BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Chatbot

Chatbot merupakan sistem perangkat yang dibuat dengan tujuan untuk melakukan interaksi dan mendapatkan informasi yang terjadi antara pengguna, baik antar pengguna dengan admin ataupun antara sistem chatbot berbasis AI dengan pengguna. Chatbot adalah layanan berbasis AI yang berfungsi untuk mensimulasikan interaksi manusia secara virtual (Hikmah et al., 2022).

2.1.2 Natural Language Processing (NLP)

Natural Language Processing adalah cabang kecerdasan buatan yang fokus pada pemrosesan dan pemahaman bahasa alami oleh komputer. NLP berperan penting dalam chatbot untuk mengenali maksud pengguna dan memberikan respons yang relevan (Jurafsky & Martin, 2020). NLP mencakup berbagai teknik seperti :

1. **Tokenisasi** merupakan proses memecah teks menjadi bagian-bagian kecil yang disebut *token*, seperti kata atau tanda baca. Proses ini penting agar komputer bisa mengenali struktur teks dan memprosesnya lebih lanjut.
2. **Named Entity Recognition (NER)** merupakan teknik untuk mengenali kata-kata penting dalam teks yang merujuk pada nama orang, tempat, organisasi,

atau tanggal. Contohnya, dalam kalimat “Presiden Jokowi mengunjungi Jakarta”, sistem mengenali “Jokowi” sebagai nama orang dan “Jakarta” sebagai lokasi.

3. **Text Classification** merupakan proses mengelompokkan teks ke dalam kategori tertentu, seperti membedakan apakah sebuah kalimat merupakan permintaan, keluhan, atau pertanyaan. Teknik ini membantu chatbot memahami maksud pengguna.
4. **Semantic Matching** merupakan teknik untuk mencocokkan dua kalimat berdasarkan arti, meskipun kata-katanya berbeda. Contohnya, pertanyaan “Kapan kantor buka?” bisa dicocokkan dengan jawaban “Kantor mulai beroperasi pukul 08.00 pagi.” Sistem memahami maknanya agar bisa memberikan jawaban yang tepat.

