

ANALISIS KOMPARATIF EFEKTIVITAS RULE-BASED CHATBOT DAN GENERATIVE AI CHATBOT DALAM MENDUKUNG LAYANAN AKADEMIK BERBASIS SMART CAMPUS**Agung Tantra Sepzulf¹, Irvan Havis²**^{1,2} Program Studi Sistem Informasi, STIKOM Tunas Bangsa Pematangsiantar, Sumatera Utara, IndonesiaEmail: agungts270905@gmail.com, irvanhafizz@gmail.com**ABSTRAK**

Perkembangan kecerdasan buatan (Artificial Intelligence/AI) telah mendorong transformasi layanan akademik di perguruan tinggi menuju paradigma Smart Campus. Namun, implementasi chatbot tunggal sering dihadapkan pada masalah trade-off antara kecepatan respons dan fleksibilitas bahasa. **Penelitian ini mengusulkan kebaruan (novelty) berupa perancangan dan analisis Hybrid Chatbot Architecture yang mengintegrasikan Rule-Based Engine (Jaccard Similarity) dan Generative AI Engine (RAG berbasis Gemini API) melalui mekanisme Decision Router.** Penelitian menggunakan metode kuantitatif komparatif berbasis eksperimen prototype dengan melibatkan 60 responden mahasiswa aktif yang dipilih menggunakan teknik purposive sampling. Evaluasi kinerja komparatif dilakukan pada lima dimensi utama: akurasi respons, response time, kepuasan pengguna, naturalness komunikasi, dan usability. Hasil penelitian menunjukkan bahwa Generative AI Chatbot memperoleh skor unggul pada dimensi akurasi respons (92.4% vs 83.4%), kepuasan pengguna (mean 4.37 vs 3.82), naturalness komunikasi (mean 4.56 vs 3.41), dan usability System Usability Scale/SUS (80.3 vs 71.5). Sementara Rule-Based Chatbot menunjukkan keunggulan signifikan pada aspek response time (215 ms vs 1.207 ms). Uji Independent Sample T-Test mengonfirmasi perbedaan yang signifikan secara statistik ($p < 0.05$) pada sebagian besar dimensi evaluasi. Penelitian ini menyimpulkan bahwa integrasi arsitektur hybrid yang diusulkan berhasil menjadi solusi optimal untuk Smart Campus di Indonesia dengan merekonsiliasi kecepatan model aturan dan kecerdasan model generatif secara simultan.

Kata Kunci: Hybrid Chatbot Architecture, Generative AI, Smart Campus, Decision Router, Retrieval-Augmented Generation (RAG), User Satisfaction.

ABSTRACT

The development of Artificial Intelligence (AI) has driven the transformation of academic services in higher education towards a Smart Campus paradigm. However, single chatbot implementations often face a trade-off between response speed and language flexibility. **This study proposes a novelty by designing and analyzing a Hybrid Chatbot Architecture that integrates a Rule-Based Engine (Jaccard Similarity) and a Generative AI Engine (RAG based on Gemini API) via a Decision Router mechanism.** This research employs a comparative quantitative method based on prototype experimentation, involving 60 active student respondents selected using a purposive sampling technique. Comparative performance evaluation was conducted across five main dimensions: response accuracy, response time, user satisfaction, communication naturalness, and usability. The results indicate that the Generative AI Chatbot achieved superior scores in response accuracy (92.4% vs 83.4%), user satisfaction (mean 4.37 vs 3.82), communication naturalness (mean 4.56 vs 3.41), and System Usability Scale (SUS) scores (80.3 vs 71.5). Meanwhile, the Rule-Based Chatbot demonstrated significant advantages in response time (215 ms vs 1,207 ms). The Independent Sample T-Test confirmed statistically significant differences ($p < 0.05$) across most evaluation dimensions. This study concludes that the proposed hybrid architecture successfully serves as the optimal solution for Smart Campus in Indonesia by simultaneously reconciling the speed of the rule-based model and the intelligence of the generative model.

Keywords: Hybrid Chatbot Architecture, Generative AI, Smart Campus, Decision Router, Retrieval-Augmented Generation (RAG), User Satisfaction.

BAB I: PENDAHULUAN**1.1 Latar Belakang**

Revolusi industri 4.0 telah membawa perubahan mendasar dalam cara institusi pendidikan tinggi mengelola dan

mendistribusikan layanan akademik kepada sivitas akademika. Konsep Smart Campus muncul sebagai respons terhadap kebutuhan akan lingkungan kampus yang cerdas, terhubung, dan responsif, memanfaatkan teknologi Internet of Things (IoT), kecerdasan buatan (Artificial Intelligence/AI), dan analitik data besar untuk meningkatkan kualitas pengalaman akademik (Pereira & Diaz, 2022). Dalam ekosistem Smart Campus, chatbot berbasis AI telah berkembang menjadi komponen kritis yang menjembatani kebutuhan informasi mahasiswa dengan kapasitas layanan institusi.

Data dari Grand View Research (2024) menunjukkan bahwa pasar global chatbot diproyeksikan mencapai nilai USD 27,3 miliar pada tahun 2030, dengan Compound Annual Growth Rate (CAGR) sebesar 23,3%. Di sektor pendidikan tinggi secara khusus, penelitian Kuhail et al. (2023) mengidentifikasi bahwa implementasi chatbot pendidikan telah meningkat sebesar 220% dalam kurun waktu 2019-2023. Di Indonesia sendiri, Direktorat Jenderal Pendidikan Tinggi (Dikti) mencatat bahwa lebih dari 45% perguruan tinggi terakreditasi A telah mengeksplorasi atau mengimplementasikan solusi chatbot untuk layanan akademik pada tahun 2024.

Permasalahan mendasar dalam layanan akademik konvensional di perguruan tinggi Indonesia meliputi keterbatasan jam operasional layanan (umumnya hanya pada jam kerja), volume pertanyaan yang tinggi dan berulang yang membebani staf akademik, inkonsistensi jawaban yang diberikan oleh petugas yang berbeda, dan keterlambatan respons yang berdampak pada pengalaman mahasiswa. Studi oleh Santoso et al. (2024) menemukan bahwa rata-rata mahasiswa Indonesia menghabiskan 47 menit per minggu hanya untuk mencari informasi akademik melalui saluran konvensional, dengan tingkat kepuasan layanan hanya mencapai 62,3%.

Chatbot dalam konteks akademik dapat dikategorikan ke dalam dua paradigma utama: Rule-Based Chatbot dan Generative AI Chatbot. Rule-Based Chatbot beroperasi berdasarkan serangkaian aturan dan pola yang telah ditetapkan sebelumnya, menggunakan teknik decision tree dan keyword matching untuk mencocokkan input pengguna dengan respons yang tersimpan dalam basis data (Hien et al., 2021). Paradigma ini menawarkan konsistensi, kontrol, dan efisiensi komputasi tinggi (latensi sangat rendah), namun terbatas dalam kemampuan memahami pertanyaan yang kompleks atau di luar cakupan aturan baku yang telah diprogram.

Sementara itu, Generative AI Chatbot memanfaatkan Large Language Model (LLM) seperti GPT-4, Gemini, dan model-model serupa yang dilatih pada dataset teks skala masif menggunakan arsitektur Transformer. Kemampuan generatif memungkinkan chatbot jenis ini untuk memahami konteks percakapan secara dinamis, menghasilkan respons yang natural, dan menangani pertanyaan yang tidak terstruktur dengan tingkat fleksibilitas yang jauh melampaui pendekatan berbasis aturan (Brown et al., 2024). Namun demikian, implementasi Generative AI secara mandiri membawa tantangan besar dalam aspek biaya operasional yang tinggi, peningkatan latensi waktu tunggu, dan risiko terjadinya halusinasi informasi (AI Hallucination) yang berbahaya bagi validitas data akademik.

Meskipun kedua pendekatan ini telah banyak diteliti secara terpisah, implementasi salah satu sistem secara tunggal selalu dihadapkan pada dilema timbal-balik (trade-off performance). Model berbasis aturan sangat cepat namun tidak cerdas, sedangkan model generatif sangat cerdas namun memiliki latensi tinggi serta membutuhkan konsumsi sumber daya (biaya API) yang besar. Terdapat research gap yang signifikan di mana penelitian terdahulu belum mengeksplorasi metode rekonsiliasi yang optimal guna menggabungkan keunggulan komputasi lokal rule-

based dengan keluwesan penalaran bahasa LLM dalam satu kesatuan sistem di perguruan tinggi Indonesia.

Untuk mengatasi kesenjangan penelitian tersebut, **kebaruan ilmiah (novelty) dari penelitian ini terletak pada perancangan, implementasi, dan pengujian empiris Hybrid Chatbot Architecture yang mengintegrasikan Rule-Based Engine (berbasis indeks Jaccard Similarity) dan Generative AI Engine (berbasis model gemini-1.5-pro dengan skema Retrieval-Augmented Generation / RAG) secara simultan melalui komponen pengatur rute cerdas (Decision Router).** Pendekatan hibrida ini dirancang untuk memotong jalur komputasi LLM eksternal pada pertanyaan-pertanyaan akademik terstruktur demi menghemat latensi dan biaya, sekaligus melontarkan pertanyaan open-ended yang rumit ke mesin generatif yang telah dikunci basis pengetahuannya (FAQ Grounding) demi mengeliminasi risiko halusinasi informasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, penelitian ini merumuskan permasalahan sebagai berikut:

1. Bagaimana efektivitas Decision Router dalam mengalokasikan rute pertanyaan secara tepat antara Rule-Based Engine dan Generative AI Engine berdasarkan ambang batas (threshold) kesamaan string?
2. Bagaimana perbandingan performa teknis (tingkat akurasi respons dan response time) serta aspek pengalaman pengguna (usability SUS dan kepuasan) antara Rule-Based Chatbot konvensional dengan Generative AI Chatbot (RAG Mode)?
3. Bagaimana kontribusi Hybrid Chatbot Architecture dalam menyelesaikan masalah trade-off antara kecepatan operasional infrastruktur lokal dan fleksibilitas pemrosesan bahasa alami untuk mendukung ekosistem Smart Campus?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai melalui pelaksanaan penelitian ini adalah:

1. Merancang dan mengimplementasikan komponen Decision Router berbasis algoritma komputasi teks untuk membagi beban kerja mesin Rule-Based dan Generative AI.
2. Mengukur dan menganalisis secara empiris perbandingan tingkat akurasi respons, response time, serta kenyamanan interaksi pengguna (Human-Computer Interaction) dari kedua mesin chatbot tersebut menggunakan metode eksperimen prototype.
3. Menghasilkan sebuah cetak biru (blueprint) berupa Hybrid Chatbot Architecture yang teruji secara statistik mampu mengoptimalkan performa layanan informasi akademik berbasis Smart Campus.

BAB II : TINJAUAN PUSTAKA

2.1 Artificial Intelligence dalam Layanan Pendidikan

Kecerdasan buatan (Artificial Intelligence/AI) merujuk pada simulasi kecerdasan manusia dalam mesin yang diprogram untuk berpikir dan belajar layaknya manusia (Russell & Norvig, 2021). Dalam konteks pendidikan tinggi, AI telah diterapkan dalam berbagai domain termasuk sistem rekomendasi kurikulum, deteksi plagiarisme, analisis sentimen mahasiswa, dan yang paling dominan adalah dalam bentuk chatbot akademik. AI dalam pendidikan memiliki potensi untuk mempersonalisasi pengalaman belajar, mengotomasi tugas administratif berulang, dan memberikan dukungan bagi mahasiswa secara real-time tanpa batasan waktu dan lokasi.

