telah ditetapkan dalam *dataset* tolok ukur. Jawaban yang dihasilkan oleh LLM kemudian dibandingkan dengan jawaban referensi yang ada dalam *dataset* tersebut untuk mengukur tingkat akurasinya.

Sebagai pembanding, LLM berbasis DeepSeekMoE tanpa Self-RAG juga diuji pada *dataset* tolok ukurdengan metrik evaluasi yang sama yaitu akurasi, guna mengetahui sejauh mana Self-RAG memberikan pengaruh terhadap kualitas jawaban yang dihasilkan.

## Tahapan Penelitian

Berikut gambar 3.2, langkah-langkah yang dijalankan dalam penelitian ini.



Gambar 3.2 Tahapan Penelitian

Berikut adalah uraian penjelasan dari tahapan-tahapan pada gambar di atas:

1. Studi Pendahuluan

Peneliti melakukan studi literatur dan analisis penelitian sebelumnya yang berkaitan dengan Self-RAG dan DeepSeekMoE. Studi ini dilakukan untuk mendapatkan pemahaman awal tentang metode yang digunakan.

1. Perumusan Masalah

Berdasarkan tinjauan pustaka, masalah utama yang dapat dirumuskan adalah belum adanya penelitian yang mengeksplorasi pendekatan Self-Reflective Retrieval-Augmented Generation (Self-RAG) pada large language model (LLM) berbasis DeepSeekMoE.

1. Pengumpulan Data

Data yang dikumpulkan berupa dokumen-dokumen sumber pengetahuan tekstual, yaitu dokumen berbasis teks yang digunakan dalam proses *retrieval* Self-RAG, seperti korpus Wikipedia yang telah diproses sebelumnya serta *dataset* pelatihan.

Selain itu, dikumpulkan juga *dataset* tolok ukur untuk mengevaluasi akurasi LLM. Dataset benchmark yang digunakan adalah TriviaQA-unfiltered, PopQA *long-tailed Subset*, dan ARC-Challenge.

1. Pengolahan Data

Pada tahap ini, data yang telah dikumpulkan diolah agar dapat digunakan dalam programLangChain yang telah dibuat.

1. Desain Programmenggunakan LangChain

Setelah tahap pengolahan data, Self-RAG diimplementasikan pada LLM berbasis DeepSeekMoE menggunakan *framework* LangChain dengan tahapan sebagai berikut:

1. Membangun komponen *retrieval* yang digunakan LLM untuk mencari dokumen dengan *embedding* yang mirip dengan *embedding* input.
2. Integrasi DeepSeekMoE pada programagar dapat diinjeksikan dokumen hasil *retrieval* dan kueriinput.
3. Implementasi Self-RAG dengan memungkinkan *output* LLM untuk sekaligus mengeluarkan token *reflections*.
4. Hasil dan Pembahasan

Menguji akurasi model terhadap *dataset* tolok ukur. LLM berbasis DeepSeekMoE tanpa Self-RAG juga diuji akurasinya pada *dataset* tolok ukuruntuk perbandingan.

1. Kesimpulan dan Saran

Berdasarkan hasil dan pembahasan, peneliti memberikan kesimpulan mengenai kinerja akurasi jawaban yang dihasilkan LLM berbasis DeepSeekMoE saat menggunakan Self-RAG, serta perbandingan akurasinya dengan LLM berbasis DeepSeekMoE yang tidak menggunakan Self-RAG. Peneliti juga memberikan saran untuk perbaikan model di masa depan.

## Metode Penyelesaian

Berikut gambar 3.3, metode penyelesaian yang digunakan dalam penelitian ini.

A diagram of a company

AI-generated content may be incorrect.