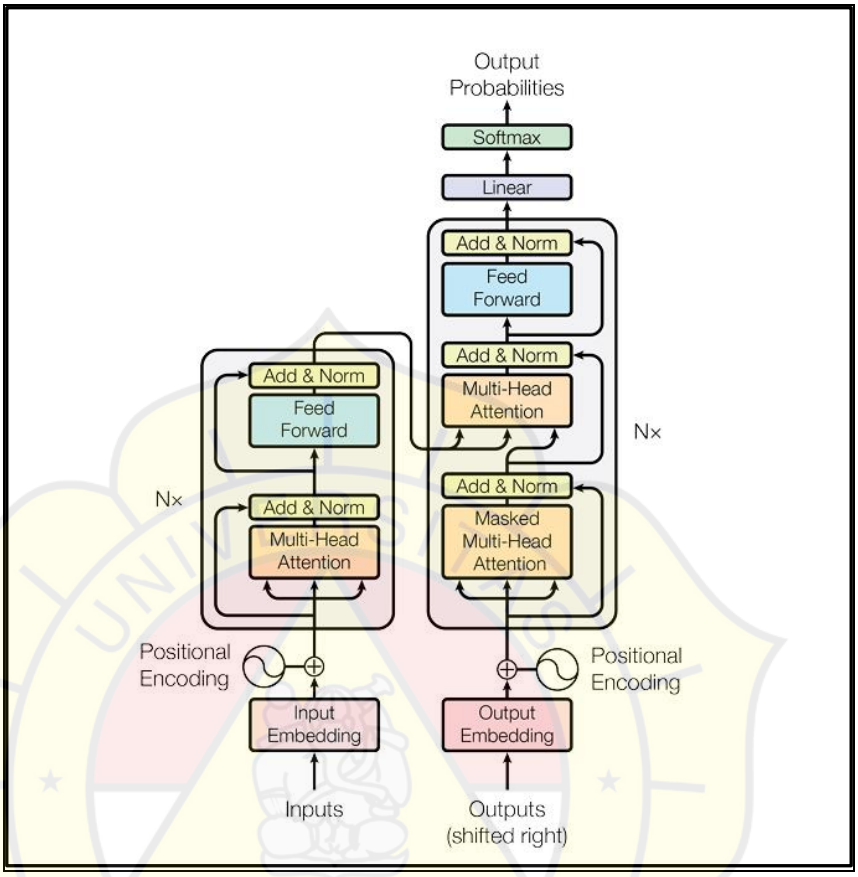
2.1.3 Deep learning

Deep learning merupakan cabang dari *machine learning* yang menggunakan arsitektur jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) untuk mengekstraksi representasi data secara otomatis dan hierarkis. *Deep learning* telah menjadi fondasi utama dalam pengembangan kecerdasan buatan modern, termasuk dalam bidang *Natural Language Processing* yang menjadi dasar dari *Large Language Models* (Bengio et al., 2021). Teknologi ini memungkinkan sistem untuk belajar langsung dari data dalam skala besar, tanpa ketergantungan penuh pada fitur buatan manusia. Dengan pendekatan

representasi yang mendalam, *deep learning* telah menunjukkan performa unggul dalam berbagai tugas AI, seperti pengenalan gambar, pengenalan suara, dan pemahaman bahasa alami, yang semuanya relevan untuk pengembangan chatbot cerdas. Perkembangan arsitektur seperti *transformer* dan *self-supervised learning* juga menjadi pendorong utama peningkatan kapabilitas sistem berbasis LLM.

2.1.4 Transformer

Transformer merupakan arsitektur jaringan saraf yang didasarkan pada mekanisme *attention* tanpa memanfaatkan unit berulang (*recurrent*) maupun konvolusi (*convolutional*). Mekanisme *attention* ini memungkinkan model untuk memberikan bobot berbeda pada setiap elemen input, di mana koefisien pembobotan tersebut bergantung secara dinamis pada nilai-nilai input itu sendiri. Dengan demikian, *Transformer* mampu menangkap bias induktif yang melekat secara kuat pada data sekuensial serta bentuk data lainnya, yang memungkinkan pemahaman konteks yang lebih baik dibandingkan model berbasis RNN atau CNN konvensional (Tay et al., 2023).



Gambar 2. 1 Arsitektur *Transformer*

Dari gambar 2.1 arsitektur *Transformer* terdiri dari dua komponen utama, yaitu *encoder* dan *decoder*. *Encoder* bertugas memproses input mentah dan mengubahnya menjadi representasi internal yang kaya secara semantik, sedangkan *decoder* menggunakan representasi ini untuk menghasilkan output yang diinginkan, seperti teks terjemahan atau rangkuman (Vaswani et al., 2017).

Berdasarkan penggunaan dan konfigurasi kedua komponen diatas ini, model berbasis *Transformer* umumnya dibagi ke dalam tiga kategori utama (Tunstall et al., 2022):

1. Encoder-only

Model ini hanya terdiri dari bagian *encoder* dan berfungsi untuk mengubah input teks menjadi representasi numerik yang bermakna. Representasi ini kemudian digunakan untuk berbagai tugas seperti klasifikasi teks, *named entity recognition* (NER), dan *analisis sentimen*. Contoh model *encoder-only* yang populer adalah BERT (*Bidirectional Encoder Representations from Transformers*) yang mengandalkan pembelajaran dua arah (*bidirectional*) untuk memahami konteks kalimat secara lebih mendalam.

2. Decoder-only

Model ini hanya terdiri dari bagian *decoder* dan dirancang untuk tugas generatif yang memprediksi *token* berikutnya dalam sebuah urutan teks. *Decoder* secara berurutan menghasilkan output dengan memanfaatkan konteks *token* sebelumnya, menjadikannya sangat cocok untuk aplikasi *language modeling*, *text generation*, dan *autocompletion*. GPT (*Generative Pre-trained Transformer*) adalah contoh utama model *decoder-only* yang telah menunjukkan performa unggul dalam berbagai aplikasi chatbot dan pemrosesan bahasa alami.

3. Encoder-Decoder

Model yang menggabungkan kedua komponen ini, *encoder* dan *decoder*,

biasanya digunakan untuk tugas yang melibatkan transformasi satu bentuk teks ke bentuk lain, seperti penerjemahan bahasa, peringkasan teks, dan tanya jawab (*question answering*). Model-model ini memanfaatkan kemampuan *encoder* dalam memahami konteks input secara komprehensif, kemudian *decoder* menerjemahkan representasi tersebut menjadi output yang sesuai. Contoh terkenal dari kategori ini adalah BART (*Bidirectional and Auto-Regressive Transformers*) dan T5 (*Text-to-Text Transfer Transformer*).

