BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini, tentunya dilakukan pencarian sumber-sumber melalui pembacaan naskah-naskah penelitian terdahulu. Penelitian mengenai chatbot, LangChain, dan LLM Gemini telah banyak dilakukan di kalangan akademisi. Melalui pencarian sumber-sumber yang ada sebelumnya ditemukan beberapa penelitian berikut :

Pertama yaitu penelitian yang dilakukan oleh G. K. Santhosh Ram dan V. Muthumanikandan (2024), dengan topik di mana pengembangan *chatbot* percakapan bernama Visistant digunakan untuk menghasilkan visualisasi data interaktif dari bahasa alami. Dimana metode analisis data tradisional seringkali kurang efektif dalam menyampaikan informasi yang kompleks, sehingga dibutuhkan cara untuk mempermudah akses analisis data bagi non-ahli. Perancangan sistem dilakukan dengan menggunakan *prompt engineering* untuk memandu model bahasa besar (LLM) Gemini dari Google dalam menghasilkan kode visualisasi, serta memanfaatkan LangChain untuk mengelola memori percakapan.

Kedua yaitu penelitian yang dilakukan oleh Mamta Mittal, Gopi Battineni, Dharmendra Singh, Thakursingh Nagarwal, dan Prabhakar Yadav (2021), dengan topik di mana perancangan *chatbot* cerdas berbasis web digunakan untuk mengelola permintaan pengguna dan menyediakan akses cepat ke informasi rumah sakit.

Dimana data rumah sakit yang tersedia secara daring seringkali tidak transparan dan dapat diandalkan, sehingga dibutuhkan sistem yang dapat memberikan respons instan selama 24/7 untuk mengurangi antrean di meja bantuan rumah sakit. Perancangan sistem dilakukan dengan mengintegrasikan beberapa pendekatan *machine learning* seperti algoritma *gradient descent* (GD) dan *Natural Language Processing* (NLP) untuk memproses input pengguna, serta menggunakan *feed-forward neural network* (FNN) untuk menghasilkan respons yang akurat dari data yang tersimpan dalam format JSON.

Ketiga yaitu website dokumentasi resmi dari LangChain (2025). Dalam dokumentasi ini, pengembangan aplikasi yang memanfaatkan LLM diidentifikasi sebagai sebuah tantangan teknis yang kompleks. Tantangan ini muncul karena aplikasi semacam itu harus mampu mengorkestrasi berbagai komponen secara efisien, mulai dari manajemen *prompt*, interaksi dengan API model, pengambilan data eksternal, hingga pembuatan alur kerja yang logis.

Keempat yaitu penelitian yang dilakukan oleh Mochammad Fariz Syah Lazuardy dan Dyah Anggraini (2022), dengan topik dimana arsitektur *front-end web* modern dibahas melalui perbandingan penggunaan *library* React.js dan *framework* Next.js. Dimana pengembangan aplikasi web Sistem Informasi Aparatur Sipil Negara (SIASN) menjadi sebuah tantangan karena harus dapat dioptimalkan dari berbagai aspek seperti antarmuka, pengalaman pengguna, dan performa untuk seluruh ASN di Indonesia. Perancangan sistem dilakukan melalui studi kasus pada pengembangan aplikasi SIASN untuk menganalisis kelebihan dan kekurangan antara React.js yang mendukung *Client-Side Rendering* dan Next.js yang

dibutuhkan untuk menerapkan *Server-Side Rendering*.

Kelima yaitu penelitian yang dilakukan oleh Gerald Santoso, Johan Setiawan, dan Agus Sulaiman (2023), dengan topik di mana perancangan *chatbot* inovatif berbasis OpenAI API digunakan untuk meningkatkan interaksi pengguna di situs web *Journal of Business, Management, and Social Studies* (JBMS). Dimana pengguna sering mengalami kesulitan dalam menemukan informasi spesifik di dalam situs web jurnal, dan banyak situs jurnal terakreditasi di Indonesia belum memiliki sistem layanan seperti *chatbot*. Perancangan sistem dilakukan dengan menggunakan metodologi prototipe terstruktur yang melibatkan partisipasi pengguna dalam evaluasi, serta memanfaatkan teknologi seperti *Artificial Intelligence* (AI) dan *Natural Language Processing* (NLP).

