

BAB III

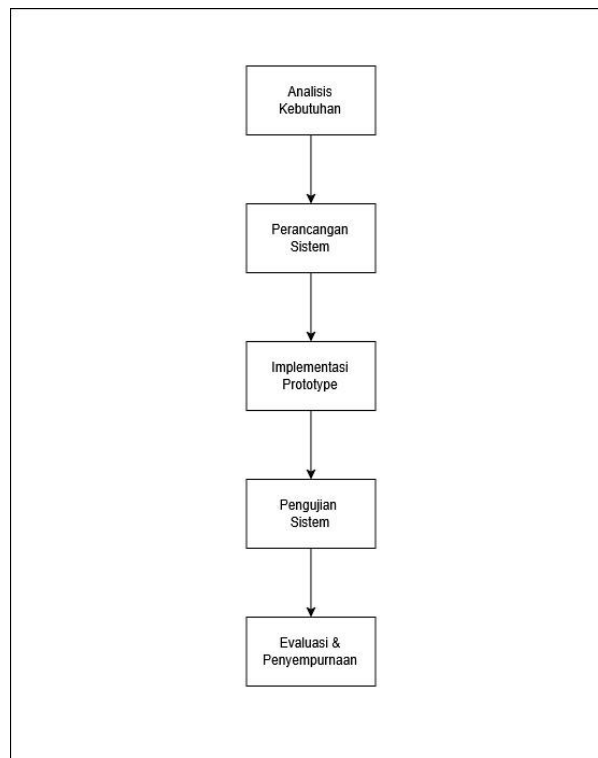
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan *Research and Development* (R&D) dengan model *Prototyping*. Pendekatan R&D dipilih karena penelitian ini bertujuan untuk menghasilkan produk berupa sistem chatbot berbasis *Large Language Model* (LLM) dengan teknologi *Retrieval Augmented Generation* (RAG) untuk YPI Al-Azhar Jakarta. Model *Prototyping* memungkinkan pengembangan sistem secara bertahap melalui pembuatan *prototype* awal, pengujian terhadap fungsionalitas dan kualitas respons, serta perbaikan berulang hingga sistem sesuai dengan kebutuhan pengguna. Pendekatan iteratif ini sangat sesuai untuk pengembangan sistem chatbot yang memerlukan penyesuaian berkelanjutan terhadap pola pertanyaan pengguna, kualitas retrieval dokumen, dan akurasi respons yang dihasilkan.

3.2 Tahapan Penelitian

Tahapan penelitian yang dilakukan meliputi analisis kebutuhan, perancangan sistem, implementasi prototype, pengujian sistem, serta evaluasi dan penyempurnaan sistem. Alur tahapan penelitian ditunjukkan pada Gambar berikut:



3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk memastikan bahwa sistem chatbot yang akan dikembangkan mampu menyelesaikan permasalahan pada sistem layanan informasi dan transaksi yang ada saat ini, serta memenuhi tujuan penelitian yang telah ditetapkan. Analisis ini mencakup:

1. Analisis Sistem Berjalan, yaitu evaluasi terhadap kondisi dan kendala sistem yang sudah ada.
2. Analisis Kebutuhan Pengguna, untuk memahami kebutuhan pengguna dalam menggunakan layanan informasi dan layanan transaksional.
3. Kebutuhan Fungsional, fitur utama yang harus dimiliki oleh sistem chatbot agar mampu beroperasi sesuai tujuan penelitian.
4. Kebutuhan Non Fungsional, aspek kualitas sistem yang memengaruhi keamanan, kinerja, dan kenyamanan penggunaan.
5. Kebutuhan Teknis, teknologi, arsitektur, dan infrastruktur yang dibutuhkan untuk mendukung pengembangan sistem chatbot berbasis *Large Language Model* (LLM) dengan pendekatan *Retrieval Augmented Generation* (RAG).

3.3.1 Analisis Sistem Berjalan

Sistem layanan online di YPI Al-Azhar saat ini telah tersedia dalam bentuk aplikasi yang berfungsi sebagai media penyampaian informasi dan layanan transaksional bagi pengguna, yang meliputi calon siswa, orang tua, dan staf administrasi. Berdasarkan hasil observasi yang dilakukan, sistem yang berjalan memiliki beberapa karakteristik sebagai berikut.

1. Layanan Informasi.

Informasi terkait prosedur, fasilitas, persyaratan administrasi, serta layanan akademik telah disediakan melalui website resmi. Namun demikian, beberapa informasi penting, seperti biaya pendidikan, belum disajikan secara rinci dan terstruktur. Akibatnya, pengguna masih perlu mencari atau mengonfirmasi informasi tersebut melalui sumber lain. Selain itu, sistem belum menyediakan mekanisme tanya jawab interaktif secara real-time, sehingga pertanyaan spesifik harus diklarifikasi melalui telepon, email, atau kunjungan langsung.

2. Layanan Transaksional

Proses pendaftaran dan pengelolaan data dilakukan melalui formulir online yang bersifat statis. Pengguna, baik calon siswa maupun orang tua, diharuskan mengisi seluruh data yang dibutuhkan sekaligus tanpa adanya panduan interaktif atau bantuan langsung selama proses pengisian.

3. Sistem Informasi

Sistem informasi yang ada telah mengelola data pendaftaran dan informasi layanan pendidikan. Namun, sistem tersebut masih bersifat terpisah dan belum terintegrasi secara menyeluruh.

Berdasarkan analisis terhadap sistem yang berjalan, terdapat beberapa kendala utama, yaitu sebagai berikut.