Seiring perkembangan teknologi, *Transformer* telah mengalami berbagai inovasi yang berfokus pada peningkatan efisiensi komputasi dan kemampuan menangani data yang sangat panjang. Beberapa varian seperti *Reformer*, *Linformer*, dan *Longformer* dirancang untuk mengatasi keterbatasan kompleksitas komputasi dan penggunaan memori, sehingga memungkinkan pemrosesan dokumen dengan panjang sangat besar secara efisien (Tay et al., 2023). Kemampuan *Transformer* untuk memproses data secara paralel, memahami konteks *global* dari seluruh input, serta menghasilkan representasi semantik yang mendalam, menjadikannya pilihan utama dalam pengembangan chatbot berbasis *Large Language Models* (LLM). Oleh karena itu, pemahaman yang mendalam terhadap prinsip kerja, struktur, serta evolusi arsitektur *Transformer* menjadi dasar yang penting dalam penelitian dan implementasi sistem chatbot cerdas berbasis LLM.

2.1.5 Large Language Models (LLM)

Large Language Models (LLM) merupakan model kecerdasan buatan berskala besar yang dibangun menggunakan arsitektur *Transformer* dan dilatih pada kumpulan data teks dalam jumlah sangat besar. Model ini memiliki kemampuan untuk memahami dan menghasilkan bahasa alami secara otomatis berdasarkan konteks sebelumnya, sehingga mampu menjalankan berbagai tugas pemrosesan bahasa alami secara efektif (Minaee et al., 2025). Dengan jumlah parameter yang mencapai ratusan juta hingga ratusan miliar, LLM dapat mengenali pola bahasa dan makna semantik secara mendalam, melebihi kemampuan model bahasa tradisional. Beberapa model terkemuka yang mewakili kemajuan LLM saat ini antara lain GPT-3 dan GPT-4 dari OpenAI, PaLM dari Google, serta LLaMA dari Meta AI. Model-model tersebut telah digunakan secara luas dalam berbagai aplikasi seperti sistem tanya jawab, peringkasan teks, dan chatbot (Minaee et al., 2025).

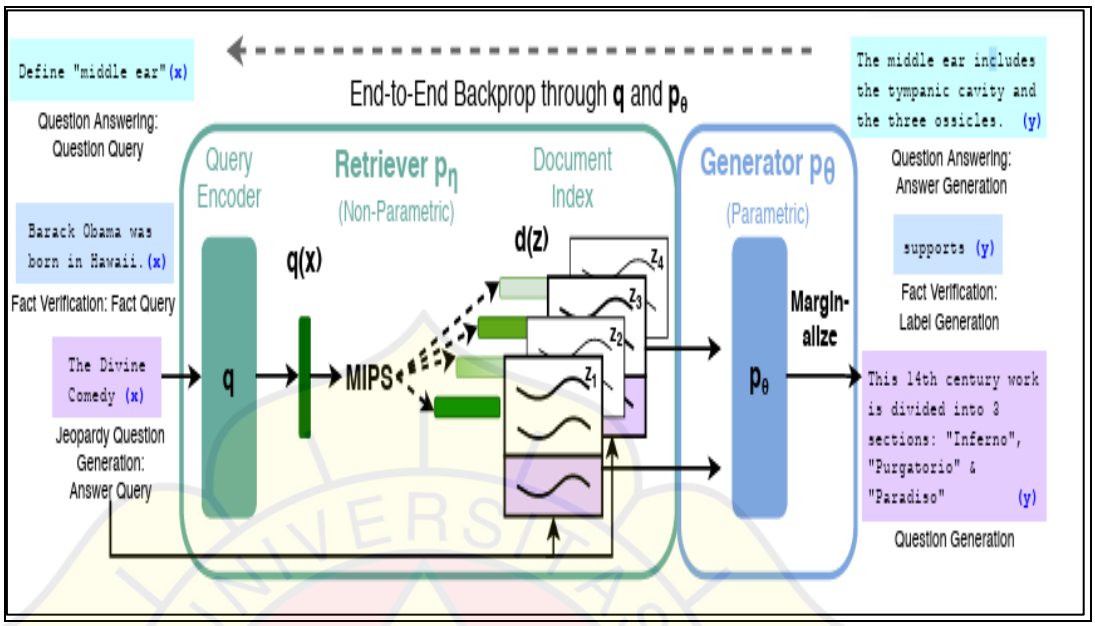
Kemampuan LLM dalam menghasilkan respons yang relevan dan menyerupai gaya bahasa manusia membuatnya banyak diterapkan dalam aplikasi dunia nyata, termasuk chatbot, asisten virtual, sistem ekstraksi informasi, dan rekomendasi berbasis teks.