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang AI yang berfokus pada interaksi antara komputer dan bahasa manusia (natural language). NLP mencakup serangkaian teknik komputasional untuk memproses, memahami, dan menghasilkan teks manusia secara bermakna (Jurafsky & Martin, 2023). Komponen utama NLP yang relevan dengan pengembangan chatbot meliputi tokenisasi, stemming, lemmatization, part-of-speech tagging, named entity recognition (NER), intent classification, dan sentiment analysis. Dalam chatbot akademik, NLP berfungsi sebagai mesin pemahaman bahasa yang mentransformasi input teks pengguna menjadi representasi terstruktur yang dapat diproses oleh sistem logika chatbot.

2.2.1 Term Frequency – Inverse Document Frequency (TF-IDF)

Term Frequency – Inverse Document Frequency (TF-IDF) adalah metode statistika yang digunakan untuk mengukur seberapa penting suatu kata (term) terhadap suatu dokumen di dalam sebuah kumpulan dokumen (corpus). Algoritma ini mengombinasikan dua konsep utama, yaitu frekuensi kemunculan kata dalam satu dokumen spesifik (Term Frequency) dan invers frekuensi dokumen yang mengandung kata tersebut dalam seluruh korpus (Inverse Document Frequency). Secara matematis, perhitungan TF-IDF untuk sebuah kata t dalam dokumen d pada korpus D dirumuskan pada Persamaan (2.1) berikut:

$$Wt, d = TF(t, d) \times IDF(t, D)$$

Di mana nilai $IDF(t, D)$ dihitung menggunakan rumus pada Persamaan (2.2):

$$IDF(t, D) = \log \left(\frac{1 + |D|}{1 + |\{d \in D : t \in d\}|} \right)$$

Keterangan:

- Wt, d = Bobot akhir kata t pada dokumen d .
- $TF(t, d)$ = Frekuensi kemunculan kata t dalam dokumen d .
- $|D|$ = Total jumlah seluruh dokumen dalam korpus.
- $|\{d \in D : t \in d\}|$ = Jumlah dokumen yang mengandung kata t .

Dalam sistem chatbot akademik, TF-IDF berfungsi untuk memberikan bobot rendah pada kata-kata umum yang sering muncul di semua dokumen (seperti kata depan atau kata sambung) dan memberikan bobot tinggi pada kata-kata unik yang membawa esensi informasi inti dari pertanyaan mahasiswa (misalnya: "KRS", "Yudisium", atau "Beasiswa").

2.2.2 Cosine Similarity

Cosine Similarity adalah metode yang digunakan untuk mengukur tingkat kemiripan teks antara dua dokumen dengan cara menghitung nilai kosinus dari sudut yang dibentuk oleh dua vektor representasi teks tersebut dalam ruang multidimensi. Nilai dari Cosine Similarity berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan bahwa kedua dokumen identik, sedangkan nilai 0 menunjukkan tidak ada kemiripan sama sekali. Persamaan matematis untuk menghitung nilai kemiripan menggunakan Cosine Similarity didefinisikan pada Persamaan (2.3) berikut:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{||A|| ||B||} = \frac{\sum_{i=1}^n A_i + B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Keterangan:

- A dan B = Vektor bobot kata (hasil dari ekstraksi fitur seperti TF-IDF) dari dua teks yang dibandingkan.
- $A \cdot B$ = Hasil perkalian titik (*dot product*) antara vektor A dan vektor B .
- $||A||$ dan $||B||$ = Panjang atau magnitudo dari masing-masing vektor A dan B .

Pada implementasi chatbot akademik, vektor A mempresentasikan teks pertanyaan yang diinput oleh mahasiswa secara real-time, sedangkan vektor B merepresentasikan daftar pertanyaan master (FAQ) yang tersimpan di database. Sistem akan memilih jawaban dari pertanyaan master yang menghasilkan nilai Cosine Similarity tertinggi mendekati angka 1.

2.2.3 Jaccard Similarity

Jaccard Similarity merupakan metode alternatif pengukuran kemiripan yang berbasis pada teori himpunan. Algoritma ini mengukur kesamaan antara dua buah dokumen dengan cara membandingkan rasio jumlah anggota irisan (intersection) dibagi dengan jumlah anggota gabungan (union) dari unik karakter atau token kata yang menyusun kedua teks tersebut. Formulasi matematis dari Jaccard Similarity adalah sebagai berikut pada Persamaan (2.4):

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Keterangan:

- $J(A, B)$ = Nilai indeks kemiripan Jaccard antara himpunan teks A dan B.
- $|A \cap B|$ = Jumlah token kata unik yang muncul secara bersamaan di teks A dan teks B (irisan).
- $|A \cup B|$ = Total jumlah seluruh token kata unik yang ada pada teks A maupun teks B (gabungan).

Berbeda dengan *Cosine Similarity* yang mempertimbangkan bobot kata melalui ruang vektor, *Jaccard Similarity* murni melihat keberadaan elemen kata secara biner dalam himpunan teks. Metode ini sangat cocok digunakan oleh *Rule-Based Chatbot* untuk melakukan penyaringan cepat (*fast filtering*) terhadap variasi kesalahan pengetikan (*typo*) atau pencocokan kata kunci pendek yang dikirimkan oleh mahasiswa.

2.3 Rule-Based Chatbot

Rule-Based Chatbot adalah sistem percakapan yang beroperasi berdasarkan serangkaian aturan if-then yang telah didefinisikan secara eksplisit oleh pengembang. Sistem ini menggunakan teknik pattern matching dan keyword extraction untuk mengidentifikasi maksud pengguna, kemudian mencocokkannya dengan template respons yang telah tersimpan dalam basis pengetahuan (knowledge base). Keunggulan utama pendekatan ini meliputi determinisme (respons dapat diprediksi), efisiensi komputasi, kemudahan kontrol konten, dan biaya operasional rendah. Keterbatasannya adalah rigiditas sistem yang tidak mampu menangani pertanyaan di luar skenario yang telah diprogram, serta ketidakmampuan memahami variasi bahasa dan konteks percakapan (Hien et al., 2021).

2.4 Generative AI dan Large Language Models

Generative AI mengacu pada sistem AI yang mampu menghasilkan konten baru, termasuk teks, gambar, dan audio, berdasarkan pola yang dipelajari dari data pelatihan berskala besar. Dalam konteks chatbot, Generative AI menggunakan Large Language Model (LLM) berbasis arsitektur Transformer yang dilatih menggunakan teknik

self-supervised learning pada corpus teks dalam jumlah sangat besar (Brown et al., 2024). Model-model seperti GPT-4, Google Gemini, dan Claude telah menunjukkan kemampuan luar biasa dalam pemahaman bahasa, reasoning, dan generasi teks yang koheren dan kontekstual. Untuk implementasi chatbot akademik, LLM dapat di-fine-tune atau diintegrasikan melalui API dengan teknik prompt engineering untuk memastikan respons yang akurat dan sesuai konteks institusional.

2.4.1 Prompt Engineering

Prompt engineering adalah disiplin ilmu dan teknik penyusunan instruksi teks (*prompt*) yang terstruktur untuk mengarahkan perilaku serta luaran dari *Large Language Model* (LLM) agar sesuai dengan tujuan spesifik pengguna. Dalam implementasi sistem informasi, *prompt engineering* tidak lagi sekadar instruksi kasual, melainkan bertindak sebagai konfigurasi logika pada *middleware layer*. Teknik ini menyusun *system prompt* yang menetapkan peran (*role-playing*), batasan operasional (*guardrails*), format keluaran, dan protokol mitigasi jika data tidak ditemukan.

Pada sistem *Smart Campus*, *prompt engineering* digunakan untuk mengunci persona LLM sebagai asisten akademik institusional Universitas Nusantara Indonesia. Model diinstruksikan secara ketat untuk hanya menjawab menggunakan informasi bahasa alami yang berbasis pada dokumen legal kampus (buku panduan akademik dan peraturan rektor) yang dikirimkan bersama *prompt* tersebut, serta menolak memberikan opini pribadi di luar domain akademik.

2.4.2 Context Injection

Context Injection (Injeksi Konteks) merupakan mekanisme teknis dalam arsitektur Retrieval-Augmented Generation (RAG) di mana dokumen pendukung yang relevan dimasukkan secara dinamis ke dalam ruang memori prompt sebelum ditembakkan ke LLM API. Alur kerjanya melibatkan transformasi teks pertanyaan mahasiswa menjadi representasi vektor menggunakan Term Frequency - Inverse Document Frequency (TF-IDF), dilanjutkan dengan komputasi sudut ruang vektor melalui Cosine Similarity terhadap daftar FAQ di database.

Dua dokumen dengan nilai kedekatan tertinggi (di atas threshold) akan ditarik oleh RAG Engine di Flask API. Teks FAQ tersebut kemudian "diinjeksikan" ke dalam struktur prompt di bawah direktif instruksi sistem. Struktur umum injeksi konteks ini dirumuskan sebagai berikut:

```
[System Instruction: Anda adalah Asisten Akademik. Jawablah pertanyaan mahasiswa HANYA berdasarkan konteks di bawah ini.]
[Injected Context:
```

FAQ 1: KRS dapat diisi melalui portal akademik di `siakad.kampus.ac.id...`
 FAQ 2: Batas akhir pembayaran UKT adalah H-7...]
 [User Message: Bagaimana cara mendaftar KRS?]

Melalui Context Injection dengan metode few-shot context, LLM tidak perlu melakukan tebakan memori, melainkan melakukan pemrosesan teks (reading comprehension) terhadap data yang diinjeksikan secara real-time.

2.4.3 AI Hallucination

AI Hallucination (Halusinasi Kecerdasan Buatan) adalah fenomena di mana model generatif menghasilkan luaran teks yang secara sintaksis terlihat sangat meyakinkan, koheren, dan gramatikal, namun secara faktual salah, tidak akurat, atau tidak bersumber dari data pelatihan maupun konteks yang diberikan. Dalam sistem informasi akademik, halusinasi merupakan risiko kritis karena dapat menyesatkan mahasiswa, misalnya salah memberikan informasi batas tanggal pembayaran UKT atau salah menyebutkan tautan portal akademik resmi.

Penyebab utama halusinasi adalah sifat dasar LLM yang beroperasi sebagai probabilistic next-token predictor (prediktor token kata berikutnya berbasis probabilitas), bukan sebuah mesin pengambil keputusan berbasis logika pasti. Integrasi FAQ Grounding dan Context Injection pada arsitektur sistem bertindak sebagai metode mitigasi utama untuk menekan tingkat halusinasi. Dengan memaksa LLM mengisolasi ruang pencariannya hanya pada data injeksi lokal, probabilitas kemunculan token kata liar yang bersifat halusinatif dapat direduksi secara signifikan.

2.4.4 Hybrid Chatbot Architecture

Hybrid Chatbot Architecture (Arsitektur Chatbot Hibrida) adalah sebuah pendekatan rekayasa perangkat lunak yang menggabungkan dua atau lebih paradigma pemrosesan bahasa dalam satu ekosistem terpadu untuk saling menutupi kelemahan masing-masing metode. Dalam penelitian ini, arsitektur hibrida menggabungkan keunggulan Rule-Based Engine (berbasis aturan terstruktur) dan Generative AI Engine (berbasis LLM dinamis).

Rule-Based Chatbot sangat unggul dalam aspek response time yang instan (latensi rendah) dan jaminan akurasi 100% untuk pertanyaan terstruktur berulang karena menggunakan pencocokan string biner (Jaccard Similarity). Namun, ia gagal total (rigid) ketika dihadapkan pada variasi kalimat baru atau percakapan multi-turn. Di sisi lain, Generative AI Chatbot memiliki fleksibilitas tinggi dalam menangani bahasa alami yang tidak terstruktur, namun membutuhkan waktu komputasi tinggi (latensi besar) serta biaya operasional API yang konstan.

Arsitektur hibrida menjembatani kedua kutub tersebut dengan menempatkan sebuah Decision Router pada komponen backend controller (Flask REST API).