Gambar 3.3 Metode Penyelesaian

Langkah-langkah untuk mengimplementasi Self-RAG pada LLM berbasis DeepSeekMoE menggunakanLangChain adalah sebagai berikut:

1. Implementasi *Retrieval* pada Program LangChain

Modul pengambilan informasi atau *retrieval* diintegrasikan pada program LangChain. Komponen ini memungkinkan LLM untuk mengakses sumber data eksternal yang relevan dengan pertanyaan yang diajukan, sehingga konteks tambahan dapat digunakan untuk meningkatkan kualitas jawaban.

1. Implementasi Token *Reflection*

Token *reflection* memungkinkan LLM untuk melakukan penilaian internal atas jawaban yang dihasilkan. Pada langkah ini, program dikonfigurasi untuk menghasilkan dan menggunakan token refleksi saat menjalankan proses *retrieval* dan penghasilan jawaban. Token ini mengarahkan model untuk memeriksa relevansi dokumen yang diambil dan menilai kualitas jawaban secara mandiri.

1. Menghubungkan Program LangChain ke LLM berbasis DeepSeekMoE

Setelah komponen pengambilan informasi dan token *reflection* terpasang, langkah selanjutnya adalah menghubungkan program LangChainke LLM berbasis DeepSeekMoE. Integrasi ini memastikan bahwa LLM tersebut dapat menerima *input* dari program, melakukan inferensi, serta menghasilkan jawaban berdasarkan data dan refleksi yang tersedia.

1. Evaluasi menggunakan *Dataset* Benchmark

Pada langkah ini, evaluasi dimulai dengan menggunakan beberapa *benchmark dataset,* yaitu TriviaQA-unfiltered, PopQA *long-tailed subset*, dan ARC-Challenge. Program berupa implementasi Self-RAG dieksekusi dan dikirim pertanyaan-pertanyaan dalam *dataset* tolok ukurtersebut sebagai input, dan jawaban yang dihasilkan dicatat.

1. Membandingkan Akurasi Jawaban LLM berbasis DeepSeekMoE dengan Self-RAG terhadap Jawaban Referensi.

Pada langkah ini, jawaban yang dihasilkan LLM berbasis DeepSeekMoE dengan Self-RAG dibandingkan dengan jawaban referensi yang sudah ditetapkan dalam *dataset* tolok ukur. Metrik yang digunakan adalah akurasi. Evaluasi dilakukan untuk mengetahui seberapa sering model menghasilkan jawaban yang benar dan relevan.

1. Membandingkan dengan LLM berbasis DeepSeekMoE tanpa Self-RAG.

Pada langkah ini, analisis komparatif antara LLM berbasis DeepSeekMoE dengan Self-RAG dan tanpa Self-RAG dilakukan. Perbandingan ini melibatkan pengukuran metrik performa yang sama pada kedua model, yaitu akurasi pada *dataset* tolok ukur.

## Metode Pengukuran Akurasi

Untuk mengukur akurasi dari LLM berbasis DeepSeekMoE dengan Self-RAG, digunakan 3 (tiga) *dataset* tolok ukur *(benchmark dataset),* yaitu:

1. TriviaQA-unfiltered

*Dataset open-domain question-answering* dengan 11.313 pertanyaan yang menguji kemampuan LLM dalam menjawab pertanyaan faktual. Metrik yang digunakan adalah akurasi. Mengacu pada penelitian Self-RAG sebelumnya, suatu jawaban dinyatakan benar apabila terdapat kecocokan *subset* antara jawaban yang dihasilkan dan jawaban referensi.

1. PopQA long-tailed subset

*Dataset open-domain question-answering* dengan 1.399 pertanyaan mengenai entitas langka yang memiliki kurang dari 100 tampilan halaman Wikipedia per bulan. *Dataset* ini digunakan untuk menguji kemampuan LLM dalam menjawab pertanyaan yang memerlukan pengetahuan langka. Metrik yang digunakan adalah akurasi. Mengacu pada penelitian Self-RAG sebelumnya, suatu jawaban dinyatakan benar apabila terdapat kecocokan *subset* antara jawaban yang dihasilkan dan jawaban referensi.

