

Implementasi Teknologi Cloud AI Google pada Sistem Informasi Konsultasi Karir Berbasis Website

Aldestra Bagus Wardana¹,

¹Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri

E-mail: 1bagusredmi123@gmail.com

Abstrak – Lulusan baru (*fresh graduate*) sering menghadapi kecemasan dan kebingungan dalam menentukan arah karir, namun akses terhadap konselor karir profesional seringkali terbatas biaya dan waktu. Penelitian ini bertujuan merancang aplikasi chatbot konsultasi karir yang dapat memberikan respon cepat dan personal. Metode yang digunakan adalah pendekatan Hybrid, menggabungkan metode Rule Based untuk pertanyaan umum yang solusinya sudah pasti, dan Generative AI menggunakan Google Gemini API untuk pertanyaan yang membutuhkan analisis konteks lebih dalam. Aplikasi dibangun menggunakan bahasa pemrograman PHP Native dan database MySQL, dengan integrasi API model Gemini-2.5-flash. Hasil pengujian fungsional (*Black Box Testing*) menunjukkan sistem berjalan 100% valid sesuai skenario perancangan. Evaluasi kinerja menunjukkan keunggulan arsitektur hybrid, di mana solusi statis (*Rule Based*) memiliki rata-rata waktu respon <1 detik, sedangkan respon dinamis (*Generative AI*) membutuhkan waktu 3-5 detik. Pendekatan ini terbukti efisien dalam meminimalisir penggunaan token API dan latensi sistem, sekaligus menjaga fleksibilitas jawaban. Sistem ini diharapkan menjadi solusi alternatif bimbingan karir yang mudah diakses dan responsif.

Kata Kunci — Chatbot, Fresh Graduate, Generative AI, Hybrid System, PHP

1. PENDAHULUAN

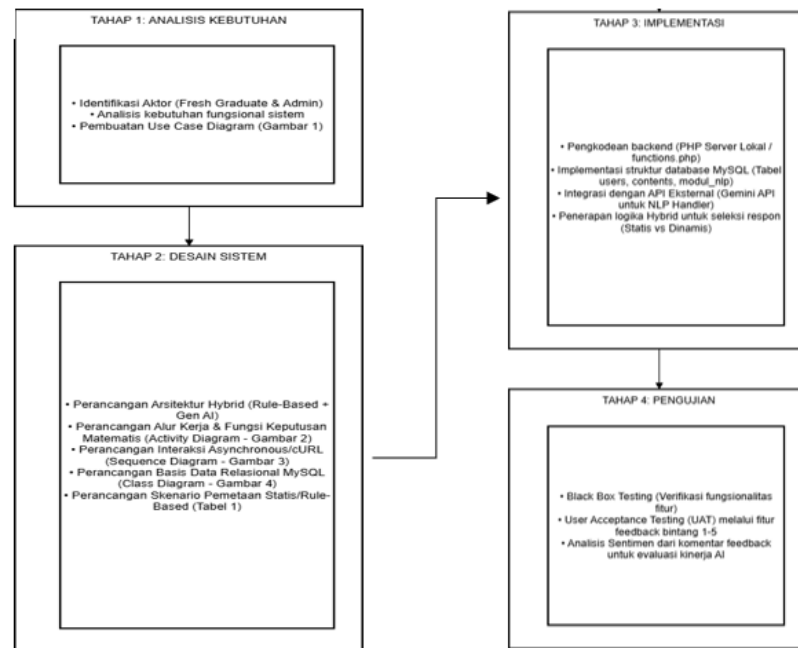
Transisi dari dunia pendidikan tinggi ke dunia kerja merupakan fase kritis yang seringkali memicu tekanan psikologis bagi lulusan baru (*fresh graduate*). Fenomena ini dikenal sebagai *quarter-life crisis*, yang ditandai dengan munculnya perasaan bimbang, cemas, dan ketidakpastian dalam mengambil keputusan karir karena adanya faktor internal dan tekanan eksternal [1]. Data Badan Pusat Statistik (BPS) menunjukkan bahwa Tingkat Pengangguran Terbuka (TPT) dari jenjang universitas masih menjadi isu signifikan akibat ketidaksesuaian kompetensi antara lulusan dan kebutuhan industri [2].