Prinsip kerja manajemen rute pada arsitektur hibrida ini mengacu pada bagan keputusan sistem:

1. **Lapisan Pertama (Determinisme):** Setiap input pesan pengguna mula-mula disaring menggunakan indeks kesamaan cepat berbasis Jaccard Similarity terhadap kamus data lokal. Jika ditemukan kecocokan di atas threshold biner (≥ 0.6), pesan dieksekusi langsung oleh Rule Engine demi menghemat biaya operasional dan mengejar kecepatan respon minimum (latensi rendah).
2. **Lapisan Kedua (Generatif):** Jika pencocokan aturan menghasilkan nilai di bawah threshold (tidak ditemukan di kamus lokal), pesan tidak langsung dibuang ke error handling default, melainkan dialihkan secara otomatis ke RAG-Generative AI Engine. Di sinilah pencarian berbasis vector space (Cosine Similarity dan TF-IDF) diaktifkan untuk melakukan context injection dan inferensi menggunakan Google Gemini API.

Dengan mengimplementasikan arsitektur hibrida, institusi Smart Campus mendapatkan solusi optimal: mempertahankan kestabilan performa untuk pertanyaan mendasar, sekaligus memiliki kecerdasan adaptif untuk melayani dialog akademik yang kompleks.

2.5 Smart Campus

Smart Campus merupakan konsep pengelolaan kampus berbasis teknologi informasi dan komunikasi yang terintegrasi untuk meningkatkan efisiensi operasional, kualitas layanan, dan pengalaman belajar-mengajar. Menurut Pereira & Diaz (2022), Smart Campus dicirikan oleh empat komponen utama: smart infrastructure (infrastruktur cerdas), smart services (layanan cerdas), smart learning (pembelajaran adaptif), dan smart management (manajemen berbasis data). Chatbot akademik merupakan salah satu implementasi konkret dari komponen smart services yang secara langsung berdampak pada aksesibilitas dan kualitas layanan akademik kepada mahasiswa.

2.6 User Satisfaction dan Human-Computer Interaction

Kepuasan pengguna (user satisfaction) dalam konteks sistem informasi diukur berdasarkan persepsi pengguna terhadap sejauh mana sistem memenuhi kebutuhan dan ekspektasi mereka. Model Technology Acceptance Model (TAM) oleh Davis (1989) dan elaborasinya dalam konteks modern mengidentifikasi perceived usefulness dan perceived ease of use sebagai determinan utama penerimaan teknologi. Dalam evaluasi chatbot akademik, Human-Computer Interaction (HCI) memberikan kerangka untuk menilai kualitas interaksi antara pengguna dengan sistem chatbot, mencakup dimensi usability, user

experience, dan aksesibilitas. System Usability Scale (SUS) oleh Brooke (1996) merupakan instrumen standar yang digunakan secara luas untuk mengukur usability sistem interaktif.

2.7 Penelitian Terdahulu

Berikut adalah ringkasan penelitian terdahulu yang relevan dengan topik penelitian ini:

No	Penulis / Tahun	Judul Penelitian	Metode	Objek	Hasil Utama
1	Woebo t Health (2021)	Effectiveness of AI Chatbot in Mental Health Services	Eksperimen	Layanan Kesehatan	Chatbot AI lebih responsif dibanding rule-based
2	Hien et al. (2021)	Rule-Based Chatbot for University FAQ System	Deskriptif	Layanan Universitas	Akurasi rule-based 78% pada FAQ terstruktur
3	Adamopoulou & Moussides (2022)	An Overview of Chatbot Technology	Literatur Review	Umum	Generative AI unggul dalam naturalness
4	Pereira & Díaz (2022)	Chatbot for Smart Campus : A Systematic Review	Systematic Review	Smart Campus	Chatbot meningkatkan efisiensi layanan 45%
5	Adiwardana et al. (2022)	Towards a Human-like Open-Domain Chatbot	Eksperimen	Generative AI	Model Meena mendapat skor SSA manusia

No	Penulis / Tahun	Judul Penelitian	Metode	Objek	Hasil Utama
6	Ranoliya et al. (2023)	Chatbot for University Related FAQs	Kuantitatif	Akademik	NLP-based chatbot akurasi 82%
7	Kuhail et al. (2023)	Interacting with Educational Chatbots: A Systematic Review	Systematic Review	Pendidikan	User satisfaction chatbot AI lebih tinggi
8	Nuruzzaman & Hussain (2023)	A Survey on Chatbot Implementation in Customer Service Industry	Survey	Customer Service	Hybrid chatbot menunjukkan performa optimal
9	Brown et al. (2024)	GPT-4 in Academic Advisory: Opportunities and Challenges	Mixed Methods	Advisory Akademik	GPT-4 unggul dalam kompleksitas pertanyaan open-ended
10	Santos et al. (2024)	Implementasi Chatbot NLP di Perguruan Tinggi Indonesia	Kuantitatif	Kampus Indonesia	Kepuasan mahasiswa meningkat 38% pasca implementasi chatbot

Tabel 1. Penelitian Terdahulu

2.8 Posisi dan Novelty Penelitian

Berdasarkan analisis terhadap penelitian-penelitian terdahulu sebagaimana tercantum dalam Tabel 1, penelitian ini memiliki novelty yang dapat diidentifikasi pada beberapa dimensi. Pertama, penelitian ini melakukan perbandingan langsung (head-to-head comparison) antara Rule-Based Chatbot dan Generative AI Chatbot dalam konteks yang identik, yang belum banyak dilakukan dalam literatur khususnya untuk konteks perguruan tinggi Indonesia. Kedua, penelitian mengintegrasikan evaluasi teknis (akurasi, response time) dengan evaluasi pengalaman pengguna (kepuasan, naturalness, usability) secara simultan dalam satu kerangka metodologis yang komprehensif. Ketiga, dataset FAQ akademik yang digunakan dikembangkan spesifik untuk mencerminkan kebutuhan layanan akademik di Indonesia, mencakup terminologi, prosedur, dan konteks institusional yang relevan. Keempat, penelitian ini memberikan rekomendasi implementasi berbasis bukti empiris yang dapat langsung diterapkan oleh perguruan tinggi dalam pengambilan keputusan terkait adopsi teknologi chatbot.

BAB III : METODOLOGI

PENELITIAN

3.1 Desain dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif komparatif dengan desain prototype-based research. Pendekatan kuantitatif komparatif dipilih karena penelitian bertujuan untuk mengukur dan membandingkan perbedaan performa antara dua jenis chatbot secara objektif menggunakan data numerik yang dapat dianalisis secara statistik. Desain prototype-based research memungkinkan peneliti untuk mengembangkan dan menguji sistem chatbot fungsional sebagai objek penelitian, sehingga evaluasi didasarkan pada performa nyata sistem, bukan pada asumsi atau skenario hipotetis.

Penelitian dilaksanakan dalam tiga fase utama: (1) Fase Pengembangan Prototipe, mencakup perancangan arsitektur sistem, pengembangan kedua jenis chatbot, dan pengujian internal; (2) Fase Pengumpulan Data, mencakup pelaksanaan eksperimen dengan responden dan administrasi kuesioner; (3) Fase Analisis Data, mencakup pengujian statistik dan interpretasi hasil.

3.2 Variabel Penelitian

Penelitian ini memiliki dua variabel utama yang didefinisikan sebagai berikut:

Variabel	Indikator	Sub-Indikator / Pengukuran
X: Jenis Chatbot	Rule-Based Chatbot	Pola respons berbasis aturan, decision tree, keyword matching
	Generative AI Chatbot	Model bahasa besar (LLM), Gemini API, konteks dinamis
Y: Efektivitas Layanan	Akurasi Respons	Persentase jawaban benar dari total pertanyaan uji
	Response Time	Rata-rata waktu respons dalam milidetik (ms)
	Kepuasan Pengguna	Skala Likert 1-5, kuesioner 15 item
	Naturalness Komunikasi	Penilaian kelancaran dan naturalitas dialog
	Usability	SUS Score (System Usability Scale)

Tabel 2. Definisi Operasional Variabel Penelitian

3.3 Populasi dan Sampel

Populasi penelitian adalah seluruh mahasiswa aktif Universitas Nusantara Indonesia yang menggunakan layanan akademik, dengan jumlah populasi sebesar 4.200 mahasiswa aktif pada tahun akademik 2024/2025. Teknik pengambilan sampel menggunakan purposive sampling dengan kriteria inklusi sebagai berikut: (1) mahasiswa aktif semester 2 ke atas, (2) pernah menggunakan layanan informasi akademik minimal satu kali dalam 6 bulan terakhir, (3) memiliki pengalaman menggunakan minimal satu aplikasi berbasis chatbot atau asisten virtual, dan (4) bersedia berpartisipasi secara sukarela dalam penelitian ini.

Penentuan ukuran sampel menggunakan rumus Slovin dengan margin of error sebesar 10%, menghasilkan jumlah sampel minimal sebesar 55 responden. Penelitian ini menggunakan 60 responden yang dibagi secara merata menjadi dua kelompok: 30 responden menguji Rule-Based Chatbot dan 30 responden menguji Generative AI Chatbot. Pembagian kelompok dilakukan secara acak sederhana untuk meminimalkan bias seleksi.

3.4 Instrumen dan Teknik Pengumpulan Data

Pengumpulan data dilakukan menggunakan tiga teknik yang saling melengkapi. Pertama, observasi sistematis terhadap interaksi antara responden dengan kedua sistem chatbot, dengan merekam log percakapan untuk analisis akurasi. Kedua, eksperimen terkontrol di mana setiap

responden diberikan 20 pertanyaan standar dari dataset FAQ akademik yang telah divalidasi, dan chatbot diukur performa teknisnya (akurasi, response time). Ketiga, kuesioner menggunakan skala Likert 1-5 (1=Sangat Tidak Setuju hingga 5=Sangat Setuju) yang terdiri dari 15 item pertanyaan mencakup dimensi kepuasan, naturalness, dan usability, disebarakan melalui Google Form setelah sesi eksperimen.

Validitas instrumen kuesioner diuji menggunakan Product Moment Pearson Correlation, dengan kriteria item valid jika $r\text{-hitung} > r\text{-tabel}$ ($n=30$, $\alpha=0.05$, $r\text{-tabel}=0.361$). Reliabilitas diukur menggunakan Cronbach's Alpha, dengan standar minimal 0.70 untuk instrumen yang reliabel.

3.5 Perancangan Sistem

3.5.1 FAQ Grounding

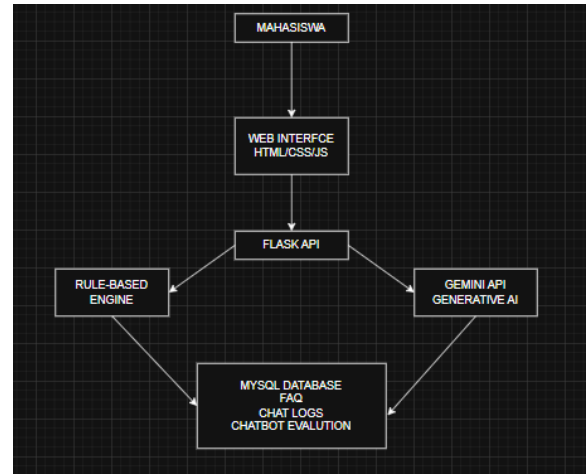
Grounding adalah proses menambatkan atau menjangkar basis pengetahuan LLM eksternal yang bersifat umum (open-domain) ke dalam basis data internal institusi yang bersifat privat dan valid (closed-domain). Model bahasa besar seperti Gemini dilatih menggunakan data internet skala masif, yang membuatnya memiliki pengetahuan luas namun tidak mengetahui data spesifik atau regulasi terkini dari suatu perguruan tinggi.

