

PROJEK UAS

BAB I

PENDAHULUAN

1.1 Latar Belakang

Universitas Indonesia (UI) merupakan salah satu perguruan tinggi negeri terkemuka di Indonesia yang memiliki jumlah mahasiswa, calon mahasiswa, alumni, dan masyarakat umum yang sangat besar. Setiap tahunnya, UI menyediakan beragam informasi terkait akademik, administrasi, fasilitas kampus, kegiatan mahasiswa, hingga layanan institusi melalui berbagai media seperti situs web resmi, media sosial, dan pusat layanan informasi.

Namun, informasi yang tersedia sering kali tersebar di berbagai platform dan tidak selalu mudah diakses secara cepat dan efisien, terutama bagi pengguna yang membutuhkan jawaban instan dan spesifik. Hal ini dapat menimbulkan kesulitan, seperti kebingungan dalam memahami prosedur akademik, keterlambatan memperoleh informasi, serta meningkatnya beban kerja pada pihak administrasi kampus.

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence), khususnya dalam bidang *Natural Language Processing* (NLP), memungkinkan pengembangan chatbot sebagai sistem tanya-jawab otomatis yang mampu memberikan respons cepat, akurat, dan interaktif. Chatbot dapat menjadi solusi inovatif untuk membantu pengguna memperoleh informasi terkait Universitas Indonesia secara efisien melalui percakapan alami berbasis teks.

Oleh karena itu, diperlukan pengembangan sebuah chatbot yang mampu menjawab pertanyaan seputar Universitas Indonesia dengan memanfaatkan teknologi kecerdasan buatan, sehingga dapat meningkatkan kualitas layanan informasi serta pengalaman pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian/pengembangan ini adalah:

1. Bagaimana merancang dan membangun chatbot yang mampu menjawab pertanyaan seputar Universitas Indonesia?
2. Bagaimana chatbot dapat memahami pertanyaan pengguna dalam bahasa alami?
3. Bagaimana tingkat keakuratan dan relevansi jawaban yang dihasilkan oleh chatbot?
4. Bagaimana efektivitas chatbot dalam membantu pengguna memperoleh informasi terkait UI?

1.3 Tujuan

Tujuan dari penelitian/pengembangan ini adalah sebagai berikut:

1. Mengembangkan chatbot berbasis kecerdasan buatan untuk menjawab pertanyaan tentang Universitas Indonesia.
2. Mengimplementasikan teknik *Natural Language Processing* dalam memahami pertanyaan pengguna.
3. Menyediakan sistem tanya-jawab yang cepat, akurat, dan mudah digunakan.
4. Mengevaluasi kinerja chatbot dalam memberikan layanan informasi kepada pengguna.

1.4 Manfaat

Manfaat yang diharapkan dari pengembangan chatbot ini meliputi:

1. **Bagi Pengguna**

Mempermudah mahasiswa, calon mahasiswa, dan masyarakat umum dalam memperoleh informasi terkait Universitas Indonesia secara cepat dan interaktif.

2. **Bagi Universitas**

Mengurangi beban layanan informasi manual serta meningkatkan kualitas pelayanan publik berbasis teknologi.

3. **Bagi Peneliti/Pengembang**

Menjadi sarana penerapan dan pengembangan ilmu di bidang kecerdasan buatan, khususnya NLP dan sistem chatbot.

1.5 Batasan Masalah

Agar penelitian lebih terfokus, batasan masalah dalam pengembangan chatbot ini adalah:

1. Chatbot hanya menjawab pertanyaan yang berkaitan dengan Universitas Indonesia.
2. Informasi yang digunakan bersumber dari data resmi UI.
3. Chatbot berbasis teks dan tidak mencakup layanan suara.
4. Evaluasi chatbot difokuskan pada akurasi dan kepuasan pengguna.

BAB II

LANDASAN TEORI

2.1 Konsep User Interface (UI)

User Interface (UI) merupakan bagian dari sistem yang berfungsi sebagai penghubung antara pengguna dan sistem komputer. UI dirancang untuk memungkinkan pengguna berinteraksi dengan sistem secara efektif, efisien, dan mudah dipahami. Dalam konteks sistem berbasis chatbot, UI biasanya berbentuk antarmuka percakapan berbasis teks yang memungkinkan pengguna mengajukan pertanyaan dan menerima jawaban secara langsung. Desain UI yang baik harus memperhatikan aspek usability, learnability, dan kepuasan pengguna (Shneiderman et al., 2016). UI yang sederhana dan intuitif sangat penting agar pengguna dapat memanfaatkan chatbot tanpa memerlukan pelatihan khusus.

2.2 Chatbot dan Sistem Tanya Jawab

Chatbot adalah sistem berbasis kecerdasan buatan yang dirancang untuk berinteraksi dengan manusia menggunakan bahasa alami. Dalam konteks question answering system, chatbot menerima pertanyaan dalam bentuk teks dan menghasilkan jawaban yang relevan berdasarkan pengetahuan yang dimiliki sistem. Konsep sistem tanya jawab modern banyak dibahas dalam kajian Natural Language Processing dan Information Retrieval. Chatbot bekerja sebagai sistem tanya-jawab (*question answering system*) yang menerima masukan berupa pertanyaan dan menghasilkan keluaran berupa jawaban yang relevan. Sistem tanya jawab umumnya terdiri dari beberapa tahap utama, yaitu pemrosesan input pengguna, pemahaman maksud pertanyaan, pencarian atau penentuan jawaban, serta penyajian jawaban kepada pengguna. Dalam penelitian ini, chatbot digunakan sebagai media untuk menyampaikan informasi Universitas Indonesia secara otomatis dan interaktif.