Dalam proses generatif, LLM memprediksi *token* berikutnya berdasarkan konteks yang diberikan. Proses ini melibatkan pemilihan *token* dengan

kemungkinan kemunculan tertinggi dalam distribusi *probabilitas* yang dihasilkan, dan dilakukan secara berulang untuk menyusun kalimat atau respons yang utuh.

2.1.6 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) merupakan salah satu pendekatan dalam bidang pemrosesan bahasa alami (*Natural Language Processing*/NLP) yang mengintegrasikan proses pengambilan informasi dari basis data dengan kemampuan generatif model bahasa besar (*Large Language Model*). Berbeda dari metode *fine-tuning* yang mengharuskan pelatihan ulang model dengan *dataset* tertentu, RAG memungkinkan penggunaan informasi eksternal secara langsung untuk menghasilkan jawaban yang relevan berdasarkan konteks pertanyaan pengguna. Efektivitas pendekatan ini terlihat dari penerapan RAG pada pengembangan chatbot dibidang layanan informasi kesehatan menunjukkan bahwa integrasi RAG dalam chatbot mampu meningkatkan ketepatan dan kesesuaian informasi yang diberikan kepada pengguna, terutama pada pertanyaan yang memerlukan rujukan dari sumber terpercaya (Samudra et al., 2025).

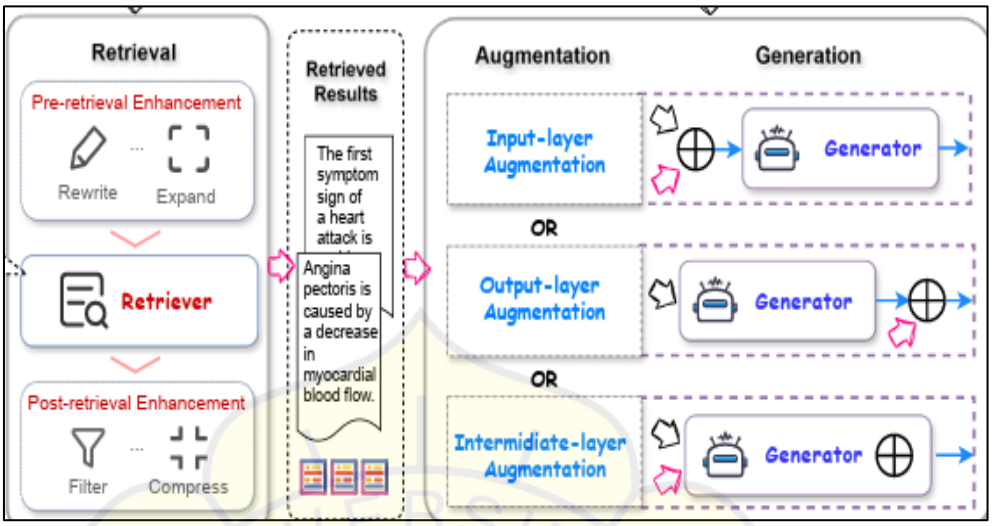


Gambar 2. 2 RAG

Gambar 2.2 secara teknis arsitektur RAG terdiri dari dua elemen utama (Lewis et al., 2020):

1. **Retriever** merupakan komponen yang bertugas menelusuri dokumen atau informasi relevan dari basis data sesuai dengan pertanyaan yang diajukan.
2. **Generator** merupakan komponen yang bertugas menyusun jawaban berdasarkan informasi yang diperoleh *retriever*.

Dalam penerapan RAG yang lebih modern arsitektur dibagi dalam 3 tahap utama (Fan et al., 2024).



