

**PENGEMBANGAN *HYBRID CHATBOT* UNTUK LAYANAN
INFORMASI AKADEMIK PROGRAM STUDI
INFORMATIKA UII**



Disusun Oleh:

N a m a : Ahmad Fauzan Zaky

NIM : 21523201

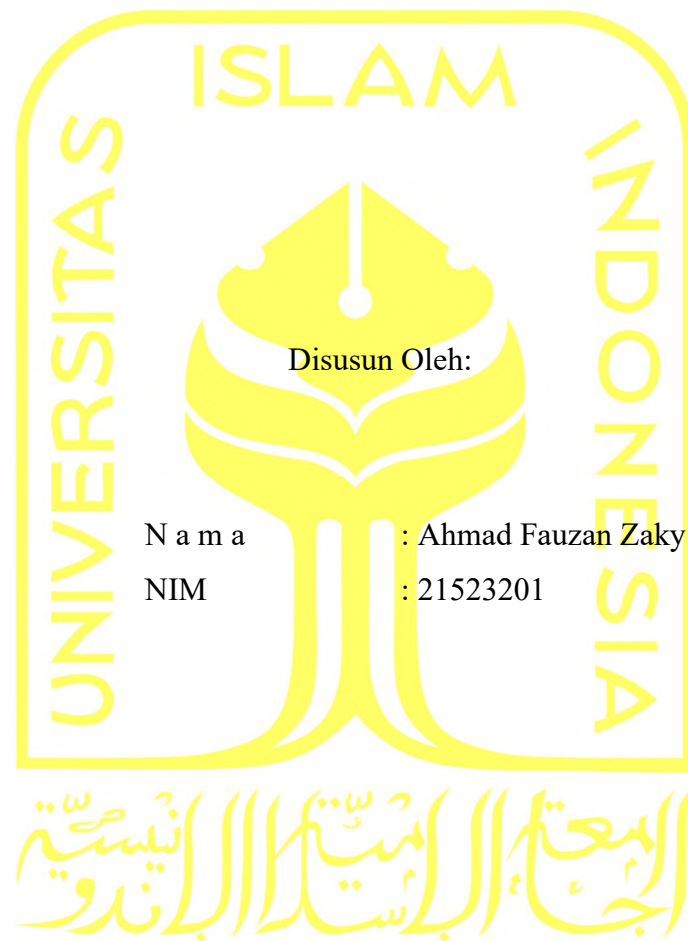
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN *HYBRID CHATBOT* UNTUK LAYANAN
INFORMASI AKADEMIK PROGRAM STUDI
INFORMATIKA UII

TUGAS AKHIR



Yogyakarta, 13 Januari 2026

Pembimbing,

(Chanifah Indah Ratnasari, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN *HYBRID CHATBOT* UNTUK LAYANAN
INFORMASI AKADEMIK PROGRAM STUDI
INFORMATIKA UII**

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 13 Januari 2026

Tim Penguji

Chanifah Indah Ratnasari, S.Kom., M.Kom.

Anggota 1

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

Anggota 2

Erika Ramadhani, S.T., M.Eng.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ahmad Fauzan Zaky

NIM : 21523201

Tugas akhir dengan judul:

**PENGEMBANGAN *HYBRID CHATBOT* UNTUK LAYANAN
INFORMASI AKADEMIK PROGRAM STUDI
INFORMATIKA UII**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 13 Januari 2026



(Ahmad Fauzan Zaky)

HALAMAN PERSEMBAHAN

Allah Swt., atas segala rahmat, karunia, dan hidayah-Nya yang senantiasa menyertai setiap langkah penulis, sehingga tugas akhir ini dapat diselesaikan dengan baik.

Orang tua tercinta, sebagai sumber doa dan kasih sayang. Terima kasih atas segala pengorbanan, kesabaran, dan dukungannya.

Semoga karya ini dapat bermanfaat dan dapat berkontribusi untuk kebaikan dunia.

HALAMAN MOTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا، إِنَّ مَعَ الْعُسْرِ يُسْرًا

“Maka sesungguhnya bersama kesulitan ada kemudahan. Sesungguhnya bersama kesulitan ada kemudahan”

(QS. Al-Insyirah: 5-6)

KATA PENGANTAR

Puji syukur ke hadirat Allah Swt. atas segala rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir ini dengan baik. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem *chatbot* informasi akademik yang mampu mendukung kebutuhan layanan informasi mahasiswa melalui pendekatan *dual-mode*, yaitu mengombinasikan mekanisme *rule-based* dan *hybrid*. Diharapkan sistem yang dikembangkan dapat menjadi solusi alternatif dalam meningkatkan aksesibilitas dan efisiensi penyampaian informasi akademik, khususnya dalam menghadapi variasi cara bertanya yang digunakan oleh mahasiswa.

Penulis menyadari bahwa laporan tugas akhir ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk penyempurnaan penelitian ini di masa yang akan datang. Semoga tugas akhir ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi, serta menjadi referensi bagi penelitian selanjutnya. Untuk itu, penulis ingin menyampaikan terima kasih dan penghargaan yang setinggi-tingginya kepada:

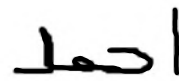
1. Allah Swt. yang selalu memberikan kekuatan dan kelapangan hati dalam setiap langkah kehidupan.
2. Bapak, Ibu, dan kakak tercinta, atas kasih sayang, doa, serta dukungan yang tiada henti.
3. Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
4. DThomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Chanifah Indah Ratnasari, S.Kom., M.Kom., selaku dosen pembimbing tugas akhir, yang telah memberikan bimbingan, arahan, dan ilmu yang sangat berharga dalam proses penyusunan tugas akhir ini.
6. Seluruh dosen Program Studi Informatika Universitas Islam Indonesia, atas segala ilmu, dedikasi, dan pengalaman yang telah diberikan selama masa studi.
7. Seluruh pihak yang telah memberikan bantuan dan dukungan dalam bentuk apa pun selama proses penyusunan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang membangun guna perbaikan di masa mendatang.

Akhir kata, penulis berharap karya ini dapat memberikan manfaat dan menjadi kontribusi positif, khususnya dalam bidang teknologi kecerdasan buatan dan interaksi digital, serta menjadi pijakan awal bagi penelitian-penelitian selanjutnya. Semoga apa yang tertuang dalam tugas akhir ini dapat memberikan inspirasi dan membuka cakrawala baru bagi pembaca yang tertarik dengan topik yang dibahas.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 13 Januari 2026



(Ahmad Fauzan Zaky)

SARI

Perkembangan teknologi informasi mendorong perguruan tinggi untuk menyediakan layanan informasi akademik yang lebih cepat, mudah diakses, dan responsif terhadap kebutuhan mahasiswa. Namun, sistem informasi akademik konvensional yang bersifat statis, seperti laman web atau *Frequently Asked Questions* (FAQ), masih memiliki keterbatasan dalam menangani variasi cara bertanya pengguna, khususnya ketika pertanyaan disampaikan dalam bahasa alami, tidak baku, atau mengandung parafrase. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem *chatbot* informasi akademik berbasis *dual-mode* yang mampu menangani pertanyaan mahasiswa baik yang bersifat terstruktur maupun tidak terstruktur. Sistem *chatbot* dikembangkan untuk Program Studi Informatika Universitas Islam Indonesia dengan mengombinasikan dua pendekatan penalaran, yaitu *rule-based* dan *hybrid*. Mode *rule-based* digunakan untuk memberikan respons yang cepat dan deterministik terhadap pertanyaan akademik yang terdefinisi dengan baik, sedangkan mode *hybrid* mengintegrasikan logika *rule-based* dengan penalaran *generative* berbasis *Large Language Model* untuk menangani variasi bahasa dan pertanyaan yang lebih fleksibel.

Metode penelitian yang digunakan adalah *prototyping*, yang memungkinkan pengembangan sistem dilakukan secara iteratif melalui tahapan pengumpulan kebutuhan, perancangan awal, pembangunan prototipe, evaluasi, perbaikan, dan implementasi. Proses pengumpulan kebutuhan dilakukan melalui wawancara dengan staf akademik dan observasi terhadap pertanyaan mahasiswa. Sistem *chatbot* dibangun menggunakan arsitektur *multi-layer* yang memisahkan *frontend* berbasis PHP dan *backend* berbasis *Python* yang dijalankan melalui Flask sebagai *Application Programming Interface* (API). Mode *hybrid* memanfaatkan model *Mistral-7B-Instruct-v0.2.Q4_K_M* yang dijalankan secara lokal. Evaluasi sistem dilakukan dengan menguji *chatbot* menggunakan pertanyaan nyata dari mahasiswa dan staf akademik. Hasil pengujian menunjukkan bahwa *chatbot rule-based* memiliki keunggulan dalam kecepatan dan konsistensi respons, dengan waktu respons yang sangat rendah dan perilaku yang stabil untuk pertanyaan terstruktur. Namun, mode ini memiliki keterbatasan dalam menangani variasi bahasa, parafrase, dan kesalahan penulisan. Sebaliknya, *chatbot hybrid* menunjukkan peningkatan akurasi dalam menangani pertanyaan yang tidak terstruktur, meskipun dengan konsekuensi waktu respons yang lebih tinggi dan munculnya beberapa respons yang bersifat halusinasi. Berdasarkan hasil penelitian, dapat disimpulkan bahwa pendekatan *dual-mode* memberikan solusi yang saling melengkapi dalam layanan informasi akademik. Mode *rule-based* efektif untuk kebutuhan informasi yang cepat dan pasti, sementara

mode *hybrid* memberikan fleksibilitas linguistik yang lebih baik. Penelitian ini menunjukkan bahwa penerapan arsitektur *hybrid* ringan dengan mekanisme *generative* dapat meningkatkan ketahanan dan adaptabilitas *chatbot* akademik tanpa sepenuhnya mengorbankan keandalan sistem deterministik.

Kata kunci: *chatbot*, informasi akademik, *rule-based*, *hybrid*, *prototyping*.

GLOSARIUM

<i>Chatbot</i>	Sistem perangkat lunak yang dirancang untuk berinteraksi dengan pengguna melalui percakapan berbasis teks guna memberikan informasi atau layanan secara otomatis.
<i>Generative</i>	Pendekatan pada sistem kecerdasan buatan yang menghasilkan respons secara dinamis
<i>Hybrid</i>	Pendekatan sistem yang mengombinasikan mekanisme <i>rule-based</i> dengan pendekatan <i>generative</i> .
<i>Prototyping</i>	Metode pengembangan sistem yang dilakukan secara iteratif melalui pembuatan dan penyempurnaan prototipe berdasarkan umpan balik pengguna hingga diperoleh sistem yang sesuai dengan kebutuhan.
<i>Rule-based</i>	Pendekatan sistem yang menghasilkan respons berdasarkan aturan, pola, atau kata kunci yang telah ditentukan sebelumnya untuk menjamin konsistensi dan keandalan jawaban.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	xi
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II KAJIAN PUSTAKA	5
2.1 <i>Rule-Based Chatbot</i>	5
2.2 <i>Generative Chatbot</i>	5
2.3 <i>Hybrid Chatbot</i>	6
2.4 <i>Metode Prototyping</i>	6
2.5 Penelitian Terdahulu	6
BAB III METODOLOGI PENELITIAN	9
3.1 <i>Communication</i>	9
3.2 <i>Quick Plan and Quick Design</i>	9
3.3 <i>Prototyping</i>	10
3.4 <i>Deployment, Delivery, and Feedback</i>	10
BAB IV HASIL DAN PEMBAHASAN	11
4.1 Iterasi Pertama	11
4.1.1 Hasil <i>Communication</i>	11
4.1.2 Hasil <i>Quick Plan and Quick Design</i>	11
4.1.3 Hasil dari <i>Prototyping</i>	16
4.1.4 Hasil dari <i>Deployment, Delivery, and Feedback</i>	17
4.2 Iterasi Kedua	17
4.2.1 Hasil <i>Communication</i>	18
4.2.2 Hasil <i>Quick Plan and Quick Design</i>	18
4.2.3 Hasil dari <i>Prototyping</i>	22
4.2.4 Hasil dari <i>Deployment, Delivery, and Feedback</i>	23
4.3 Analisis Hasil Penelitian	24
4.3.1 Umpan Balik	24
4.3.2 Konfigurasi	24
4.3.3 Pengambilan Data	25
4.3.4 Kendala	31
BAB V KESIMPULAN DAN SARAN	35

5.1 Kesimpulan	35
5.2 Saran	36
DAFTAR PUSTAKA	39
LAMPIRAN.....	41

DAFTAR TABEL

Tabel 2.1 Analisis Performa <i>Chatbot Rule-based</i> dan <i>Hybrid</i>	7
Tabel 4.1 Umpan Balik	24
Tabel 4.2 Perbandingan Model Mistral	25
Tabel 4.3 <i>Sample</i> Pertanyaan	26
Tabel 4.4 Contoh Respons <i>Chatbot Rule-based</i> dan <i>Hybrid</i>	27
Tabel 4.5 Tabel Akurasi.....	28
Tabel 4.6 Tabel Waktu.....	30

DAFTAR GAMBAR

Gambar 3.1 Metode <i>Prototyping</i>	9
Gambar 4.1 Struktur <i>Knowledge</i>	12
Gambar 4.2 Contoh <i>Knowledge</i>	12
Gambar 4.3 <i>Sample Knowledge</i>	12
Gambar 4.4 Diagram <i>Use Case</i>	13
Gambar 4.5 <i>Activity Diagram</i>	15
Gambar 4.6 Kode <i>Prompting Mode Hybrid</i> Iterasi Pertama	16
Gambar 4.7 Tampilan Iterasi Pertama	16
Gambar 4.8 <i>Testing Pengguna</i>	17
Gambar 4.9 Struktur <i>Chatbot</i> Sempel	19
Gambar 4.10 Struktur <i>Chatbot</i>	19
Gambar 4.11 Alur <i>Chatbot</i>	20
Gambar 4.12 Sesi Flask	21
Gambar 4.13 Tampilan Iterasi Pertama	23
Gambar 4.14 Kode <i>Prompting Mode Hybrid</i> Iterasi Kedua.....	23
Gambar 4.15 <i>Testing Waktu Hybrid</i> (Mode Handal)	29
Gambar 4.16 <i>Testing Waktu Rule-based</i> (Mode Cepat).....	30
Gambar 4.17 Perbandingan Waktu	31

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai bidang, termasuk dalam dunia pendidikan tinggi. Pada masa sebelumnya, informasi akademik seperti pengumuman, jadwal perkuliahan, dan peraturan akademik umumnya disampaikan melalui papan pengumuman atau komunikasi manual. Seiring dengan kemajuan teknologi digital, penyampaian informasi tersebut mulai beralih ke media daring seperti situs web resmi institusi dan media sosial. Meskipun demikian, metode penyampaian informasi berbasis web dan media sosial masih memiliki keterbatasan, antara lain mengharuskan pengguna melakukan pencarian secara manual, tidak bersifat personal, serta tidak menyediakan bantuan interaktif untuk menjawab pertanyaan spesifik mahasiswa.