1. ARC-Challenge

*Dataset benchmark* berbentuk pilihan ganda yang diambil dari soal ujian ilmu pengetahuan tingkat sekolah menengah. *Dataset ini* digunakan untuk menguji kemampuan penalaran logis dan pemahaman konsep ilmiah LLM. Metrik yang digunakan adalah akurasi.

Akurasi dinyatakan sebagai rasio jawaban yang benar dibandingkan dengan jumlah total jawaban yang dibuat (Vasques, 2024). Rumus yang digunakan yaitu rumus akurasi pada umumnya, yaitu:

dengan keterangan sebagai berikut:

* 1. Jawaban benar = Jumlah jawaban LLM yang sama dengan jawaban referensi.
  2. Jawaban total = Jumlah total jawaban yang dihasilkan oleh LLM.

# DAFTAR PUSTAKA

Ahmed, A. A. A., Agarwal, S., Kurniawan, Im. G. A., Anantadjaya, S. P. D., & Krishnan, C. (2022). Business boosting through sentiment analysis using Artificial Intelligence approach. *International Journal of System Assurance Engineering and Management*, *13*, 699–709. https://doi.org/10.1007/s13198-021-01594-x

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. http://arxiv.org/abs/2310.11511

Atkinson‑Abutridy, J. (2025). *Large Language Models; Concepts, Techniques and Applications* (1st ed.). The CRC Press, LLC.

Bishop, C. M., & Bishop, H. (2024). *Deep Learning: Foundations and Concepts*. Springer.

Chavan, T., & Patil, S. (2024). NAMED ENTITY RECOGNITION (NER) FOR NEWS ARTICLES. *International Journal of Artificial Intelligence Research and Development (IJAIRD)*, *2*(1), 103–112.

Clarido, C. B., Tangarorang, R. C., Soberano, K. T., & Condes, K. A. (2023). A Survey of Natural Language Processing Applications in Healthcare System. In *International Journal of Research in Engineering and Science (IJRES) ISSN* (Vol. 11). www.ijres.org

Dai, D., Deng, C., Zhao, C., Xu, R. X., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., Xie, Z., Li, Y. K., Huang, P., Luo, F., Ruan, C., Sui, Z., & Liang, W. (2024). *DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models*. http://arxiv.org/abs/2401.06066

DeepSeek AI. (2025, April 3). *What is DeepSeek AI? Unveiling China’s Groundbreaking Open-Source Revolution in Artificial Intelligencehat-is-deepseek-ai*. https://deepseek.ai/what-is-deepseek-ai

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., … Zhang, Z. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. http://arxiv.org/abs/2501.12948

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., … Pan, Z. (2024). *DeepSeek-V3 Technical Report*. http://arxiv.org/abs/2412.19437

Ferawaty, Damanik, R. R., Robin, & Pratama, Y. A. (2021). *PEDOMAN PENULISAN TUGAS AKHIR & TATA TERTIB PELAKSANAAN SIDANG AKHIR* (2nd ed.). Universitas Pelita Harapan Kampus Medan.

Grand View Research. (2024). *Artificial Intelligence Market Size, Share & Trends Analysis Report By Solution, By Technology (Deep Learning, Machine Learning, NLP, Machine Vision, Generative AI), By Function, By End-Use, By Region, And Segment Forecasts, 2025 - 2030*.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. de Las, Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. van den, Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). *Training Compute-Optimal Large Language Models*. http://arxiv.org/abs/2203.15556

Hosseini, P., Castro, I., Ghinassi, I., & Purver, M. (2024). *Efficient Solutions For An Intriguing Failure of LLMs: Long Context Window Does Not Mean LLMs Can Analyze Long Sequences Flawlessly*. http://arxiv.org/abs/2408.01866

Islam, S. Bin, Rahman, A., Hossain, T., Hoque, E., Joty, S., & Parvez, R. (2024). *OPEN-RAG: Enhanced Retrieval-Augmented Reasoning with Open-Source Large Language Models*. 14231–14244. https://openragmoe.github.io/

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, *3*(1), 79–87. https://doi.org/10.1162/neco.1991.3.1.79

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Chan, H. S., Madotto, A., & Fung, P. (2022). *Survey of Hallucination in Natural Language Generation*. https://doi.org/10.1145/3571730

Kamath, U., Keenan, K., Somers, G., & Sorenson, S. (2025). *Large Language Models: A Deep Dive Bridging Theory and Practice*. Springer Nature Switzerland AG.