Gambar 2. 3 RAG

Gambar 2.3 di atas menunjukan alur proses RAG sebagai berikut :

1. Tahap Retrieval

Tahap ini bertujuan untuk mengekstraksi informasi yang relevan dari basis data atau sumber eksternal berdasarkan pertanyaan atau perintah yang diberikan oleh pengguna. Proses *retrieval* dalam RAG modern mencakup beberapa komponen penting yaitu:

Pre-retrieval enhancement bertujuan untuk menyempurnakan tahapan input kueri sebelum proses pencarian dilakukan, misalnya dengan teknik *query rewriting* dan *query expansion* guna meningkatkan relevansi hasil pencarian.

Retriever merupakan modul yang bertugas menelusuri dokumen atau fragmen teks yang sesuai dari basis data menggunakan teknik pencarian berbasis semantik atau *dense retrieval*.

Post-retrieval enhancement yaitu proses penyaringan (*filtering*) dan peringkasan (*compression*) terhadap hasil pencarian agar hanya informasi yang paling relevan dan esensial yang digunakan pada tahap berikutnya.

2. Tahap Augmentation

Setelah informasi berhasil diperoleh melalui proses *retrieval*, langkah selanjutnya adalah mengintegrasikan informasi tersebut ke dalam alur kerja LLM. Tahap *augmentation* dalam RAG dapat dilakukan melalui tiga pendekatan teknis:

Input-layer augmentation yaitu menyisipkan informasi hasil *retrieval* secara langsung ke dalam input *prompt* yang dikirim ke model.

Intermediate-layer augmentation yaitu menyisipkan informasi ke dalam lapisan tersembunyi model melalui mekanisme seperti *cross-attention* atau *adapter modules*, sehingga informasi eksternal dapat berinteraksi lebih dalam dengan representasi internal model.

Output-layer augmentation yaitu penggunaan informasi hasil *retrieval* untuk memodulasi atau mengarahkan proses generasi output pada tahap akhir.

3. Tahap Generation

Tahap akhir dalam arsitektur RAG adalah proses generasi, di mana model LLM menghasilkan respons atau jawaban berdasarkan pertanyaan awal pengguna yang telah diperkaya dengan informasi hasil *retrieval*. Dan dengan

adanya *augmentasi* yang terstruktur, respons yang dihasilkan menjadi lebih kontekstual, relevan, dan akurat, terutama dalam skenario *knowledge-intensive* seperti chatbot, asisten medis, atau sistem tanya-jawab berbasis dokumen.

2.1.7 Chunking dan Preprocessing Teks

Dalam sistem berbasis LLM dan *Retrieval-Augmented Generation* (RAG), proses *chunking* atau pemecahan dokumen menjadi bagian-bagian pendek sangat memengaruhi akurasi pencarian informasi. *Chunk* digunakan karena model *retriever* seperti *Sentence-BERT* atau MiniLM lebih optimal saat memproses teks pendek. Potongan teks yang lebih pendek memungkinkan perhitungan kemiripan semantik yang lebih presisi.

Strategi *chunking* yang efektif meliputi penggunaan panjang *chunk* antara 64 hingga 512 *token* dan penerapan *overlap* sebesar 20–25%. *Overlap* ini penting untuk menjaga kesinambungan konteks antar *chunk*, sehingga informasi penting tidak terputus antar segmen teks. *Chunking* yang terlalu besar menyebabkan hilangnya presisi dalam pencarian, sedangkan *chunk* yang terlalu kecil bisa kehilangan konteks penting (Bhat et al., 2025).

Setelah proses *chunking*, dokumen biasanya dinormalisasi melalui konversi ke huruf kecil, penghilangan tanda baca, dan proses *tokenisasi*. Hasil *chunk* ini kemudian diubah menjadi *vector embedding* sebagai input untuk tahap *retrieval* dalam sistem RAG.

2.1.8 Embedding untuk Pencarian Semantik

Embedding merupakan proses mengubah teks menjadi representasi vektor numerik berdimensi tetap yang berada dalam ruang semantik. Dalam sistem RAG, *embedding* digunakan untuk merepresentasikan:

1. Pertanyaan pengguna (*query*)
2. *Chunk* dokumen dari basis pengetahuan

Tujuan *embedding* adalah untuk memungkinkan pencocokan semantik menggunakan kemiripan vektor, seperti *cosine similarity*, yang dirumuskan sebagai:

$$\cos(\theta) = \frac{A.B}{||A||||B||} \quad (1)$$

Di mana A adalah vektor *query* dan B adalah vektor dokumen. Model *embedding* yang sering digunakan dalam *retrieval* modern antara lain:

1. Sentence-BERT
2. MiniLM
3. Instructor-XL dan BGE-M3 (untuk kebutuhan multibahasa dan domain spesifik)

Sentence-BERT dilatih secara khusus agar dua kalimat dengan makna serupa memiliki representasi vektor yang berdekatan dalam ruang vektor (Reimers & Gurevych, 2019). Ini berbeda dari *word embedding* klasik seperti Word2Vec yang beroperasi pada level kata.