2.3 Komputasi Intelegensia

Komputasi Intelegensi merupakan cabang ilmu komputer yang mempelajari metode komputasi untuk meniru kecerdasan manusia dalam menyelesaikan masalah. Pendekatan dalam komputasi intelegensia meliputi pembelajaran mesin (*machine learning*), sistem berbasis pengetahuan, logika fuzzy, dan jaringan saraf tiruan. Dalam pengembangan chatbot, komputasi intelegensia berperan dalam proses pengambilan keputusan, pengenalan pola bahasa, serta penentuan jawaban yang paling sesuai terhadap pertanyaan pengguna. Penerapan komputasi intelegensia memungkinkan sistem untuk bekerja secara adaptif dan cerdas.

2.4 Large Language Model dan Risiko Halusinasi

Large Language Model (LLM) adalah model generatif yang mempelajari pola bahasa dari data skala besar sehingga mampu menghasilkan teks yang koheren dan kontekstual. Kekuatan utama LLM terletak pada kemampuan memahami variasi pertanyaan, melakukan parafrase, menyusun jawaban ringkas, serta menjaga alur dialog. Namun, LLM tidak secara inheren “mengetahui” fakta spesifik dari dokumen

internal organisasi kecuali fakta tersebut disediakan dalam konteks input atau sudah tertanam pada parameter model (yang sering kali tidak relevan untuk data lokal/privat).

Dalam aplikasi informasi kampus, risiko utama penggunaan LLM murni adalah halusinasi: model dapat menghasilkan jawaban yang secara linguistik meyakinkan tetapi tidak didukung oleh sumber internal. Karena itu, dibutuhkan mekanisme pengendalian (guardrails) dan strategi penyediaan konteks yang membuat LLM bekerja sebagai “penyusun jawaban” (answer synthesizer) berbasis dokumen, bukan sebagai “sumber pengetahuan” utama.

2.5 Representasi Semantik dengan Embedding

Embedding adalah representasi vektor berdimensi tetap dari teks yang menangkap kedekatan semantik. Dengan embedding, kemiripan antar-teks dapat dihitung menggunakan metrik seperti cosine similarity sehingga sistem dapat menemukan potongan dokumen yang relevan terhadap pertanyaan meskipun kata-katanya tidak identik. Pendekatan embedding sangat penting untuk bahasa alami karena pertanyaan pengguna sering berupa parafrasa, singkatan, atau campuran istilah formal-informal.

Pada skenario berbahasa Indonesia, pemilihan embedding multilingual menjadi relevan agar representasi semantik stabil untuk kosakata Indonesia dan entitas lokal. Model embedding multilingual umumnya dilatih pada korpus lintas bahasa sehingga lebih robust ketika pertanyaan pengguna menggunakan bahasa Indonesia, campuran istilah Inggris, maupun variasi penulisan nama

Pada penelitian ini digunakan pendekatan embedding berbasis Sentence Transformer untuk merepresentasikan teks menjadi vektor semantik. Model embedding yang digunakan adalah “sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2” karena mendukung kueri berbahasa Indonesia dan variasi penulisan yang umum muncul pada pertanyaan pengguna. Embedding digunakan baik pada tahap indexing dokumen (chunk → embedding → disimpan pada vector database) maupun tahap query-time (pertanyaan → embedding → similarity search). Selain itu, embedding dinormalisasi (normalize embeddings) agar perhitungan kemiripan kosinus lebih konsisten saat proses retrieval.

2.6 Vector Database dan Similarity Search

Vector database adalah penyimpanan terstruktur untuk embedding yang mendukung pencarian kemiripan (similarity search) secara efisien. Proses yang umum adalah: teks dokumen dipecah menjadi unit kecil (chunk), setiap chunk diubah menjadi embedding, lalu disimpan beserta metadata. Ketika pengguna bertanya, pertanyaan di-embedding dan sistem melakukan pencarian top-k chunk paling mirip sebagai kandidat konteks.

Penggunaan metadata (misalnya domain “event”, “fasilitas”, “dosen”, “umum”) membantu membatasi ruang pencarian sehingga konteks yang diambil lebih fokus dan mengurangi “campur aduk” antar-topik. Selain itu, penyimpanan persisten (persisted index) memungkinkan sistem digunakan berulang tanpa mengulang proses indexing dari awal, selama sumber data tidak berubah.

2.7 Chunking dan Dampaknya terhadap Kualitas Retrieval

Chunking adalah proses membagi dokumen panjang menjadi potongan yang lebih kecil agar lebih mudah diindeks dan diambil kembali. Chunk yang terlalu besar berisiko memasukkan banyak informasi tidak relevan dan menurunkan presisi retrieval, sedangkan chunk yang terlalu kecil berisiko kehilangan konteks dan menurunkan koherensi jawaban. Praktik umum adalah menggunakan overlap antar-chunk agar informasi yang “terpotong” di batas chunk tetap terbawa pada chunk berikutnya, sehingga konteks yang diambil lebih lengkap.

Dalam basis pengetahuan berupa PDF profil/kompilasi informasi kampus, chunking menjadi krusial karena struktur PDF sering panjang dan beragam (judul, daftar, profil orang, serta paragraf naratif). Dengan chunking dan overlap yang tepat, sistem retrieval dapat menangkap jawaban yang tersebar pada bagian-bagian berbeda tanpa mengandalkan pencarian kata kunci mentah semata.