Keterbatasan tersebut juga sejalan dengan permasalahan yang banyak ditemukan pada sistem informasi administratif konvensional, di mana proses manual atau semi-digital rentan terhadap kesalahan manusia, keterlambatan penyampaian informasi, serta kehilangan data (Gumilang et al., 2024). Dalam konteks akademik, kondisi ini dapat menyebabkan kesulitan bagi mahasiswa dalam memperoleh informasi yang akurat dan tepat waktu, yang pada akhirnya berpotensi memengaruhi proses pengambilan keputusan akademik.

Salah satu pendekatan teknologi yang dapat digunakan untuk mengatasi permasalahan tersebut adalah *chatbot*. *Chatbot* merupakan perangkat lunak percakapan yang mampu berinteraksi dengan pengguna melalui bahasa alami dalam bentuk teks maupun suara (Ilieva et al., 2023). Dalam lingkungan akademik, *chatbot* telah dimanfaatkan untuk menjawab pertanyaan yang sering diajukan terkait jadwal perkuliahan, peraturan akademik, prosedur administrasi, serta layanan akademik lainnya (Labadze et al., 2023; Ula et al., 2023). Beberapa penelitian juga menunjukkan bahwa *chatbot* dapat meningkatkan keterlibatan pengguna karena mampu menyediakan interaksi yang lebih alami dan responsif dibandingkan sistem informasi berbasis web (Skjuve et al., 2021).

Sebagian besar *chatbot* akademik yang telah dikembangkan menggunakan pendekatan *rule-based*, yaitu dengan memanfaatkan kumpulan aturan dan kata kunci yang telah ditentukan sebelumnya (Adamopoulou & Moussiades, 2020). Pendekatan ini memiliki keunggulan dalam hal konsistensi dan keandalan jawaban, terutama untuk pertanyaan yang terstruktur dan berada dalam lingkup domain yang jelas. Namun, *chatbot rule-based* cenderung kurang fleksibel

dalam menangani variasi bahasa, parafrase, maupun pertanyaan dengan struktur kalimat yang berbeda, yang umum digunakan oleh mahasiswa dalam interaksi sehari-hari.

Sebagai upaya untuk meningkatkan fleksibilitas tersebut, beberapa penelitian terbaru mengusulkan penggunaan *chatbot hybrid* yang mengombinasikan pendekatan *rule-based* dengan model generatif berbasis *Large Language Models* (LLMs) (Halvonik & Kapusta, 2024; Thorat & Jadhav, 2020). Pendekatan *hybrid* memungkinkan *chatbot* untuk memahami variasi bahasa yang lebih luas, namun di sisi lain juga menimbulkan tantangan baru, seperti berkurangnya kendali terhadap jawaban, meningkatnya latensi, serta risiko munculnya jawaban yang tidak akurat atau bersifat halusinasi. Oleh karena itu, diperlukan perancangan *chatbot hybrid* yang mampu menyeimbangkan antara fleksibilitas linguistik dan keandalan jawaban, khususnya pada domain akademik yang menuntut ketepatan informasi.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan *chatbot* informasi akademik untuk Program Studi Informatika Universitas Islam Indonesia (UII) dengan pendekatan *dual-mode*, yaitu mode *rule-based* murni dan mode *hybrid* yang mengintegrasikan logika *rule-based* dengan penalaran generatif secara terbatas. Pendekatan *dual-mode* ini memungkinkan pemisahan yang jelas antara respons deterministik dan respons yang melibatkan model generatif, sehingga dapat dilakukan perbandingan yang terkontrol terhadap kedua pendekatan tersebut. Dengan memanfaatkan pengetahuan akademik yang bersifat spesifik institusi, *chatbot* ini diharapkan dapat memberikan layanan informasi akademik yang lebih akurat, terpusat, dan mudah diakses.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

- a. Bagaimana membangun *chatbot* informasi akademik dengan pendekatan *dual-mode* yang memisahkan mode *rule-based* dan mode *hybrid*?
- b. Bagaimana perbandingan kinerja *chatbot* pada mode *rule-based* dan mode *hybrid* dalam menangani pertanyaan akademik?
- c. Bagaimana efektivitas penggunaan penalaran generatif dalam meningkatkan kemampuan *chatbot* tanpa mengurangi keandalan jawaban?

1.3 Batasan Masalah

Beberapa batasan masalah dari penelitian ini adalah sebagai berikut:

- a. *Chatbot* hanya digunakan untuk menyampaikan informasi akademik Program Studi Informatika – Program Sarjana Universitas Islam Indonesia.
- b. Interaksi *chatbot* dibatasi pada percakapan berbasis teks.
- c. Mode generatif digunakan sebagai tambahan ketika mode *rule-based* tidak memberikan hasil yang memuaskan.
- d. Evaluasi *chatbot* difokuskan pada ketepatan dan relevansi jawaban.

1.4 Tujuan Penelitian

Adapun tujuan penelitian dari penelitian ini adalah:

- a. Mengembangkan *chatbot* informasi akademik dengan pendekatan *dual-mode*.
- b. Membandingkan performa *chatbot rule-based* dan *chatbot hybrid* dalam konteks layanan akademik.
- c. Menganalisis dampak penggunaan generatif terbatas terhadap keandalan dan fleksibilitas *chatbot*.

1.5 Manfaat Penelitian

Beberapa manfaat penelitian dari penelitian ini adalah:

- a. Sebagai referensi dalam pengembangan dan evaluasi *chatbot* akademik berbasis *dual-mode*.
- b. Membantu mahasiswa memperoleh informasi akademik secara cepat dan terpusat, di mana saja dan kapan saja.
- c. Adanya alternatif layanan informasi akademik bagi Program Studi Informatika UII.

1.6 Sistematika Penulisan

Berikut adalah struktur laporan yang terdiri dari 5 bab:

a. BAB 1 PENDAHULUAN

Bab ini memaparkan landasan awal penelitian yang melatarbelakangi pengembangan sistem *chatbot* informasi akademik. Pembahasan dimulai dengan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

b. BAB II KAJIAN PUSTAKA

Bab ini berisi kajian teoritis dan penelitian terdahulu yang relevan dengan pengembangan *chatbot* informasi akademik. Pembahasan mencakup *chatbot* berbasis *rule-based*, *chatbot* berbasis *generative*, *chatbot hybrid*.

c. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan serta perancangan sistem *chatbot* secara rinci seperti pemilihan aspek *chatbot* dan pengukuran performa *chatbot*.

d. BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi sistem *chatbot* informasi akademik serta analisis kinerja sistem berdasarkan pengujian yang telah dilakukan. Pembahasan mencakup konfigurasi sistem, hasil iterasi, arsitektur, basis pengetahuan alur, dan performa *chatbot*

e. BAB V KESIMPULAN DAN SARAN

Bab ini menyimpulkan seluruh hal penelitian dan memberikan saran untuk penelitian serupa di masa depan.

BAB II

KAJIAN PUSTAKA

Pada tahap ini dilakukan kajian pustaka mengenai landasan teori terkait dan penelitian *chatbot* terdahulu. Kajian ini dilakukan guna menentukan tipe *chatbot* yang cocok dikembangkan sebagai *chatbot* Informasi Akademik Prodi Informatika Universitas Islam Indonesia.

2.1 *Rule-Based Chatbot*

Rule-based chatbot bekerja berdasarkan pencocokan *keyword* terhadap aturan yang telah ditentukan (Singh et al., 2019). Ranoliya et al. (2017) mengembangkan *chatbot* akademik berbasis *Artificial Intelligence Markup Language* (AIML) dengan komponen *rule-based* untuk menjawab pertanyaan terkait informasi kampus. Meskipun sistem tersebut mampu memberikan jawaban yang akurat untuk pertanyaan yang sesuai dengan aturan yang telah didefinisikan, *chatbot* mengalami kesulitan dalam menangani masukan pengguna yang tidak sesuai dengan pola yang tersedia bila hanya mengandalkan *rule-based*. Permasalahan serupa yang sering ditemukan pada *chatbot* yang menerapkan komponen *rule-based*, juga ditemukan pada *chatbot* berbasis Dialogflow untuk layanan FAQ perpustakaan universitas, di mana variasi bahasa, singkatan, dan penggunaan bahasa campuran menyebabkan terjadinya kesalahan pencocokan dan respons yang kurang relevan (Barus & Suriyati, 2021). Temuan ini menunjukkan bahwa *chatbot rule-based* memiliki keterbatasan dalam menghadapi keragaman ekspresi bahasa alami (*natural language*).

2.2 *Generative Chatbot*

Chatbot berbasis *generative* memanfaatkan model kecerdasan buatan, khususnya *Large Language Models* (LLMs), untuk menghasilkan respons secara dinamis berdasarkan konteks percakapan. Berbeda dengan *chatbot rule-based*, pendekatan *generative* tidak bergantung pada aturan baku, melainkan pada kemampuan model dalam memahami dan menghasilkan bahasa alami. Hal ini memungkinkan *chatbot generative* untuk menangani variasi bahasa, parafrase, serta pertanyaan yang bersifat terbuka. Ula et al. (2023) mengembangkan *chatbot* berbasis AI untuk meningkatkan layanan informasi akademik dengan memanfaatkan pemrosesan bahasa alami (*natural language processing*). Hasil penelitian menunjukkan bahwa *chatbot generative* dapat meningkatkan efisiensi layanan dengan mengotomatisasi pertanyaan. Namun, penelitian

tersebut juga mencatat keterbatasan dalam pemahaman konteks dan ketepatan jawaban untuk pertanyaan yang kompleks. Selain itu, Ji et al. (2023) mengidentifikasi fenomena halusinasi pada model *generative*, yaitu kondisi di mana *chatbot* menghasilkan jawaban yang tampak meyakinkan namun tidak akurat secara faktual. Permasalahan ini menjadi tantangan serius dalam penerapan *chatbot generative* pada domain akademik yang menuntut ketepatan informasi.

2.3 Hybrid Chatbot

Untuk mengatasi keterbatasan masing-masing pendekatan, beberapa penelitian mengusulkan penggunaan *chatbot hybrid* yang mengombinasikan logika *rule-based* dengan *generative*. Halvonik & Kapusta (2024) menunjukkan bahwa integrasi kedua pendekatan tersebut dapat meningkatkan kelancaran percakapan tanpa mengorbankan ketepatan informasi. Adamopoulou & Moussiades (2020) juga menyatakan bahwa *chatbot hybrid* mampu menjembatani respons deterministik dan interaksi yang lebih natural. Labadze et al. (2023), melalui tinjauan sistematis terhadap *chatbot* di pendidikan tinggi, menekankan pentingnya keseimbangan antara akurasi dan fleksibilitas dalam pengembangan *chatbot* akademik.

2.4 Metode Prototyping

Model *prototyping* yang digunakan dalam penelitian ini mengacu pada Pressman (2005), yang menekankan visualisasi awal sistem, pembangunan cepat, serta penyempurnaan berulang hingga sistem memenuhi kebutuhan pengguna. Pendekatan *prototyping* sesuai untuk pengembangan *chatbot*, mengingat karakteristik sistem percakapan yang menuntut ketepatan respons, alur dialog yang alami, serta kemudahan penggunaan antarmuka.

Beberapa penelitian sebelumnya juga menunjukkan bahwa metode *prototyping* efektif dalam pengembangan sistem interaktif berbasis pengguna karena memungkinkan penyesuaian dini terhadap kebutuhan dan ekspektasi pemangku kepentingan (*stakeholder*) (Abigail & Haryono, 2025; Camburn et al., 2017; Rachma & Muhlas, 2022).

2.5 Penelitian Terdahulu

Pada kajian ini dilakukan pendalaman terhadap penelitian-penelitian sebelumnya guna membangun landasan teori bagi penelitian ini. Kajian ini juga mengidentifikasi kelebihan dan kekurangan dari metode yang digunakan dalam penelitian terdahulu. Pencarian literatur terdahulu dilakukan dengan melakukan pencarian melalui Google Scholar dengan

menggunakan kata kunci “*Chatbot*”, “*Rule-based*”, “*Generative*”, dan “*Web*” diterapkan filter pencarian tahun dari 2015 hingga 2025 .