Terakhir yaitu penelitian yang dilakukan oleh Jaemin Jeong, Daeyoung Gil, Daeho Kim, dan Jaewook Jeong (2024), dengan topik dimana evaluasi keuntungan konstruksi *off-site* dilakukan untuk mengkonsolidasikan penelitian saat ini dan memetakan arah masa depan. Dimana proses tinjauan pustaka secara tradisional sangat memakan waktu dan tenaga, sehingga dibutuhkan solusi yang lebih efisien. Perancangan sistem dilakukan dengan memanfaatkan *Large Language Model* (LLM) yang diintegrasikan dengan kerangka kerja LangChain untuk meringkas 47 makalah penelitian secara cepat dan efisien, serta melakukan analisis jaringan kata kunci untuk mengidentifikasi tren penelitian.

Tabel 2. 1 Tinjau Pustaka

No	Penulis	Objek	Metode	Hasil
1	(Muthumanikandan dkk., 2024)	Gemini LLM, LangChain, Chatbot	Kuantitatif	LLM seperti Gemini Pro menawarkan solusi lengkap, mulai dari pemahaman

				bahasa alami hingga pembuatan kode, yang mengurangi biaya pengembangan dan meningkatkan akurasi dibandingkan dengan pendekatan berbasis aturan atau jaringan saraf yang lebih tua.
2	(Mittal dkk., 2021)	Chatbot, Machine Learning, *Natural Language* Processing (NLP), Web Technologies	SDLC (System Development Life Cycle), kualitatif	*Chatbot* yang diusulkan dapat menjadi solusi untuk mengekstrak data dari rumah sakit lokal dan berfungsi sebagai saluran komunikasi yang baik bagi pengguna dan staf rumah sakit, serta membantu mengurangi kerumunan
3	(LangChain., 2025)	LangChain, RAG, LLM	Studi Dokumentasi, Analisis Arsitektur	Menyajikan sebuah *framework* (LangChain) yang menyederhanakan dan menstandarisasi proses pengembangan aplikasi berbasis LLM. Memberikan arsitektur modular yang memungkinkan pengembang membangun alur kerja yang kompleks.
4	(Fariz dkk., 2022)	React.js, Next.js, Arsitektur frontend web	Kualitatif	Memberikan gambaran tentang langkah-langkah yang dilakukan pengembang saat membangun aplikasi dari sisi *front-end*. Menyajikan poin-poin kelebihan dan kekurangan dari*ibrary* React.js dan *framework* Next.js sebagai teknologi utama dalam membangun antarmuka pengguna aplikasi.
5	(Santoso dkk., 2023)	Chatbot, OpenAI API, User interaction,	SDLC (System	Pengujian Penerimaan

		Website journal	Development Life Cycle), kuantitatif	Pengguna (UAT) memperoleh skor rata-rata 4,14, yang menandakan kepuasan pengguna yang tinggi. Integrasi OpenAI API berhasil meningkatkan ekstraksi informasi dari artikel jurnal dan memberikan rekomendasi artikel yang dipersonalisasi.
6	(Jeong dkk., 2024)	LangChain, *Large Language Model* (LLM)	Kualitatif	LangChain terbukti dapat meringkas penelitian dengan cepat dan efektif, menjadikannya alat yang berharga untuk tinjauan pustaka. Penelitian saat ini menunjukkan keunggulan konstruksi *off-site* melalui teknologi mutakhir, namun kekurangan data masih menjadi tantangan

2.2 Dasar Teori

2.2.1 Large Language Model (LLM)

(Jeong dkk., 2024) menyatakan *Large Language Model* (LLM) adalah model Kecerdasan Buatan (AI) generatif yang dilatih menggunakan data teks dalam jumlah yang sangat besar. LLM semakin banyak digunakan untuk tugas-tugas seperti peringkasan dokumen untuk membuat proses yang padat karya, seperti tinjauan pustaka, menjadi lebih efisien. Model-model ini dapat dilatih dengan dokumen spesifik seperti makalah penelitian untuk mengekstrak jawaban yang diinginkan.

(Muthumanikandan & Ram, 2024) menyatakan Kemampuan LLM dapat

disetarakan dengan performa manusia dalam beberapa bidang tertentu. Mereka menawarkan solusi lengkap yang mencakup interpretasi bahasa hingga pembuatan kode, yang dapat mengurangi biaya pengembangan dan meningkatkan akurasi. Kemampuan ini berasal dari basis pengetahuan luas yang terakumulasi selama proses pelatihannya. Namun, LLM memiliki sifat non-deterministik, yang berarti mereka dapat menghasilkan output yang berbeda meskipun diberikan p*rompt* atau masukan yang identik. Kinerjanya pun bervariasi; model yang berbeda memiliki tingkat akurasi dan biaya yang berbeda pula.