2.8 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) adalah arsitektur yang menggabungkan komponen retrieval (mengambil konteks dari basis pengetahuan) dan komponen generation (LLM menyusun jawaban). Prinsip RAG adalah memaksa jawaban bersandar pada potongan dokumen yang diambil, sehingga mengurangi halusinasi dan meningkatkan relevansi faktual untuk domain spesifik organisasi.

Secara konseptual, RAG memecah tugas tanya-jawab menjadi dua: pertama, memilih bukti (evidence selection) melalui similarity search; kedua, menyusun respons (response synthesis) melalui LLM dengan batasan “hanya gunakan konteks.” Efektivitas RAG ditentukan oleh kualitas indexing, kualitas retrieval (top-k, deduplikasi, domain routing), serta kualitas prompt dan aturan keluaran (output constraints).

Komponen generatif pada sistem ini menggunakan Large Language Model (LLM) sebagai penyusun jawaban berbasis konteks hasil retrieval. LLM tidak dijadikan sumber fakta utama; model dibatasi untuk merangkum dan menyajikan informasi hanya dari konteks yang diberikan oleh modul retrieval. Dengan pola tersebut, RAG menurunkan risiko halusinasi dan menjaga jawaban tetap ter-grounding pada basis pengetahuan internal.

2.9 Guardrails, Kebijakan Jawaban, dan Deterministic Handling

Guardrails adalah aturan dan mekanisme yang memastikan keluaran sistem mengikuti kebijakan aplikasi, misalnya “jawab hanya berdasarkan konteks” dan “jika tidak ada di basis pengetahuan, nyatakan tidak ditemukan.” Guardrails dapat diterapkan pada tiga level: pra-pemrosesan input (membatasi panjang, membersihkan noise), desain prompt (instruksi eksplisit), dan pasca-pemrosesan output (merapikan jawaban, menghapus format yang tidak diinginkan).

Selain itu, untuk kebutuhan tertentu seperti “daftar fakultas” atau “daftar dosen,” pendekatan deterministik (tanpa LLM) sering lebih tepat. Daftar adalah keluaran terstruktur yang sebaiknya konsisten dan tidak bergantung pada kreativitas model. Dengan deterministic handler, sistem dapat menjamin format, urutan, dan deduplikasi list, serta mengurangi beban token dan risiko drift jawaban.

BAB III

METODOLOGI

3.1 Pendekatan Pengembangan

Metodologi yang digunakan adalah pengembangan sistem secara iteratif dengan fokus pada kualitas jawaban berbasis dokumen internal. Iterasi dilakukan dengan pola: membangun pipeline ingestion dan indexing, menguji kemampuan retrieval terhadap pertanyaan tipikal pengguna, memperbaiki aturan routing/normalisasi untuk kasus-kasus nyata (misalnya nama dosen dan daftar fakultas), lalu menyempurnakan kebijakan jawaban dan keamanan aplikasi web. Pendekatan ini dipilih karena permasalahan utama bukan hanya “bisa menjawab,” tetapi “bisa menjawab secara benar dan konsisten berdasarkan dataset.”

3.2 Lingkungan Implementasi dan Komponen Sistem

Sistem dibangun menggunakan Python sebagai bahasa pemrograman utama. Layanan aplikasi berjalan pada server web berbasis Flask yang menyediakan endpoint percakapan dan pengelolaan state chat. Basis pengetahuan disimpan pada vector database Chroma yang dipersistensikan pada direktori lokal per domain. Komponen embedding menggunakan model sentence-transformers multilingual untuk menghasilkan representasi vektor dari potongan dokumen dan pertanyaan. Komponen generasi jawaban menggunakan Gemini sebagai LLM, dengan konfigurasi melalui environment variable untuk kunci API dan pemilihan model, sehingga deployment dapat dipindahkan tanpa mengubah kode inti.

3.3 Data dan Sumber Pengetahuan

Sumber pengetahuan utama berupa dokumen PDF internal (dataset kampus) yang memuat berbagai kategori informasi seperti profil umum, event/kegiatan, fasilitas, serta profil dosen. Untuk meningkatkan cakupan atau menambal bagian PDF yang sulit diekstrak (misalnya daftar yang berbentuk gambar/tabel), sistem menyediakan sumber tambahan berbentuk teks (misalnya file tambahan untuk pengetahuan umum dan daftar fakultas). Seluruh sumber tersebut diperlakukan sebagai “ground truth” sehingga chatbot dibatasi untuk menjawab hanya dari konten tersebut.

3.4 Pra-pemrosesan: Ekstraksi Teks dan Normalisasi

Tahap pra-pemrosesan dimulai dengan ekstraksi teks dari PDF menggunakan loader, dengan strategi fallback apabila salah satu metode ekstraksi menghasilkan teks yang terlalu sedikit. Teks hasil ekstraksi kemudian dinormalisasi secara ringan untuk mengurangi variasi whitespace, karakter non-standar, dan noise yang dapat mengganggu chunking maupun retrieval. Pada entitas dosen, normalisasi nama dilakukan lebih ketat, termasuk pembersihan gelar atau tanda baca tertentu agar pencocokan nama lebih stabil. Normalisasi ini bertujuan mengurangi kegagalan retrieval akibat perbedaan kecil penulisan (misalnya “Prof.” vs tanpa gelar, atau variasi kapitalisasi).