Penelitian serupa sebelumnya telah membahas aspek *rule-based*, *generative*, *hybrid*, dan pengembangan *chatbot* FAQ serupa. Konsensus yang ditemukan dari berbagai peneliti adalah komponen *rule-based* bersifat kaku, deterministik, dan terstruktur. Dan komponen *generative* membawa risiko halusinasi, terpersonalisasi dan fleksibel. Berikut Tabel 2.1 mengenai literatur terdahulu.

Tabel 2.1 Analisis Performa *Chatbot Rule-based* dan *Hybrid*

Perbandingan	Chatbot	
	<i>Rule-based</i>	<i>Hybrid</i>
Fleksibilitas Bahasa	Rendah, terbatas pada pola yang tersedia. (Barus & Surijati, 2021)	Lebih fleksibel dibanding <i>rule-based</i> namun tetap terkontrol. (Halvonik & Kapusta, 2024)
Akurasi Informasi	Tinggi selama aturan tersedia. (Ranoliya et al., 2017)	Sama seperti <i>rule-based</i> . Pada bagian <i>generative</i> tinggi apabila <i>knowledge</i> bagus (Labadze et al., 2023) Adaptif saat <i>generatif</i> digunakan. (Adamopoulou & Moussiades, 2020)
Risiko Halusinasi	Tidak ditemukan karena respons bersifat deterministik. (Ranoliya et al., 2017)	Ada apabila memanfaatkan <i>generative</i> . (Ji et al., 2023)
Kesesuaian untuk Domain Akademik	Cocok untuk informasi terstruktur dan FAQ. (Barus & Surijati, 2021; Ranoliya et al., 2017)	Dapat memberikan informasi yang terpersonalisasi. (Labadze et al., 2023; Thorat & Jadhav, 2020)

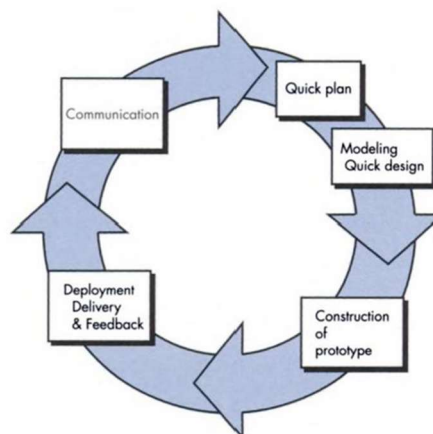
Hal ini dilakukan karena kebanyakan *chatbot* menerapkan salah satu dari dua pendekatan ini. Untuk *hybrid* yang dimaksud di sini adalah tipe *chatbot* yang menggabungkan *rule-based*

dan *generative*. Hal ini perlu digaris bawahi karena beberapa *chatbot* menggunakan *hybrid* dari dua jenis *chatbot* lain, seperti *rule-based* dan *retrieval*.

Fleksibilitas bahasa kemampuan sistem *chatbot* dalam menerima *input* dalam berbagai bentuk: seperti *slang*, bahasa tidak formal, dan *typo*. Dalam bidang ini *chatbot hybrid* lebih unggul karena dapat memahami berdasarkan data *generative*. Untuk akurasi, *rule-based chatbot* lebih unggul karena lebih tepat pada *knowledge*, tetapi agak lebih baku. Sedangkan *hybrid* dapat terjadi halusinasi. Untuk risiko halusinasi pada *rule-based* seharusnya apabila *knowledge* tepat, tidak akan terjadi halusinasi. Untuk *generative* selalu ada risiko halusinasi.

BAB III METODOLOGI PENELITIAN

Model *prototyping* yang digunakan dalam penelitian ini mengacu pada Pressman (2005), yang menerapkan tahap seperti yang dapat dilihat pada Gambar 3.1., yaitu *Communication*, *Quick Plan and Quick Design*, *Prototyping*, dan *Deployment, Delivery, and Feedback*. Metode *prototyping* dipilih karena pengembangan *chatbot* sering kali melibatkan kebutuhan yang belum sepenuhnya jelas di awal sehingga pendekatan iteratif memungkinkan pembuatan prototipe awal yang dapat langsung dievaluasi oleh pengguna. Melalui siklus berulang berbasis umpan balik, fitur dapat disempurnakan secara bertahap hingga mencapai versi iterasi yang memenuhi kebutuhan pengguna.



Gambar 3.1 Metode *Prototyping*

3.1 *Communication*

Dilakukan dengan melalui wawancara dengan staf akademik prodi informatika Universitas Islam Indonesia, sekretaris program studi, dan melakukan observasi pada pertanyaan yang sering muncul oleh mahasiswa untuk menentukan *knowledge chatbot* yang akan dikembangkan.

3.2 *Quick Plan and Quick Design*

Pada tahap ini dilakukan perancangan awal sistem alur komunikasi antar komponen, serta desain antarmuka pengguna. Selain itu, dibuat sketsa untuk memvisualisasikan hubungan dan alur program dan juga tata letak sistem sebelum tahap implementasi.

3.3 *Prototyping*

Tahap pembangunan prototipe dilakukan secara bertahap dengan mengimplementasikan fungsi-fungsi utama *chatbot* sesuai dengan desain yang telah dibuat. Prototipe awal difokuskan pada implementasi mode *rule-based*, komunikasi API antara *frontend* dan *backend*, serta pemrosesan teks dasar. Selanjutnya, prototipe dikembangkan dengan menambahkan mode *rule-based* dan kemudian *hybrid*.

3.4 *Deployment, Delivery, and Feedback*

Setiap prototipe yang dihasilkan, dilakukan interaksi dan evaluasi oleh staf akademik dan pengguna uji (mahasiswa). Evaluasi mencakup ketepatan jawaban, kejelasan antarmuka, kecepatan respons, serta kesesuaian alur percakapan. Masukan dari tahap ini digunakan sebagai dasar untuk melakukan perbaikan dan penyempurnaan pada siklus *prototyping* berikutnya. *Feedback* didapatkan dengan pengujian kepada calon pengguna, dalam hal ini mahasiswa serta staf pada prodi tersebut.

Pengukuran performa *chatbot* pada tahap *Deployment, Delivery, and Feedback* dilakukan dengan menargetkan akurasi dan kecepatan respons *chatbot*. *Chatbot* di uji dengan melakukan *testing* mandiri oleh *programmer* dan dilakukannya pengujian bersama pihak terkait seperti staf prodi dan mahasiswa.

Akurasi jawaban *chatbot* dilakukan dengan melakukan *input*, dan kemudian membandingkan *output* yang ada dengan jawaban yang semestinya. Jawaban yang benar ini di dapat dengan melihat *knowledge base* yang ada dan dengan bertanya langsung pada staf prodi yang merupakan ahli dalam informasi mengenai prodi.

Kecepatan *chatbot* diukur dengan melakukan beberapa kali *input*, dan kemudian melihat kecepatan pemrosesan *output*. Fitur sementara berupa fitur waktu yang dikeluarkan bersama *output* untuk memudahkan melihat waktu prosesnya. *Testing* kecepatan dilakukan dengan mandiri oleh *programmer* dan pengujian bersama pihak dan mahasiswa guna melihat kecepatan *chatbot*. Nilai rata-rata dan standar deviasi menjadi data yang penting untuk kemajuan pengembangan performa *chatbot*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Iterasi Pertama

Pada iterasi pertama, sistem *chatbot* dikembangkan dengan fokus pada implementasi dasar arsitektur *rule-based* dan *hybrid*. *Knowledge base* masih bersifat terbatas dan mekanisme evaluasi respons belum sepenuhnya mempertimbangkan kondisi ketika informasi tidak tersedia. Hasil pengujian pada tahap ini menunjukkan bahwa *chatbot* mampu menjawab pertanyaan yang sesuai dengan data yang ada, namun masih menghasilkan beberapa keluaran yang kurang relevan, terutama pada mode generatif ketika jawaban tidak ditemukan dalam *knowledge base*.

4.1.1 Hasil *Communication*

Tahap paling pertama yang dilakukan dalam penelitian ini adalah tahap *communication*. Pada iterasi pertama, tahap *communication* menghasilkan beberapa hal penting dari wawancara dengan pemangku kepentingan. Hasil dari tahap ini berupa didapatkannya pertanyaan yang sering ditanyakan mahasiswa dan pertanyaan yang diinginkan oleh pihak staf. Pertanyaan yang ada ini berupa pertanyaan seputar informasi akademik informatika UII dan kemudian di masukan ke dalam *knowledge chatbot*. Hasil dari tahap *communication* ini di jadikan sebagai pedoman yang pada tahap tersebut, diperoleh beberapa kebutuhan utama sistem, antara lain:

- a. Pengadaan informasi akademik berbentuk *knowledge* yang sering ditanyakan mahasiswa
- b. Basis data pengetahuan terstruktur berbasis teks.
- c. Diketahui bahwa dibutuhkannya antarmuka yang menyerupai aplikasi pesan instan.

4.1.2 Hasil *Quick Plan and Quick Design*

Tahap kedua adalah *quick plan and quick design*. Hasil dari tahap *Quick Plan and Quick Design* menjadi hal yang penting untuk keseluruhan sistem. Aspek yang penting dalam bagian ini berupa:

- a. *Knowledge*
- b. *Use case*
- c. *Activity Diagram*

```
keywords: [keyword1] [keyword2]
response:
[Informasi]
---selesai---
```

Gambar 4.1 Struktur *Knowledge*

```
keywords: Kalender akademik FTI UII
response:
Kalender akademik FTI dapat dilihat di:
https://fit.uui.ac.id/blog/2022/07/05/kalender-akademik/
---selesai---
```

Gambar 4.2 Contoh *Knowledge*

```
keywords: apa itu jurusan informatika
response:
Mendidik mahasiswa-mahasiswanya sehingga menghasilkan profesional teknologi
informasi yang berjiwa entrepreneur serta menjunjung tinggi etika merupakan
tekad kami. Didukung dengan sumber daya manusia yang kompeten dan berkomitmen
tinggi serta fasilitas pembelajaran yang memadai, kami siap menyambut Anda
untuk bergabung dan ikut merespon perkembangan teknologi informasi terkini.
---selesai---
```

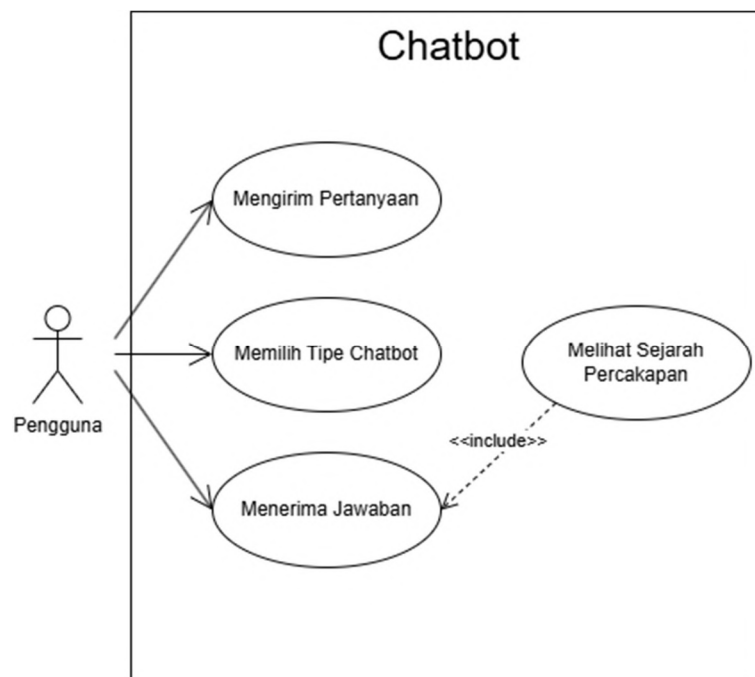
```
keywords: form checklist
response:
Untuk memudahkan dalam menge-trace MK yang sudah diambil (beserta nilainya)
ataupun yang belum, dapat mengunduh form checklist MK pada tautan berikut:
https://tinyurl.com/FormChecklistMatkulInformatika
(apabila nilai Minimum kosong, nilai minimal adalah D, dengan catatan nilai
D maksimal 10% dari total SKS).
---selesai---
```

Gambar 4.3 *Sample Knowledge*

Basis pengetahuan *chatbot* disimpan dalam bentuk teks yang berisi pasangan kata kunci dan respons. Pengadaan *knowledge* ini sendiri berdasar pada wawancara dengan staf prodi. Pengembangan lanjutan basis pengetahuan (*knowledge base*) dilakukan secara iteratif berdasarkan hasil pengujian dan wawancara lanjut dengan pihak staf, observasi pertanyaan mahasiswa, serta masukan yang diperoleh selama proses evaluasi pengguna. Informasi akademik yang disimpan mencakup kategori mata kuliah, jalur tugas akhir, mekanisme key-in, serta informasi akademik umum lainnya. Contoh isi basis pengetahuan ditampilkan pada Gambar 4.1 yang berisikan struktur, Gambar 4.2 yang berisikan salah satu entri, dan Gambar 4.3 yang berupa sampel.

Cara kerja *knowledge chatbot* menghasilkan jawaban berupa pengguna melakukan *input* pertanyaan yang kemudian *chatbot* akan mencocokkan kandungan kata *input* dengan *keywords* pada entri yang paling mirip. Kemudian *chatbot* mengeluarkan *output*.