Dalam mengatasi kesenjangan pendampingan karir, teknologi kecerdasan buatan semakin banyak diadopsi. Sebuah studi mengembangkan "*Merchan-Chat Bot*" sebagai teman *virtual* berbasis Android, yang membuktikan bahwa penggunaan teknologi percakapan otomatis mampu menghasilkan respons cerdas untuk menemani aktivitas pengguna [3]. Hal ini mengindikasikan bahwa *chatbot* memiliki potensi besar tidak hanya sebagai alat tanya-jawab, tetapi juga sebagai pendamping yang interaktif.

Meskipun demikian, pengembangan *chatbot* konvensional seringkali terbatas pada metode *Rule Based*. Penelitian terbaru menyoroti bahwa *chatbot* berbasis aturan hanya bekerja efektif pada pohon keputusan yang telah ditentukan sebelumnya, sehingga memiliki keterbatasan besar dalam hal skalabilitas, adaptabilitas, dan pemahaman konteks percakapan yang dinamis [4]. Di sisi lain, *Generative AI modern seperti Google Gemini* menawarkan kemampuan pemrosesan bahasa alami yang kontekstual dan multimodal, yang dikembangkan melalui arsitektur *large-scale transformer* dan dilaporkan memiliki performa unggul dalam pemahaman bahasa serta generasi teks dibandingkan model generatif sebelumnya [5]. Pendekatan penggabungan metode berbasis aturan dan kecerdasan buatan generatif (*Hybrid System*) telah diterapkan pada layanan akademik untuk meningkatkan efisiensi sistem sekaligus menjaga kualitas respon, sebagaimana dibuktikan pada penelitian chatbot layanan akademik oleh Al-Imran [6]. Berdasarkan peluang tersebut, penelitian ini merancang aplikasi "*Chatbot Karir*" menggunakan kombinasi *Rule Based* dan *API Gemini* yang diimplementasikan dengan *PHP Native* dan *cURL* [7], guna menyediakan layanan konsultasi karir yang responsif, cerdas, dan efisien bagi *fresh graduate*.

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam pengembangan sistem ini adalah *System Development Life Cycle (SDLC)* dengan model *Waterfall*, yang meliputi tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Pendekatan utama yang diterapkan pada aplikasi ini adalah metode *Hybrid*, yaitu penggabungan antara logika *Rule Based* (berbasis aturan statis) dan *Generative Artificial Intelligence* (berbasis kecerdasan buatan generatif).