3.5 Ekstraksi Terarah untuk Domain Dosen dan Daftar Fakultas

Selain indexing umum berbasis chunking, dilakukan ekstraksi terarah (targeted extraction) untuk bagian dosen dan daftar fakultas. Untuk dosen, sistem mengekstrak section spesifik “Dosen Departemen

Matematika” lalu mem-parsing profil berdasarkan penomoran 1–2 digit agar tahun seperti “2018.” tidak salah terbaca sebagai entri dosen baru. Hasil parsing membentuk dokumen profil dosen yang lebih rapi, kemudian di deduplikasi menggunakan “nama yang sudah dinormalisasi” agar satu orang tidak terindeks ganda. Untuk daftar fakultas, sistem mencoba mengambil section “daftar fakultas/program/sekolah” bila tersedia dalam teks PDF; jika tidak tersedia, daftar dapat diperkaya dari sumber tambahan berbentuk teks agar chatbot tetap mampu menjawab pertanyaan daftar dengan konsisten.

3.6 Pembangunan Index: Chunking, Embedding, dan Penyimpanan Chroma

Setelah teks siap, dokumen dibagi menjadi chunk menggunakan metode splitting berbasis karakter dengan ukuran chunk yang relatif sedang dan overlap untuk menjaga kesinambungan konteks antar-batas potongan. Setiap chunk kemudian diubah menjadi embedding menggunakan model multilingual, lalu disimpan dalam Chroma dengan persist directory per domain, yaitu domain umum, event, fasilitas, dan dosen. Pemisahan domain dilakukan untuk meningkatkan presisi retrieval: pertanyaan terkait event diprioritaskan mengambil konteks dari domain event, namun tetap memiliki fallback ke domain lain jika hasil utama tidak memadai. Setelah index terbentuk, proses re-index diperlukan ketika sumber PDF berubah atau strategi ekstraksi/embedding diperbarui, karena perubahan tersebut dapat mengubah representasi vektor maupun struktur chunk.

Sistem menerapkan embedding multilingual untuk mengubah potongan dokumen dan pertanyaan pengguna menjadi vektor semantik. Fungsi embedding diimplementasikan melalui Hugging Face Embeddings dengan model “sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2”. Pada implementasi, embedding dihitung pada device yang tersedia (GPU jika ada, selain itu CPU). Embedding juga dinormalisasi (normalize_embeddings=True) untuk meningkatkan kestabilan perilaku cosine similarity pada proses similarity search. Perubahan model embedding mengharuskan re-indexing karena vektor yang tersimpan di vector database bergantung langsung pada model embedding yang dipakai.

3.7 Metodologi Query-Time: Routing, Retrieval, dan Penyusunan Konteks

Saat pengguna mengajukan pertanyaan, sistem melakukan pembersihan input dan pembatasan panjang untuk menjaga stabilitas layanan. Untuk kasus tertentu yang bersifat daftar (misalnya “daftar fakultas” atau “daftar dosen”), sistem menjalankan handler deterministik yang langsung menyusun jawaban dari cache/metadata tanpa melibatkan LLM. Untuk pertanyaan dosen yang menyebut nama, sistem mencoba direct match dengan lookup berbasis nama yang dinormalisasi; jika ditemukan, profil dosen dipakai sebagai konteks utama sehingga mengurangi risiko dosen “tidak terambil” oleh similarity search biasa.

Jika tidak masuk jalur deterministik maupun direct match, sistem melakukan intent detection sederhana berbasis kata kunci untuk memilih primary domain (umum/event/fasilitas/dosen). Kemudian sistem membentuk beberapa varian query retrieval, termasuk versi yang menghapus kata tanya umum (siapa/apa/kapan/dimana dan sejenisnya) agar embedding lebih menonjolkan kata kunci substansial. Retrieval dilakukan terhadap primary domain dengan top-k lebih besar dan domain lain sebagai fallback dengan top-k lebih kecil. Hasil retrieval di deduplikasi dan dirangking, lalu disusun menjadi blok konteks dengan penanda domain agar LLM memiliki struktur yang jelas saat menyusun jawaban.

Komponen generasi jawaban menggunakan layanan Google Generative AI (Gemini). Aplikasi membaca kredensial melalui environment variable (GEMINI_API_KEY) dan nama model melalui

GEMINI_MODEL, dengan default “gemini-2.5-flash”, sehingga model dapat diganti tanpa memodifikasi kode inti. Model diinisialisasi melalui konfigurasi API key dan pembuatan objek GenerativeModel, kemudian dipanggil pada runtime menggunakan metode generate_content untuk menghasilkan jawaban.

3.8 Penyusunan Prompt, Guardrails, dan Generasi Jawaban

Prompt disusun dengan memasukkan focus hint (indikasi topik), ringkasan riwayat percakapan, blok konteks hasil retrieval, dan pertanyaan pengguna. Di dalam prompt diterapkan kebijakan yang menekankan bahwa jawaban harus berbasis konteks dan menggunakan bahasa Indonesia yang ringkas. Sistem juga menetapkan aturan format keluaran agar jawaban tidak menghasilkan markdown atau formatting yang tidak diinginkan. Setelah LLM menghasilkan respons, dilakukan pasca-pemrosesan untuk merapikan jawaban dan memastikan kebijakan fallback berjalan, yaitu ketika konteks kosong atau tidak mendukung jawaban, sistem mengembalikan pernyataan bahwa informasi tidak ditemukan di dataset.

3.9 Manajemen Konteks Percakapan dan Follow-up Handling

Aplikasi menyimpan state percakapan yang relevan, misalnya “active person” untuk menangani pertanyaan lanjutan berbasis pronomina (“dia”, “beliau”, “orang itu”). Ketika pertanyaan terdeteksi sebagai follow-up pronominal dan state aktif tersedia, sistem me-rewrite query retrieval dengan menambahkan identitas orang/topik sebelumnya agar retrieval kembali relevan. Pendekatan ini membantu menjaga kesinambungan dialog tanpa memaksa pengguna mengulang nama atau konteks di setiap pertanyaan.