Struktur *knowledge* itu sendiri terdiri dari *keywords* dan *response*. Kumpulan *keywords* yang berperan sebagai *tags* untuk *chatbot* dapat mencocokkan pertanyaan dengan jawaban yang sesuai. *Response* berperan untuk menampung jawaban, yang kemudian akan diambil *chatbot* berdasarkan *keywords*.



Gambar 4.4 Diagram *Use Case*

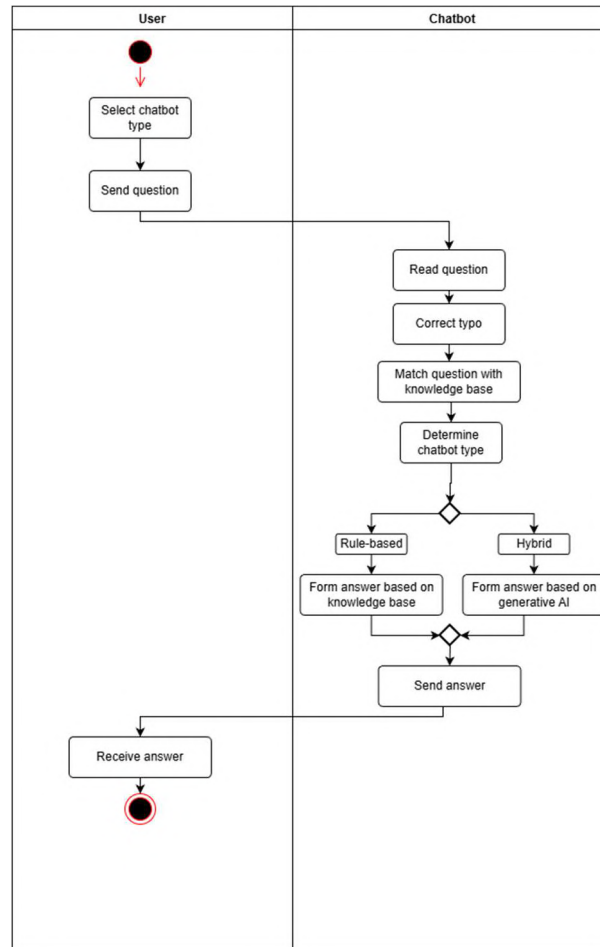
Use case diagram pada Gambar 4.4 menggambarkan interaksi antara aktor Pengguna dengan sistem *Chatbot* dalam konteks layanan informasi akademik. Diagram ini menunjukkan fungsionalitas utama yang dapat diakses oleh pengguna serta hubungan antar *use case* yang mendukung proses interaksi *chatbot* secara menyeluruh. Aktor Pengguna merepresentasikan pihak yang menggunakan sistem *chatbot* untuk memperoleh informasi akademik. Pengguna berinteraksi langsung dengan sistem melalui antarmuka web *chatbot*. *Use case* Mengirim Pertanyaan menggambarkan proses ketika pengguna memasukkan dan mengirimkan teks pertanyaan ke dalam sistem *chatbot*. Pertanyaan ini dapat berupa informasi terkait kegiatan akademik, prosedur administrasi, maupun layanan kemahasiswaan. *Use case* ini merupakan fungsi utama sistem karena menjadi titik awal pemrosesan informasi oleh *chatbot*. *Use case*

Memilih Tipe *Chatbot* menunjukkan bahwa pengguna memiliki kendali untuk menentukan mode *chatbot* yang akan digunakan, yaitu *rule-based* atau *hybrid*. Pemilihan tipe *chatbot* ini memengaruhi mekanisme pemrosesan pertanyaan dan pembentukan jawaban. Dengan adanya *use case* ini, sistem memberikan fleksibilitas kepada pengguna untuk memilih tingkat keandalan atau fleksibilitas respons yang diinginkan.

Use case Menerima Jawaban merepresentasikan proses di mana pengguna memperoleh respons dari *chatbot* setelah sistem memproses pertanyaan yang dikirimkan. Jawaban yang diterima dapat berasal dari basis pengetahuan (*rule-based*) atau hasil pemrosesan tambahan menggunakan *generative AI* pada mode *hybrid*. *Use case* ini menandai keberhasilan satu siklus interaksi antara pengguna dan sistem.

Use case Melihat Sejarah Percakapan memungkinkan pengguna untuk meninjau kembali percakapan yang telah dilakukan sebelumnya dengan *chatbot*. *Use case* ini memiliki relasi <<*include*>> dengan *use case* Menerima Jawaban, yang berarti setiap kali pengguna menerima jawaban dari *chatbot*, sistem secara otomatis menyimpan percakapan tersebut ke dalam riwayat. Dengan demikian, fungsi melihat sejarah percakapan tidak berdiri sendiri, melainkan selalu terkait dengan proses penerimaan jawaban.

Secara keseluruhan, *use case* diagram ini menunjukkan bahwa sistem *chatbot* dirancang untuk mendukung interaksi yang sederhana namun fleksibel, dengan memberikan kendali kepada pengguna dalam memilih tipe *chatbot* serta menyediakan fitur pendukung berupa penyimpanan dan peninjauan sejarah percakapan. Desain ini sejalan dengan tujuan sistem, yaitu menyediakan layanan informasi akademik yang interaktif, mudah diakses, dan dapat diandalkan.



Gambar 4.5 Activity Diagram

Activity diagram pada Gambar 4.5 menggambarkan alur interaksi antara pengguna (*User*) dan sistem *chatbot* dalam proses pemberian jawaban terhadap pertanyaan akademik. Diagram ini dibagi menjadi dua *swimlane*, yaitu *User* dan *Chatbot*, untuk memperjelas alur proses.

Alur dimulai ketika pengguna memasuki sistem *chatbot*, yang ditandai dengan *initial node*. Selanjutnya, pengguna melakukan proses pemilihan tipe *chatbot*, yaitu *rule-based* atau *hybrid*. Pemilihan ini memungkinkan pengguna menentukan jenis respons yang diinginkan sesuai dengan kebutuhan interaksi. Setelah tipe *chatbot* dipilih, alur bergabung kembali dan pengguna mengirimkan pertanyaan melalui sistem. Proses pengiriman pertanyaan ini menjadi titik transisi dari aktivitas pengguna ke aktivitas *chatbot*.

Setelah pertanyaan diterima, sistem *chatbot* melakukan beberapa tahapan pemrosesan. Pertama, *chatbot* membaca pertanyaan yang dikirim oleh pengguna. Selanjutnya, sistem melakukan proses koreksi kesalahan penulisan (*typo correction*) untuk meningkatkan akurasi pemahaman terhadap masukan pengguna. Setelah itu, pertanyaan dicocokkan dengan basis pengetahuan (*knowledge base*) yang dimiliki sistem. Berdasarkan hasil pencocokan tersebut,

sistem menentukan tipe *chatbot* yang akan digunakan dalam pembentukan jawaban, sesuai dengan pilihan pengguna atau mekanisme internal sistem. Apabila tipe *chatbot* yang digunakan adalah *rule-based*, sistem akan membentuk jawaban berdasarkan aturan dan data yang tersedia dalam basis pengetahuan. Sementara itu, jika tipe *chatbot* yang digunakan adalah *hybrid*, sistem akan membentuk jawaban dengan memanfaatkan kemampuan kecerdasan buatan *generative* sebagai pelengkap ketika aturan tidak menemukan kecocokan yang sesuai. Setelah jawaban dibentuk, alur dari kedua pendekatan tersebut digabungkan kembali, dan *chatbot* mengirimkan jawaban kepada pengguna. Pada sisi pengguna, jawaban yang dikirim oleh *chatbot* diterima dan ditampilkan kepada pengguna. Proses ini ditandai dengan *final node*, yang menunjukkan bahwa satu siklus interaksi telah selesai..

4.1.3 Hasil dari *Prototyping*

Pada iterasi pertama, *chatbot* hanya berfokus mengimplementasikan mekanisme *rule-based* dengan mode *generative* yang masih terbatas, seperti kode pada Gambar 4.6 yang menampilkan *prompt generative* yang masih sederhana dibandingkan iterasi selanjutnya. Total entri *knowledge* dalam iterasi ini berjumlah 15. Antarmuka pada tahap ini masih bersifat sederhana dan belum menyerupai belum mencerminkan lingkungan percakapan seperti aplikasi *chat*, seperti yang di tunjukkan pada Gambar 4.7.

```
prompt = (
    "Berikut adalah kumpulan informasi akademik:\n\n"
    f"{self.knowledge}\n\n"
    f"Pertanyaan: {user_input}\nJawaban:"
)
```

Gambar 4.6 Kode *Prompting Mode Hybrid* Iterasi Pertama

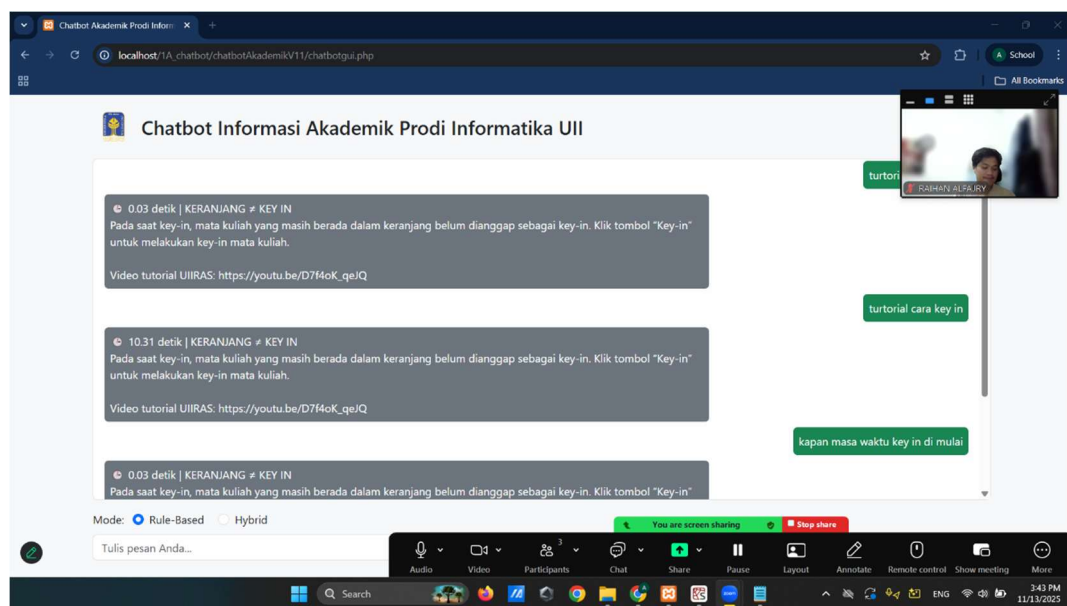


Gambar 4.7 Tampilan Iterasi Pertama

4.1.4 Hasil dari *Deployment, Delivery, and Feedback*

Deployment, Delivery, and Feedback, yang dilakukan dengan staf prodi dan mahasiswa seperti Gambar 4.8 menghasilkan berbagai macam masukan yang digunakan untuk meningkatkan kualitas *chatbot*. Masukan ini berupa:

- a. Bahasa *input*
Bahasa yang digunakan untuk bertanya ke *chatbot*.
- b. Sapaan
Ucapan sapaan pada *output chatbot*.
- c. Bahasa tampilan
Bahasa tampilan terlalu teknis.
- d. Grafik Tampilan
Tampilan antarmuka *chatbot*.



Gambar 4.8 *Testing Pengguna*

4.2 Iterasi Kedua

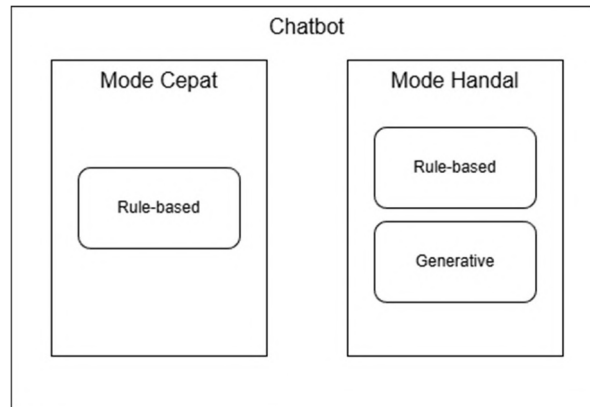
Pada iterasi kedua, dilakukan penyempurnaan terhadap sistem *chatbot* berdasarkan hasil evaluasi iterasi sebelumnya. Perbaikan difokuskan pada penyempurnaan sistem sesuai dengan umpan balik yang di dapatkan pada iterasi sebelumnya.

4.2.1 Hasil *Communication*

Pada *Communication* di iterasi kedua, dilakukan klarifikasi dari hasil *Deployment, Delivery, and Feedback* iterasi pertama. Dari klarifikasi ini ditemukan bahwa penguji/pengguna sering mengalami ketidaksesuaian respons ketika pertanyaan disampaikan dalam bentuk parafrase, bahasa informal, bahasa daerah, bahasa luar negeri, atau terdapat kesalahan penulisan. Pengguna juga menginginkan *chatbot* untuk dapat menjawab kalimat sapaan seperti selamat pagi agar *chatbot* terasa ramah. Di dalam *chatbot*, ada dua mode yang dapat di gunakan. Yaitu mode *rule-based* dan mode *hybrid* yang di cerminkan pada *chatbot* dalam adanya dua opsi yang dapat dipilih. Pengguna merasa bahwa kedua nama mode ini berkesan terlalu teknis dan membingungkan. Tampilan antarmuka *chatbot* iterasi pertama yang di tunjukkan pada Gambar 4.6 menurut pengguna terasa sulit untuk di baca dan dilihat. Pengguna menyatakan bahwa paduan warna kurang sesuai dan perlu diperbaiki. Permintaan untuk menghilangkan waktu pemrosesan pada *output* seperti yang dapat dilihat pada tampilan iterasi pertama dan dapat dikliknya *link* yang dibagikan juga disebutkan.