2.1.9 Prompt Engineering dalam LLM

Prompt engineering adalah praktik penyusunan input dengan tujuan meningkatkan kualitas respons LLM. Brown et al. (2020) menjelaskan bahwa teknik *prompt few-shot* dan *instruction prompting* secara efektif meningkatkan performa model tanpa *fine-tuning*, strategi lainnya mencangkup:

1. *Zero-shot* memasukan instruksi langsung tanpa contoh.
2. *Few-shot* memasukkan beberapa contoh input-output.
3. *Instruction prompting* mengarahkan model secara eksplisit melalui instruksi dalam bahasa alami.

Dengan rancangan *prompt* yang tepat, output menjadi lebih akurat, koheren, dan relevan dengan konteks pengguna (Brown et al., 2020).

2.1.10 LangChain

LangChain merupakan kerangka kerja (*framework*) *open-source* yang dirancang untuk memudahkan pengembang dalam membangun aplikasi berbasis model bahasa besar (*Large Language Models*/LLM). *Framework* ini memungkinkan integrasi antara LLM dengan berbagai komponen eksternal,

seperti basis data vektor, API, file dokumen, dan alat pencarian, guna membentuk alur kerja pemrosesan bahasa yang lebih kompleks dan kontekstual (Soygazi & Oguz, 2023).

Dalam implementasinya, LangChain menyediakan berbagai modul seperti LLMChain, *Prompt Template*, *Retriever*, dan *Agent* yang dapat disusun menjadi alur kerja (*pipeline*) untuk menjawab pertanyaan berbasis dokumen atau sumber eksternal. Penelitian ini memanfaatkan modul *RetrievalQA* untuk menggabungkan hasil pencarian dokumen dengan respons generatif dari LLM.

2.1.11 Hugging Face

Hugging Face adalah *platform* dan pustaka *open-source* yang menyediakan akses terhadap berbagai model bahasa besar (*Large Language Models*) seperti BERT, GPT-2, RoBERTa, dan T5. Pustaka ini tidak hanya menyediakan model-model siap pakai, tetapi juga mendukung proses pelatihan lanjutan (*fine-tuning*), evaluasi, serta integrasi model ke dalam berbagai aplikasi *Natural Language Processing* (NLP) melalui antarmuka berbasis Python yang mudah diimplementasikan (Wolf et al., 2020).

Selain menyediakan model-model bahasa besar, Hugging Face juga menawarkan ekosistem yang lengkap untuk mendukung pengembangan aplikasi berbasis NLP. Melalui pustaka transformers, pengguna dapat dengan mudah mengakses dan mengintegrasikan model *open-source* seperti GPT, BERT, dan

LLaMA ke dalam sistem, termasuk sistem berbasis *Retrieval-Augmented Generation* (RAG). Fitur-fitur seperti pemanggilan model pralatih, *tokenisasi* otomatis, serta antarmuka API berbasis Python menjadikan Hugging Face sebagai standar dalam implementasi LLM dalam berbagai penelitian dan aplikasi industri (Wolf et al., 2020).

2.1.12 WebSocket

WebSocket adalah protokol komunikasi berbasis TCP yang memungkinkan terjadinya pertukaran data dua arah (*full-duplex*) secara *real-time* antara client dan server. Dibandingkan dengan protokol HTTP tradisional, WebSocket hanya memerlukan satu koneksi yang tetap terbuka sehingga lebih efisien dalam penggunaan *bandwidth* dan mengurangi latensi komunikasi. Karakteristik ini membuat WebSocket sangat ideal untuk aplikasi interaktif seperti chatbot, yang membutuhkan pertukaran data secara cepat dan kontinu (Murley et al., 2021).