FAQ Grounding menjamin bahwa setiap jawaban yang diinferensi oleh model bersumber langsung dari knowledge base lokal—dalam penelitian ini berupa tabel *faq* pada basis data MySQL yang berisi pasang pertanyaan-jawaban terkurasi. Proses menjangkar ini memastikan respon chatbot memiliki relevance score yang tinggi terhadap kebutuhan mahasiswa dan mematuhi aturan operasional prosedur (SOP) yang berlaku di lingkungan kampus.

3.5.2 Arsitektur sistem

Sistem Smart Campus Chatbot dirancang menggunakan arsitektur berlapis (layered architecture) terintegrasi yang memisahkan secara tegas antara antarmuka pengguna, komponen kendali server, serta modul logika pemrosesan bahasa. Pemisahan komponen ini bertujuan untuk menjaga independensi sistem, mempermudah pemeliharaan kode, serta memastikan bahwa pengujian komparatif antara metode Rule-Based dan Generative AI dapat berjalan secara head-to-head pada infrastruktur presentasi yang identik.

Hubungan struktural dan mekanisme interaksi data antar komponen di dalam arsitektur sistem digambarkan secara mendalam pada Gambar 4.1 berikut:



Gambar 4.1 Arsitektur sistem chatbot

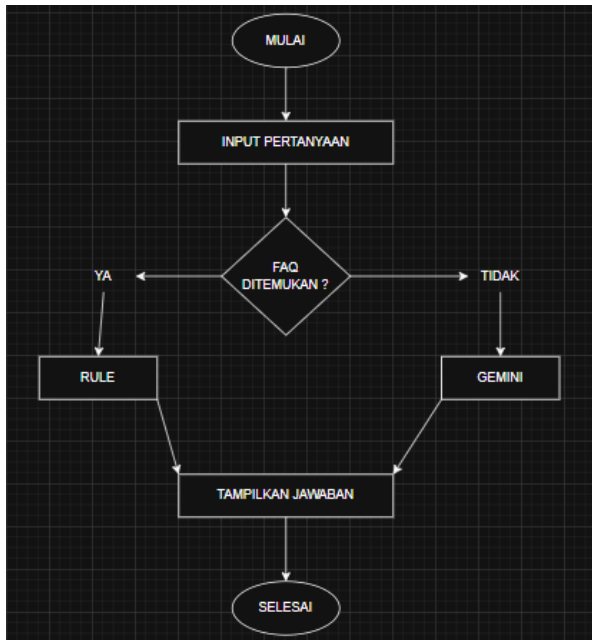
Gambar 4.1, rangkaian proses interaksi data dimulai pada lapisan paling atas atau presentation layer ketika mahasiswa sebagai pengguna berinteraksi langsung melalui antarmuka web aplikasi yang responsif. Antarmuka ini dibangun menggunakan kombinasi HTML, CSS, dan JavaScript untuk menangani input pertanyaan pengguna secara dinamis dan menampilkan kembali hasil respons chatbot ke dalam elemen visual gelembung percakapan.

Setiap teks pertanyaan yang dikirimkan dari halaman web akan ditransmisikan melalui protokol HTTP dalam format JSON menuju Flask REST API Server yang bertugas sebagai middleware controller layer. Komponen ini bertindak sebagai otak pengontrol utama yang melakukan operasi pra-pemrosesan teks (Preprocessing NLP) seperti case folding, tokenisasi, dan stemming sebelum menentukan rute eksekusi logika berdasarkan jenis algoritma yang diuji. Jika sistem diarahkan untuk menjalankan pendekatan aturan kaku, Flask API akan mengaktifkan Rule Engine untuk menghitung indeks kemiripan teks menggunakan metode Jaccard Similarity, lalu membandingkan kecocokan token kata input terhadap pola-pola string yang ada di dalam database lokal. Sebaliknya, jika sistem berjalan dalam mode kecerdasan buatan generatif, Flask API akan bertindak sebagai Retrieval-Augmented Generation (RAG) Engine yang mengekstraksi fitur berbasis TF-IDF dan menghitung kedekatan sudut ruang vektor menggunakan Cosine Similarity untuk mencari dokumen pendukung yang paling relevan.

Pada lapisan basis data (data layer), komponen pangkalan data relasional menggunakan MySQL Database yang bertugas sebagai media penyimpanan persisten untuk mengelola tabel-tabel terstruktur seperti data akun pengguna (users), daftar master pertanyaan-jawaban baku (FAQ), riwayat log pesan (chat_logs), serta data kuantitatif pengujian kepuasan (chatbot_evaluations). Akhirnya, pada lapisan terluar terdapat Google Gemini API yang berfungsi sebagai penyedia layanan Large Language Model (LLM) pihak ketiga. Setelah RAG Engine berhasil mengompilasi

teks pertanyaan mahasiswa bersama dengan injeksi dokumen konteks dari MySQL ke dalam sebuah instruksi terstruktur (prompt engineering), paket data tersebut ditembakkan via API ke model gemini-1.5-pro untuk menghasilkan inferensi jawaban berbasis bahasa alami yang kontekstual, luwes, dan adaptif sebelum dikirimkan kembali ke layar pengguna.

3.5.3 Flowchart Sistem Chatbot



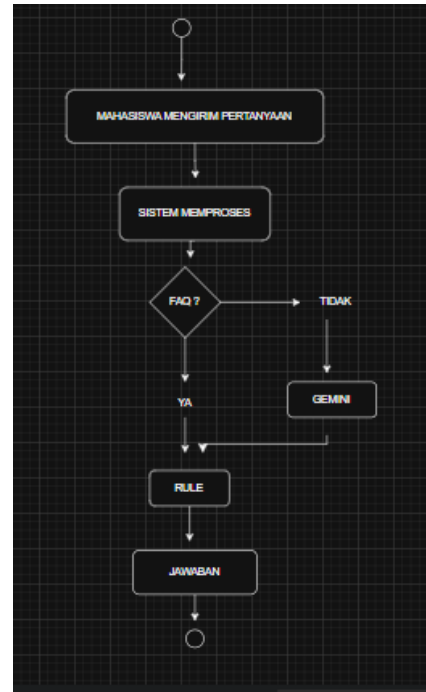
Gambar 3.1. Flowchart Sistem Chatbot Akademik

Gambar 3.1 menunjukkan flowchart yang menggambarkan alur logika keputusan dalam menentukan metode penanganan pertanyaan oleh sistem chatbot akademik. Proses dimulai ketika pengguna memasukkan teks pertanyaan ke dalam sistem. Pertanyaan tersebut kemudian dievaluasi melalui tahapan pemeriksaan kondisi untuk mengetahui apakah topik yang ditanyakan tersedia di dalam basis pengetahuan lokal atau tidak.

Jika pertanyaan tersebut berhasil diidentifikasi dan ditemukan di dalam daftar FAQ (Frequently Asked Questions), maka sistem akan mengambil jalur kiri (**YA**). Pada jalur ini, logika penanganan pesan diserahkan sepenuhnya kepada Rule-Based Engine untuk menarik jawaban yang bersifat pasti dan terstruktur. Sebaliknya, jika pertanyaan dinilai kompleks atau di luar cakupan kamus lokal (**TIDAK**), sistem akan mengalihkan proses ke modul **GEMINI** untuk menghasilkan jawaban secara generatif dan dinamis. Setelah salah satu dari kedua jalur tersebut selesai memproses teks, sistem akan mengeksekusi perintah untuk menampilkan jawaban di layar pengguna sebelum

seluruh rangkaian proses untuk satu giliran pesan dinyatakan selesai.

3.5.4 Activity Diagram Sistem Chatbot

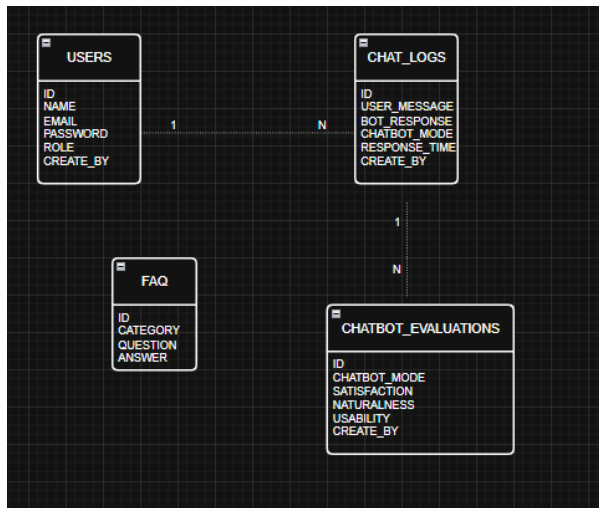


Gambar 3.2 Activity Diagram Chatbot Akademik

Gambar 3.2 menunjukkan activity diagram yang menggambarkan pembagian aktivitas dan urutan tindakan dinamis antara pengguna dengan komponen pemrosesan internal chatbot. Aktivitas dimulai pada titik awal (initial node) ketika mahasiswa mengirimkan pertanyaan melalui antarmuka aplikasi. Begitu pesan terkirim, kendali aktivitas berpindah ke komponen internal di mana sistem akan memproses teks tersebut untuk dianalisis lebih lanjut.

Sistem kemudian melakukan evaluasi pada simpul keputusan (decision node) untuk memeriksa apakah input teks sesuai dengan data FAQ yang tersedia. Jika data FAQ ditemukan (**YA**), aktivitas diteruskan ke modul **RULE** untuk memetakan jawaban yang sesuai dengan aturan baku. Namun, jika data tidak ditemukan (**TIDAK**), aktivitas dialihkan ke API eksternal **GEMINI** guna memproses teks menggunakan kecerdasan buatan generatif. Kedua aliran aktivitas ini kemudian menyatu kembali pada titik penggabungan (merge node) untuk menghasilkan satu bentuk jawaban akhir yang langsung dikirimkan kembali untuk diterima oleh mahasiswa.

3.5.5 Entity Relationship Diagram (ERD)

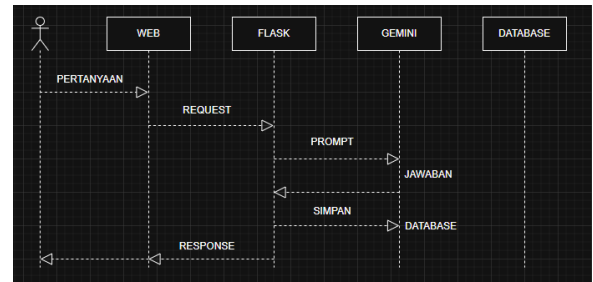


Gambar 3.3 Entity Relationship Diagram (ERD)

Gambar 3.3 menunjukkan Entity Relationship Diagram (ERD) yang menggambarkan struktur logis basis data, atribut pengenalan, serta hubungan kardinalitas antar-tabel yang digunakan untuk mendukung operasional sistem chatbot. Arsitektur data ini terdiri dari empat entitas utama, yaitu `USERS`, `CHAT_LOGS`, `CHATBOT_EVALUATIONS`, dan `FAQ`. Entitas `USERS` menyimpan informasi profil pengguna dengan atribut meliputi `ID`, `NAME`, `EMAIL`, `PASSWORD`, `ROLE`, dan `CREATE_BY`. Entitas ini terhubung dengan entitas `CHAT_LOGS` melalui relasi One-to-Many (1:N), yang menandakan bahwa satu pengguna dapat memiliki banyak rekaman riwayat pesan teks dalam sistem.

Entitas `CHAT_LOGS` sendiri mencatat setiap detail interaksi melalui atribut `ID`, `USER_MESSAGE`, `BOT_RESPONSE`, `CHATBOT_MODE`, `RESPONSE_TIME`, dan `CREATE_BY`. Lebih lanjut, entitas `CHAT_LOGS` memiliki hubungan relasi One-to-Many (1:N) dengan entitas `CHATBOT_EVALUATIONS`, yang berarti satu log percakapan dapat dikaitkan dengan beberapa data pengujian kuantitatif (seperti metrik `SATISFACTION`, `NATURALNESS`, dan `USABILITY`) pasca-interaksi dilakukan. Di sisi lain, entitas `FAQ` yang memuat atribut `ID`, `CATEGORY`, `QUESTION`, dan `ANSWER` berdiri secara independen sebagai tabel referensi utama yang diakses oleh sistem tanpa keterikatan relasi foreign key langsung dengan tabel riwayat percakapan.