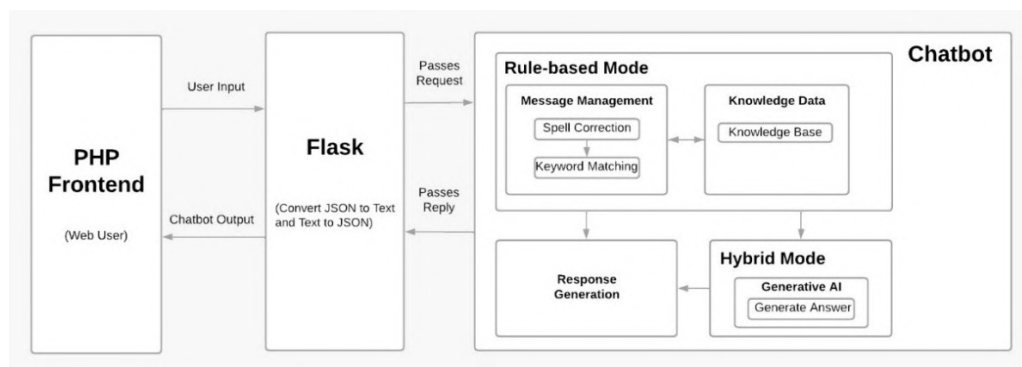
4.2.2 Hasil *Quick Plan and Quick Design*

Pada tahap ini, dilakukan kembali perencanaan dan perancangan berdasarkan informasi yang di dapatkan dari hasil *Communication* dan dilakukan juga penggambaran diagram sistem. Dilakukan penambahan kemampuan *chatbot* untuk dapat menjawab kalimat sapaan. Hal ini mungkin karena jawaban untuk kalimat sapaan ditambahkan ke dalam *knowledge chatbot*. Mode *chatbot* lebih dikembangkan lagi seperti yang dapat dilihat di gambar 4.9. Perkembangan ini berupa perubahan nama mode “*rule-based*” menjadi “Mode Cepat” dan mode “*Hybrid*” menjadi “Mode Handal”. Waktu pemrosesan di pindah keluar *output* dan menampilkan hitungan berapa lama *chatbot* berpikir. Beberapa perubahan pada tampilan ini, termasuk warna *bubble* teks dilakukan demi kenyamanan pengguna mengoperasikan *chatbot*.



Gambar 4.9 Struktur *Chatbot* Sempel

Seperti yang di tunjukkan Gambar 4.9, Tipe *chatbot* berupa “Mode Cepat” yang terdiri dari *rule-based* dan “Mode Handal” yang berupa *chatbot hybrid* dan terdiri dari *rule-based* dan *generative*. Kemudian pengguna mengirim pertanyaan. Di sini *chatbot* memproses pertanyaan, dan kemudian menjawabnya berdasarkan mode *chatbot* yang dipilih sebelumnya. Terakhir jawaban ditampilkan ke pengguna.



Gambar 4.10 Struktur *Chatbot*

Gambar 4.10 menjelaskan struktur *chatbot* secara rinci dan terdiri dari tiga tahap. Tahap-tahap ini berupa:

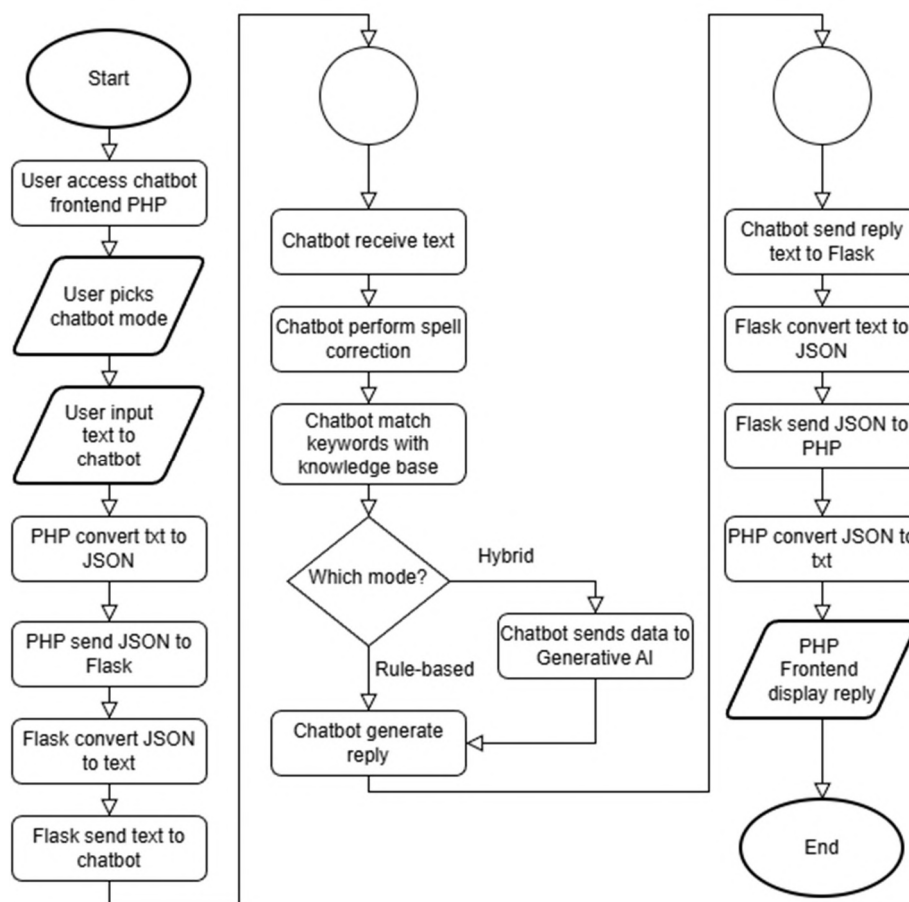
- a. Frontend
- b. Middleware
- c. Backend

Frontend

Dalam pengembangan aplikasi atau web, *frontend* merujuk pada bagian klien yang berinteraksi langsung dengan pengguna. *Frontend* mencakup semua elemen yang terlihat dan

interaktif di layar, termasuk tata letak, desain, warna, tombol, formulir, gambar, teks, dan animasi. Pada pengembangan ini, *frontend* menjadi perantara pengguna dengan sistem *chatbot*.

Flowchart pada Gambar 4.11 menggambarkan alur pemrosesan pertanyaan pengguna pada sistem *chatbot* berbasis web yang mengintegrasikan *frontend* berbasis PHP, *backend* berbasis Flask, serta dua mode *chatbot*, yaitu *rule-based* dan *hybrid*. Diagram ini menjelaskan aliran proses secara menyeluruh, mulai dari interaksi awal pengguna dengan sistem hingga penyajian jawaban *chatbot*, serta menunjukkan peran masing-masing komponen dalam mendukung layanan informasi akademik.



Gambar 4.11 Alur *Chatbot*

Proses dimulai ketika pengguna mengakses antarmuka *chatbot* melalui *frontend* berbasis PHP, yang ditandai dengan simbol Start. Antarmuka ini telah melalui beberapa iterasi perancangan berdasarkan kebutuhan dan umpan balik pengguna, dengan tujuan meningkatkan kemudahan penggunaan, kejelasan alur interaksi, serta kenyamanan dalam menyampaikan pertanyaan. Pada tahap ini, pengguna memilih mode *chatbot* yang akan digunakan, yaitu *rule-*

based atau *hybrid*, sebagai bentuk kendali pengguna terhadap karakteristik interaksi yang diinginkan. Setelah mode dipilih, pengguna memasukkan teks pertanyaan ke dalam sistem *chatbot* melalui antarmuka.

Gambar 4.12 Menunjukkan *session* untuk dimasukkan ke dalam sejarah percakapan akar bisa dilihat. Teks pertanyaan yang diterima kemudian dikonversi oleh *frontend* PHP ke dalam format JSON. Penggunaan JSON dipilih karena merupakan standar pertukaran data yang ringan, terstruktur, dan kompatibel dengan berbagai platform serta bahasa pemrograman, sehingga memudahkan komunikasi antara *frontend* PHP dan *backend*. Setelah proses konversi selesai, *frontend* mengirimkan data JSON tersebut ke server Flask untuk diproses lebih lanjut.

```
chat_history|a:2:{i:0;a:2:{s:6:"sender";s:4:"user";s:7:"message";s:36:"habis selesai pendadaran ngapain ya?";}i:1;a:2:{s:6:"sender";s:3:"bot";s:7:"message";s:293:"Informasi tentang alur pasca pendadaran dapat dilihat di: https://drive.google.com/file/d/1W3XkpMZB4OYXBNaZbYZBJ6c414_IBAlq/view Untuk pendaftaran pendadaran reguler dapat di lakukan di: https://docs.google.com/forms/d/15mVPvaJHGmM5DB-bee0l702zDnOwdKqgznpmACXgpzI/viewform?edit_requested=true";}}
```

Gambar 4.12 Sesi Flask

Middleware

Dalam pengembangan aplikasi atau web, *middleware* adalah lapisan *software* yang berada di antara *client requests* dan *backend*. Komunikasi antara *frontend* PHP dan Python difasilitasi melalui *framework* Flask versi 3.1.2 sebagai *Application Programming Interface* (API) yang berfungsi sebagai *middleware* untuk menangani pertukaran data berbasis JSON. Penggunaan Flask dipilih karena sifatnya yang minimalis, fleksibel, dan efisien dalam mengelola komunikasi *client-server* tanpa menambah kompleksitas sistem yang tidak diperlukan.

Backend

Backend merujuk pada bagian *server-side* dari sebuah situs web atau aplikasi yang beroperasi di belakang layar, ini ditunjukkan dengan detail di Gambar 4.11. Perannya di sini adalah menerima teks pertanyaan dari Flask dan melakukan tahap pra-pemrosesan, yaitu koreksi ejaan (*spell correction*). Proses koreksi ejaan ini menggunakan algoritma Ratcliff/Obershelp, yang bekerja dengan mengukur tingkat kemiripan *string* berdasarkan karakter terpanjang yang sama. Pemilihan algoritma ini bertujuan untuk meningkatkan toleransi sistem terhadap kesalahan penulisan, variasi ejaan, maupun perbedaan kecil dalam

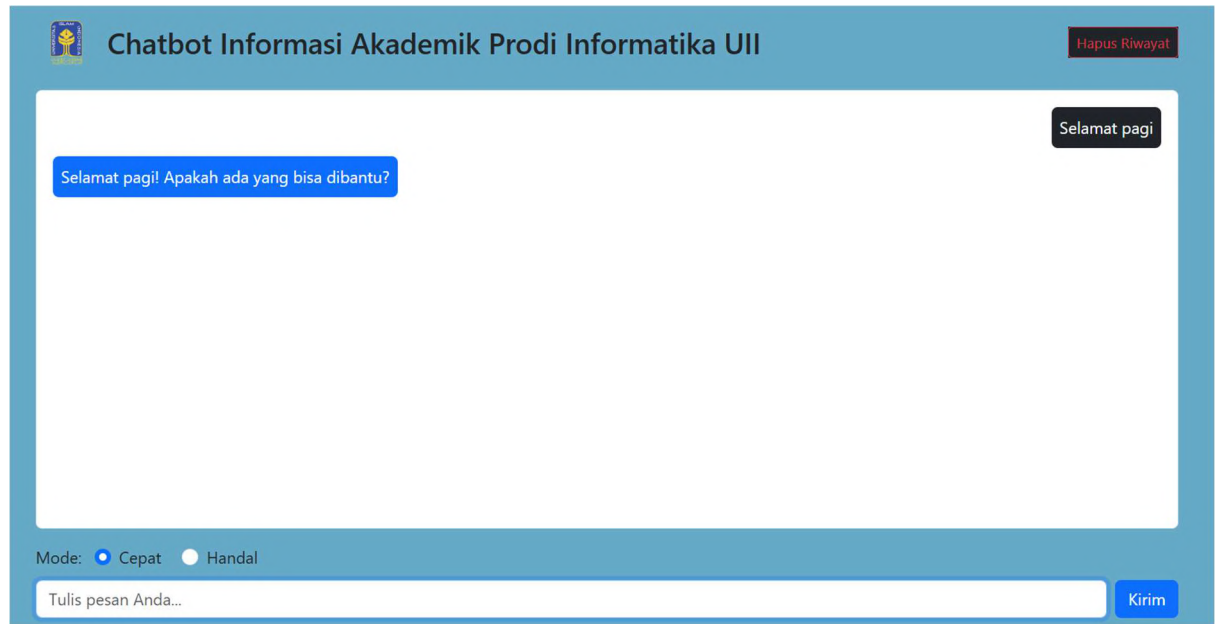
masukannya, sehingga pertanyaan dapat diproses secara lebih akurat pada tahap selanjutnya. Ada beberapa alasan lain yang menyebabkan terpilihnya algoritma ini. Yaitu adanya *import difflib* di *Python* yang memudahkan penerapannya secara langsung. Nilai *cutoff* 0.75 dipilih karena nilai similaritas 75% sudah ideal untuk menangkap *typo*.