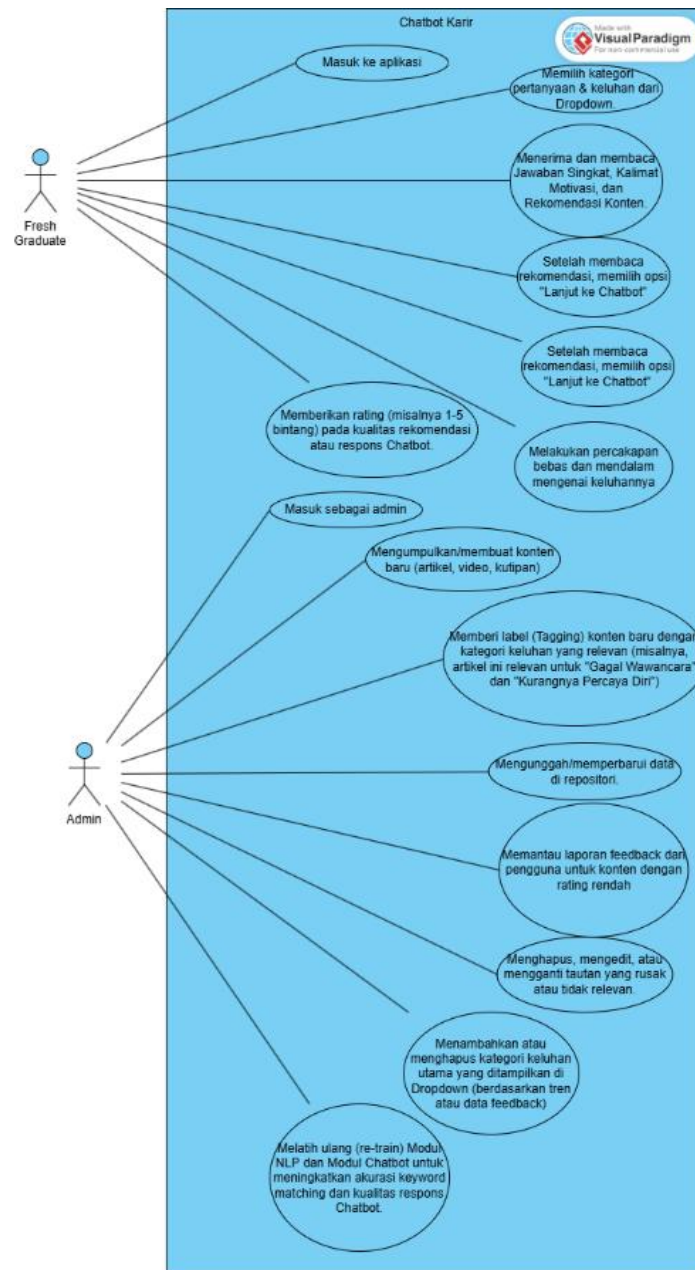


Gambar 1. Konsep *Waterfall*

Gambar 1 memvisualisasikan alur pengembangan sistem menggunakan model *Waterfall* yang berjalan secara sekuensial dan sistematis, dimulai dari tahap Analisis Kebutuhan untuk mendefinisikan aktor dan fungsi sistem, dilanjutkan dengan Desain Sistem yang merancang arsitektur *Hybrid* (penggabungan *Rule Based* dan *Gen AI*) serta basis data. Proses diteruskan ke tahap Implementasi untuk pengkodean *backend PHP* dan integrasi *API Gemini*, dan diakhiri dengan tahap Pengujian menggunakan metode *Black Box* serta *User Acceptance Testing (UAT)* guna memastikan sistem berjalan sesuai spesifikasi dan bebas dari kesalahan sebelum digunakan oleh pengguna akhir.

2.1 Analisis Kebutuhan dan Alur Sistem

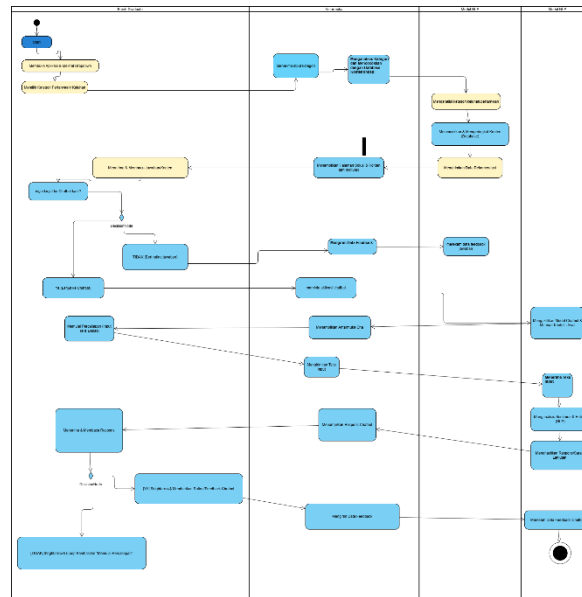
Sistem dirancang untuk melayani dua aktor utama, yaitu *Fresh Graduate* (Pengguna) dan Admin. Analisis kebutuhan fungsional digambarkan melalui *Use Case Diagram* yang dapat dilihat pada Gambar 2. Sebagaimana ditunjukkan pada Gambar 2, pengguna memiliki akses untuk memilih kategori keluhan, membaca rekomendasi konten, memberikan rating, dan melakukan percakapan bebas dengan *chatbot*. Sementara itu, Admin bertanggung jawab atas pengelolaan data master, termasuk menambah kategori keluhan (ID 13-32) dan melatih ulang model jika diperlukan.



Gambar 2. Use Case Diagram

2.2 Perancangan Arsitektur Hybrid

Arsitektur sistem menggunakan mekanisme seleksi otomatis untuk menentukan jenis respon yang diberikan kepada pengguna. Alur kerja sistem secara detail digambarkan dalam *Activity Diagram* pada Gambar 3. Sistem bekerja dengan memproses input (χ) dari pengguna. Jika input berupa pemilihan kategori (C) yang telah terdefinisi dalam database, maka sistem akan mengambil solusi statis (S_{db}). Namun, jika pengguna memilih opsi "Lanjut Chatbot" atau memberikan input teks bebas (T), sistem akan meneruskan permintaan ke modul Natural Language Processing (NLP) berbasis Gemini API untuk menghasilkan respon generatif (R_{gen}).