2.2.2 Gemini API

(Muthumanikandan & Ram, 2024) menyatakan bahwa Gemini adalah model bahasa besar pra-terlatih dari Google. Gemini disajikan sebagai alternatif dari model OpenAI seperti GPT-3, di mana beberapa *benchmark* menunjukkan bahwa Gemini Pro dapat melampaui GPT-3.5 dalam hal kinerja pemrosesan. Hal ini salah satunya dimungkinkan karena Gemini dilatih pada sumber data yang lebih luas, yang berpotensi memberikannya kemampuan penalaran yang lebih kuat.

2.2.3 LangChain

(Jeong dkk., 2024) menyatakan bahwa LangChain adalah sebuah kerangka kerja (*framework*) *open-source* yang dirancang untuk memfasilitasi pengembangan aplikasi yang didukung oleh *Large Language Models* (LLM). Konsep utama dari LangChain adalah 'merangkai' (*chaining*), yaitu menghubungkan LLM dengan berbagai sumber data eksternal untuk memungkinkan fungsionalitas seperti peringkasan, terjemahan, serta layanan tanya-jawab (Q&A). Kerangka kerja ini

memiliki berbagai modul untuk memproses data, mulai dari pengumpulan dan ekstraksi, memecah teks menjadi bagian-bagian yang dapat dikelola (*chunks*), hingga mengubahnya menjadi representasi vektor (*embedding*) dan menyimpannya dalam *vector database* untuk pencarian yang efisien. Salah satu penerapan spesifiknya adalah untuk memberikan 'memori' pada agen percakapan atau chatbot. Sebagai contoh (Muthumanikandan & Ram, 2024) menyatakan bahwa modul *ConversationBufferWindowMemory* digunakan untuk menyimpan dan mengelola riwayat percakapan, yang memungkinkan *chatbot* mengingat beberapa interaksi terakhir untuk menjaga kesinambungan dan konteks dalam percakapan.

2.2.4 React JS

(Fariz dkk., 2022) menyatakan React.js, atau dikenal juga sebagai React, adalah sebuah *library front-end* berbasis JavaScript yang digunakan untuk membangun antarmuka pengguna (*userinterface* atau UI) atau komponen UI. *Library* ini dikelola oleh Facebook bersama dengan komunitas pengembang dan perusahaan. Paradigma utama yang digunakan React adalah komponen, di mana UI dipecah menjadi beberapa bagian kecil yang dapat digunakan kembali (*reusable*). Komponen-komponen ini mirip dengan fungsi dalam JavaScript; mereka menerima input yang disebut "*props*" dan mengembalikan elemen React yang mendeskripsikan apa yang harus ditampilkan di layar. *Virtual DOM* oleh React mengoptimalkan proses rendering, menghasilkan *interface* yang sangat responsif. Suatu keharusan untuk aplikasi percakapan yang dinamis di mana pesan terus ditambahkan ke layar.