Jika mode yang digunakan adalah *rule-based*, *chatbot* akan langsung menghasilkan jawaban berdasarkan aturan dan informasi yang telah didefinisikan sebelumnya dalam basis pengetahuan. Pendekatan ini memastikan konsistensi dan keandalan jawaban, terutama untuk pertanyaan akademik yang bersifat faktual dan terstruktur. Sebaliknya, jika mode yang digunakan adalah *hybrid*, *chatbot* akan mengirimkan data pertanyaan yang relevan ke modul *generative AI*. Modul ini berperan sebagai mekanisme pendukung untuk membentuk jawaban yang lebih fleksibel dan kontekstual ketika aturan yang tersedia tidak sepenuhnya mampu menangani variasi pertanyaan pengguna. Hasil dari pemrosesan *generative AI* kemudian dikembalikan ke *chatbot* dan digunakan sebagai bagian dari jawaban akhir, dengan tetap mempertahankan kendali sistem untuk meminimalkan risiko kesalahan informasi.

Setelah jawaban akhir dihasilkan, *chatbot* mengirimkan teks jawaban tersebut ke *backend* Flask. Flask kemudian mengonversi jawaban ke dalam format JSON dan mengirimkannya kembali ke *frontend* PHP. Pada sisi *frontend*, PHP menerima data JSON tersebut, mengonversinya kembali menjadi teks, dan menampilkan jawaban *chatbot* kepada pengguna melalui antarmuka web yang telah dirancang secara iteratif. Proses ini menandai akhir dari satu siklus interaksi antara pengguna dan sistem *chatbot*, yang ditunjukkan dengan simbol *End* pada *flowchart*.

4.2.3 Hasil dari *Prototyping*

Prototipe tahap kedua berupa perbaikan tampilan dan penambahan kode dapat dilihat pada gambar di bawah. Gambar 4.13 menggambarkan tampilan *chatbot* akhir yang lengkap berdasarkan hasil konsiderasi umpan balik pengguna. Pada gambar 4.14 Ditampilkan kode bagian *prompting* yang lebih diperinci agar hasil *output* sesuai dengan yang diinginkan.



Gambar 4.13 Tampilan Iterasi Kedua

```

prompt = (
    "Jawab dalam bahasa Indonesia.\n"
    "Kamu adalah asisten akademik.\n"
    "Jawablah HANYA berdasarkan informasi berikut.\n"
    "Jika informasi tidak tersedia di teks, jawab: "
    "'Maaf, saya tidak tahu berdasarkan data yang tersedia.'\n\n"
    f"Informasi:\n{best_response}\n\n"
    f"Pertanyaan: {user_input}\n"
    "Jawaban yang sopan dan ringkas:\n"
)

```

Gambar 4.14 Kode *Prompting Mode Hybrid* Iterasi Kedua

4.2.4 Hasil dari *Deployment, Delivery, and Feedback*

Staf Program Studi menyatakan bahwa tampilan antarmuka *chatbot* yang dikembangkan sudah baik, rapi, dan mudah digunakan. Desain visual serta alur interaksi yang didapat dinilai memuaskan dan telah mampu mendukung kebutuhan pengguna dalam memperoleh informasi akademik.

Program Studi berharap agar kemampuan *chatbot* dalam menjawab pertanyaan dapat terus ditingkatkan pada pengembangan selanjutnya. Peningkatan terutama diharapkan pada aspek kelengkapan informasi, ketepatan jawaban, serta kemampuan *chatbot* dalam menangani variasi pertanyaan yang lebih kompleks agar dapat memberikan layanan yang lebih ideal.

4.3 Analisis Hasil Penelitian

Analisis hasil penelitian berisikan informasi berguna yang didapatkan dari proses iterasi berdasarkan proses pengujian dan data-data yang didapatkan dari proses pengujian tersebut.

4.3.1 Umpan Balik

Hasil umpan balik pada Tabel 4.1 yang digunakan pada tahap *Communication* dan *Deployment, Delivery, and Feedback*. Proses ini dilakukan guna mendapatkan umpan balik dari pihak terkait untuk kegunaan penelitian.

Tabel 4.1 Umpan Balik

Kantor Informatika FTI		Zoom Mahasiswa	
Tanggal	Umpan Balik	Tanggal	Umpan Balik
15/09/2025	Chatbot dapat menjawab pertanyaan seputar akademik prodi dan dapat dengan mudah di gunakan.	11/12/2025	Penambahan <i>knowledge</i> . <i>Output</i> dari <i>chatbot</i> tidak usah panjang. Bahasa Tampilan jangan teknis (mode <i>rule-based, hybrid</i>) Kalimat sapaan
13/11/2025	Penambahan <i>knowledge</i> Dihilangkannya waktu dan dapat dikliknya <i>link</i> pada <i>output chatbot</i> . Warna kurang serasi	13/11/2025	Bahasa Tampilan jangan teknis. Waktu di <i>output chatbot</i> tidak usah.

4.3.2 Konfigurasi

Bagian ini menjelaskan konfigurasi eksperimen dan prosedur pengujian yang digunakan untuk mengevaluasi kinerja *chatbot* informasi akademik *dual-mode* yang diusulkan. Evaluasi difokuskan pada tingkat akurasi respons, waktu respons, serta perilaku sistem dalam memproses pertanyaan nyata dari mahasiswa yang mencerminkan kebutuhan informasi akademik umum pada UII.

Lingkungan pengujian menggunakan *workstation* dengan sistem operasi Windows 11 (64-bit) yang dilengkapi prosesor AMD Ryzen 5 7535HS (enam inti dan dua belas *thread*, 3,3 GHz) serta memori sebesar 16 GB RAM. Antarmuka *frontend* dikembangkan menggunakan PHP

versi 8.2.12, sedangkan mesin *backend* yang mencakup komponen *rule-based* dan modul penalaran *hybrid* diimplementasikan menggunakan *Python* versi 3.11.11.

Komunikasi antara *frontend* PHP dan mesin *Python* difasilitasi melalui *framework* Flask versi 3.1.2 yang berfungsi sebagai *middleware* ringan untuk menangani pertukaran data berbasis JSON. Penggunaan Flask dipilih karena sifatnya yang minimalis, fleksibel, dan efisien dalam mengelola komunikasi *client-server* tanpa menambah kompleksitas sistem yang tidak diperlukan. Pada mode *hybrid*, sistem memanfaatkan model Mistral-7B-Instruct-v0.2.Q4_K_M yang dijalankan secara lokal dalam format GGUF terkuantisasi dan didistribusikan di bawah lisensi Apache 2.0 (Apache Software Foundation, 2004; TheBloke, 2023) Model Mistral dipilih karena menawarkan keseimbangan antara performa dan kebutuhan sumber daya, sehingga dapat dianggap sebagai *middle ground* dibandingkan model berukuran lebih kecil yang cenderung memiliki keterbatasan kemampuan penalaran, maupun model berukuran lebih besar yang membutuhkan sumber daya komputasi yang signifikan. Dengan ukuran parameter yang relatif moderat dan hasil inferensi yang cukup baik untuk tugas pemahaman bahasa alami, Mistral-7b-Instruct-v0.2.Q4_K_M dinilai mampu memberikan respons yang kontekstual tanpa mengorbankan efisiensi sistem. Selain itu, penerapan memungkinkan pengurangan penggunaan memori dan peningkatan kecepatan inferensi, sehingga model tetap dapat dijalankan secara lokal sesuai dengan keterbatasan perangkat keras yang tersedia, tanpa bergantung pada layanan komputasi eksternal. Detail perbandingan model dapat dilihat pada Tabel 4.2 yang didapatkan langsung dari situs Hugging Face.

Tabel 4.2 Perbandingan Model Mistral

<i>Name</i>	<i>Quant method</i>	<i>Bits</i>	<i>Size (GB)</i>	<i>Max RAM required (GB)</i>	<i>Use case</i>
mistral-7b-instruct-v0.2.Q2_K.gguf	Q2_K	2	3.08	5.58	<i>smallest, significant quality loss - not recommended for most purposes</i>
mistral-7b-instruct-v0.2.Q4_K_M.gguf	Q4_K_M	4	4.37	6.87	<i>medium, balanced quality - recommended</i>
mistral-7b-instruct-v0.2.Q8_0.gguf	Q8_0	8	7.70	10.20	<i>very large, extremely low quality loss - not recommended</i>

4.3.3 Pengambilan Data

Dilakukan berbagai analisis dan penjabaran data dalam bentuk tabel dan penjelasan dari data-data tabel tersebut. Analisis ini mencakup:

- a. Analisis Perilaku *Chatbot* pada Mode *Rule-based* dan *Hybrid*
- b. Analisis Akurasi *Chatbot*
- c. Analisis Waktu Respons

Analisis Perilaku *Chatbot* pada Mode *Rule-based* dan *Hybrid*

Perilaku *chatbot* pada kedua mode penalaran dianalisis secara kualitatif melalui contoh interaksi yang disajikan pada Tabel 4.3. Contoh-contoh tersebut menunjukkan perbedaan karakteristik respons antara mode *rule-based* dan mode *hybrid* terhadap berbagai jenis masukan pengguna.

Tabel 4.3 *Sample* Pertanyaan

No	<i>User Input</i>	Hasil <i>Rule-based</i>	Hasil <i>Hybrid</i>
1	Halo selamat pagi	Benar	Benar
2	informasi matakuliah smester ahir	Salah	Benar
3	informasi matakuliah smester ahir deh misalnya	Salah	Benar
4	tutorial key in	Benar	Benar
5	bagikan kalender akademik	Benar	Benar
6	kontak dosen di mana ya? terimakasih	Benar	Salah
7	jadwal	Benar	Salah
8	Syarat skripsi	Salah	Salah
9	Kapan masa waktu key in dimulai	Salah	Benar
10	KKN	Benar	Benar

Penilai benar atau salah menggunakan metode tertentu. Jawaban *chatbot* dikategorikan sebagai *benar* apabila:

- a. *Chatbot* memberikan informasi yang sesuai dengan *knowledge base* ketika data yang ditanyakan tersedia di *knowledge base*.
- b. *Chatbot* memberikan respons “Tidak tahu” atau respons serupa ketika informasi yang ditanyakan tidak tersedia dalam *knowledge base*.

Sebaliknya, jawaban dikategorikan sebagai salah apabila *chatbot* memberikan informasi yang tidak sesuai atau memberikan jawaban “Tidak tahu” walaupun sebenarnya *chatbot* tahu dan memiliki jawabannya di *knowledge base*. Contoh *output* dapat dilihat pada Tabel 4.4.

Pada kasus di mana pertanyaan pengguna sesuai dengan kata kunci yang tersedia dalam basis pengetahuan, kedua mode mampu memberikan jawaban yang benar dan konsisten. Hal ini menunjukkan bahwa basis pengetahuan yang dikurasi telah mencakup sebagian besar pertanyaan akademik umum. Namun, pada pertanyaan yang disampaikan dalam bentuk

parafrase, bahasa informal, atau mengandung kesalahan penulisan, mode *rule-based* sering gagal memberikan respons yang tepat. Sebaliknya, mode *hybrid* mampu menghasilkan jawaban yang relevan melalui pemanfaatan mekanisme *generative*.

Di sisi lain, terdapat beberapa kasus di mana mode *rule-based* memberikan respons yang akurat, sedangkan mode *hybrid* menghasilkan jawaban yang kurang tepat atau terlalu berhati-hati. Hal ini mengindikasikan bahwa meskipun mode *hybrid* memiliki fleksibilitas linguistik yang lebih tinggi, interpretasi *generative* terhadap konteks yang terbatas dapat menyebabkan respons yang kurang optimal. Temuan ini sejalan dengan karakteristik umum sistem berbasis LLM, yang mampu menghasilkan respons kaya konteks namun berpotensi mengalami ketidaktepatan respons (Ji et al., 2023).

Tabel 4.4 Contoh Respons *Chatbot Rule-based* dan *Hybrid*

No	User Input	Rule-based Output	Hasil Rule-based	Hybrid Output	Hasil Hybrid
1	Bagikan kalender akademik	Kalender akademik FTI dapat dilihat di: https://fit.uui.ac.id/blog/2022/07/05/kalender-akademik/	Benar	Saya akan menyediakan Anda dengan referensi terhadap kalender akademik FTI yang bisa dibaca secara detail di link berikut: https://fit.uui.ac.id/blog/2022/07/05/kalender-akademik/ Note: Pastikan link yang disediakan benar dan aktif.	Benar
2	jadwal	Dapat dilihat di isian gateway: Gateway > RAS > Isian. Juga dapat dilihat di jadwal kuliah: Gateway > Akademik > Jadwal Kuliah.	Benar	Maaf, saya tidak tahu jadwal kuliah berdasarkan data yang tersedia saat ini.	Salah

3	Kapan masa waktu key in	KERANJANG \neq KEY IN Pada saat kuliah yang masih berada dalam keranjang belum dianggap sebagai key-in. Klik tombol “Key-in”	Salah	Maaf, saya tidak tahu berdasarkan data yang tersedia. Waktu mulai key-in mata kuliah di UIIRAS tidak disediakan dalam informasi yang diberikan. Silakan melihat video tutorial untuk memperoleh lebih banyak informasi.	Benar
4	Syarat skripsi	Syarat kehadiran kuliah perkelas adalah hadir 75% perkuliahan.	Salah	Untuk memenuhi syarat skripsi, mahasiswa harus hadir dalam 75% dari semester kuliah skripsi	Salah

Analisis Akurasi *Chatbot*

Hasil pengujian akurasi *chatbot* ditunjukkan pada Tabel 4.5. Pengujian dilakukan dengan membandingkan jumlah respons yang benar terhadap total pertanyaan yang diajukan pada mode masing-masing. Persamaan (4.1) digunakan untuk mendapatkan nilai akurasi setiap mode. Mode *rule-based* dapat menjawab enam dari sepuluh pertanyaan dan mode *hybrid* dapat menjawab tujuh dari sepuluh pertanyaan. Jumlah data uji yang digunakan masih terbatas, sehingga hasil evaluasi bersifat indikatif dan belum dapat merepresentasikan performa sistem secara menyeluruh.