1. Waktu respons layanan relatif lambat karena masih bergantung pada layanan customer service manual dengan jam operasional terbatas.
2. Terjadi inkonsistensi informasi yang diberikan oleh petugas customer service.
3. Informasi layanan tersebar di berbagai sumber sehingga sulit diakses secara cepat dan efisien.
4. Formulir pendaftaran tidak bersifat interaktif, yang menyebabkan tingginya tingkat kesalahan input serta tidak tersedianya panduan secara real-time.
5. Sistem belum memiliki kemampuan untuk menjawab pertanyaan pengguna secara otomatis berbasis bahasa alami.
6. Sistem informasi belum terintegrasi sepenuhnya, sehingga proses monitoring dan manajemen data belum berjalan secara optimal.

3.3.2 Analisis Kebutuhan Pengguna

Berdasarkan analisis terhadap sistem yang berjalan, kebutuhan pengguna dalam penelitian ini dikelompokkan menjadi dua kategori utama, yaitu calon siswa/orang tua dan staf administrasi YPI Al-Azhar.

1. Calon Siswa / Orang Tua

Calon siswa dan orang tua sebagai pengguna utama layanan pendaftaran memiliki kebutuhan sebagai berikut.

- a. Membutuhkan akses terhadap informasi yang cepat, akurat, dan terstruktur mengenai prosedur pendaftaran, biaya pendidikan per jenjang, persyaratan

dokumen, serta fasilitas sekolah tanpa harus melakukan pencarian manual di berbagai halaman atau menunggu respons dari customer service.

- b. Membutuhkan bantuan interaktif selama proses pendaftaran berupa panduan bertahap (step-by-step) agar proses pengisian data menjadi lebih mudah dipahami dan dapat mengurangi kesalahan input.
- c. Membutuhkan fitur pelacakan (tracking) status pendaftaran secara real-time sehingga pengguna dapat mengetahui tahapan proses pendaftaran yang sedang berlangsung.
- d. Membutuhkan layanan sistem yang dapat diakses selama 24 jam sehari dan 7 hari seminggu (24/7), tanpa bergantung pada jam operasional kantor.

2. Staf Administrasi YPI Al-Azhar

Staf administrasi sebagai pengelola sistem memiliki kebutuhan sebagai berikut.

- a. Membutuhkan sistem yang mampu mengurangi beban pelayanan pertanyaan berulang dari calon siswa dan orang tua, sehingga staf dapat lebih fokus pada pekerjaan administrasi yang bersifat kompleks dan strategis.
- b. Membutuhkan sistem yang dapat memandu calon siswa dalam melengkapi formulir pendaftaran secara benar, sehingga dapat menekan tingkat kesalahan input data dan mengurangi kebutuhan verifikasi ulang.
- c. Membutuhkan dashboard terpadu yang dapat digunakan untuk memantau jumlah pendaftaran yang masuk serta status proses pendaftaran secara real-time.
- d. Membutuhkan sistem yang mudah diperbarui ketika terjadi perubahan kebijakan, informasi, atau ketentuan pendaftaran tanpa harus melakukan perubahan langsung pada kode program.

3.3.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan kemampuan utama yang harus dimiliki oleh sistem chatbot agar dapat berfungsi sesuai dengan tujuan penelitian, yaitu sebagai sistem layanan informasi dan layanan transaksional dalam proses Penerimaan Siswa Baru (PSB) di YPI Al-Azhar.

Berdasarkan hasil analisis kebutuhan pengguna dan ruang lingkup penelitian, sistem chatbot yang akan dibangun harus mampu memenuhi kebutuhan fungsional sebagai berikut.

1. Menyediakan Layanan Informasi Terpadu

Sistem chatbot mampu menjawab pertanyaan pengguna secara otomatis terkait layanan umum, akademik, administratif, dan non-akademik di lingkungan YPI Al-Azhar berdasarkan data internal yang tersimpan dalam knowledge base sistem.

2. Mengakses dan Memproses Data Internal Organisasi

Sistem chatbot mampu melakukan pencarian dan pengambilan informasi dari berbagai sumber data internal YPI Al-Azhar melalui mekanisme *Retrieval Augmented Generation* (RAG) tanpa memerlukan proses retraining pada model *Large Language Model* (LLM).

3. Memandu Proses Pendaftaran Siswa Baru Secara Interaktif

Sistem chatbot mampu berperan sebagai agen transaksional yang memandu calon siswa atau orang tua dalam proses pendaftaran siswa baru melalui percakapan interaktif berbasis formulir bertahap (*multi-step form*).

4. Mengelola Pengisian Formulir Pendaftaran

Sistem chatbot mampu mengumpulkan data pendaftaran siswa baru secara bertahap, melakukan validasi input dasar, serta menyimpan data pendaftaran ke dalam basis data sistem.

5. Menerima dan Memvalidasi Dokumen Pendaftaran

Sistem chatbot mampu menerima unggahan dokumen pendaftaran, seperti identitas diri, dokumen akademik, dan berkas pendukung lainnya, melakukan validasi format dan kelengkapan dokumen, serta menyimpan dokumen secara terstruktur dalam sistem.

6. Menampilkan Status Pendaftaran Secara Real-Time

Sistem chatbot mampu menampilkan informasi status pendaftaran siswa secara real-time, mulai dari tahap pengisian formulir, verifikasi dokumen, hingga status akhir pendaftaran.

7. Mendukung Pembaruan Knowledge Base Secara Adaptif

Sistem chatbot mampu memperbarui basis pengetahuan secara otomatis ketika terdapat data atau dokumen baru tanpa memerlukan proses *retraining* pada model LLM utama.

3.3.4 Kebutuhan Non Fungsional

Kebutuhan non-fungsional berkaitan dengan karakteristik kualitas sistem chatbot yang memengaruhi kenyamanan pengguna, tingkat keamanan, serta kinerja sistem secara keseluruhan. Kebutuhan non-fungsional ini menjadi faktor pendukung agar sistem dapat beroperasi secara optimal dan berkelanjutan.

Berdasarkan tujuan pengembangan sistem, kebutuhan non-fungsional sistem chatbot yang akan dibangun meliputi sebagai berikut.