3.10 Pengujian dan Evaluasi

Evaluasi dilakukan melalui skenario uji berbasis pertanyaan nyata pengguna yang dikelompokkan per domain (umum, event, fasilitas, dosen, dan daftar). Setiap pertanyaan memiliki ekspektasi: apakah seharusnya “answerable” dari dataset atau seharusnya memicu fallback “tidak ditemukan.” Kriteria penilaian meliputi relevansi konteks yang diambil, kepatuhan terhadap kebijakan “grounded,” kebenaran fakta terhadap dokumen, konsistensi format jawaban, serta stabilitas untuk variasi parafrasa. Temuan dari pengujian digunakan untuk memperbaiki normalisasi, kata kunci intent, strategi query variants, dan deterministic handler untuk kasus-kasus yang berulang.

Pada saat query-time, konteks hasil retrieval (multi-domain) disisipkan ke dalam prompt sebagai satu-satunya sumber informasi yang boleh digunakan LLM. Dengan kebijakan ini, ketika konteks kosong atau tidak mendukung jawaban, sistem diarahkan untuk memberikan respons fallback (menyatakan informasi tidak ditemukan di dataset) alih-alih mengarang jawaban. Prosedur ini memastikan keluaran LLM berperan sebagai “penyusun jawaban” berbasis bukti, bukan sebagai basis pengetahuan mandiri.

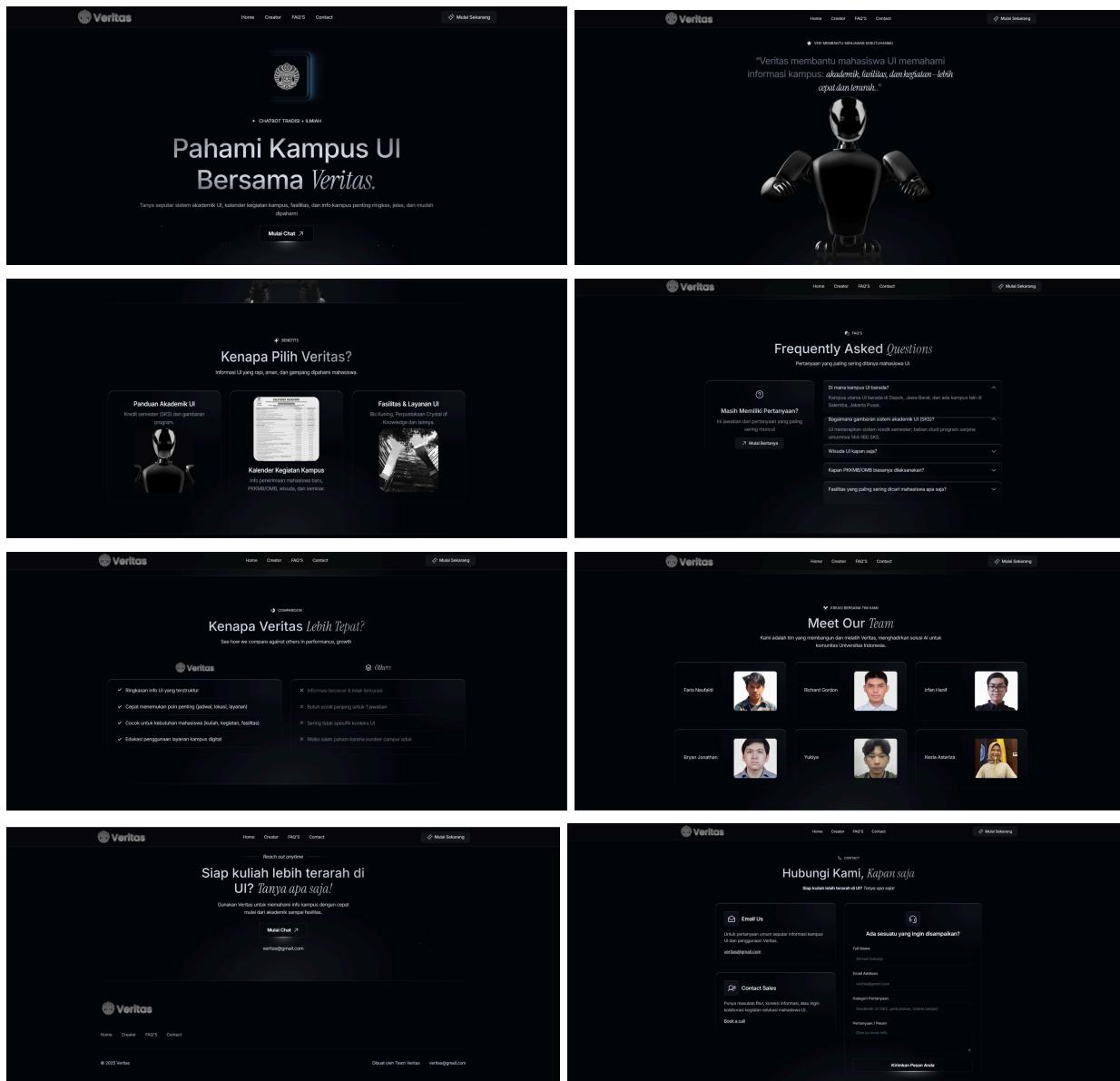
3.11 Aspek Keamanan dan Kesiapan Layanan

Untuk meningkatkan kesiapan layanan, aplikasi menerapkan proteksi CSRF pada endpoint berbasis form dan pembatasan laju (rate limiting) untuk mengurangi risiko penyalahgunaan. Selain itu, konfigurasi reverse proxy dipertimbangkan melalui middleware agar aplikasi tetap berjalan benar saat dideploy di belakang proxy. Langkah-langkah ini tidak mengantikan praktik keamanan menyeluruh, namun menjadi baseline yang penting untuk aplikasi chatbot yang berpotensi diakses banyak pengguna.