Dalam sistem yang dikembangkan pada penelitian ini, WebSocket digunakan sebagai saluran utama komunikasi antara antarmuka pengguna (*frontend*) dan server (*backend*). Setiap pertanyaan dari pengguna dikirimkan langsung melalui koneksi WebSocket ke *backend*, yang kemudian diteruskan ke model LLM untuk diproses. Setelah respons dihasilkan, jawaban dikirim kembali ke pengguna melalui koneksi yang sama, tanpa perlu membuka koneksi baru.

Penerapan WebSocket memberikan keuntungan signifikan dalam hal efisiensi, waktu tanggap yang rendah, serta pengalaman interaksi yang menyerupai percakapan manusia secara langsung.

2.1.13 Pengujian

Pengujian merupakan salah satu tahapan yang penting untuk mengetahui akurasi jawaban dan tingkat kepuasan pengguna terhadap chatbot yang dikembangkan. Pengujian diperlukan untuk memastikan sistem berfungsi sesuai dengan harapan (Hikmah et al., 2022). Berikut ini adalah metode yang digunakan dalam melakukan pengujian :

1. Pengujian Tingkat akurasi

Pengujian akurasi dilakukan dengan menyusun 30 pertanyaan valid seputar CV DSN. Masing-masing pertanyaan memiliki variasi yang berbeda namun mengandung inti pokok yang sama. Pengujian dilakukan melalui aplikasi chatbot, dimana sistem menampilkan respons pada antarmuka pengguna. Setiap kali pengguna mengajukan pertanyaan baru, sistem akan memprosesnya bersama dengan konteks yang tersimpan untuk menghasilkan respons yang relevan. Tingkat akurasi ditentukan dengan rumus berikut:

$$Akurasi = \frac{Jumlah\ jawaban\ sesuai}{Total\ pertanyaan} \times 100\% \quad (2)$$

2.2 Penelitian terdahulu

Beberapa penelitian sebelumnya telah membahas topik chatbot dan llm, yang relevan dengan fokus penelitian ini. Dalam penelitian ini penulis mengkaji tentang implementasi chatbot berbasis *Large Language Model* (LLM) dalam konteks *e-commerce*. Penelitian-penelitian terdahulu memberikan gambaran mengenai perkembangan teknologi tersebut, serta membantu mengidentifikasi gap yang dapat dijadikan dasar untuk melanjutkan penelitian ini. Berikut beberapa penelitian yang dijadikan rujukan pada penyusunan penelitian ini:

Tabel 2. 1 Penelitian terdahulu

No.	Judul penelitian	Authors	Fokus penelitian	Metodologi penelitan	Hasil penelitian
1.	Implementasi Chatbot Sebagai Virtual Assistant di Universitas Panca Marga Probolinggo menggunakan Metode TF-IDF	Nuzul Hikmah, Dyah Ariyanti, Ferry Agus Pratama, Tahun 2022 Jurnal JTIM	penelitian ini berfokus pada pengunaan chatbot berbasis AI sebagai assisten virtual didalam lingkungan kampus	Metode TF-IDF dalam pemrosesan pertanyaan, pengujian akurasi dan *User Acceptance Test* (UAT).	Chatbot dapat menjawab pertanyaan mahasiswa secara otomatis dan memiliki tingkat akurasi yang cukup tinggi. Pengguna merasa

					terbantu dengan adanya chatbot.
	Penelitian ini menggunakan metode TF-IDF tradisional yang terbatas pada pencocokan kata kunci, tidak memiliki pemahaman kontekstual seperti LLM. Sementara penelitian ini mengusulkan sistem RAG berbasis LLM yang mampu memahami konteks dan menjawab pertanyaan berdasarkan pengetahuan dokumen secara dinamis.				
2.	CODEFUSION: A Pre-trained Diffusion Model for Code Generation	Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, Gust Verbruggen. Tahun 2023. Publisher:	Pemanfaatan metode llm dalam pembuatan dan pemerograman secara otomatis	*Model pre-trained* berbasis *diffusion* untuk pemrosesan kode.	Model LLM mampu memahami konteks pemrograman dan menghasilkan kode yang sesuai, meningkatk