1. Keamanan

Sistem harus mampu melindungi data pribadi calon siswa dan pengguna sesuai dengan prinsip keamanan data. Pengamanan mencakup kontrol akses terhadap *Application Programming Interface* (API), perlindungan data sensitif, serta pengelolaan dan penyimpanan dokumen pendaftaran secara aman.

2. Kinerja

Sistem chatbot harus memiliki waktu respons yang cepat dan stabil, baik dalam menjawab pertanyaan informasional maupun dalam memproses transaksi pendaftaran siswa baru, sehingga dapat memberikan pengalaman pengguna yang baik.

3. Skalabilitas

Sistem harus mampu menangani peningkatan jumlah pengguna yang mengakses layanan secara bersamaan, khususnya pada periode puncak pelaksanaan Penerimaan Siswa Baru.

4. Ketersediaan

Sistem chatbot diharapkan dapat diakses selama 24 jam sehari tanpa bergantung pada jam operasional kantor, guna mendukung layanan informasi yang tersedia setiap saat.

5. Kemudahan Penggunaan

Antarmuka chatbot harus dirancang secara sederhana, intuitif, dan mudah dipahami oleh pengguna awam, termasuk orang tua dan calon siswa dengan latar belakang kemampuan teknologi yang beragam.

6. Kemudahan Pemeliharaan (Maintainability)

Sistem harus mudah dipelihara dan dikembangkan lebih lanjut, terutama dalam hal penambahan sumber data baru, pembaruan knowledge base, serta pengembangan fitur lanjutan di masa mendatang.

3.3.5 Analisis Kebutuhan Teknis

Analisis kebutuhan teknis dilakukan untuk menentukan teknologi, arsitektur, serta infrastruktur yang mendukung pengembangan sistem chatbot berbasis *Large Language Model* (LLM) dan *Retrieval Augmented Generation* (RAG). Analisis ini dilakukan melalui studi literatur, observasi lapangan, serta inventarisasi data dan kebutuhan sistem.

Berdasarkan hasil analisis tersebut, kebutuhan teknis sistem chatbot yang akan dibangun adalah sebagai berikut.

1. Large Language Model (LLM)

Sistem menggunakan model LLM publik berbasis Application Programming Interface (API) untuk menghasilkan respons bahasa alami. Model LLM tidak dilakukan proses *retraining*, melainkan diintegrasikan dengan pendekatan *Retrieval Augmented Generation* guna menyesuaikan jawaban dengan konteks data internal.

2. Retrieval Augmented Generation (RAG)

Pendekatan RAG digunakan untuk menggabungkan kemampuan generatif LLM dengan sistem pengambilan informasi berbasis *vector database*, sehingga chatbot dapat mengakses dan memanfaatkan data internal YPI Al-Azhar secara dinamis dan kontekstual.

3. Vector Database

Vector database digunakan untuk menyimpan representasi embedding dari dokumen internal YPI Al-Azhar. Basis data ini memungkinkan proses pencarian semantik dilakukan secara lebih efisien dan akurat dibandingkan pencarian berbasis kata kunci.

4. Data Ingestion dan Processing Pipeline

Sistem memerlukan *pipeline* pemrosesan data yang mencakup proses ekstraksi teks dari dokumen, pemecahan dokumen (*chunking*), pembuatan embedding (*embedding generation*), serta penyimpanan hasil embedding ke dalam *vector database*.

5. Arsitektur Microservices

Sistem dirancang menggunakan arsitektur *microservices* yang terdiri atas beberapa modul utama, yaitu *Data Ingestion Layer*, *Processing Pipeline*, *Retrieval and Generation Layer*, *Adaptive Learning Module*, *Transactional Module*, dan *User Interface Layer*. Pendekatan ini dipilih untuk meningkatkan skalabilitas dan kemudahan pengembangan sistem.

6. Backend dan API Services

Sistem backend dikembangkan menggunakan layanan *RESTful API* untuk mendukung layanan informasional dan transaksional, termasuk proses pendaftaran siswa baru, unggah dokumen, serta pengecekan status pendaftaran.

7. User Interface Chat Interaktif

Antarmuka chatbot dirancang untuk mendukung percakapan berbasis teks, pengisian formulir, unggah dokumen, serta visualisasi status pendaftaran. Antarmuka ini juga dapat diintegrasikan ke berbagai platform layanan digital yang digunakan oleh YPI Al-Azhar.

3.4 Perancangan Sistem

Tahap perancangan sistem dilakukan berdasarkan hasil analisis kebutuhan yang telah diperoleh pada tahap sebelumnya. Tujuan dari tahap ini adalah untuk merancang struktur, komponen, dan alur kerja sistem chatbot secara menyeluruh agar sistem yang dibangun sesuai dengan kebutuhan pengguna dan tujuan penelitian.

Perancangan sistem mencakup perancangan arsitektur sistem, alur pemrosesan data, mekanisme Retrieval Augmented Generation (RAG), alur percakapan chatbot, basis data, antarmuka pemrograman aplikasi (API), serta antarmuka pengguna. Pada tahap ini, rancangan dibuat sebagai acuan utama dalam proses implementasi prototype.