BAB IV

HASIL DAN PEMBAHASAN

<https://veritass.vercel.app>



4.1 Hasil Implementasi Halaman Pembuka

Halaman pembuka Veritas dirancang sebagai titik awal interaksi pengguna sebelum memasuki fitur utama chatbot. Pada bagian atas halaman tersedia navigasi utama yang memuat menu **Home**, **Creator**, **FAQ's**, dan **Contact**, serta tombol ajakan bertindak (call-to-action) “**Mulai Sekarang**”. Struktur navigasi ini

berfungsi untuk memisahkan informasi produk (profil, manfaat, tim, dan kontak) dari area percakapan chatbot, sehingga pengguna tidak langsung “dilempar” ke chat tanpa konteks.

Bagian hero menampilkan identitas produk “Veritas” dengan judul besar “**Pahami Kampus UI Bersama Veritas.**” disertai deskripsi singkat yang menegaskan lingkup layanan, seperti informasi akademik, kalender kegiatan kampus, fasilitas, dan informasi kampus yang disajikan secara ringkas. Di bagian ini terdapat tombol “**Mulai Chat**” yang menjadi jalur utama menuju halaman chatbot. Perlu ditegaskan bahwa pada halaman pembuka ini **pengguna belum dapat melakukan tanya-jawab**, karena halaman ini berfungsi sebagai pengantar. Akses ke chatbot baru dilakukan setelah menekan tombol “**Mulai Chat**” yang mengarahkan pengguna ke antarmuka percakapan.

Di bawah hero, halaman menampilkan bagian pengantar yang menekankan manfaat Veritas untuk membantu mahasiswa memahami informasi kampus secara lebih cepat dan terarah. Selanjutnya terdapat bagian **Benefits** berisi tiga kartu fitur utama, yaitu **Panduan Akademik UI**, **Kalender Kegiatan Kampus**, serta **Fasilitas & Layanan UI**. Masing-masing kartu berfungsi sebagai ringkasan area pengetahuan yang dapat ditangani oleh sistem, sekaligus memberi ekspektasi jenis pertanyaan yang relevan saat pengguna masuk ke chatbot.

Halaman pembuka juga menyediakan modul **Frequently Asked Questions (FAQ)** dalam bentuk daftar pertanyaan yang dapat dibuka-tutup (accordion). Konten FAQ menampilkan contoh pertanyaan umum (misalnya lokasi kampus, gambaran SKS, jadwal wisuda, PKKMB/OMB, dan fasilitas yang sering dicari). Bagian ini memberikan contoh cakupan informasi serta membantu pengguna yang hanya membutuhkan jawaban cepat tanpa harus masuk ke percakapan panjang.

Selain itu, tersedia bagian perbandingan “**Kenapa Veritas Lebih Tepat?**” yang membedakan pendekatan Veritas dengan pencarian informasi umum. Bagian ini menonjolkan nilai tambah berupa ringkasan yang terstruktur, kemudahan menemukan poin penting, dan pengurangan risiko salah paham akibat sumber yang campur aduk. Di bagian akhir halaman, ditampilkan **Meet Our Team** untuk memperkenalkan tim pengembang, serta bagian **Contact** yang menyediakan email dan formulir masukan. Halaman ditutup dengan call-to-action lanjutan yang kembali menegaskan tombol “**Mulai Chat**” sebagai pintu masuk ke chatbot.

4.2 Pembahasan Desain dan Alur Penggunaan

Secara fungsional, halaman pembuka berperan sebagai “lapisan orientasi” (onboarding) yang membentuk pemahaman pengguna mengenai apa yang bisa dan tidak bisa dilakukan sistem. Pemisahan antara landing page dan halaman chatbot mengurangi kebingungan pengguna, karena pengguna diberi konteks terlebih dahulu: apa tujuan Veritas, topik apa yang didukung, dan bagaimana cara memulai interaksi. Penempatan tombol “**Mulai Chat**” pada hero dan pada bagian penutup juga merupakan strategi untuk meminimalkan hambatan (friction) menuju fitur inti, tanpa menghilangkan informasi pengantar yang penting.

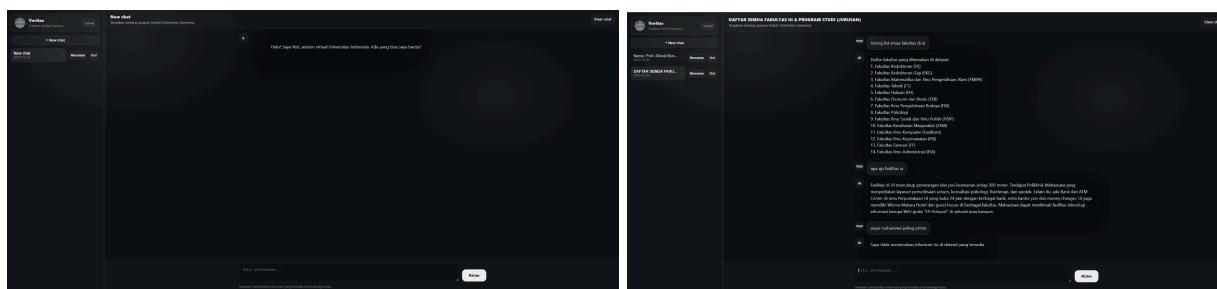
Pemilihan komponen **Benefits** dan **FAQ** memperkuat komunikasi cakupan sistem. Benefits mengelompokkan layanan ke tiga domain besar (akademik, kegiatan, fasilitas) sehingga pengguna memahami “peta” topik yang disediakan. Sementara itu, FAQ berfungsi sebagai contoh konkret dalam bentuk pertanyaan, yang biasanya lebih mudah dipahami pengguna dibanding deskripsi teknis. Dengan