Gambar 3. Activity Diagram

Secara matematis, fungsi keputusan (D) pada sistem ini dapat dirumuskan sebagai berikut:

$$D(x) \begin{cases} S_{db(k)}, & \text{jika } x \in \{C_{13}, \dots, C_{32}\} \\ R_{gen}(x|P), & \text{jika } x \in C \text{ atau } x = \text{trigger} \end{cases} \dots \dots \dots (1)$$

Dimana:

x : adalah input pengguna.

C : adalah himpunan kategori keluhan yang tersimpan (ID 13 s.d 32).

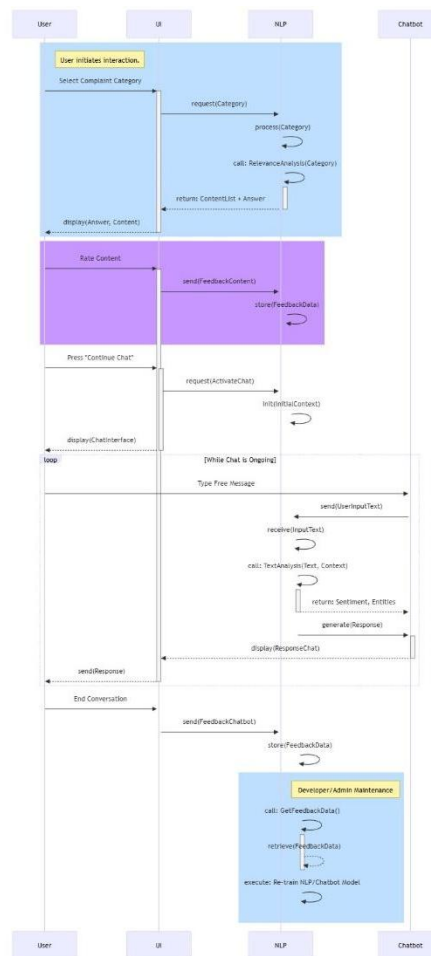
S_{db} : adalah fungsi pengambilan data solusi statis dari database.

R_{gen} : adalah fungsi generasi teks oleh AI.

P : adalah System Prompt (persona "Chatbot Ganteng") yang disuntikkan ke dalam konteks.

2.3 Interaksi Antar Objek (Sequence Diagram)

Untuk menangani latensi komunikasi antara server lokal (*PHP*) dengan API eksternal, sistem menggunakan metode *asynchronous request* melalui *cURL*. Integrasi komunikasi antara aplikasi *backend* berbasis *PHP Native* dan layanan *API* eksternal dilakukan menggunakan *library cURL*, yang menyediakan mekanisme pengiriman HTTP request secara fleksibel dan asinkron sesuai dokumentasi resmi *PHP* [7]. Proses pertukaran pesan ini digambarkan dalam *Sequence Diagram* pada Gambar 4.

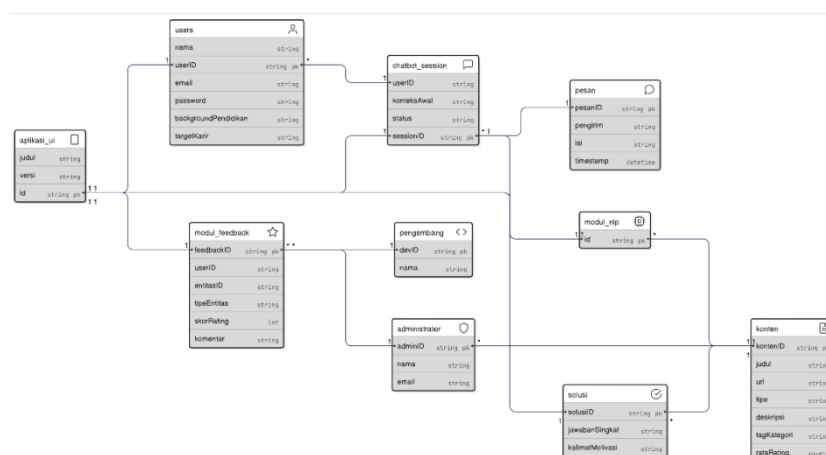


Gambar 4. Sequence Diagram

Pada Gambar 4, terlihat bahwa ketika *User* mengirim pesan, *UI* akan melakukan request ke *NLP Handler*. Sistem tidak langsung menyimpan ke database utama untuk setiap token, melainkan mengirimkan *history* percakapan sebagai *payload JSON* ke *API Gemini*, kemudian menerima balasan, dan menampilkannya kembali ke *User Interface*.