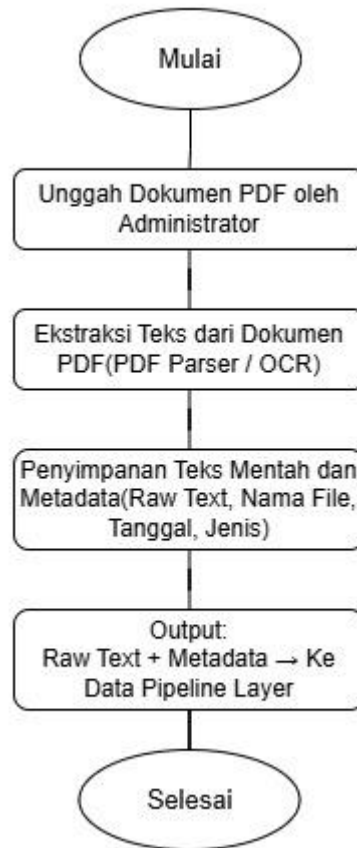
3.4.1 Arsitektur Sistem

Sistem dirancang menggunakan arsitektur microservices berlapis yang terdiri dari beberapa komponen utama yang saling terintegrasi. Arsitektur ini memisahkan secara jelas tanggung jawab antara proses ingestion data, pemrosesan dan pengelolaan pengetahuan, serta layanan aplikasi dan antarmuka pengguna. Selain itu, sistem dilengkapi dengan mekanisme quality assurance berbasis kecerdasan buatan yang diterapkan pada tahap pemrosesan pengetahuan untuk meningkatkan kualitas, keterbacaan, dan konsistensi data sebelum digunakan oleh komponen *Retrieval Augmented Generation* (RAG). Pendekatan ini dirancang untuk mendukung peningkatan modularitas, skalabilitas, serta kemudahan pemeliharaan sistem chatbot secara keseluruhan.

Layer	Komponen dan Fungsi
Data Ingestion Layer	Menangani proses unggah dokumen PDF, ekstraksi teks dari dokumen digital maupun hasil pemindaian menggunakan PDF parser, serta penyimpanan teks mentah dan metadata dokumen ke sistem backend.
Processing Pipeline Layer	Melakukan pemrosesan lanjutan terhadap hasil ekstraksi dokumen, termasuk penulisan ulang dan penataan konten secara otomatis untuk membentuk knowledge base yang lebih terstruktur dan mudah dipahami, pemecahan teks dengan skema overlap untuk menjaga kesinambungan konteks, penambahan metadata dokumen seperti sumber, kategori informasi, lokasi atau cabang, jenjang pendidikan, serta periode berlakunya informasi, dan pembentukan representasi vector untuk mendukung pencarian semantik.
Database Layer	ChromaDB untuk vector storage dengan metadata filtering, PostgreSQL untuk relational data dan audit trail
RAG Engine Layer	Mengelola proses pencarian dan penyusunan jawaban chatbot melalui mekanisme <i>query rewriting</i> berbasis konteks dan metadata, pencarian dokumen menggunakan

	<i>semantic retrieval</i> pada vector database yang diperkaya dengan penyaringan metadata, penyusunan dan perankingan konteks hasil pencarian, serta integrasi dengan model bahasa besar melalui teknik <i>prompt engineering</i> untuk menghasilkan jawaban yang relevan dan kontekstual.
Application Layer	FastAPI backend dengan REST API endpoints, business logic untuk intent classification (<i>informational vs transactional</i>), session management dan conversation history
Presentation Layer	Chat interface untuk interaksi user-chatbot, document management interface, bulk review system untuk knowledge base, progress tracking untuk pendaftaran siswa

3.4.2 Perancangan Data Ingestion Layer



Data Ingestion Layer berfungsi sebagai pintu masuk data ke dalam sistem chatbot dan bertanggung jawab dalam mengelola dokumen sumber yang digunakan sebagai basis pengetahuan. Pada penelitian ini, sumber data utama berupa dokumen PDF yang berisi informasi resmi terkait biaya pendidikan serta dokumen pendukung lainnya.

Proses pada Data Ingestion Layer difokuskan pada pengambilan dan ekstraksi informasi tekstual dari dokumen PDF agar dapat diproses lebih lanjut oleh layer berikutnya. Alur proses pada layer ini terdiri dari tahapan sebagai berikut.

1. Unggah Dokumen PDF

Administrator sistem mengunggah dokumen PDF melalui antarmuka sistem. Dokumen yang diunggah merupakan dokumen resmi yang telah diverifikasi secara internal sehingga dapat dijadikan sumber informasi yang valid bagi sistem chatbot.

2. Ekstraksi Teks dari Dokumen PDF

Sistem melakukan ekstraksi konten tekstual dari dokumen PDF menggunakan mekanisme PDF parser. Proses ini bertujuan untuk memperoleh teks yang terkandung di dalam dokumen secara terstruktur sebagai bahan awal pemrosesan.