		Association for Computational Linguistics			an produktifitas developer.
	Meskipun menggunakan LLM, penelitian ini berfokus pada generasi kode secara otomatis, bukan interaksi berbasis percakapan. Penelitian ini tidak membahas arsitektur sistem layanan pelanggan, berbeda dengan penelitian ini yang membangun chatbot untuk interaksi langsung dengan pelanggan *e-commerce.*				
3.	An Analysis of Large Language Models and LangChain in Mathematics Education	Soygazi F, Oguz D Tahun 2023. Publisher: Association for Computing Machinery	Analisis penggunaan LLM dan framework LangChain dalam edukasi matematika	Analisis eksperimen dan studi literatur terhadap implementasi LLM.	Metode LLM efektif dalam proses pembelajaran matematika dan dapat digunakan untuk mengembangkan chatbot edukatif
	Penelitian ini menelaah LLM dan LangChain dalam pembelajaran matematika. Fokusnya adalah edukasi dan penyampaian materi ajar. Berbeda dengan penelitian ini yang menerapkan LangChain dan LLM dalam sistem tanya-jawab untuk keperluan layanan pelanggan toko online.				
4.	HuggingFace's Transformers: State-of-the-art	Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi	Pemanfaatan transformers dalam NLP,	Eksperimen dan pengembangan pustaka open-	*Transformers* seperti GPT dan

	Natural Language Processing	A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, Von Platen P, ,Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A. Tahun 2020. Publisher: Association for Computational Linguistics	termasuk pengembangan chatbot	source seperti Hugging Face	BERT dapat meningkatkan performa chatbot dalam memahami dan merespons bahasa alami.
	Penelitian ini merupakan studi literatur tentang *Transformer* tanpa implementasi langsung ke sistem chatbot untuk layanan pelanggan. Penelitian ini memiliki kontribusi praktis dalam pengembangan aplikasi nyata berbasis RAG yang terintegrasi ke sistem *e-commerce.*				

5.	Implementasi Chatbot Menggunakan Framework Langchain Berbasis LLM GPT (Studi Kasus: Panduan Akademik UTM)	Muhammad Dimas Arya Muhajir, Novi Prastiti, Meidya Koeshardianto. Tahun 2025. Publisher: JATI (Jurnal Mahasiswa Teknik Informatika)	Chatbot untuk layanan akademik menggunakan GPT-3.5 + LangChain	RAG, LangChain, Streamlit, GPT-3.5 Turb	Akurasi tinggi (96.66%), UAT 82.79%, respons terasa natural
	Penelitian ini sangat relevan karena menggunakan GPT-3.5 dan LangChain untuk menjawab pertanyaan mahasiswa secara otomatis. Namun, fokusnya terbatas pada konteks akademik, bukan domain komersial. Sistem yang diusulkan dalam penelitian ini menggunakan pendekatan serupa tetapi diterapkan pada chatbot layanan pelanggan toko online, dengan cakupan dokumen dan pertanyaan yang lebih variatif dan kontekstual.				
6.	Implementasi RAG dalam Perancangan Chatbot Kesehatan Pencernaan	Gufranaka Samudra, Ahmad Turmudi Zy, Ermanto. Tahun 2025,	Chatbot kesehatan dengan RAG dan LLaMA	RAG, Indo-SBERT, HNSW, LLaMA 3	MRR 93%, Semantic Similarity 81%

		Jurnal: JSAI: Journal Scientific and Applied Informatics			
	Penelitian ini juga menggunakan metode RAG, tetapi diterapkan pada domain kesehatan, dengan sumber data yang lebih sensitif dan struktural. Model yang digunakan adalah LLaMA 3.1:8B, sama seperti yang diusulkan dalam penelitian ini. Namun, konteks dan tujuan sistem sangat berbeda—penelitian ini menargetkan peningkatan pelayanan konsumen dalam e-commerce lokal, yang menuntut sistem untuk merespons data operasional toko dan pertanyaan seputar produk.				

Dari tabel 2.1 diatas pada penelitian terdahulu yang pernah diteliti terdapat perbedaan pada penelitian yang akan diteliti oleh penulis. Sebagian besar penelitian sebelumnya berfokus pada implementasi chatbot menggunakan metode tradisional seperti TF-IDF, atau pada penerapan LLM dalam konteks umum dan edukatif. Sementara itu, penelitian ini secara khusus akan mengimplementasikan chatbot berbasis *Large Language Model* (LLM), serta diintegrasikan langsung dengan web toko online milik CV Darma Sentosa Nuswantoro. Penelitian ini juga menitikberatkan pada cara kerja chatbot untuk meningkatkan pelayanan terhadap pelanggan dan meringankan beban admin, yang belum banyak dibahas secara mendalam dalam penelitian-penelitian terdahulu.