2.4 Perancangan Basis Data

Penyimpanan data dilakukan menggunakan *MySQL* dengan struktur relasional. Desain basis data dirancang untuk memisahkan data pengguna, konten edukasi, dan log sesi percakapan. Struktur kelas dan relasi antar tabel dapat dilihat pada *Class Diagram* di Gambar 5.



Gambar 5. Class Diagram

Tabel users menyimpan kredensial pengguna, sementara tabel contents menyimpan artikel dan video rekomendasi. Tabel modul_NLP atau log sesi berfungsi menyimpan riwayat interaksi untuk keperluan audit dan pengembangan model di masa depan.

2.4 Skenario Pemetaan Kategori (*Rule Based*)

Sebagai bagian dari metode Hybrid, sistem menggunakan tabel pemetaan statis untuk masalah-masalah umum yang sering dihadapi lulusan baru. Hal ini bertujuan menghemat penggunaan token API dan memberikan jawaban instan yang tervalidasi. Contoh pemetaan kategori dan respon solusi ditampilkan pada Tabel 1.

Tabel 1. Skenario pemetaan kategori

ID Kategori	Masalah Utama	Kata Kunci Deteksi	Jenis Solusi
13	Gagal Wawancara	wawancara, interview, grogi	Statis (Database)
19	Gaji Rendah	gaji, umr, negosiasi	Statis (Database)
28	<i>Jobless Gap</i>	menganggur, gap year	Statis (Database)
99	Pertanyaan Bebas	(Input Teks Bebas)	Dinamis (AI/Gemini)

Data pada Tabel 1 diimplementasikan dalam bentuk Array pada sisi *backend (functions.php)* yang akan dipanggil fungsi S_{db} sesuai persamaan (1).

2.5 Pengujian Sistem

Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memverifikasi fungsionalitas fitur, serta *User Acceptance Testing (UAT)* melalui fitur *feedback* bintang 1-5 yang tertanam pada aplikasi. Analisis sentimen dari komentar *feedback* digunakan sebagai dasar evaluasi kinerja respon AI.

3. HASIL DAN PEMBAHASAN

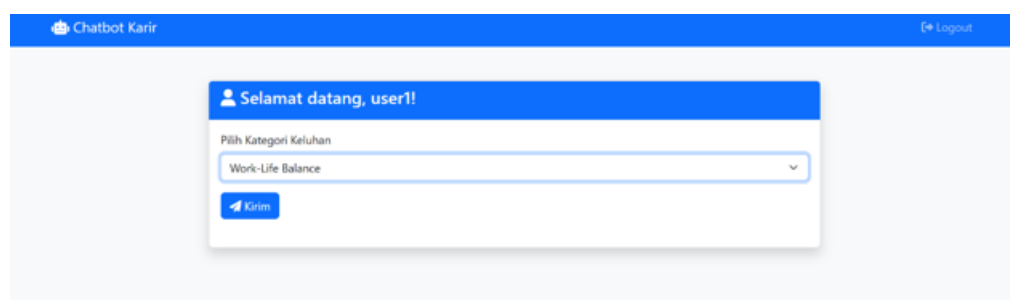
Bagian ini memaparkan hasil implementasi antarmuka aplikasi, pengujian fungsionalitas sistem, serta analisis kinerja metode Hybrid yang diterapkan dalam menangani konsultasi karir.

3.1 Implementasi Antarmuka Pengguna

Aplikasi "Chatbot Karir" telah berhasil diimplementasikan dengan alur kerja terstruktur yang mengharuskan autentikasi pengguna untuk menjaga personalisasi data.