pendekatan ini, ekspektasi pengguna terhadap chatbot menjadi lebih realistik: pengguna terdorong untuk mengajukan pertanyaan yang relevan ketika sudah memasuki halaman chat.

Bagian perbandingan “Veritas vs Others” menekankan diferensiasi produk pada sisi pengalaman pengguna: informasi disajikan ringkas dan terarah, serta membantu menemukan poin penting tanpa perlu menelusuri dokumen panjang. Bagian ini penting dalam laporan karena menunjukkan bahwa desain antarmuka tidak hanya estetika, tetapi juga mengkomunikasikan value proposition sistem. Sementara itu, bagian tim dan kontak mendukung aspek kredibilitas dan pemeliharaan sistem: pengguna dapat mengetahui pihak pengembang dan memiliki kanal untuk menyampaikan koreksi atau masukan, yang berguna untuk iterasi kualitas basis pengetahuan maupun pengalaman penggunaan.

Dengan demikian, halaman pembuka Veritas dapat dipandang sebagai komponen hasil implementasi yang berfokus pada keterbacaan, struktur informasi, dan pengarah tindakan menuju chatbot. Halaman ini belum menjalankan fungsi tanya-jawab, tetapi menyediakan konteks, edukasi singkat, dan jalur navigasi yang jelas agar pengguna siap memasuki antarmuka percakapan melalui tombol “**Mulai Chat**.”

4.3 Hasil dan Pembahasan: Antarmuka Chatbot Veritas



Tampilan Awal Halaman Chatbot

Setelah pengguna menekan tombol “**Mulai Chat**” pada halaman pembuka, sistem menampilkan antarmuka percakapan Veritas. Pada tampilan awal, halaman chat terdiri atas dua area utama, yaitu panel sisi kiri dan area percakapan di bagian tengah. Panel sisi kiri berfungsi sebagai manajemen sesi percakapan (chat session) yang memungkinkan pengguna membuat sesi baru melalui tombol “+ New chat”, serta melihat daftar sesi yang pernah dibuat. Setiap sesi dapat diberi nama (rename) atau dihapus (delete), sehingga pengguna dapat memisahkan topik percakapan yang berbeda (misalnya “Fakultas”, “Fasilitas”, atau “Dosen”) tanpa mencampur riwayatnya.

Di area percakapan, sistem menampilkan pesan pembuka otomatis dari asisten “Veri” sebagai sinyal bahwa chatbot siap menerima pertanyaan. Pada bagian bawah terdapat kotak input untuk menulis pertanyaan, tombol “**Kirim**” untuk mengirim pesan, serta teks penanda yang menegaskan bahwa jawaban diberikan berdasarkan informasi yang tersedia pada knowledge base. Selain itu, tersedia tombol “**Clear chat**” pada bagian kanan atas untuk mengosongkan percakapan dalam sesi aktif, sehingga pengguna dapat memulai ulang interaksi tanpa perlu membuat sesi baru.

Interaksi Saat Pengguna Bertanya dan Sistem Menjawab

Ketika pengguna memasukkan pertanyaan dan menekan tombol “**Kirim**”, pertanyaan akan ditampilkan sebagai pesan pengguna, kemudian sistem memproses pertanyaan tersebut dan mengembalikan jawaban pada balon pesan asisten. Dari hasil uji coba, chatbot mampu menangani pertanyaan yang berkaitan dengan informasi kampus UI yang tersedia di basis pengetahuan, misalnya permintaan daftar fakultas, informasi fasilitas, maupun topik-topik akademik dan kegiatan kampus.

Contoh pertama adalah saat pengguna meminta **daftar semua fakultas di UI**. Sistem menampilkan jawaban dalam bentuk daftar terstruktur (bernomor) yang memuat nama fakultas dan singkatannya. Bentuk keluaran daftar ini memberikan pengalaman yang lebih jelas dan mudah dibaca, serta memudahkan pengguna melakukan scanning informasi tanpa membaca paragraf panjang.

Contoh kedua adalah ketika pengguna menanyakan **fasilitas di UI**. Sistem mengembalikan jawaban berupa ringkasan deskriptif dalam format paragraf, menjelaskan beberapa fasilitas yang relevan. Pola ini menunjukkan bahwa chatbot tidak hanya mengeluarkan daftar, tetapi juga dapat menyusun jawaban naratif yang merangkum informasi dari basis pengetahuan agar lebih mudah dipahami.