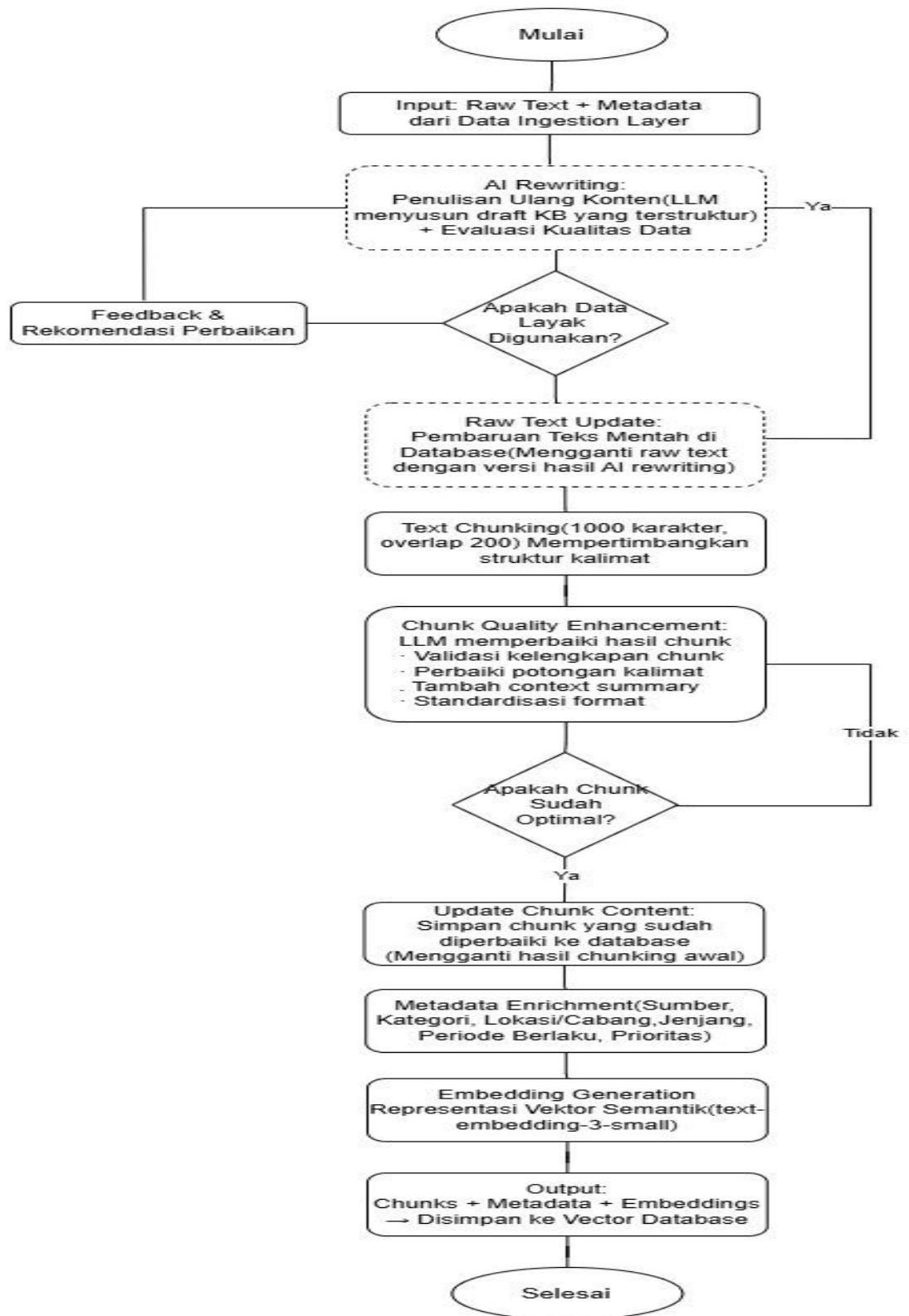
3. Penyimpanan Teks Mentah dan Metadata

Hasil ekstraksi teks disimpan ke dalam sistem backend sebagai teks mentah, disertai metadata dokumen yang meliputi nama file, tanggal unggah, dan jenis dokumen. Penyimpanan ini dilakukan untuk mendukung audit trail, pelacakan sumber data, serta menjaga keterlacakan informasi.

4. Output Data Ingestion Layer

Sebagai hasil akhir dari Data Ingestion Layer, sistem menghasilkan teks mentah beserta metadata dokumen yang selanjutnya diteruskan ke Data Pipeline Layer untuk dilakukan pemrosesan lanjutan.

3.4.3 Perancangan Data Pipeline



Data Pipeline Layer bertugas melakukan pemrosesan lanjutan dan transformasi data dari teks mentah hasil ekstraksi menjadi basis pengetahuan yang terstruktur serta representasi vektor yang siap digunakan dalam proses pencarian semantik pada sistem chatbot. Layer ini menjadi komponen penting yang membedakan pendekatan penelitian ini dengan sistem chatbot konvensional karena mengintegrasikan pemrosesan berbasis kecerdasan buatan untuk meningkatkan kualitas dan konsistensi data.

Tahapan proses pada Data Pipeline Layer dijelaskan sebagai berikut.

1. Penulisan Ulang dan Penataan Konten Secara Otomatis

Sistem memanfaatkan *Large Language Model* untuk melakukan penulisan ulang terhadap konten teks mentah agar menjadi lebih terstruktur, jelas, dan mudah dipahami. Proses ini mencakup penyusunan draft basis pengetahuan berdasarkan topik dan kategori informasi yang relevan.

Selain itu, model AI melakukan evaluasi kualitas data secara menyeluruh yang meliputi pemeriksaan kelengkapan informasi, deteksi duplikasi konten, penilaian relevansi terhadap domain pendidikan, konsistensi istilah dan format, serta identifikasi potensi kesalahan akibat proses ekstraksi. Apabila hasil evaluasi menunjukkan bahwa data belum memenuhi standar kualitas, sistem menyediakan mekanisme umpan balik berupa rekomendasi perbaikan dan memungkinkan proses iterasi hingga kualitas data dinyatakan layak digunakan.

2. Pembaruan Teks Mentah

Setelah proses AI Rewriting dan validasi kualitas data, sistem melakukan pembaruan terhadap teks mentah di dalam basis data. Teks mentah yang telah diperbarui merupakan versi hasil penulisan ulang dan perbaikan kualitas, sehingga menggantikan teks hasil ekstraksi awal. Tahapan ini bertujuan untuk memastikan bahwa seluruh proses pemecahan teks dan pembentukan representasi vektor selanjutnya menggunakan data yang telah tervalidasi dan bebas dari inkonsistensi maupun kesalahan ekstraksi.

3. Pemecahan Teks (Text Chunking)

Basis pengetahuan yang telah diperbarui dan tervalidasi dipecah menjadi bagian-bagian kecil (chunks) dengan ukuran sekitar 1000 karakter dan overlap sebesar 200

karakter antar chunks. Penggunaan overlap bertujuan untuk menjaga kesinambungan konteks antar potongan teks dan mencegah hilangnya informasi penting pada batas pemotongan. Proses chunking dilakukan dengan mempertimbangkan struktur kalimat agar koherensi makna tetap terjaga.