Tabel 4.5 Tabel Akurasi

<i>Mode</i>	<i>Accuracy</i>
<i>Rule-based</i>	60%
<i>Hybrid</i>	70%

$$\text{Akurasi} = \frac{\text{Jumlah Jawaban Benar}}{\text{Jumlah Data Uji}} \cdot 100\% \quad (4.1)$$

Berdasarkan hasil pengujian, *chatbot rule-based* mencapai tingkat akurasi sebesar 60%. Kesalahan yang terjadi umumnya disebabkan oleh keterbatasan pencocokan kata kunci, terutama pada pertanyaan yang mengandung parafrase, variasi bahasa, atau informasi yang belum tercakup dalam basis pengetahuan. Keterbatasan ini merupakan karakteristik dari

pendekatan deterministik, sebagaimana juga dilaporkan dalam penelitian sebelumnya (Adamopoulou & Moussiades, 2020).

Sementara itu, *chatbot hybrid* mencapai tingkat akurasi sebesar 70%. Peningkatan akurasi ini diperoleh dari kemampuan mekanisme *generative* dalam menangani variasi bahasa dan struktur pertanyaan yang lebih fleksibel. Namun, pada beberapa kasus ditemukan respons yang bersifat halusinasi, di mana jawaban yang dihasilkan tampak meyakinkan tetapi kurang sesuai dengan informasi akademik yang sebenarnya. Fenomena ini sejalan dengan temuan Ji et al. (2023) mengenai keterbatasan model *generative*.

Hasil ini menunjukkan adanya *trade-off* antara fleksibilitas linguistik dan keandalan respons, di mana *chatbot hybrid* unggul dalam pemahaman bahasa tetapi tetap memiliki risiko kesalahan *generative*.

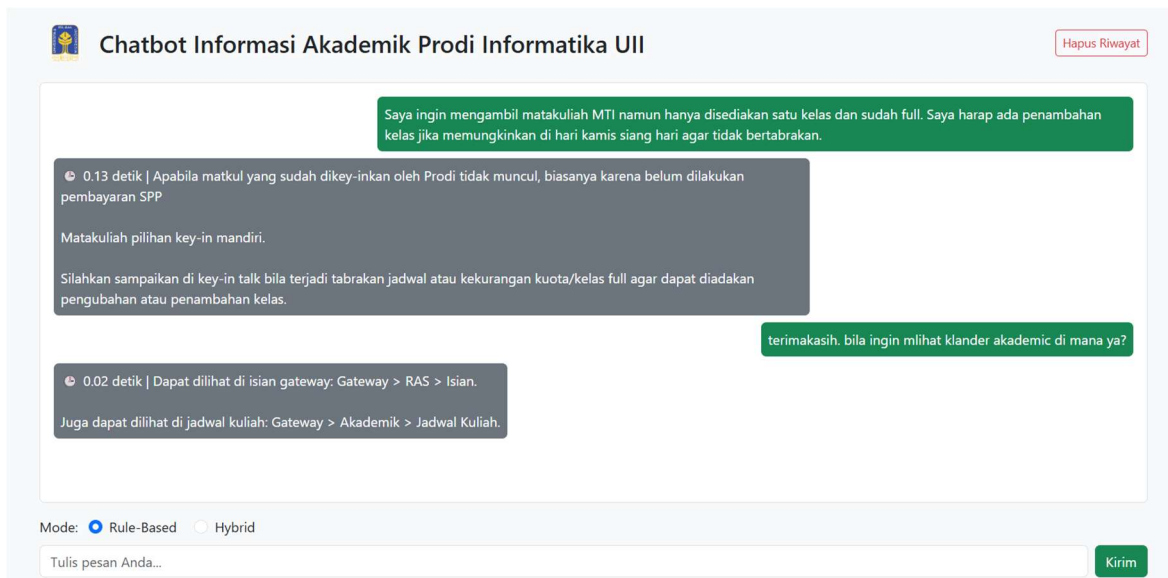
Analisis Waktu Respons

The screenshot displays a chatbot interface titled "Chatbot Informasi Akademik Prodi Informatika UII". The chat history shows the following interaction:

- User:** Saya ingin mengambil matakuliah MTI namun hanya disediakan satu kelas dan sudah full. Saya harap ada penambahan kelas jika memungkinkan di hari kamis siang hari agar tidak bertabrakan.
- Chatbot:** 99.78 detik | Jika ada kesempatan, Anda dapat mengirim permintaan penambahan kelas melalui form penambahan kelas di UIIRAS atau melalui key-in talk. Jika masa key-in belum tertutup, prodi akan melakukan perubahan jadwal atau penambahan kelas sesuai kebutuhan.
- User:** terimakasih. bila ingin mlihat klander akademik di mana ya?
- Chatbot:** 48.68 detik | Untuk melihat kalender akademik FTI, silahkan kunjungi link <https://fit.uui.ac.id/blog/2022/07/05/kalender-akademik/>. Informasi lebih lanjut tentang kalender akademik dapat Anda dapatkan secara langsung melalui website atau kontak Dekan FTI. Terima kasih atas pertanyaan Anda.

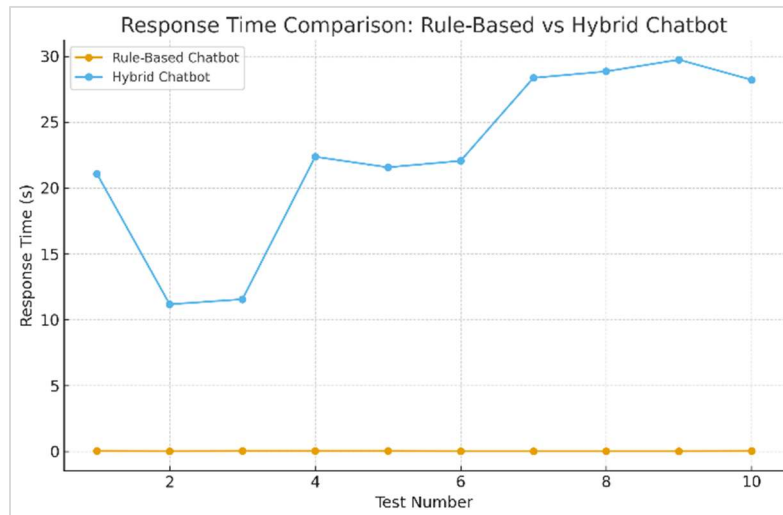
At the bottom of the interface, there is a "Mode" selector with "Rule-Based" and "Hybrid" options, and a "Kirim" button.

Gambar 4.15 Testing Waktu *Hybrid* (Mode Handal)

Gambar 4.16 Testing Waktu *Rule-based* (Mode Cepat)

Tabel 4.6 Tabel Waktu

No	<i>Rule-based Chatbot</i> (detik)	<i>Hybrid Chatbot</i> (detik)
1	0.03	21.08
2	0.02	11.17
3	0.03	11.55
4	0.03	22.37
5	0.03	21.57
6	0.02	22.06
7	0.02	28.36
8	0.02	28.85
9	0.02	29.74
10	0.03	28.21
Rata-rata	0.025	25.268
Standar Deviasi	0.005	6.758



Gambar 4.17 Perbandingan Waktu

Pengujian waktu respons dilakukan dengan mengukur selang waktu antara pengiriman permintaan dalam format JSON oleh *frontend* dan penerimaan respons JSON dari Flask API seperti yang dapat dilihat pada Gambar 4.15 dan Gambar 4.16. Hasil pengujian ditampilkan pada Tabel 4.6 dan terlihat lebih jelas pada grafik Gambar 4.17, di mana *rule-based* lebih rendah dan rata yang menunjukkan cepat dan konsistennya waktu menjawab pada setiap tes, dan *hybrid* lebih tinggi dan beragam yang menunjukkan lebih lama dan berbedanya kecepatan di setiap tes.

Chatbot rule-based menunjukkan waktu respons yang sangat cepat dan stabil, dengan rata-rata waktu sebesar 0,025 detik. Hal ini disebabkan oleh proses pencocokan kata kunci yang ringan dan tidak melibatkan komputasi berat. Sebaliknya, *chatbot hybrid* memiliki waktu respons yang jauh lebih tinggi, dengan rata-rata 25,268 detik. Waktu ini dipengaruhi oleh proses pemuatan model, *tokenisasi*, dan generasi teks pada model Mistral-7B-Instruct, yang memang membutuhkan sumber daya komputasi yang lebih besar (Halvonik & Kapusta, 2024).

Perbedaan waktu respons ini menunjukkan bahwa mode *rule-based* sangat cocok untuk layanan informasi akademik yang membutuhkan jawaban cepat, sedangkan mode *hybrid* lebih sesuai untuk pertanyaan yang kompleks dan tidak terstruktur meskipun dengan latensi yang lebih tinggi.

4.3.4 Kendala

Beberapa kendala yang ditemukan dapat mempengaruhi performa *chatbot* yang dikembangkan melalui metode *prototyping* dapat berupa:

- a. Halusinasi.
- b. *Typo* saat melakukan *input*.
- c. Menggunakan bahasa yang tidak ada pada *chatbot* saat melakukan *input*. Seperti bahasa gaul, bahasa daerah, dan bahasa luar negeri.
- d. Kurangnya data *knowledge*.
- e. Kurang rincinya data *knowledge*.
- f. Kesalahan pada data *knowledge*.
- g. Spesifikasi sistem.
- h. Arsitektur sistem.
- i. Antarmuka sistem.
- j. Kekeliruan saat melakukan proses *prototyping*.
- k. Kekeliruan *chatbot* saat memberikan informasi.

Halusinasi menjadi salah satu tantangan besar pada pengembangan *chatbot* berbasis *generative*. Hal ini dapat di hindari dengan tidak menggunakan unsur *generative*, seperti menggunakan *chatbot hybrid rule-based* saja atau menerapkan *chatbot hybrid* paduan dari *rule-based* dan *retrieval* dan kombinasi paduan lainnya. Hal ini juga dapat diperkecil dampaknya dengan menerapkan standar *input* yang andal: seperti *prompting input* ke sistem *generative* yang baik dan tuntas, penambahan data *knowledge* yang lebih banyak, penambahan data *knowledge* yang lebih rinci.

Typo merupakan sesuatu yang tidak dapat di hindari. Pada sistem *chatbot* ini, diterapkan algoritma perbaikan *typo* pada semua mode *chatbot*. Alurnya berupa *chatbot* menerima *input*, *chatbot* memproses *input* dengan menerapkan algoritme pembenaran ejaan Ratcliff/Obershelp, seterusnya *input* akan diproses sesuai mode *chatbot* masing-masing seperti yang sudah dijelaskan di bagian sebelumnya.

Knowledge adalah hal terpenting kedua pada sistem *chatbot* setelah *engine chatbot* itu sendiri. Alangkah baiknya bila *Knowledge* diperlakukan dengan sangat hati-hati. Banyak hal yang bisa gagal pada komponen ini seperti kurangnya data, kurang rincinya data, dan kesalahan memasukkan data *knowledge* yang akan membuat *debugging* sulit untuk menemukan kesalahan sistem keseluruhannya berdasarkan pengalaman penulis.

Backend merupakan sesuatu yang penting pada suatu perangkat lunak. Spesifikasi sistem tergantung pada sistem server. Untuk penelitian ini, laptop dengan spesifikasi yang sudah

dicantumkan pada bab sebelumnya berperan sebagai server. Hal ini dapat mempengaruhi kecepatan *chatbot* untuk membalas pertanyaan.

Arsitektur sistem juga dapat mempengaruhi performa *chatbot*. Pada penelitian ini dua *file* Python untuk dua mode yang berbeda dijalankan secara bersamaan untuk mempermudah pembuatan sistem, mendapatkan data performa, dan mempermudah menemukan eror saat *debugging*.

Bagian *frontend* sering terlihat kurang penting. Tetapi komponen ini bisa memiliki dampak pada persepsi pengguna dan bahkan sistem *chatbot* itu sendiri. Pengguna dapat merasa kewalahan bila terlalu banyak fitur yang ada. Terlalu banyak komponen dekoratif seperti animasi JavaScript juga dapat memperlambat fungsionalitas sistem pada perangkat yang kurang modern.

Untuk tampilan disengajakan untuk berbentuk sederhana layaknya aplikasi pesan pada umumnya. Tampilan sederhana ini diharapkan dapat mempermudah pengguna dalam menggunakan dan menerima informasi dari *chatbot*. Untuk fitur yang relatif cukup minimalis diterapkan untuk membuat pengguna tidak terintimidasi atau kewalahan dan menggunakan *chatbot*.