2.2.5 Chatbot

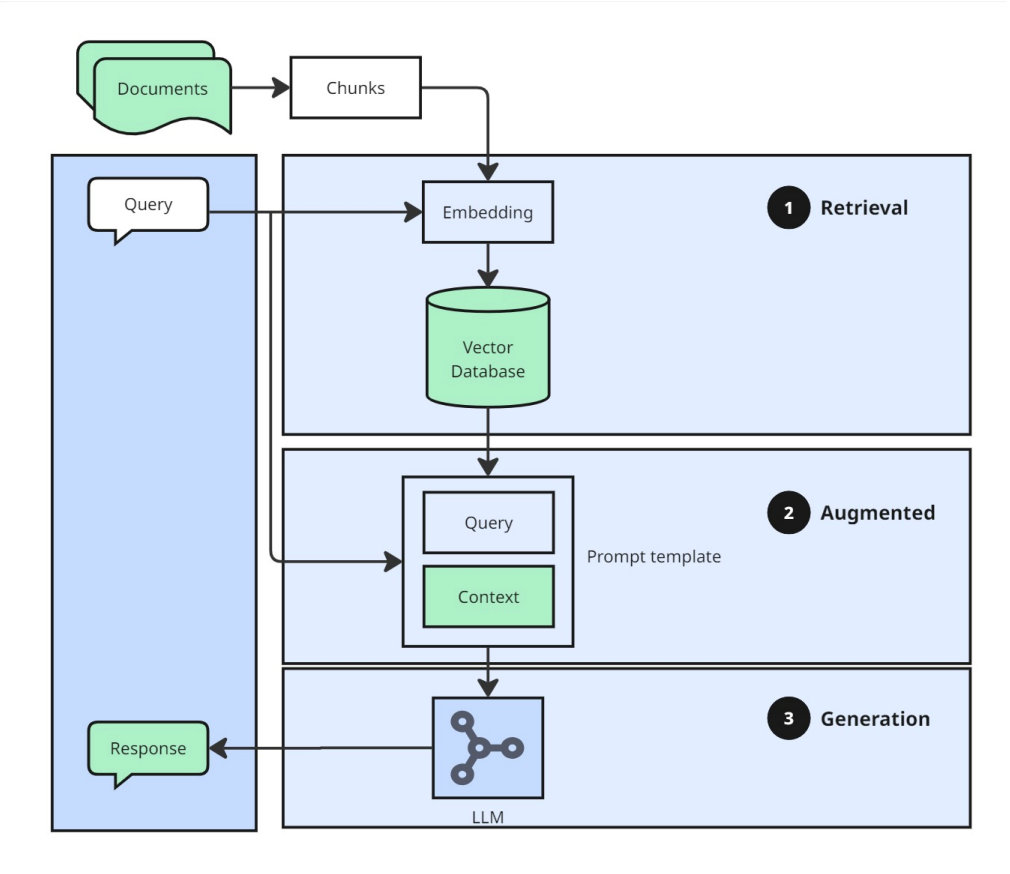
(Santoso dkk., 2023) menyatakan bahwa *Chatbot* adalah sebuah perangkat lunak atau program komputer yang dirancang untuk dapat berkomunikasi dan melakukan interaksi interaktif dengan manusia melalui teks, suara, dan elemen visual. (Mittal dkk., 2021) juga menyatakan *Chatbot* digambarkan sebagai agen percakapan yang dirancang sebagai sebuah antarmuka pengguna. Teknologi ini terintegrasi dengan teknik Kecerdasan Buatan (AI) seperti *Machine Learning* (ML) dan *Natural Language Processing* (NLP). ML memungkinkan *chatbot* untuk belajar dan menganalisis berbagai bahasa, sementara NLP membantunya memahami bahasa manusia dan memberikan respons yang sesuai.

2.2.6 Retrieval-Augmented Generation (RAG)

(Jeong dkk., 2024) menyatakan dan tertulis dalam dokumentasi (LangChain, 2025) bahwa *Retrieval-Augmented Generation* (RAG) adalah sebuah arsitektur yang dirancang untuk menghubungkan *Large Language Models* (LLM) dengan sumber data eksternal, sehingga memungkinkan LLM untuk menghasilkan jawaban yang lebih akurat dan relevan dengan memanfaatkan informasi dari basis pengetahuan yang spesifik. Alih-alih hanya mengandalkan data pre-training yang bersifat statis, RAG memungkinkan LLM untuk mengakses dan menggunakan informasi terkini dari dokumen atau database yang disediakan. Mengandalkan LLM saja berisiko menghasilkan jawaban yang tidak akurat. RAG mengatasi masalah ini dengan mengambil informasi yang relevan dari basis pengetahuan dan menyediakannya sebagai konteks bagi LLM saat menjawab pertanyaan. Hal ini memberikan jawaban LLM pada fakta yang ada, tanpa keluar dari konteks. Berikut

tahapan RAG.

- Pengambilan (*Retrieval*): Pertanyaan pengguna diubah menjadi vektor. LangChain menggunakan *Retriever* untuk mencari *chunks* dokumen yang paling relevan secara semantik dari *vector store* berdasarkan vektor pertanyaan tersebut.
- Augmentasi (*Augment*): *Chunks* dokumen yang berhasil diambil digabungkan dengan pertanyaan asli pengguna ke dalam sebuah *prompt template*. *Prompt* ini secara eksplisit menginstruksikan Gemini untuk menjawab pertanyaan hanya berdasarkan konteks yang disediakan.
- Generasi (*Generate*): *Prompt* ini dikirim ke model gemini-2.5-flash melalui API. Gemini kemudian akan memproses informasi tersebut dan menghasilkan jawaban yang kontekstual.



Gambar 2. 1 Diagram Alur Retrieval Augmented Generation (RAG)

2.2.7 Asrama Mahasiswa Kabupaten Kotabaru Sa-ijaan

Asrama Mahasiswa Kabupaten Kotabaru Sa-ijaan adalah asrama daerah yang disediakan dan dinaungi oleh pemerintah daerah Kabupaten Kotabaru, Kalimantan Selatan. Fasilitas ini diperuntukkan secara khusus bagi mahasiswa asal Kabupaten Kotabaru yang sedang menempuh pendidikan tinggi di Yogyakarta. Asrama ini beralamat di Jl. Timoho No.138, Demangan, Kec. Gondokusuman, Kota Yogyakarta, Daerah Istimewa Yogyakarta.