4. Peningkatan Kualitas Chunk

Setelah proses pemecahan teks, sistem melakukan tahap tambahan untuk memastikan kualitas setiap chunk yang dihasilkan. *Large Language Model* kembali digunakan untuk melakukan validasi dan perbaikan terhadap hasil chunking dengan beberapa aspek berikut:

a. Validasi Kelengkapan Chunk

Sistem memeriksa apakah setiap chunk memiliki informasi yang lengkap dan dapat dipahami secara mandiri. Chunk yang terpotong di tengah kalimat atau kehilangan konteks penting akan diidentifikasi untuk diperbaiki.

b. Perbaikan Potongan Kalimat

Apabila ditemukan chunk yang dimulai atau diakhiri dengan potongan kalimat yang tidak lengkap, sistem akan melakukan penyesuaian batas chunk atau menambahkan konteks minimal yang diperlukan agar chunk tetap koheren.

c. Penambahan Context Summary

Setiap chunk diperkaya dengan ringkasan konteks singkat yang menjelaskan posisi informasi dalam dokumen keseluruhan. Context summary ini membantu sistem RAG memahami hubungan antar chunk dan meningkatkan akurasi retrieval.

d. Standardisasi Format

Sistem memastikan konsistensi format penulisan di seluruh chunk, termasuk penggunaan istilah, kapitalisasi, dan struktur kalimat, sehingga meminimalkan variasi yang dapat mengganggu proses embedding.

Apabila hasil evaluasi menunjukkan chunk belum optimal, sistem dapat melakukan iterasi ulang proses chunking dengan parameter yang disesuaikan, seperti mengubah ukuran chunk atau overlap, hingga kualitas chunk memenuhi standar yang ditetapkan.

5. Pembaruan Konten Chunk

Setelah proses Chunk Quality Enhancement, sistem melakukan pembaruan terhadap konten chunk di dalam basis data. Chunk yang telah diperbaiki dan diperkaya menggantikan hasil chunking awal, memastikan bahwa data yang akan digunakan untuk metadata enrichment dan embedding generation merupakan versi terbaik yang telah melalui validasi berlapis.

6. Penambahan Metadata Dokumen

Setiap chunk teks diperkaya dengan metadata yang komprehensif untuk meningkatkan akurasi proses pencarian dan penyaringan informasi. Metadata yang ditambahkan meliputi sumber dokumen, kategori informasi, lokasi atau cabang sekolah, jenjang pendidikan, periode berlakunya informasi, serta tingkat prioritas informasi.

7. Pembentukan Representasi Vektor

Setiap chunk teks yang telah diperkaya metadata diubah menjadi representasi vektor numerik yang merepresentasikan makna semantik dari teks. Representasi vektor ini memungkinkan sistem melakukan pencarian berbasis kesamaan makna sehingga hasil retrieval menjadi lebih relevan dibandingkan pencarian berbasis kecocokan kata kunci semata.

8. Output Data Pipeline Layer

Sebagai hasil akhir dari Data Pipeline Layer, sistem menghasilkan kumpulan chunk teks yang telah dilengkapi dengan metadata dan embedding. Data ini kemudian disimpan ke dalam vector database dan siap digunakan oleh RAG Engine dalam proses pengambilan informasi dan generasi jawaban chatbot.

3.4.4 Perancangan Infrastruktur dan Skema Basis Data

Perancangan infrastruktur basis data pada penelitian ini bertujuan untuk mendukung proses penyimpanan, pengelolaan, dan pengambilan data secara efisien dalam sistem chatbot berbasis kecerdasan buatan. Infrastruktur basis data dirancang menggunakan dua pendekatan penyimpanan, yaitu vector database untuk mendukung pencarian semantik dan relational database untuk pengelolaan data transaksional dan terstruktur. Pendekatan ini memungkinkan sistem untuk menangani kebutuhan pencarian informasi berbasis konteks secara akurat, sekaligus menjaga konsistensi dan integritas data.

1. ChromaDB

ChromaDB digunakan sebagai vector database untuk menyimpan representasi vector yang dihasilkan dari proses pemrosesan dokumen. Embeddings ini digunakan dalam proses semantic retrieval pada chatbot.

ChromaDB mendukung beberapa fitur utama, antara lain:

- a. Similarity search berbasis cosine similarity untuk menemukan potongan dokumen yang paling relevan secara semantik.
- b. Metadata filtering untuk penyaringan data berdasarkan konteks tertentu, seperti lokasi, jenjang pendidikan, kategori dokumen, dan periode waktu.
- c. Hybrid search, yaitu kombinasi pencarian semantik dan pencarian berbasis kata kunci guna meningkatkan relevansi hasil pencarian.

Field	Deskripsi
Id	Identitas Unik Embedding
Document_id	Referensi ke dokumen Sumber
Chunk_id	Referensi Ke Potongan dokumen
Embedding_vector	Representasi Vektor Hasil Embedding
Metadata	Informasi Pendukung

Skema ini memungkinkan sistem melakukan pencarian berbasis makna sekaligus mempertahankan konteks dokumen sumber.

2. PostgreSQL

PostgreSQL digunakan sebagai *relational database* untuk menyimpan data terstruktur dan relasional yang diperlukan oleh sistem. Basis data ini berfungsi dalam pengelolaan data operasional utama, meliputi informasi pendaftaran peserta didik, riwayat percakapan chatbot, serta pencatatan aktivitas sistem.

Perancangan Skema Basis Data Relasional

Skema basis data dirancang untuk menyimpan document chunks, embeddings metadata, data pendaftaran siswa, serta status dan riwayat interaksi pengguna dengan chatbot. Struktur tabel dirancang untuk menjaga integritas data dan mendukung proses analisis serta pelacakan status secara berkelanjutan.