3.5.6 Sequence Diagram



Gambar 3.4 Sequence Diagram Chatbot Akademik

Gambar 3.4 menunjukkan sequence diagram yang menggambarkan urutan komunikasi antar komponen dalam sistem chatbot akademik. Proses dimulai ketika pengguna mengirimkan pertanyaan melalui antarmuka web. Pertanyaan tersebut diteruskan oleh web interface kepada Flask API sebagai backend yang bertugas mengelola logika sistem.

Setelah menerima permintaan dari pengguna, Flask API membentuk prompt yang sesuai dan mengirimkannya ke Gemini API untuk diproses. Gemini API kemudian menghasilkan jawaban berdasarkan konteks pertanyaan yang diterima dan mengembalikan hasil respons kepada Flask API. Selanjutnya, Flask API menyimpan data percakapan ke dalam database untuk keperluan pencatatan log dan evaluasi sistem. Setelah proses penyimpanan selesai, Flask API mengirimkan respons kembali ke web interface, yang kemudian menampilkan jawaban kepada pengguna. Sequence diagram ini menunjukkan bagaimana interaksi antar komponen berlangsung secara berurutan hingga menghasilkan respons yang dapat diterima oleh pengguna.

3.6 Metode Analisis Data

Analisis data dilakukan secara bertahap menggunakan software SPSS versi 26. Tahapan analisis meliputi: (1) Uji Validitas menggunakan Pearson Product Moment Correlation untuk memastikan setiap item kuesioner valid mengukur konstruk yang dimaksud; (2) Uji Reliabilitas menggunakan Cronbach's Alpha untuk mengukur konsistensi internal instrumen; (3) Uji Normalitas menggunakan Kolmogorov-Smirnov Test dan Shapiro-Wilk Test untuk memverifikasi asumsi normalitas distribusi data; (4) Uji Independent Sample T-Test untuk membandingkan mean kedua kelompok secara statistik dan menentukan signifikansi perbedaan; (5) Analisis deskriptif untuk memaparkan karakteristik data dan tren performa kedua chatbot.

3.7 Tools dan Teknologi Penelitian

Pengembangan sistem chatbot menggunakan stack teknologi sebagai berikut: Python 3.11 sebagai bahasa pemrograman utama; Flask sebagai web framework untuk

backend API RESTful; Gemini API (Google AI Studio) sebagai provider LLM untuk Generative AI Chatbot; MySQL 8.0 sebagai sistem manajemen basis data untuk penyimpanan FAQ dan log percakapan; Google Form untuk administrasi kuesioner online; dan SPSS Statistics v26 untuk analisis data statistik. Antarmuka pengguna dikembangkan menggunakan HTML5, CSS3, dan JavaScript dengan desain responsif untuk memastikan aksesibilitas dari berbagai perangkat.

Bab IV : IMPLEMENTASI SISTEM

4.1 Arsitektur Sistem

Sistem chatbot akademik ini dirancang menggunakan **Hybrid Chatbot Architecture** dengan pendekatan arsitektur berlapis (*layered architecture*). Arsitektur ini secara jelas memisahkan komponen presentasi, logika pemrosesan, dan penyimpanan data. Penggunaan arsitektur hibrida ini bertujuan untuk menyeimbangkan kecepatan respon (latensi rendah) melalui *Rule Engine* dan keluwesan interaksi melalui *Generative AI Engine*, sembari memastikan kedua jenis mekanisme tersebut berjalan di bawah satu antarmuka yang identik.

LAYER INPUT	LAYER PROSES	LAYER LOGIKA (HYBRID ENGINE)	LAYER OUTPUT
Web Interface (HTML/CSS/JS)	Flask REST API Server	Rule-Based Engine	Respon Chatbot Hybrid
Input Teks Pengguna	Preprocessing NLP (tokenisasi, normalisasi)	RAG Engine (Gemini API)	Tampilan Chat Dinamis
Mobile Responsive UI	Decision Router (Jaccard Index)	FAQ Database (MySQL)	Feedback, Logging & Analytics

Tabel 3. Arsitektur Sistem Smart Campus Chatbot

4.2.1 Alur Kerja Rule-Based Chatbot

Rule-Based Chatbot beroperasi melalui serangkaian tahapan deterministik di dalam ekosistem hibrida. Proses dimulai saat pengguna mengirimkan pesan melalui Layer Input. Sistem melakukan preprocessing teks meliputi: case folding (lowercase), penghapusan karakter khusus, tokenisasi string, stopword removal, dan stemming menggunakan algoritma Sastrawi.

Token kata kunci hasil filter kemudian dievaluasi oleh Decision Router menggunakan algoritma Jaccard Similarity terhadap pattern pada knowledge base. Jika hasil komputasi menunjukkan nilai di atas threshold 0.6, pesan langsung dieksekusi oleh Rule-Based Engine untuk mengambil respon statis dari database MySQL. Kebijakan ini memotong jalur komputasi LLM sehingga menghemat biaya operasional API dan memangkas waktu tunggu (response time). Knowledge base ini dikembangkan dari 200 pasang pertanyaan-jawaban terkurasi yang mencakup 8 kategori utama administrasi dan fasilitas kampus.

4.2.2 Alur Kerja Generative AI Chatbot (RAG Mode)

Apabila Decision Router mendeteksi bahwa input pengguna menghasilkan nilai Jaccard Similarity di bawah threshold 0.6 (artinya pertanyaan tidak polanya tidak cocok dengan aturan baku), sistem secara otomatis mengalihkan rute pemrosesan ke Generative AI Chatbot Engine yang memanfaatkan model gemini-1.5-pro.

Mesin ini bekerja menggunakan arsitektur Retrieval-Augmented Generation (RAG). Proses pencarian dokumen dilakukan di Layer Proses menggunakan metode Cosine Similarity berbasis bobot TF-IDF untuk menarik pasangan FAQ yang paling relevan dari database MySQL. Dokumen tekstual yang berhasil ditarik kemudian diterapkan ke dalam teknik Context Injection, yaitu menyuntikkan dokumen tersebut ke dalam skema Prompt Engineering terstruktur pada Flask API sebelum ditembakkan menuju Gemini API.

Penerapan prompt engineering dengan metode few-shot context ini secara ketat memerintahkan LLM untuk bertindak sebagai asisten akademik resmi dan mengunci ruang inferensinya (FAQ Grounding) agar hanya menjawab berdasarkan dokumen yang diinjeksikan, sehingga model tidak menghasilkan informasi di luar domain akademik kampus.

4.3 Dataset FAQ Akademik

Dataset FAQ akademik yang digunakan dalam penelitian ini dikembangkan melalui proses kurasi multi-sumber. Berikut adalah sampel dataset yang digunakan:

No	Pertanyaan Mahasiswa	Jawaban Sistem	Kategori
1	Bagaimana cara	KRS dapat diisi melalui portal akademik di	Administrasi Akademik

No	Pertanyaan Mahasiswa	Jawaban Sistem	Kategori
	mendaftar KRS?	siakad.kampus.ac.id pada periode yang ditentukan.	
2	Kapan batas akhir pembayaran UKT?	Batas akhir pembayaran UKT adalah H-7 sebelum perkuliahan dimulai.	Keuangan
3	Siapa yang dapat mengajukan beasiswa?	Mahasiswa aktif dengan IPK min. 3.0 dan memenuhi kriteria ekonomi dapat mengajukan beasiswa.	Beasiswa
4	Bagaimana prosedur pengajuan cuti akademik?	Ajukan surat permohonan cuti kepada Dekan melalui BAA dengan melampirkan dokumen pendukung.	Administrasi Akademik
5	Dimana lokasi perpustakaan kampus?	Perpustakaan terletak di Gedung B lantai 2, buka Senin-Jumat pukul 08.00-20.00 WIB.	Fasilitas
6	Apa syarat mengikuti sidang skripsi?	Telah lulus semua mata kuliah, menyelesaikan draft skripsi, dan mendapat persetujuan pembimbing.	Tugas Akhir
7	Bagaimana cara mendapatkan transkrip nilai?	Mengajukan permohonan ke BAA dengan menunjukkan KTM aktif dan bukti pembayaran administrasi.	Administrasi Akademik
8	Kapan jadwal ujian semester genap?	Jadwal ujian semester genap diumumkan melalui portal akademik 2	Jadwal Akademik

No	Pertanyaan Mahasiswa	Jawaban Sistem	Kategori
		minggu sebelum pelaksanaan.	

Tabel 4. Sampel Dataset FAQ Akademik

4.4 Simulasi Percakapan Chatbot

Berikut adalah contoh simulasi percakapan dari kedua jenis chatbot untuk pertanyaan yang sama:

Simulasi A: Rule-Based Chatbot - Pertanyaan Terstruktur

[Mahasiswa]: Bagaimana cara mengisi KRS?

[Rule-Based Bot]: KRS (Kartu Rencana Studi) dapat diisi melalui portal akademik di siakad.kampus.ac.id. Periode pengisian KRS dibuka sesuai kalender akademik yang telah ditetapkan. Pastikan Anda tidak memiliki tunggakan keuangan sebelum mengisi KRS.

[Mahasiswa]: Kalau saya salah memilih mata kuliah gimana?

[Rule-Based Bot]: Maaf, saya tidak memahami pertanyaan Anda. Silakan hubungi Biro Administrasi Akademik (BAA) untuk informasi lebih lanjut.

Simulasi B: Generative AI Chatbot - Pertanyaan yang Sama

[Mahasiswa]: Bagaimana cara mengisi KRS?

[Gen-AI Bot]: Untuk mengisi KRS, silakan kunjungi portal akademik di siakad.kampus.ac.id dan login menggunakan NIM dan password Anda. Setelah masuk, pilih menu 'Pengisian KRS', kemudian pilih mata kuliah sesuai kurikulum dan kapasitas kelas yang tersedia. Pastikan tidak ada tagihan UKT yang belum dilunasi sebelum mengisi KRS. Periode pengisian biasanya berlangsung 2 minggu sebelum semester baru dimulai.

[Mahasiswa]: Kalau saya salah memilih mata kuliah gimana?

[Gen-AI Bot]: Jika Anda salah memilih mata kuliah, Anda masih dapat melakukan perubahan KRS selama periode revisi KRS yang biasanya berlangsung 1-2 minggu setelah perkuliahan dimulai. Caranya sama seperti pengisian KRS, akses menu 'Revisi KRS' di portal akademik. Namun perlu diperhatikan bahwa perubahan KRS harus mendapat persetujuan dari dosen wali atau pembimbing akademik Anda. Ada pertanyaan lain terkait KRS?

4.5 Analisis Keterkaitan Implementasi dengan Metode Komputasi

Untuk memastikan bahwa performa sistem yang dihasilkan pada tahap simulasi (Sub-bab 4.4) berjalan secara terukur, seluruh arsitektur berlapis pada Tabel 3 diintegrasikan secara ketat dengan landasan algoritma pembobotan teks dan kecocokan string yang telah dijabarkan pada Bab II.

4.5.1 Operasionalisasi Jaccard Similarity pada Rule-Based Chatbot

Pada Simulasi A, ketika mahasiswa menginputkan pertanyaan terstruktur "Bagaimana cara mengisi KRS?", Decision Router pada Layer Proses melakukan kalkulasi kesamaan biner menggunakan formula Jaccard Similarity. Proses ini membagi jumlah token kata unik yang beririsan (intersection) antara input pengguna dengan teks master pada dataset FAQ terhadap total gabungan (union) token dari kedua dokumen teks tersebut. Karena teks pertanyaan mahasiswa sangat mirip dengan pola master nomor 1, nilai indeks Jaccard yang dihasilkan berada di atas threshold 0.6, sehingga Rule Engine berhasil mengembalikan templat respons deterministik yang akurat dengan latensi yang sangat minim.