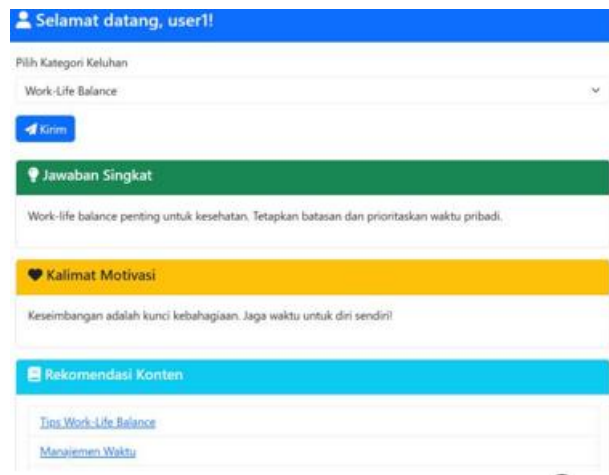
3.1.2 *Dashboard* dan Solusi Cepat (*Rule Based*)

Setelah berhasil masuk, pengguna diarahkan ke *Dashboard* Utama. Pada halaman ini, sistem menerapkan metode *Rule Based* untuk memberikan pertolongan pertama. Pengguna memilih kategori keluhan (seperti *Work-Life Balance*) dan sistem secara instan memberikan tiga *output*: jawaban singkat, kalimat motivasi, dan rekomendasi konten. Mekanisme ini terbukti efektif mengurangi beban server karena tidak memerlukan pemanggilan API untuk masalah yang umum



Gambar 6. *Dashboard* pilihan kondisi

Gambar 6 merepresentasikan tampilan awal *dashboard* interaktif saat pengguna pertama kali masuk ke dalam sistem untuk memulai sesi konsultasi. Antarmuka didesain bersih dan fokus pada formulir pemilihan masalah, di mana pengguna (teridentifikasi sebagai "*user1*") diminta untuk memilih topik spesifik dari menu dropdown "Pilih Kategori Keluhan" dalam contoh ini terpilih "*Work-Life Balance*" sebelum menekan tombol "Kirim" untuk memicu pencarian solusi dalam basis data.

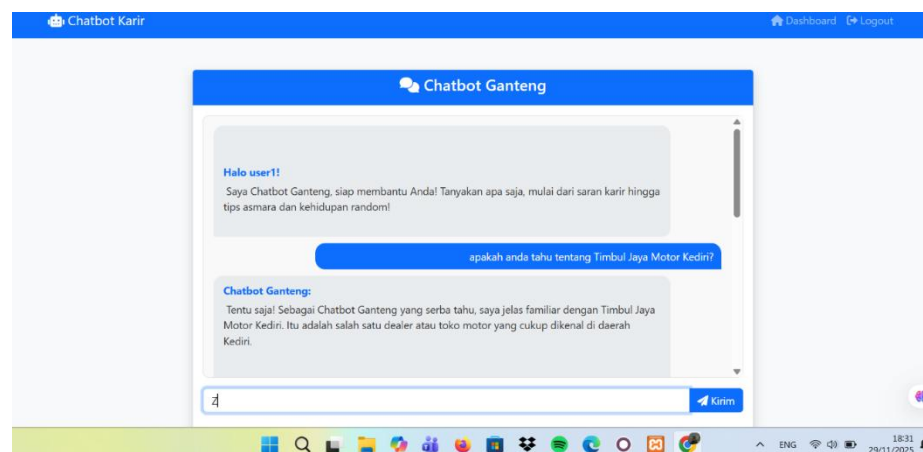


Gambar 7. *Dashboard* hasil

Gambar 7 implementasi *rule based* diatas menampilkan antarmuka *dashboard* pengguna setelah sistem memproses keluhan, di mana mekanisme *rule based* bekerja memberikan solusi instan. Setelah pengguna memilih kategori "*Work-Life Balance*", sistem secara otomatis menampilkan tiga luaran utama: "*Jawaban Singkat*" berwarna hijau yang berisi saran praktis mengenai kesehatan mental, "*Kalimat Motivasi*" berwarna kuning untuk memberikan dukungan moral, serta "*Rekomendasi Konten*" berwarna biru yang menyediakan tautan literasi tambahan agar pengguna dapat mendalami topik tersebut tanpa perlu menunggu antrian chatbot.