Tabel	Kolom Utama	Tipe Data & Constraint
document_chunks	id, filename, content, metadata_json, status, created_at, updated_at	INT PK, VARCHAR (255), TEXT, JSON, VARCHAR (50) DEFAULT 'pending', TIMESTAMP
document_embeddings	id, chunk_id, vector, created_at	INT PK, INT FK → document_chunks(id), JSON, TIMESTAMP
student_registrations	id, registration_number, student_data, parent_data, academic_data, status, created_at	INT PK, VARCHAR (20) UNIQUE, JSON, JSON, VARCHAR (50), TIMESTAMP
registration_documents	id, registration_id, document_type, filename, file_path, status, uploaded_at	INT PK, INT FK → student_registrations(id), VARCHAR (50), VARCHAR (255), TEXT, VARCHAR (50), TIMESTAMP
registration_tracking	id, registration_id, status, notes, created_at	INT PK, INT FK → student_registrations(id), VARCHAR (100), TEXT, TIMESTAMP
conversations	id, session_id, user_message, bot_response, created_at	INT PK, VARCHAR (100), TEXT, TEXT, TIMESTAMP
conversation_state	session_id, current_step, collected_data, created_at, updated_at	VARCHAR (100) PK, VARCHAR (50), JSON, TIMESTAMP, TIMESTAMP

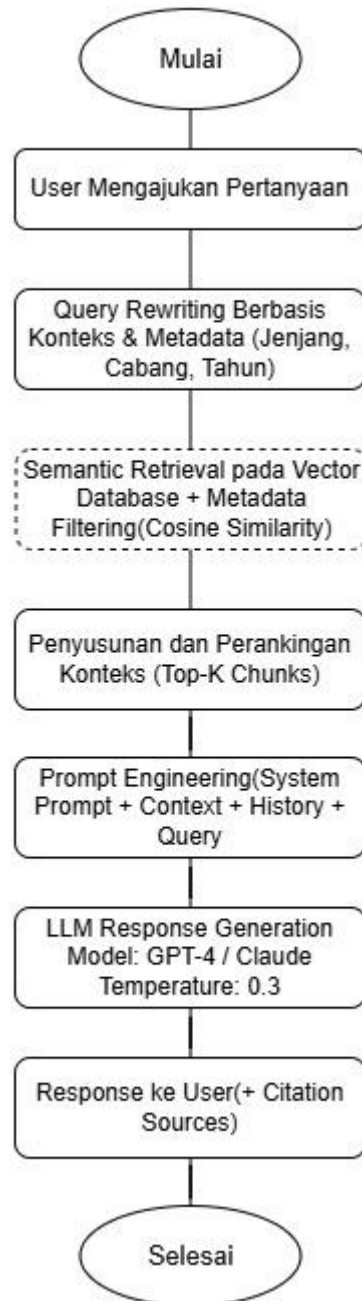
3. Output Infrastruktur Basis Data

Output dari perancangan infrastruktur dan skema basis data ini adalah sistem penyimpanan yang mampu:

- a. Mendukung operasi read/write dengan performa tinggi.
- b. Menyediakan pencarian informasi yang kompleks, baik secara semantik maupun relasional.
- c. Menjaga integritas, konsistensi, dan keamanan data.
- d. Mendukung kebutuhan monitoring dan evaluasi melalui audit trail dan riwayat interaksi.

Dengan desain ini, sistem chatbot dapat beroperasi secara optimal dalam memberikan informasi yang akurat, relevan, dan kontekstual kepada pengguna.

3.4.5 Perancangan RAG Engine



RAG Engine merupakan inti dari sistem chatbot yang dirancang untuk mengintegrasikan kemampuan pencarian informasi berbasis makna (*semantic search*) dengan kemampuan generatif dari *Large Language Model* (LLM). Tujuan utama dari layer ini adalah memastikan bahwa setiap respons yang dihasilkan chatbot bersifat relevan, kontekstual, akurat, dan dapat diverifikasi berdasarkan dokumen sumber yang tersedia dalam basis pengetahuan.

RAG Engine mengelola keseluruhan proses mulai dari pemrosesan pertanyaan pengguna, pengambilan informasi yang relevan, penyusunan konteks, hingga generasi respons berbasis LLM.

1. Pemrosesan dan Penulisan Ulang Query

Pertanyaan yang diajukan oleh pengguna terlebih dahulu melalui tahap preprocessing dan query rewriting. Pada tahap ini, sistem melakukan analisis untuk mengenali entitas penting, seperti nama, tanggal, jenjang pendidikan, lokasi atau cabang sekolah. Selain itu, sistem memperkaya query dengan informasi konteks tambahan yang diperoleh dari metadata dan conversation history, sehingga pertanyaan menjadi lebih spesifik dan terarah.

Tahap ini bertujuan untuk meningkatkan akurasi proses pencarian dokumen pada tahap selanjutnya.

2. Strategi Pengambilan Informasi

Setelah query diproses, sistem melakukan proses pengambilan informasi (retrieval) dari vector database menggunakan pendekatan semantic search. Pencarian dilakukan dengan menghitung *cosine similarity* antara embedding query dan embedding potongan dokumen (*document chunks*) yang tersimpan dalam ChromaDB. Untuk meningkatkan relevansi hasil pencarian, proses semantic retrieval diperkaya dengan metadata filtering, sehingga hanya dokumen yang sesuai dengan konteks pengguna (lokasi, jenjang pendidikan, kategori dokumen, dan periode berlaku) yang akan dipertimbangkan. Selain itu, sistem juga mendukung hybrid search, yaitu kombinasi pencarian berbasis makna dan pencarian berbasis kata kunci (*keyword-based search*), guna menangkap istilah spesifik yang mungkin tidak sepenuhnya terwakili dalam embedding.

3. Penyusunan dan Perankingan Konteks

Hasil pencarian berupa sejumlah *top-K document chunks* kemudian disusun menjadi sebuah konteks terpadu. Pada tahap ini, sistem melakukan:

- a. Penghapusan duplikasi informasi,
- b. Penyusunan urutan dokumen secara koheren berdasarkan *relevance score*,
- c. Re-ranking konteks berdasarkan keterkaitan metadata dan dependensi informasi,

- d. Penyesuaian panjang konteks agar sesuai dengan batas input LLM.

