BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan *Research and Development* (R&D) dengan menerapkan model Prototyping. Pendekatan R&D dipilih karena penelitian ini tidak hanya bertujuan untuk menganalisis suatu fenomena, tetapi juga untuk menghasilkan dan mengembangkan sebuah produk berupa sistem chatbot berbasis *Large Language Model* (LLM) yang terintegrasi dengan teknologi *Retrieval Augmented Generation* (RAG) guna mendukung layanan informasi dan administrasi di lingkungan YPI Al-Azhar Jakarta.

Pendekatan R&D memungkinkan peneliti untuk melakukan proses pengembangan secara sistematis yang mencakup perancangan, pembuatan, pengujian, dan penyempurnaan sistem. Selain menghasilkan produk, penelitian ini juga bertujuan untuk mengukur efektivitas, fungsionalitas, serta kualitas respons chatbot, khususnya dalam menjawab kebutuhan pengguna terhadap informasi dan layanan pendaftaran siswa baru.

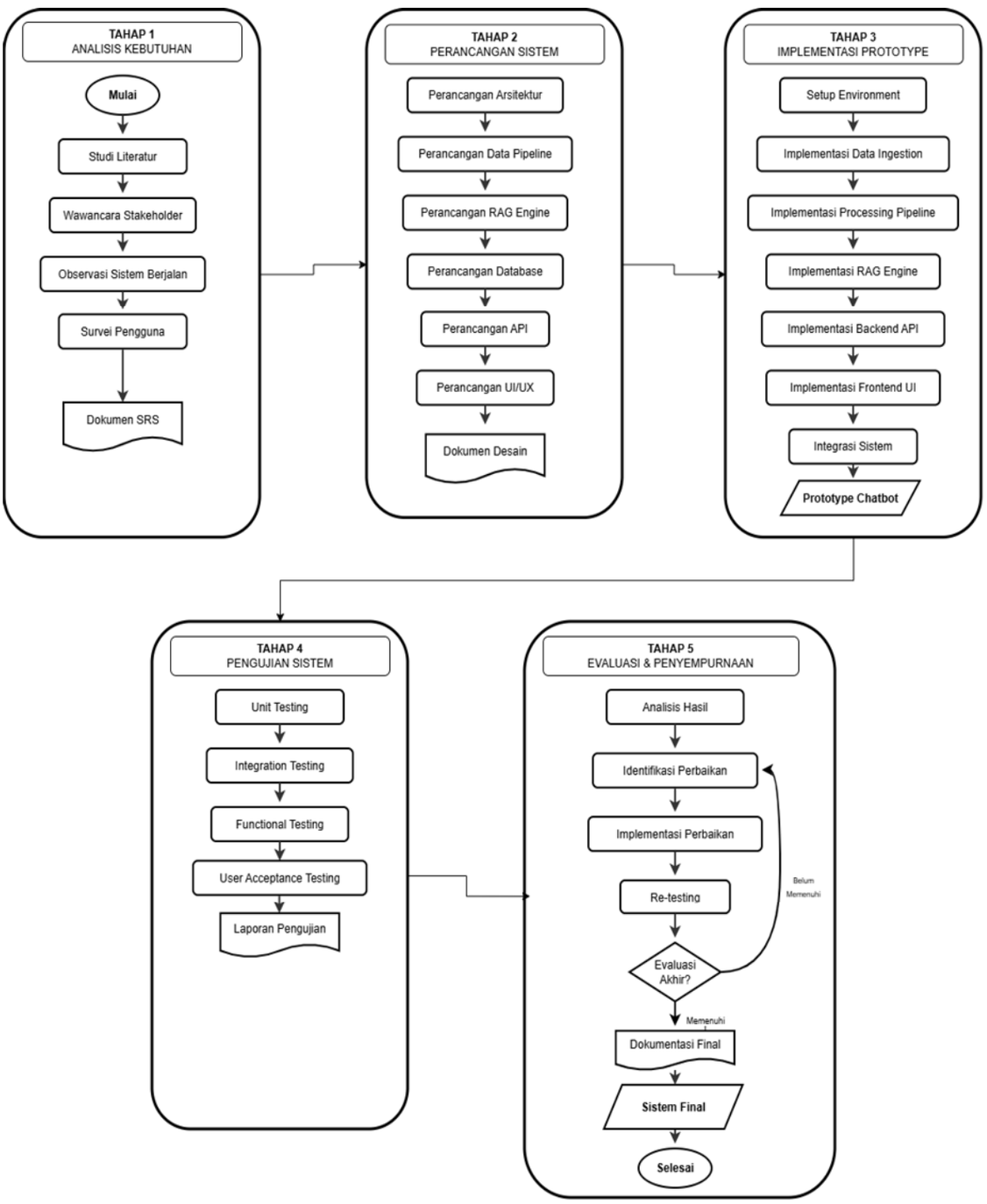
Model Prototyping digunakan karena pengembangan sistem chatbot memerlukan proses iteratif, yaitu pembuatan prototipe awal yang kemudian diuji secara langsung oleh pengguna, dievaluasi, dan disempurnakan secara berulang berdasarkan umpan balik yang diperoleh. Melalui model ini, peneliti dapat menyesuaikan sistem dengan kebutuhan nyata pengguna, terutama terkait pola interaksi, relevansi dokumen yang diambil oleh sistem RAG, serta akurasi dan kejelasan jawaban yang dihasilkan oleh LLM.