Kebijakan Jawaban Saat Informasi Tidak Tersedia pada Dataset

Sistem juga menerapkan kebijakan pembatasan jawaban untuk menjaga agar informasi tetap ter-grounding pada dataset. Jika pengguna menanyakan hal yang **tidak tercakup pada basis pengetahuan**, chatbot tidak akan mengarang jawaban. Sebagai gantinya, chatbot akan menyatakan secara eksplisit bahwa informasi tersebut **tidak ditemukan di dataset yang tersedia**. Pada pengujian, hal ini terlihat ketika pengguna mengajukan pertanyaan di luar cakupan (misalnya pertanyaan yang bersifat opini atau tidak terdapat pada dokumen sumber). Respons “tidak menemukan informasi di dataset” berperan sebagai mekanisme pencegahan halusinasi dan menjaga keandalan sistem, karena pengguna diberi sinyal bahwa pertanyaan tersebut memang tidak dapat dijawab secara faktual berdasarkan knowledge base yang dimiliki.

Secara keseluruhan, hasil implementasi antarmuka chat Veritas menunjukkan alur interaksi yang jelas: pengguna memilih/membuat sesi chat, memasukkan pertanyaan, sistem menampilkan jawaban, dan pengguna dapat mengelola percakapan melalui fitur rename, delete, serta clear chat. Dari sisi perilaku jawaban, sistem mampu menjawab pertanyaan yang sesuai cakupan data, serta memberikan fallback yang transparan ketika informasi tidak tersedia di dataset.

Kesimpulan

Penelitian dan pengembangan Veritas menghasilkan sebuah chatbot asisten kampus Universitas Indonesia yang mampu memberikan jawaban ringkas dan terarah berdasarkan basis pengetahuan internal. Sistem dirancang untuk menangani kebutuhan informasi yang umum dicari mahasiswa, seperti informasi akademik, fasilitas kampus, kegiatan, serta informasi dosen, melalui antarmuka web yang mendukung manajemen sesi percakapan (pembuatan chat baru, penamaan, penghapusan, dan pembersihan percakapan). Dengan demikian, Veritas tidak hanya berfokus pada kemampuan menjawab, tetapi juga pada pengalaman penggunaan yang terstruktur dan mudah dipahami.

Dari sisi teknis, implementasi menggunakan arsitektur Retrieval-Augmented Generation (RAG) yang memadukan proses retrieval berbasis embedding dan vector database dengan proses generasi jawaban menggunakan LLM. Dokumen sumber yang tidak terstruktur (PDF) diproses melalui ekstraksi teks dan chunking, kemudian diindeks ke vector store secara terpisah per domain (umum, event, fasilitas, dan dosen) untuk meningkatkan relevansi konteks yang diambil. Pada saat query-time, sistem memanfaatkan domain routing, variasi query retrieval, serta penanganan deterministik untuk permintaan daftar agar keluaran lebih stabil dan konsisten.

Hasil pengujian menunjukkan bahwa chatbot dapat menjawab pertanyaan yang sesuai dengan cakupan dataset secara relevan, serta memiliki mekanisme pengendalian untuk mencegah halusinasi. Ketika informasi yang diminta tidak tersedia pada basis pengetahuan, sistem memberikan respons transparan bahwa informasi tidak ditemukan di dataset, sehingga menjaga keandalan dan kredibilitas jawaban. Secara keseluruhan, Veritas menunjukkan bahwa pendekatan RAG yang dikombinasikan dengan guardrails dan penanganan deterministik pada kasus tertentu efektif untuk membangun chatbot informasi kampus yang grounded, konsisten, dan siap dikembangkan lebih lanjut.

Sebagai arah pengembangan, peningkatan kualitas dapat difokuskan pada perluasan dan pemutakhiran dataset, evaluasi kuantitatif yang lebih sistematis (misalnya set pertanyaan uji per domain dan metrik groundedness), serta penguatan modul retrieval (misalnya reranking dan perbaikan normalisasi entitas). Langkah-langkah tersebut diharapkan dapat meningkatkan cakupan, ketepatan jawaban, dan kestabilan performa sistem ketika digunakan oleh pengguna yang lebih luas.

LAMPIRAN CHATBOT

https://drive.google.com/drive/folders/1eko2nvVY_Mv1685IsPUPshlWGMrhzxNy?usp=drive_link

DAFTAR PUSTAKA

1. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmquist, N., & Diakopoulos, N. (2016).
Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th ed.). Pearson.
2. Russell, S., & Norvig, P. (2021).
Artificial Intelligence: A Modern Approach (4th ed.). Pearson.
(Dasar AI & komputasi intelegensia)
3. Jurafsky, D., & Martin, J. H. (2023).
Speech and Language Processing (3rd ed., draft).
(Dasar NLP & language modeling)
4. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013).
Efficient Estimation of Word Representations in Vector Space.
arXiv preprint arXiv:1301.3781.
(Fondasi embedding vektor)
5. Reimers, N., & Gurevych, I. (2019).
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
Proceedings of EMNLP-IJCNLP.
(Embedding kalimat & cosine similarity)
6. Manning, C. D., Raghavan, P., & Schütze, H. (2008).
Introduction to Information Retrieval. Cambridge University Press.
(Similarity search, nearest neighbor, retrieval theory)
7. Lewis, P., Perez, E., Piktus, A., et al. (2020).
Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.
Advances in Neural Information Processing Systems (NeurIPS).

- (RAG – konsep utama)
8. Karpukhin, V., Oguz, B., Min, S., et al. (2020).
Dense Passage Retrieval for Open-Domain Question Answering.
Proceedings of EMNLP.
(Dense retrieval & vector database)
9. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017).
Attention Is All You Need.
Advances in Neural Information Processing Systems (NeurIPS).
- (Arsitektur transformer – dasar LLM)
10. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
Proceedings of NAACL-HLT.
(Pretrained language model)
11. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmquist, N., & Diakopoulos, N. (2016).
12. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing*. Pearson.