Proses ini memastikan bahwa hanya informasi yang paling relevan dan berkualitas tinggi yang digunakan sebagai dasar dalam generasi jawaban.

4. Integrasi LLM dan Prompt Engineering

Setelah konteks disusun, sistem melakukan prompt engineering untuk mengarahkan LLM dalam menghasilkan respons yang tepat. Prompt yang dikirimkan ke LLM terdiri dari beberapa komponen utama, yaitu:

- a. System prompt, yang berisi instruksi peran, batasan, dan pedoman perilaku chatbot,
- b. Retrieved context, yaitu potongan dokumen hasil proses retrieval,
- c. Conversation history, untuk menjaga kesinambungan konteks percakapan,
- d. User query, yaitu pertanyaan pengguna saat ini.

5. Generasi Respons

Pada tahap akhir, LLM menghasilkan jawaban berdasarkan konteks dan prompt yang telah disiapkan. Respons yang dihasilkan bersifat informatif, kontekstual, dan sesuai dengan kebutuhan pengguna. Jika sistem tidak menemukan dokumen yang relevan, chatbot akan memberikan jawaban yang bersifat informatif dengan menyatakan keterbatasan informasi yang tersedia.

6. Output Layer

Output dari RAG Engine adalah respons chatbot yang akurat, kontekstual, dan dapat diverifikasi, dengan dukungan sumber informasi dari dokumen asli. Dengan perancangan ini, sistem mampu mengurangi risiko *hallucination*, meningkatkan relevansi jawaban, serta memberikan pengalaman interaksi yang lebih andal bagi pengguna.

3.4.6 Perancangan Conversation Flow

Conversation flow dirancang untuk mendukung dua mode interaksi utama pada chatbot YPI Al-Azhar:

1. Mode Informational

Pada mode ini, chatbot memberikan jawaban terhadap pertanyaan pengguna terkait informasi umum YPI Al-Azhar. Alur kerjanya mencakup:

- a. Pengguna mengajukan pertanyaan.
- b. Sistem memproses pertanyaan, mengenali entitas penting dan tujuan pertanyaan.
- c. Sistem mengambil dokumen atau informasi relevan dari basis pengetahuan.
- d. Chatbot menghasilkan respons yang sesuai dan menyertakan referensi sumber jika diperlukan.
- e. Pengguna dapat melanjutkan dengan pertanyaan lanjutan secara interaktif.

2. Mode Transactional

Pada mode ini, chatbot memandu calon siswa melalui proses pendaftaran secara bertahap, yang meliputi:

- a. Menyambut pengguna dan memberikan gambaran umum tahapan pendaftaran.
- b. Pengumpulan data pribadi calon siswa secara interaktif dengan validasi setiap langkah.
- c. Pengumpulan data orang tua/wali secara interaktif dan terstruktur.
- d. Pengumpulan informasi akademik dari sekolah sebelumnya, termasuk nilai rapor terakhir.
- e. Unggah dokumen persyaratan satu per satu, dengan validasi format dan ukuran berkas.
- f. Konfirmasi data yang telah diisi dan perkiraan biaya pendaftaran.
- g. Proses pembuatan nomor registrasi.

3.4.7 Perancangan API

API dirancang untuk menyediakan antarmuka komunikasi antara frontend dan sistem chatbot secara terstruktur. Prinsip perancangan mengikuti konsep *RESTful*, sehingga setiap layanan sistem dapat diakses melalui metode HTTP yang sesuai.

Endpoint utama mencakup:

- a. Proses dokumen dan penyimpanan data ke basis pengetahuan.
- b. Pembuatan representasi data untuk mendukung pencarian informasi.
- c. Interaksi chatbot untuk menjawab pertanyaan pengguna.
- d. Layanan manajemen data dan status pendaftaran siswa.
- e. Dokumentasi interaktif yang memudahkan pengembang atau integrasi sistem lain.

API memungkinkan integrasi yang fleksibel dengan berbagai komponen sistem, mendukung pengelolaan data transaksional, pengambilan informasi berbasis pengetahuan, dan monitoring proses pendaftaran secara real-time.

3.4.8 Perancangan Antarmuka Pengguna

Antarmuka pengguna dirancang agar interaksi dengan sistem chatbot menjadi mudah, intuitif, dan responsif di berbagai platform. Pendekatan yang digunakan memungkinkan integrasi antarmuka ke web maupun aplikasi mobile melalui satu mekanisme embed. Komponen utama meliputi:

1. Chat Widget
 - a. Menyediakan sarana percakapan interaktif dengan chatbot.
 - b. Mempermudah pengguna melihat respons sistem dan menindaklanjuti pertanyaan.
2. Komponen Unggah Dokumen
 - a. Memfasilitasi pengunggahan dokumen persyaratan pendaftaran secara bertahap.
 - b. Menyediakan validasi dan umpan balik visual untuk memastikan dokumen diterima sesuai ketentuan.
3. Multi-Step Form Wizard
 - a. Membimbing pengguna melalui tahapan pendaftaran secara bertahap.

- b. Memberikan panduan visual untuk setiap langkah, validasi data, dan navigasi maju-mundur antar langkah.
- c. Memastikan proses pendaftaran dapat dilanjutkan apabila terjadi gangguan atau disconnect.

4. Status Tracking Dashboard

- a. Memberikan gambaran progres pendaftaran secara jelas.
- b. Menampilkan informasi status, catatan, dan dokumen yang sudah diunggah secara terstruktur.

Secara keseluruhan, desain antarmuka fokus pada kemudahan interaksi, transparansi proses pendaftaran, dan integrasi lancar dengan backend sistem.