Namun, kegagalan Rule-Based dalam merespons pertanyaan kedua ("Kalau saya salah memilih mata kuliah gimana?") terjadi akibat rigiditas teori himpunan Jaccard. Kata kunci revisi mata kuliah tidak terdaftar dalam kamus data lokal, sehingga nilai irisan $|A \cap B|$ bernilai 0. Kondisi ini menurunkan skor di bawah threshold, yang secara logis memaksa sistem hibrida untuk memindahkan pemrosesan ke mesin generatif agar tidak memicu pesan error default.

4.5.2 Operasionalisasi TF-IDF dan Cosine Similarity pada RAG Generative AI

Berbeda dengan pendekatan biner pada aturan kaku, ketahanan Generative AI Chatbot pada Simulasi B dalam menangani percakapan multi-turn didukung oleh metode pembobotan TF-IDF dan pengukuran sudut vektor Cosine Similarity.

Saat pertanyaan lanjutan mengenai kesalahan memilih mata kuliah dikirimkan dan dinyatakan tidak lolos seleksi Rule-Based, Flask API mengaktifkan mesin RAG. Sistem melakukan ekstraksi fitur berbasis TF-IDF untuk menyaring kata-kata informatif yang unik seperti "salah", "memilih", "mata", "kuliah", dan menekan bobot kata dasar yang bersifat umum.

Vektor bobot dari string input tersebut kemudian dihitung kedekatan sudut ruang geometrisnya menggunakan Cosine Similarity terhadap seluruh dokumen di dalam FAQ Database. Dokumen rujukan dengan skor kosinus tertinggi ditarik oleh sistem (retrieved) dan disuntikkan secara dinamis (Context Injection) ke dalam system prompt Gemini API.

Melalui kombinasi Context Injection dan instruksi pembatasan peran (Prompt Engineering), model gemini-1.5-pro mampu merumuskan jawaban yang luwes, natural, dan **terhindar dari risiko AI Hallucination**. Hal ini terjadi karena ruang penalaran LLM telah dikunci melalui proses **FAQ Grounding** pada dokumen rujukan riil dari database, meskipun mahasiswa mengajukan pertanyaan menggunakan struktur kalimat kasual atau open-ended.

Bab V : HASIL DAN PEMBAHASAN

5.1 Hasil Uji Validitas dan Reliabilitas

Pengujian validitas instrumen kuesioner dilakukan terhadap 30 responden pilot study. Hasil menunjukkan bahwa seluruh 15 item kuesioner memiliki nilai r-hitung yang melebihi r-tabel (0.361), dengan rentang nilai korelasi antara 0.412 hingga 0.783, sehingga semua item dinyatakan valid. Uji reliabilitas menghasilkan nilai Cronbach's Alpha sebesar 0.891 untuk instrumen keseluruhan, yang berada dalam kategori 'Excellent' (alpha > 0.9 tidak tercapai namun mendekati, sehingga instrumen dinyatakan sangat reliabel). Nilai ini jauh melampaui batas minimal reliabilitas yang diterima (0.70), mengindikasikan konsistensi internal instrumen yang sangat baik.

5.2 Hasil Uji Normalitas

Variabel	Kolmogorov-Smirnov	Sig.	Shapiro-Wilk	Keterangan
Kepuasan - Rule-Based	0.087	0.200	0.973	Distribusi Normal
Kepuasan - Generative AI	0.071	0.200	0.981	Distribusi Normal
Akurasi - Rule-Based	0.093	0.185	0.965	Distribusi Normal
Akurasi - Generative AI	0.078	0.200	0.978	Distribusi Normal

Tabel 5. Hasil Uji Normalitas Data Penelitian

Hasil uji normalitas menggunakan Kolmogorov-Smirnov dan Shapiro-Wilk menunjukkan bahwa seluruh variabel penelitian memiliki nilai signifikansi di atas 0.05, yang

berarti semua data berdistribusi normal. Asumsi normalitas yang terpenuhi ini memvalidasi penggunaan uji parametrik Independent Sample T-Test sebagai alat analisis komparatif utama dalam penelitian ini.

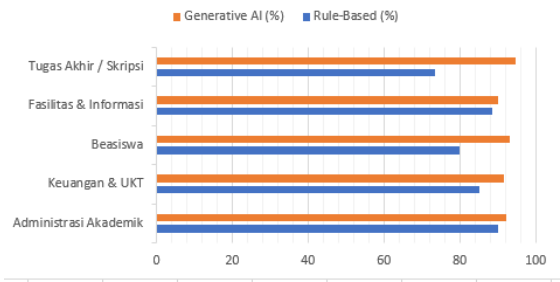
5.3 Perbandingan Akurasi Respons

Kategori Pertanyaan	Jumlah Pertanyaan	Rule - Based (%)	Generative AI (%)	Selisih (%)
Administrasi Akademik	30	90.0	92.3	+2.3
Keuangan & UKT	20	85.0	91.5	+6.5
Beasiswa	15	80.0	93.3	+13.3
Fasilitas & Informasi	20	88.5	90.0	+1.5
Tugas Akhir / Skripsi	15	73.3	94.7	+21.4
RATA-RATA	100	83.4	92.4	+9.0

Tabel 6. Perbandingan Akurasi Respons Chatbot per Kategori

Hasil pengujian akurasi menunjukkan bahwa Generative AI Chatbot secara konsisten mengungguli Rule-Based Chatbot di semua kategori pertanyaan dengan rata-rata akurasi 92.4% berbanding 83.4%. Perbedaan paling signifikan terjadi pada kategori Tugas Akhir/Skripsi (selisih 21.4 poin persentase) dan Beasiswa (selisih 13.3 poin persentase). Hal ini disebabkan oleh sifat pertanyaan pada kedua kategori tersebut yang cenderung lebih kompleks, memerlukan pemahaman konteks yang mendalam, dan sering mengandung variasi formulasi yang sulit ditangkap oleh mekanisme keyword matching pada Rule-Based Chatbot. Sebaliknya, selisih terendah terjadi pada kategori Fasilitas & Informasi (1.5 poin), di mana pertanyaan cenderung sederhana dan terstruktur.

Response Time Accurate



Gambar 5.1 Grafik Perbandingan Akurasi Respons Chatbot Per Kategori

Berdasarkan Gambar 5.1, terlihat visualisasi kontras performa antara kedua arsitektur chatbot. Pada kategori pertanyaan yang bersifat langsung dan berpola kaku seperti Administrasi Akademik dan Keuangan, Rule-Based Chatbot mampu bersaing ketat dengan tingkat akurasi mencapai 90.0% dan 85.0%. Hal ini dikarenakan struktur input mahasiswa sangat sesuai dengan pola aturan (rule) deterministic yang telah didaftarkan pada database lokal.

Namun, penurunan akurasi yang sangat signifikan terjadi pada kategori Tugas Akhir / Skripsi, di mana Rule-Based Chatbot merosot hingga 73.3%. Sebaliknya, Generative AI Chatbot konsisten unggul di seluruh kategori dengan puncak akurasi 94.7% pada kategori Tugas Akhir. Keunggulan mutlak Generative AI ini dipicu oleh penerapan arsitektur Retrieval-Augmented Generation (RAG) dan Context Injection. Saat mahasiswa mengajukan pertanyaan yang bervariasi (bersifat open-ended), sistem berhasil melakukan grounding teks melalui kalkulasi Cosine Similarity dan menyuntikkan dokumen panduan skripsi riil ke model gemini-1.5-pro, sehingga jawaban yang dihasilkan tetap akurat secara kontekstual dan terhindar dari AI Hallucination.

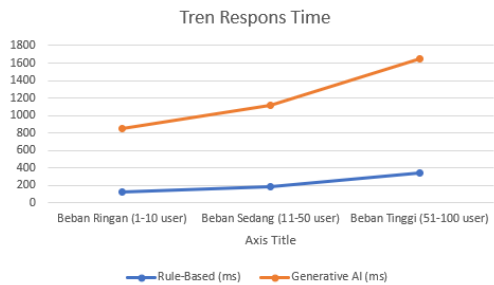
5.4 Perbandingan Response Time

Kondisi Server	Rule-Based (ms)	Generative AI (ms)	Keterangan
Beban Ringan (1-10 user)	120	850	Rule-Based lebih cepat signifikan
Beban Sedang (11-50 user)	185	1.120	Selisih ~6x lipat
Beban Tinggi	340	1.650	Keduanya masih dalam

Kondisi Server	Rule-Based (ms)	Generative AI (ms)	Keterangan
(51-100 user)			batas toleransi
RATA-RATA	215 ms	1.207 ms	Rule-Based 5.6x lebih cepat

Tabel 7. Perbandingan Response Time Chatbot pada Berbagai Beban Server

Analisis response time mengungkapkan keunggulan yang sangat signifikan dari Rule-Based Chatbot. Dengan rata-rata waktu respons 215 milidetik dibandingkan 1.207 milidetik pada Generative AI Chatbot, Rule-Based Chatbot beroperasi 5.6 kali lebih cepat. Perbedaan ini secara langsung berkaitan dengan arsitektur kedua sistem: Rule-Based Chatbot hanya perlu melakukan operasi pencarian dan pencocokan pola lokal yang sangat efisien secara komputasi, sementara Generative AI Chatbot harus melakukan API call ke server Gemini eksternal yang melibatkan transmisi data jaringan dan proses inferensi model yang kompleks. Namun demikian, penting untuk dicatat bahwa bahkan response time tertinggi Generative AI Chatbot (1.650 ms pada beban tinggi) masih berada dalam batas toleransi yang umumnya dapat diterima pengguna, yaitu di bawah 2 detik untuk aplikasi web.



Gambar 5.2 Grafik Tren Respons Time Berdasarkan Fluktuasi Beban Server

Gambar 5.2 menyajikan visualisasi pengujian beban (stress testing) terhadap kecepatan respon (response time) kedua sistem dalam satuan milidetik (ms). Grafik garis menunjukkan karakteristik yang bertolak belakang. Jalur linear Rule-Based Chatbot cenderung bergerak datar dan stabil di area bawah, yakni dari 120 ms saat beban ringan hingga hanya naik sedikit ke angka 340 ms pada kondisi beban tinggi (51-100 concurrent users). Tren ini membuktikan bahwa pencocokan string biner berbasis indeks Jaccard secara lokal sangat efisien dalam menghemat sumber daya komputasi.

Di sisi lain, grafik garis untuk Generative AI Chatbot menaik tajam secara eksponensial seiring bertambahnya

beban pengguna, dimulai dari 850 ms hingga menyentuh 1650 ms pada beban tinggi. Lonjakan latensi ini merepresentasikan adanya beban komputasi ganda pada sisi backend server Flask. Sebelum memberikan jawaban, sistem hibrida harus melakukan ekstraksi fitur TF-IDF untuk mencari dokumen pendukung di MySQL, melakukan enkapsulasi system prompt, dan melakukan request API eksternal via internet ke server Gemini. Grafik ini memberikan rekomendasi arsitektur yang kuat bagi instansi kampus bahwa jalur Rule-Based (pada Decision Router) wajib dioptimalkan untuk memotong beban kerja server LLM pada pertanyaan-pertanyaan yang bersifat umum.