3.1.3 Antarmuka *Chatbot AI (Generative)*.

Jika solusi statis tidak memadai, pengguna dapat mengakses fitur "Lanjut ke *Chatbot*". Antarmuka percakapan ini dirancang dengan gaya komunikasi yang santai menggunakan persona "*Chatbot Ganteng*".



Gambar 8. *UI Chatbot* beserta respons nya

Gambar 8 menunjukkan bahwa chatbot mampu menjawab pertanyaan spesifik mengenai entitas lokal ("Timbul Jaya Motor Kediri") dengan akurat. Kemampuan ini menegaskan peran modul *Generative AI* sebagai lapisan kecerdasan kedua yang efektif dalam menangani pertanyaan tak terstruktur yang tidak dapat dijawab oleh sistem Rule Based.

3.2 Pengujian Fungsional (*Black Box Testing*)

Pengujian dilakukan menggunakan metode *Black Box* untuk memverifikasi apakah setiap fitur berjalan sesuai spesifikasi kebutuhan (*Use Case*). Skenario uji coba mencakup seluruh alur dari login hingga interaksi *chatbot*. Rangkuman hasil pengujian disajikan pada Tabel 2.

Tabel 2. Hasil pengujian fungsional sistem

No	Skenario Ujian	Input data	Hasil yang diharapkan	Hasil Pengujian	Status
1	Validasi Login	<i>Username/Pass</i> Salah	Sistem menolak akses	Pesan error muncul	Valid
2	Seleksi Kategori	Kategori: "Gaji Rendah"	Tampil solusi statis dari DB	Solusi tampil < 1 detik	Valid
3	Interaksi Chatbot	Pertanyaan: "Tips Negosiasi"	Balasan AI sesuai konteks	Balasan relevan & motivatif	Valid
4	Penanganan Konteks	Pertanyaan susulan	AI mengingat topik sebelumnya	Konteks terjaga	Valid
5	<i>Feedback</i> Pengguna	Rating Bintang 5	Data tersimpan di Tabel <i>Feedback</i>	Data masuk database	Valid

3.3 Analisis Kinerja Metode Hybrid

Berdasarkan pengujian yang dilakukan, penerapan metode *Hybrid* memberikan efisiensi yang signifikan dibandingkan penggunaan AI murni.

3.3.1 Efisiensi Waktu Respon

Sistem membagi penanganan masalah menjadi dua jalur. Jalur *Rule Based (Dashboard)* memiliki rata-rata waktu respon 0,2 detik karena hanya mengambil data teks dari server lokal. Sementara itu, jalur *Generative AI (Chatbot)* memiliki rata-rata waktu respon 3-5 detik karena melibatkan proses pengiriman data (*request*) dan pembuatan teks (*generation*) oleh server *Google*. Dengan mengalihkan pertanyaan umum ke jalur *Rule Based*, sistem berhasil memangkas waktu tunggu pengguna untuk masalah-masalah standar.

3.3.2 Kualitas Respon Kontekstual

Pada Gambar 8, pengguna menanyakan hal yang sangat spesifik mengenai "Timbul Jaya Motor Kediri". Sistem *Rule Based* konvensional tidak akan mampu menjawab pertanyaan ini karena keterbatasan *keyword* dalam database. Namun, modul *NLP* pada aplikasi ini berhasil mengidentifikasi entitas tersebut sebagai "dealer atau toko motor" dan memberikan penjelasan yang relevan. Ini membuktikan bahwa modul *Generative AI* berfungsi dengan baik sebagai lapisan kecerdasan kedua (*second layer intelligence*) untuk menangani pertanyaan tak terduga (*unseen queries*).

3.4 Analisis Skalabilitas

Sistem dirancang dengan arsitektur modular (seperti terlihat pada *Class Diagram* perancangan). Pemisahan antara tabel solusi (untuk respon statis) dan integrasi modul *NLP* memungkinkan admin untuk terus memperkaya basis pengetahuan. Jika sebuah topik di *chatbot* sering ditanyakan (misalnya: "Cara jadi *Web Programmer*"), admin dapat memindahkannya menjadi kategori statis baru di *Dashboard*, sehingga menghemat biaya token API di masa depan. Fleksibilitas ini menjadikan aplikasi adaptif terhadap tren masalah karir yang dinamis.