Karanikolas, N., Manga, E., Samaridi, N., Tousidou, E., & Vassilakopoulos, M. (2023). Large Language Models versus Natural Language Understanding and Generation. *ACM International Conference Proceeding Series*, 278–290. https://doi.org/10.1145/3635059.3635104

Khan, A. (2024). *Artificial Intelligence: A Guide for Everyone* (1st ed.). Springer Cham.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2022). *Natural Language Processing: State of The Art, Current Trends and Challenges*. *82*, 3713–3744.

LangChain. (2024a). *Components | LangChain*. https://python.langchain.com/docs/integrations/components/

LangChain. (2024b). *Introduction | LangChain*. https://python.langchain.com/docs/introduction/

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. http://arxiv.org/abs/2005.11401

Ma, Y., Cao, Y., Hong, Y., & Sun, A. (2023). *Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!* https://doi.org/10.18653/v1/2023.findings-emnlp.710

Nur Hikmatul Auliya, Ms., Helmina Andriani, G., Roushandy Asri Fardani, Ms., Jumari Ustiawaty, Mp., Evi Fatmi Utami, Ms., Dhika Juliana Sukmana, A., Rahmatul Istiqomah, R., Oleh, D., Pustaka Ilmu Editor, C., & Abadi, H. (2020). *METODE PENELITIAN KUALITATIF & KUANTITATIF*. CV. Pustaka Ilmu.

OpenAI. (2022, November 22). *Introducing ChatGPT*. https://openai.com/index/chatgpt/

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). *GPT-4 Technical Report*. http://arxiv.org/abs/2303.08774

Shanahan, M. (2022). *Talking About Large Language Models*. http://arxiv.org/abs/2212.03551

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., & Chen, W. (2023). *Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy*. http://arxiv.org/abs/2305.15294

Shen, S., Hou, L., Zhou, Y., Du, N., Longpre, S., Wei, J., Chung, H. W., Zoph, B., Fedus, W., Chen, X., Vu, T., Wu, Y., Chen, W., Webson, A., Li, Y., Zhao, V., Yu, H., Keutzer, K., Darrell, T., & Zhou, D. (2023). *Mixture-of-Experts Meets Instruction Tuning:A Winning Combination for Large Language Models*. http://arxiv.org/abs/2305.14705

Singh, K., Muzumdar, P., & Kurian, G. (2025). *The Essentials of Machine Learning: Theory to Applications* (1st ed.). Schmidt & Bailey. https://doi.org/10.5281/zenodo.14752915

Thareja, R. (2024). *Artificial Intelligence: Beyond Classical AI*. Pearson India.

Tunstall, L., Von Werra, L., & Wolf, T. (2022). *Natural Language Processing with Transformers Building Language Applications with Hugging Face* (1st ed.). ’Reilly Media, Inc.

Vasques, X. (2024). *Machine Learning Theory and Applications* (1st ed.). Wiley.

Wang, S., Chen, Z., Li, B., He, K., Zhang, M., & Wang, J. (2024). *Scaling Laws Across Model Architectures: A Comparative Analysis of Dense and MoE Models in Large Language Models*. 5583–5595.

Zhou, X., Nie, P., Guo, Y., Wei, H., Zhang, Z., Minervini, P., Ma, R., Gui, T., Zhang, Q., & Huang, X. (2024). *Unveiling and Consulting Core Experts in Retrieval-Augmented MoE-based LLMs*. http://arxiv.org/abs/2410.15438