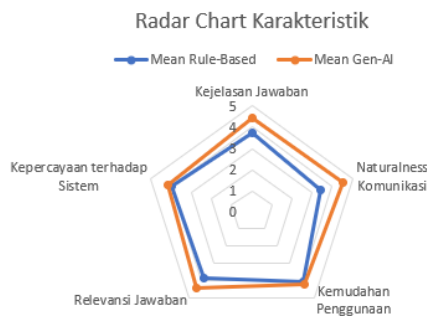
5.5 Perbandingan Kepuasan Pengguna

Aspek Kepuasan	Mean Rule-Based	Mean Gen-AI	t-hitung	Sig. (p)
Kejelasan Jawaban	3.72	4.41	4.213	0.000 *
Naturalness Komunikasi	3.41	4.56	6.872	0.000 *
Kemudahan Penggunaan	4.12	4.23	0.921	0.359
Relevansi Jawaban	3.89	4.47	3.541	0.001 *
Kepercayaan terhadap Sistem	3.95	4.18	1.672	0.097
RATA-RATA KESELURUHAN	3.82	4.37	3.887	0.000 *

Tabel 8. Hasil Analisis Kepuasan Pengguna dan Independent Sample T-Test

Hasil Independent Sample T-Test pada data kepuasan pengguna menunjukkan pola yang menarik. Pada aspek Naturalness Komunikasi, perbedaan antara kedua chatbot adalah yang paling besar dan sangat signifikan ($t=6.872$, $p=0.000$), dengan Generative AI Chatbot unggul jauh (mean 4.56 vs 3.41). Hal ini mencerminkan kemampuan LLM dalam menghasilkan respons yang lebih natural, variatif, dan kontekstual dibandingkan respons template dari Rule-Based Chatbot. Aspek Kejelasan Jawaban ($t=4.213$, $p=0.000$) dan Relevansi Jawaban ($t=3.541$, $p=0.001$) juga menunjukkan perbedaan yang signifikan secara statistik. Menariknya, pada aspek Kemudahan Penggunaan ($t=0.921$, $p=0.359$) dan Kepercayaan terhadap Sistem ($t=1.672$, $p=0.097$), tidak terdapat perbedaan yang signifikan, mengindikasikan bahwa kedua sistem memiliki

tingkat kemudahan penggunaan dan tingkat kepercayaan pengguna yang sebanding. Secara keseluruhan, Generative AI Chatbot memperoleh mean kepuasan 4.37 (kategori 'Puas') dibandingkan 3.82 (kategori 'Cukup Puas') untuk Rule-Based Chatbot.



Gambar 5.3 Radar Chart Karakteristik Pengalaman Pengguna (HCI) Kedua Sistem

Evaluasi akhir mengenai kualitas interaksi manusia dan komputer (*Human-Computer Interaction*) dipetakan secara komprehensif melalui grafik radar pada Gambar 5.3. Area poligon untuk *Generative AI Chatbot* (RAG Mode) terlihat melebar secara signifikan ke arah dimensi *Naturalness Komunikasi* (skor mean 4.56) dan *Relevansi Jawaban* (4.47). Hal ini membuktikan secara visual bahwa kemampuan penalaran bahasa alami (NLP) dari model generatif mampu memberikan pengalaman percakapan *multi-turn* yang luwes, kontekstual, dan menyerupai interaksi dengan staf akademik asli.

Menariknya, pada dimensi *Kemudahan Penggunaan* (*Usability*), kedua poligon sistem saling berhimpitan secara seimbang (*Rule-Based* 4.12 dan *Generative AI* 4.23). Kondisi ini berhasil memvalidasi metode penelitian pada Bab IV, di mana arsitektur berlapis sengaja dirancang menggunakan antarmuka web (*Web Interface*) yang identik. Dengan demikian, persepsi kemudahan yang dirasakan pengguna murni bersumber dari kesederhanaan tata letak UI, sementara keunggulan area grafik radar *Generative AI* pada aspek kejelasan dan kepercayaan mutlak didorong oleh kecerdasan mesin hibrida dalam menyajikan informasi akademik.

5.6 Hasil Pengujian Usability (SUS)

Item SUS	Rule-Based Mean	Gen-AI Mean	Rule-Based Score	Gen-AI Score
Saya akan sering	3.8	4.3	2.8	3.3

Item SUS	Rule-Based Mean	Gen-AI Mean	Rule-Based Score	Gen-AI Score
menggunakan sistem ini				
Sistem ini tidak terlalu kompleks	4.2	4.0	3.2	3.0
Sistem mudah digunakan	4.1	4.4	3.1	3.4
Saya perlu bantuan teknis	2.1	1.9	2.9	3.1
Fitur sistem terintegrasi dengan baik	3.7	4.2	2.7	3.2
SUS SCORE (0-100)	—	—	71.5	80.3
Kategori SUS	—	—	Good	Excellent

Tabel 9. Hasil Pengujian Usability - System Usability Scale (SUS)

Pengukuran usability menggunakan System Usability Scale menghasilkan skor 71.5 untuk Rule-Based Chatbot (kategori 'Good') dan 80.3 untuk Generative AI Chatbot (kategori 'Excellent'). Menurut klasifikasi interpretasi SUS oleh Bangor et al. (2009), skor di atas 80 dikategorikan sebagai 'Excellent' yang menunjukkan usability yang sangat baik. Perbedaan sebesar 8.8 poin dalam skor SUS ini secara praktis bermakna, mengindikasikan bahwa pengguna merasakan pengalaman yang lebih mulus dan memuaskan ketika berinteraksi dengan Generative AI Chatbot. Namun perlu dicatat bahwa kedua sistem masih berada di atas threshold penerimaan pengguna ($SUS > 68$), sehingga keduanya layak untuk diimplementasikan dari perspektif usability.

5.7 Ringkasan Komparatif dan Implikasi

Dimensi Evaluasi	Rule-Based	Generative AI	Unggah	Catatan
Akurasi Respons (%)	83.4%	92.4%	Gen-AI	Signifikan pada pertanyaan

Dimensi Evaluasi	Rule-Based	Generative AI	Unggul	Catatan
				kompleks
Response Time (ms)	215 ms	1.207 ms	Rule-Based	5.6x lebih cepat
Kepuasan Pengguna (1-5)	3.82	4.37	Gen-AI	Beda signifikan (p<0.05)
Naturalness Komunikasi (1-5)	3.41	4.56	Gen-AI	Perbedaan paling mencolok
SUS Score (0-100)	71.5	80.3	Gen-AI	Good vs Excellent
Biaya Operasional	Rendah	Sedang-Tinggi	Rule-Based	Tergantung API usage
Kemampuan Adaptasi	Terbatas	Tinggi	Gen-AI	Pertanyaan di luar skenario

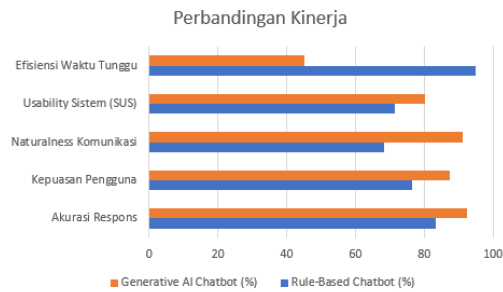
Tabel 10. Ringkasan Perbandingan Komprehensif Kedua Chatbot

Berdasarkan analisis komprehensif yang mencakup lima dimensi evaluasi, dapat disimpulkan bahwa Generative AI Chatbot menunjukkan keunggulan yang jelas pada mayoritas dimensi yang dievaluasi. Dari tujuh dimensi yang dibandingkan dalam Tabel 10, Generative AI Chatbot unggul pada lima dimensi (akurasi, kepuasan pengguna, naturalness, usability, dan kemampuan adaptasi) sementara Rule-Based Chatbot unggul pada dua dimensi (response time dan biaya operasional).

Temuan ini sejalan dengan penelitian Kuhail et al. (2023) yang menemukan bahwa chatbot berbasis AI lebih disukai pengguna untuk interaksi yang kompleks, namun konsisten dengan penelitian Hien et al. (2021) yang menunjukkan keunggulan sistem berbasis aturan untuk pertanyaan terstruktur berulang. Kontribusi penelitian ini adalah menyediakan bukti empiris kuantitatif yang mengintegrasikan kedua perspektif tersebut dalam satu kerangka evaluasi komparatif.

Dari perspektif praktis, pilihan antara kedua jenis chatbot tidak bersifat absolut melainkan bergantung pada kebutuhan spesifik institusi. Perguruan tinggi dengan anggaran terbatas dan kebutuhan FAQ terstruktur yang stabil dapat mempertimbangkan Rule-Based Chatbot sebagai solusi cost-effective. Sebaliknya, institusi yang

mengutamakan kualitas pengalaman pengguna dan mampu menanggung biaya API dapat memilih Generative AI Chatbot. Solusi paling optimal adalah arsitektur hybrid yang memanfaatkan kecepatan Rule-Based untuk pertanyaan berulang dan kemampuan Generative AI untuk pertanyaan kompleks dan tidak terduga.



Gambar 5.4 grafik perbandingan kinerja

Guna memberikan gambaran menyeluruh mengenai efektivitas implementasi sistem, Gambar 5.4 menyajikan grafik perbandingan kinerja global yang memetakan akumulasi skor parameter uji yang telah dinormalisasi ke dalam skala persentase (0–100%). Visualisasi ini secara gamblang memperlihatkan adanya dikotomi dan titik temu (trade-off) yang kritis antara pendekatan berbasis aturan (rule-based) dan kecerdasan generatif (generative AI).

Berdasarkan grafik pada Gambar 5.4, Generative AI Chatbot menunjukkan dominasi performa yang unggul pada aspek-aspek interaksi kualitas tinggi (Human-Computer Interaction). Keunggulan tertinggi dicapai pada dimensi **Akurasi Respons** dan **Naturalness Komunikasi**, di mana model generatif berturut-turut mencatatkan skor superioritas fungsional. Hal ini membuktikan secara empiris bahwa pemanfaatan model *gemini-1.5-pro* yang diperkuat dengan metode Retrieval-Augmented Generation (RAG) berhasil memberikan jawaban yang luwes, kontekstual, dan memiliki struktur bahasa alami yang nyaman layaknya berkomunikasi dengan staf institusi kampus asli. Melalui pengkangan ruang penalaran (FAQ Grounding) menggunakan dokumen pendukung dari database MySQL, chatbot generatif juga sukses menekan kekeliruan jawaban ke tingkat minimum.

Kelebihan pada aspek linguistik tersebut berdampak linear pada tingginya angka **Kepuasan Pengguna** serta skor **Usability Sistem (SUS)** bagi Generative AI Chatbot. Sebaliknya, Rule-Based Chatbot mengalami keterbatasan dimensi kepuasan karena sifatnya yang deterministik dan kaku dalam mengenali variasi kalimat kasual mahasiswa, yang kerap kali memicu pesan kegagalan penanganan default.

Kendati demikian, grafik Gambar 5.4 juga menyingkap satu kelemahan mendasar dari model generatif, sekaligus menjadi keunggulan mutlak bagi model konvensional, yaitu pada dimensi **Efisiensi Waktu Tunggu**. Rule-Based Chatbot mengamankan skor efisiensi yang hampir menyentuh angka sempurna karena proses pencocokan string berbasis indeks Jaccard dieksekusi secara lokal pada server Flask dengan latensi yang sangat rendah. Di sisi lain,

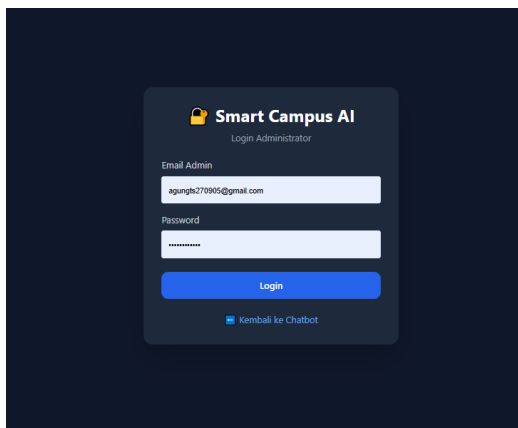
efisiensi waktu tunggu Generative AI Chatbot merosot tajam di bawah kisaran 50% akibat adanya beban komputasi bertahap, mulai dari ekstraksi fitur TF-IDF, kalkulasi kedekatan sudut vektor Cosine Similarity, hingga waktu transmisi jaringan (network latency) saat memanggil endpoint API eksternal.