Proses *prototyping* walaupun tidak terkesan teknis, sama pentingnya. Banyaknya permasalahan yang dapat muncul pada bagian ini. Contohnya proses komunikasi, bila tahap komunikasi gagal atau terjadi kesalahpahaman akan mengakibatkan tahap selanjutnya terganggu karena kurangnya info yang digunakan untuk mendasari penelitian. Tetapi metode *prototyping* akan lebih memaafkan kesalahan dibandingkan metode lain seperti *waterfall* karena metode ini utamanya adalah mengulang tahap pada iterasi yang berbeda.

Kekeliruan *chatbot* terutama *chatbot hybrid*, saat memberikan informasi dapat terjadi karena salah satu penyebab utamanya, yaitu keterbatasan *knowledge base* yang digunakan. Ketika basis pengetahuan hanya berisi entri tertentu (misalnya topik “form checklist”), *chatbot* tidak memiliki konteks yang cukup untuk menjawab pertanyaan di luar cakupan tersebut. Meskipun *prompt* telah membatasi jawaban agar hanya berdasarkan informasi yang tersedia, model tetap berusaha menghasilkan respons dari data yang paling mendekati, yaitu entri yang paling banyak mengandung kata kunci yang mirip dengan *input* pengguna.

Selain itu, kemungkinan terjadinya kesalahan interpretasi *input* juga dapat berperan. Model generatif bekerja dengan mengenali pola dan kemiripan kata kunci. Jika pertanyaan pengguna tidak eksplisit dengan isi *knowledge base*, sistem pencarian akan memilih respons dengan kemiripan tertinggi berdasarkan kata kunci terdekat.

Berbagai hal ini harus diperhatikan saat membuat sistem *chatbot* dan berbagai sistem lainnya. Dari banyak aspek yang ditemukan di penelitian ini, seperti ukuran *AI* lokal, kepuasan pengguna, fitur sistem, antarmuka, dan *backend*, pembuatan sistem harus mencari titik tengah dari berbagai komponen untuk membuat sistem yang baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

- a. Dengan menerapkan pendekatan *prototyping* dengan empat tahap utama (*Communication, Quick Plan and Quick Design, Prototyping, serta Deployment, Delivery, and Feedback*) pengembang dapat membangun *dual-mode rule-based generative chatbot*. Beberapa kesimpulan tahap *prototyping* berupa:
 1. *Communication*

Melibatkan wawancara dengan staf akademik Prodi Informatika UII dan observasi pertanyaan mahasiswa untuk menyusun *knowledge base* berupa pasangan *keywords* dan *responses* (mencakup topik prodi, mata kuliah, tugas akhir, dan key-in).
 2. *Quick Plan and Quick Design*

Merancang arsitektur *multi-layer* (*frontend PHP, middleware Flask, backend Python*) beserta *use case* dan *activity diagram*, termasuk mekanisme koreksi ejaan menggunakan algoritma Ratcliff/Obershelp.
 3. *Prototyping*

Dilakukan secara bertahap dalam empat iterasi, dimulai dari rule-based sederhana hingga integrasi *hybrid* dengan Mistral-7B-Instruct lokal (GGUF *quantized*). Di sini dapat dilihat kekurangan dan keunggulan kedua tipe *chatbot* yang kemudian akan diujikan di tahap selanjutnya.
 4. *Deployment, Delivery, and Feedback*

Dilakukan pada Windows 11 dan ditujukan untuk mendapatkan hasil uji yang konsisten. Menerima umpan balik yang akurat, dapat meningkatkan kualitas sistem di iterasi berikutnya.
- b. Perbandingan dilakukan melalui pengujian akurasi, waktu respons, dan analisis kualitatif pada 10 sampel pertanyaan akademik nyata. Beberapa jenis pengujian berupa:
 1. Akurasi

Mode *rule-based* mencapai 60%, unggul pada pertanyaan dengan kecocokan *keyword*, tetapi rentan gagal pada variasi kata atau *typo*. Mode *hybrid* mencapai

70%, lebih baik dalam menangani parafrase dan bahasa informal berkat komponen *generative*.

2. Waktu Respons

Mode *rule-based* sangat cepat dengan rata-rata 0,025 detik (SD 0,005) karena hanya melakukan pencocokan *keyword*. Mode *hybrid* jauh lebih lambat dengan rata-rata 25,268 detik (SD 6,758) akibat proses pemuatan model, *tokenisasi*, dan generasi teks oleh Mistral-7B.

3. Analisis Kualitatif

Mode *rule-based* konsisten, andal, dan bebas halusinasi untuk pertanyaan terstruktur. Mode *hybrid* lebih fleksibel secara linguistik, namun memiliki risiko halusinasi meskipun telah diminimalkan melalui prioritas *rule-based* dan *prompting* yang ketat.

c. Komponen generatif (Mistral-7B-Instruct) digunakan ketika tidak mendapatkan hasil yang memuaskan, sehingga *rule-based* tetap menjadi prioritas untuk menjaga keandalan faktual. Aspek yang dapat menunjukkan keandalan berupa:

1. Efektivitas

Terlihat dari peningkatan akurasi keseluruhan dari 60% (*rule-based* murni) menjadi 70% (*hybrid*), terutama pada kasus yang melibatkan *typo*, parafrase, atau bahasa informal yang tidak tertangani oleh pencocokan *keyword*.

2. Mengurangi halusinasi

Risiko halusinasi diminimalkan melalui strategi prioritas pengutamaan tipe *rule-based* dibandingkan *generative*, *prompting* yang *rigid* dan jelas dapat membatasi jawaban hanya pada informasi *knowledge base* yang diberikan, serta koreksi ejaan awal.

Pendekatan ini berhasil mencapai keseimbangan antara fleksibilitas bahasa *input-output* dan konsistensi faktual, ini dapat menjadikan *generative* sebagai pelengkap yang efektif tanpa mengorbankan keandalan inti pada domain informasi akademik.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasannya, pengembangan *chatbot* informasi akademik masih memiliki beberapa peluang perbaikan di masa depan:

a. Pengelolaan *Knowledge*

Perluas dan perbarui *knowledge* secara berkala untuk menangani lebih banyak variasi pertanyaan. Eksplorasi struktur alternatif seperti flat FAQ, hierarkis, *knowledge graph*, atau *vector-based retrieval* untuk membandingkan akurasi, kecepatan, dan pemahaman konteks. *Knowledge* juga dapat ditingkatkan secara dinamis dengan memahami informasi eksternal, baik dari pemberitahuan universitas sampai informasi yang disampaikan oleh pengguna.

b. Pra-pemrosesan *Input*

Meningkatkan kualitas koreksi ejaan dengan membandingkan berbagai algoritma (seperti *string similarity*, statistik, atau *machine learning*) yang paling sesuai dengan pola bahasa dan kesalahan pengguna di konteks akademik.

c. Model *Generative AI*

Uji variasi model Mistral lain (yang lebih ringan untuk efisiensi, lebih besar untuk kualitas jawaban) serta model lain di luar Mistral, untuk membandingkan performa, kebutuhan sumber daya, dan risiko halusinasi.

d. Skala Sistem

Kembangkan *chatbot* dari lingkup Program Studi Informatika menjadi skala universitas (seluruh UII), dengan integrasi data lintas fakultas serta penanganan tantangan konsistensi informasi dan otorisasi akses.

e. Optimalisasi Kinerja (khususnya mode *hybrid*)

Kurangi latensi melalui:

1. Model *generative* yang lebih ringan (respons lebih cepat, tapi akurasi bisa menurun).
2. Mengurangi jumlah dan kompleksitas entri *knowledge* (semakin sederhana, semakin cepat).

f. Optimalisasi kode, *caching*, pemrosesan asinkron, dan *dynamic model loading*.

g. Algoritme akurasi jawaban.

Meneruskan pembahasan “Benar” dan “Salah” pada tabel akurasi dan respons pada bagian *Hasil dari Deployment, Delivery, and Feedback*. Banyak hal menarik yang dapat ditarik dalam analisis tersebut. Salah satunya algoritme akurasi jawaban. Idanya adalah *chatbot* dapat mengukur persentase keyakinan akan kebenaran jawaban yang akan dikeluarkannya. Apabila persentase kebenarannya tinggi maka *chatbot* dapat memberikan jawaban yang akurat. Jika sebaliknya, maka *chatbot* akan menolak

menjawab pertanyaan yang diberikan, dengan harapan tidak menyebarkan informasi yang salah.

Dengan demikian, *chatbot* dapat menjadi lebih akurat, cepat, dan *scalable* untuk mendukung layanan informasi akademik yang lebih baik.

DAFTAR PUSTAKA

- Abigail, M. I., & Haryono, K. (2025). Building an Effective Quran Learning System with Prototyping Model. *International Journal of Informatics and Computation (IJICOM)*, 7(2). <https://doi.org/10.35842/ijicom.v7i2.138>
- Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006. <https://doi.org/10.1016/j.mlwa.2020.100006>
- Apache Software Foundation. (2004). *Apache License, Version 2.0*. <https://www.apache.org/licenses/LICENSE-2.0>
- Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K., & Wood, K. (2017). Design prototyping methods: State of the art in strategies, techniques, and guidelines. *Design Science*, 3. <https://doi.org/10.1017/dsj.2017.10>
- Gumilang, B. R., Nugroho, A., & Zy, A. T. (2024). Development of a Web-based Administration System for The Bantu Sodara Community Using the Waterfall Method. *International Journal of Informatics and Computation (IJICOM)*, 6(2). <https://doi.org/10.35842/ijicom.v6i2.95>
- Halvonik, D., & Kapusta, J. (2024). Large Language Models and Rule-Based Approaches in Domain-Specific Communication. *IEEE Access*, 12, 107046–107058. <https://doi.org/10.1109/ACCESS.2024.3436902>
- Ilieva, G., Yankova, T., Klisarova-Belcheva, S., Dimitrov, A., Bratkov, M., & Angelov, D. (2023). Effects of Generative Chatbots in Higher Education. *Information (Switzerland)*, 14(9). <https://doi.org/10.3390/info14090492>
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. In *ACM Computing Surveys* (Vol. 55, Number 12). Association for Computing Machinery. <https://doi.org/10.1145/3571730>
- Labadze, L., Grigolia, M., & Machaidze, L. (2023). Role of AI chatbots in education: systematic literature review. In *International Journal of Educational Technology in Higher Education* (Vol. 20, Number 1). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1186/s41239-023-00426-1>

- Prananta Barus, S., & Surijati, E. (2021). Chatbot with Dialogflow for FAQ Services in Matana University Library. *International Journal of Informatics and Computation (IJICOM)*, 3(2). <https://doi.org/10.35842/ijicom.v3i2.43>
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. McGraw-Hill.
- Putri, T. E., & Ramadhan, G. (2024). Penerapan Chatbot sebagai Alat Pembelajaran untuk Pengembangan Pendidikan Karakter. In *Indonesian Journal of Computer Science and Engineering (IJCSE)* (Vol. 01).
- Rachma, N., & Muhlas, I. (2022). Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design. *Jurnal Inovatif: Inovasi Teknologi Informasi Dan Informatika*, 5(1), 36–39. <https://doi.org/10.32832/inovatif>
- Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-January*, 1525–1530. <https://doi.org/10.1109/ICACCI.2017.8126057>
- Singh, J., Joesph, M. H., & Jabbar, K. B. A. (2019). Rule-based chabot for student enquiries. *Journal of Physics: Conference Series*, 1228(1). <https://doi.org/10.1088/1742-6596/1228/1/012060>
- Skjuve, M., Følstad, A., Fostervold, K. I., & Brandtzaeg, P. B. (2021). My Chatbot Companion - a Study of Human-Chatbot Relationships. *International Journal of Human Computer Studies*, 149. <https://doi.org/10.1016/j.ijhcs.2021.102601>
- TheBloke. (2023). *Mistral-7B-Instruct-v0.2-GGUF*. Hugging Face. <https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF>
- Thorat, S. A., & Jadhav, V. D. (2020). *A Review on Implementation Issues of Rule-based Chatbot Systems* (Vol. 2020). <https://ssrn.com/abstract=3567047>
- Ula, M., Hardi, R., & Hipiny, I. (2023). An Improved Structure for Academic Information Services through AI Chatbots. *Journal of Engineering Science and Technology Review*, 16(5), 164–173. <https://doi.org/10.25103/jestr.165.20>

LAMPIRAN

Lampiran A. Dampak Pembatasan *Token*

```
Request: {'message': 'kalu mau ulang jalur penelitian gimana ya?'}
llama_perf_context_print:      load time = 64917.46 ms
llama_perf_context_print: prompt eval time = 64912.12 ms / 1424 tokens (
45.58 ms per token, 21.94 tokens per second)
llama_perf_context_print:      eval time = 47807.09 ms / 199 runs (
240.24 ms per token, 4.16 tokens per second)
llama_perf_context_print:      total time = 112847.67 ms / 1623 tokens
127.0.0.1 - - [12/Nov/2025 10:47:33] "POST /chat HTTP/1.1" 200 -
Response: Bagi mahasiswa yang mengulang atau ganti Penjaluran Tugas Akhir dalam
jalur Penelitian, silakan mengusulkan judul atau mengambil judul yang ditawarkan
oleh dosen melalui link: https://s.id/FormINF. Selanjutnya, silakan melakukan
Ulang Jalur atau Ganti Jalur di Sekawan V2 (jika belum mendaftar) atau Sekawan
V3 (jika sudah mendaftar). Selanjutnya sampaikan usulan Anda kepada Dosen
Pembimbing Akademik (DPA) agar mendapat approval di Sekawan baik V2 maupun V3.
Harap berpamitan terleb
```