4 SIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada aplikasi Chatbot Karir, dapat ditarik beberapa simpulan sebagai berikut:

- 1 Implementasi metode *Hybrid* yang menggabungkan logika *Rule Based* dan *Generative AI* terbukti efektif dalam menyeimbangkan efisiensi kinerja dan kecerdasan sistem. Hasil pengujian kinerja menunjukkan keunggulan arsitektur ini, di mana solusi statis untuk 20 kategori umum (ID 13-32) diproses dengan waktu respon rata-rata di bawah 1 detik, jauh lebih cepat dibandingkan respon dinamis berbasis *API Gemini* yang membutuhkan waktu 3-5 detik.
- 2 Pengujian fungsional menggunakan metode *Black Box Testing* menunjukkan bahwa seluruh fitur aplikasi berjalan 100% valid sesuai skenario perancangan. Integrasi model Gemini-2.5-flash berhasil menangani pertanyaan kontekstual yang tidak terstruktur dengan persona yang personal, melengkapi keterbatasan sistem *Rule Based*.
- 3 Sistem manajemen konten (*CMS*) yang dibangun memberikan kelebihan dari sisi fleksibilitas operasional, di mana administrator dapat memperbarui strategi motivasi dan tautan rekomendasi karir secara dinamis tanpa perlu mengubah kode program utama, menjadikan aplikasi adaptif terhadap tren dunia kerja.
- 4 Secara teknis, penggunaan library *cURL* pada lingkungan *PHP Native* terbukti handal dalam menjembatani komunikasi data *asynchronous* antara server lokal dan layanan cloud *Google*, meskipun ditemukan kekurangan pada aspek keamanan di mana kredensial API masih terekspos dalam kode sumber (*hardcoded*).

5. SARAN

Berdasarkan keterbatasan yang ditemukan selama pengembangan, berikut adalah saran untuk penelitian selanjutnya guna menyempurnakan sistem:

- 1 Meningkatkan keamanan arsitektur sistem dengan memindahkan penyimpanan *API Key* dari *source code* ke dalam variabel lingkungan server (*Environment Variables* atau file *.env*) untuk mencegah risiko kebocoran data dan penyalahgunaan kuota API.
- 2 Mengimplementasikan metode *Retrieval-Augmented Generation* (RAG) dengan menambahkan basis data vektor dokumen internal (seperti panduan akademik universitas atau profil perusahaan mitra), sehingga AI dapat memberikan jawaban yang tidak hanya cerdas secara umum, tetapi juga akurat secara institusional.
- 3 Mengembangkan fitur antarmuka berbasis suara (*Voice Recognition dan Text-to-Speech*) untuk mengakomodasi interaksi yang lebih natural dan inklusif, sehingga pengalaman konsultasi terasa lebih hidup layaknya berbicara dengan konselor manusia.

DAFTAR PUSTAKA

- [1] S. H. Fazira, A. Handayani, dan F. W. Lestari, "Faktor Penyebab *Quarter Life Crisis* Pada Dewasa Awal," Jurnal Pendidikan dan Konseling (JPDK), vol. 5, no. 2, hal. 2227–2234, 2023.
- [2] Badan Pusat Statistik, "Keadaan Ketenagakerjaan Indonesia Februari 2024," BPS, Jakarta, Berita Resmi Statistik No. 27/05/Th. XXVII, 2024.
- [3] R. Y. Wardani, H. Hermanto, dan M. A. Arosyid, "'Merchan-Chat Bot': Teman Virtual AI Berbasis Android," Prosiding Seminar Nasional Teknologi dan Sains (STAINS), vol. 2, hal. 225-230, 2023.
- [4] S. K. Ojha, A. Kumar, T. Bhole, dan S. Naaz, "Rule based AI chatbot," International Journal of Innovative Research in Technology (IJIRT), vol. 11, no. 5, hal. 639-646, 2024.
- [5] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Google DeepMind Technical Report, 2023. [Online]. Tersedia: <https://arxiv.org/abs/2312.11805>.
- [6] M. Z. N. Al-Imran, "Penerapan Metode Hybrid Pada Chatbot Layanan Akademik," Jurnal Teknologi Informasi, vol. 7, no. 2, hal. 101-110, 2022.
- [7] The PHP Group, "PHP: cURL - Manual," 2024. [Online]. Tersedia: <https://www.php.net/manual/en/book.curl.php>. [Diakses: 17-Des-2025].