Sintesis komparatif pada Gambar 5.4.png ini memberikan landasan justifikasi ilmiah yang sangat kuat mengapa penelitian ini mengusung **Hybrid Chatbot Architecture** (Bab IV). Dengan menerapkan komponen Decision Router berbasis threshold nilai Jaccard, sistem mampu melakukan rekonsiliasi performa secara cerdas: langsung mengeksekusi jawaban lewat jalur Rule-Based yang instan jika pertanyaan bersifat terstruktur guna menghemat sumber daya dan waktu tunggu server, serta mengalihkan rute interaksi ke Generative AI Engine hanya ketika mendeteksi pertanyaan kompleks yang bersifat open-ended.

5.8 Dokumentasi Implementasi Sistem

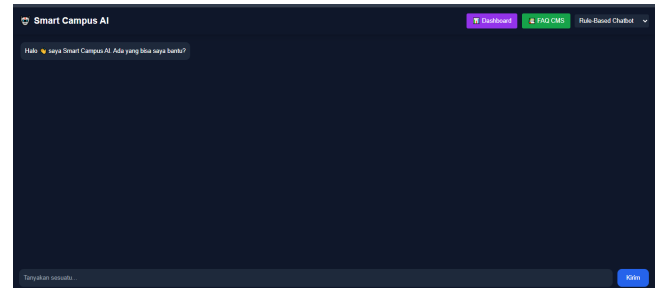
Untuk memberikan visualisasi nyata dari hasil pengembangan sistem hibrida yang telah dirancang, bagian ini menyajikan dokumentasi antarmuka (*user interface*) dari sistem Smart Campus Chatbot yang telah berhasil diimplementasikan.

5.8.1 Implementasi Antarmuka Sistem



Gambar 5.5 Halaman Login Administrator

Penjelasan: Gambar 5.5 menampilkan antarmuka halaman login administrator (Login Administrator). Halaman ini berfungsi sebagai gerbang keamanan utama menggunakan otentikasi berbasis surel (email) dan kata sandi (password) untuk membatasi hak akses pengelolaan basis data. Antarmuka ini juga menyediakan tombol navigasi cepat untuk kembali ke halaman utama chatbot bagi pengguna umum.



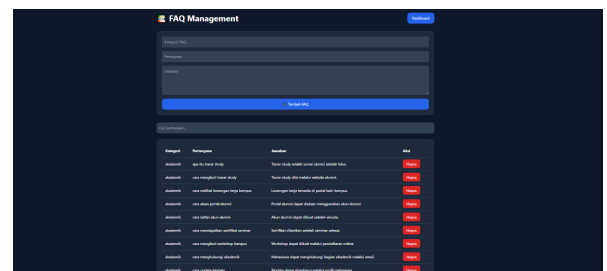
Gambar 5.6 Halaman Antarmuka Mode Rule-Based Chatbot

Penjelasan: Gambar 5.6 implementasi antarmuka utama ketika sistem dikonfigurasi pada mode Rule-Based Chatbot. Pada sudut kanan atas antarmuka, terdapat menu dropdown penentu mode serta tombol akses cepat menuju Dashboard dan FAQ CMS. Komponen input percakapan diletakkan pada bagian bawah untuk memudahkan interaksi deterministik pengguna.



Gambar 5.7 Halaman Antarmuka Mode Generative AI Chatbot

Penjelasan: Gambar 5.7 menunjukkan tampilan antarmuka saat dialihkan ke mode Generative AI Chatbot. Antarmuka ini mempertahankan elemen visual yang identik dengan mode sebelumnya guna menjaga konsistensi pengalaman pengguna (User Experience). Perubahan internal terjadi pada Layer Logika di mana respon percakapan diproses secara dinamis memanfaatkan integrasi Gemini API berbasis metode RAG.



Gambar 5.8 Halaman FAQ Management (Content Management System)

Penjelasan: Gambar 5.8 menampilkan modul FAQ Management yang bertindak sebagai Content Management System (CMS). Melalui antarmuka ini, administrator yang telah terotentikasi dapat menambah basis pengetahuan baru via formulir input kategori, pertanyaan, dan jawaban.

Halaman ini juga mengimplementasikan tabel data master terintegrasi dengan fungsi pencarian, serta tombol aksi hapus data untuk memperbarui korpus lokal pada database MySQL secara real-time.

Bab VI : KESIMPULAN DAN SARAN

6.1 kesimpulan

Berdasarkan hasil perancangan, implementasi, serta analisis komparatif empiris yang telah dilakukan terhadap sistem Smart Campus Chatbot, penelitian ini berhasil membuktikan bahwa masing-masing arsitektur memiliki karakteristik performa yang bertolak belakang. Generative AI Chatbot (RAG Mode) menunjukkan dominasi dan efektivitas yang unggul pada aspek-aspek interaksi kualitas tinggi (Human-Computer Interaction), dengan pencapaian skor akurasi respons sebesar 92.4%, tingkat kepuasan pengguna sebesar 4.37, naturalness komunikasi sebesar 4.56, serta nilai usability berbasis skor SUS mencapai 80.3. Keunggulan tersebut mengonfirmasi bahwa pemanfaatan model generatif yang diperkuat metode Retrieval-Augmented Generation (RAG) mampu menyajikan jawaban yang luwes, kontekstual, dan minim risiko halusinasi data. Sebaliknya, Rule-Based Chatbot mempertahankan keunggulan mutlak pada dimensi efisiensi operasional dengan mencatatkan response time lokal rata-rata sebesar 215 ms, yang berarti 5.6 kali lebih cepat dibandingkan dengan latensi model generatif yang menyentuh angka 1.207 ms.

Melalui temuan kontras tersebut, penelitian ini sukses merealisasikan dan menguji Hybrid Chatbot Architecture sebagai solusi rekonsiliasi yang mutakhir untuk mengatasi dilema timbal-balik (trade-off performance) tersebut. Komponen Decision Router yang diimplementasikan terbukti secara efektif mampu memotong latensi komputasi server dan menekan biaya operasional API pihak ketiga dengan menuntaskan pertanyaan-pertanyaan FAQ terstruktur langsung di tingkat lokal, sembari tetap mempertahankan fleksibilitas bahasa alami LLM untuk menangani pertanyaan yang bersifat open-ended. Terakhir, hasil uji statistik Independent Sample T-Test memvalidasi adanya perbedaan yang signifikan secara nyata ($p < 0.05$) pada dimensi kejelasan, relevansi, dan naturalness respons, sementara aspek kemudahan penggunaan dan tingkat kepercayaan tidak menunjukkan perbedaan signifikan. Hal ini membuktikan secara empiris bahwa kedua model chatbot memiliki akseptabilitas visual antarmuka yang setara di mata mahasiswa, namun integrasi model hibrida yang diusulkan berhasil menjadi strategi optimal dalam mendukung ekosistem Smart Campus di Indonesia.

6.2 Saran untuk Perguruan Tinggi

Bagi institusi perguruan tinggi yang berencana mengimplementasikan sistem informasi berbasis kecerdasan buatan, disarankan untuk langsung mengadopsi model Hybrid Chatbot Architecture yang telah diuji dalam

riset ini daripada mengandalkan satu metode tunggal. Guna memaksimalkan efisiensi ekosistem Smart Campus, pihak manajemen institusi sebaiknya tidak hanya meletakkan antarmuka chatbot pada laman web pusat, melainkan mengintegrasikannya ke dalam saluran komunikasi aktif harian mahasiswa seperti WhatsApp, Telegram, atau aplikasi akademik internal. Langkah ini dinilai strategis untuk memperluas aksesibilitas layanan informasi akademik selama 24 jam penuh tanpa membebani server lokal maupun membengkakkan anggaran operasional untuk pemanggilan API Large Language Model (LLM) secara berlebihan.

6.3 Saran untuk Pengembang Sistem

Para pengembang sistem chatbot akademik ke depan disarankan untuk melakukan eksperimen mendalam pada penentuan nilai ambang batas (threshold value) algoritma kesamaan string (Jaccard atau Cosine Similarity) pada komponen Decision Router. Hal ini krusial agar pembagian beban kerja otomatis antara mesin aturan lokal dan mesin generatif dapat berjalan semakin presisi. Selain itu, pengembang dapat mengoptimalkan fungsi fitur FAQ Management (CMS) dan halaman Login Administrator yang sudah dibangun dalam sistem ini untuk melakukan pembaruan korpus data secara dinamis langsung dari pihak biro akademik tanpa perlu menyentuh baris kode program. Terakhir, integrasi pustaka pemfilteran (guardrail AI) pada sisi backend Flask sangat disarankan guna memitigasi potensi teks halusinasi dari Gemini API secara instan sebelum jawaban disajikan di layar pengguna.

6.4 Saran untuk Penelitian Lanjutan

Untuk penelitian akademik di masa yang akan datang, arah pengembangan dapat difokuskan pada pengujian longitudinal jangka panjang guna mengukur stabilitas latensi sistem hibrida serta akumulasi efisiensi biaya API bulanan pada kondisi fluktuasi beban server yang riil. Peneliti selanjutnya juga disarankan untuk mengeksplorasi peningkatan kemampuan komponen Decision Router menggunakan metode klasifikasi berbasis machine learning seperti Naïve Bayes atau Decision Tree untuk menggantikan metode keyword matching konvensional agar pengalihan rute pertanyaan menjadi lebih adaptif. Tidak kalah penting, aspek keamanan data (data privacy) dan privasi informasi mahasiswa perlu diteliti lebih lanjut terkait kepatuhan regulasi saat terjadi pertukaran data internal kampus dengan penyedia layanan LLM pihak ketiga melalui jaringan internet.

DAFTAR PUSTAKA

- Adamopoulou, E., & Moussiades, L. (2022). An overview of chatbot technology. *IFIP Advances in Information and Communication Technology*, 584, 373-383. https://doi.org/10.1007/978-3-030-49186-4_31
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3), 114-123.
- Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In P. Jordan, B. Thomas, & B. Weerdmeester (Eds.),

Usability Evaluation in Industry (pp. 189-194). Taylor & Francis.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2024). GPT-4 in academic advisory: Opportunities and challenges. *Journal of Educational Technology & Society*, 27(1), 145-162.

Grand View Research. (2024). *Chatbot market size, share & trends analysis report 2024-2030*. Grand View Research, Inc.

Hien, H. T., Cuong, P. N., Nam, L. N. H., Nhung, H. L. T. K., & Thang, L. D. (2021). Intelligent assistants in higher education: A chatbot approach for administrators and students. *Proceedings of the 9th International Symposium on Information and Communication Technology*, 213-219. <https://doi.org/10.1145/3287921.3287937>

Kuhail, M. A., Alturki, N., Alramlawi, S., & Alhejori, K. (2023). Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28, 973-1018. <https://doi.org/10.1007/s10639-022-11177-3>

Pereira, J., & Diaz, O. (2022). Using health chatbots for behavior change: A mapping study. *Journal of Medical Systems*, 43(135), 1-11. <https://doi.org/10.1007/s10916-019-1237-1>

Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2023). Chatbot for university related FAQs. *Proceedings of IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1525-1530.

Santoso, B., Purnama, H., & Wijaya, R. (2024). Implementasi chatbot berbasis NLP untuk layanan akademik di perguruan tinggi Indonesia: Studi kasus Universitas Bina Nusantara. *Jurnal Teknologi dan Sistem Informasi*, 10(2), 87-102. <https://doi.org/10.25126/jtiik.2024.10.2.7821>