Sistem chatbot yang dikembangkan dalam penelitian ini dirancang memiliki dua mode layanan, yaitu mode informasi dan mode transaksional. Mode informasi berfungsi untuk memberikan jawaban atas pertanyaan pengguna mengenai profil YPI Al-Azhar Jakarta. Sementara itu, mode transaksional dirancang untuk mendukung proses layanan administratif, khususnya pendaftaran siswa baru.

Dengan menggunakan pendekatan R&D dan model Prototyping, diharapkan sistem chatbot yang dikembangkan dapat memenuhi kebutuhan pengguna secara optimal, mudah digunakan, serta siap diimplementasikan dalam lingkungan YPI Al-Azhar Jakarta.

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 merupakan bagian dari Tahap Awal pada model Prototyping. Tahap ini dilakukan untuk memastikan bahwa sistem chatbot yang akan dikembangkan mampu menyelesaikan permasalahan pada sistem layanan informasi dan layanan transaksional yang berjalan saat ini, serta selaras dengan tujuan penelitian yang telah ditetapkan.

Pada tahap ini, analisis dilakukan melalui studi literatur, wawancara stakeholder, observasi sistem berjalan, dan survei pengguna, sebagaimana ditunjukkan pada alur Tahap 1 pada diagram penelitian. Hasil dari seluruh aktivitas analisis kebutuhan ini dirangkum dalam bentuk Dokumen *Software Requirement Specification* (SRS) yang menjadi dasar pada tahap perancangan sistem.

Analisis kebutuhan sistem dalam penelitian ini mencakup beberapa aspek utama, yaitu:

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 LLM dengan pendekatan RAG.

3.3.1 Analisis Sistem Berjalan

Analisis sistem berjalan dilakukan sebagai bagian dari Tahap Analisis Kebutuhan untuk memahami kondisi layanan informasi dan layanan transaksional yang saat ini digunakan di YPI Al-Azhar. Sistem layanan online yang tersedia telah berfungsi sebagai media penyampaian informasi dan pengelolaan transaksi pendaftaran bagi calon siswa, orang tua, dan staf administrasi.

Berdasarkan hasil observasi terhadap sistem yang berjalan, diperoleh karakteristik sebagai berikut:

1. Layanan Informasi

Informasi terkait prosedur pendaftaran, fasilitas, persyaratan administrasi, serta layanan akademik telah tersedia melalui website resmi YPI Al-Azhar. Namun, beberapa informasi penting seperti biaya pendidikan belum disajikan secara rinci dan terstruktur. Selain itu, sistem belum menyediakan mekanisme tanya jawab interaktif secara real-time, sehingga pengguna masih harus menghubungi customer service melalui telepon, email, atau kunjungan langsung.

2. Layanan Transaksional

   Proses pendaftaran siswa baru dilakukan melalui formulir online yang bersifat statis. Pengguna diwajibkan mengisi seluruh data pendaftaran sekaligus tanpa adanya panduan interaktif atau bantuan langsung selama proses pengisian berlangsung.

3. Sistem Informasi

   Sistem informasi yang ada telah mengelola data pendaftaran dan layanan pendidikan, namun masih bersifat terpisah dan belum terintegrasi secara menyeluruh antar modul.

Berdasarkan analisis tersebut, ditemukan beberapa kendala utama, yaitu:

1. Waktu respons layanan relatif lambat karena masih bergantung pada customer service manual.
2. Terjadinya inkonsistensi informasi yang disampaikan kepada pengguna.
3. Informasi layanan tersebar di berbagai sumber sehingga sulit diakses secara cepat.
4. Formulir pendaftaran tidak bersifat interaktif dan minim panduan real-time.
5. Sistem belum mampu menjawab pertanyaan pengguna secara otomatis berbasis bahasa alami.
6. Sistem informasi belum terintegrasi secara optimal untuk mendukung monitoring dan manajemen data.

3.3.2 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan untuk mengidentifikasi kebutuhan sistem dari sudut pandang pengguna akhir. Berdasarkan hasil wawancara dan survei pada Tahap Analisis Kebutuhan, pengguna sistem 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 dan informasi memiliki kebutuhan sebagai berikut.

a. Akses terhadap informasi pendaftaran yang cepat, akurat, dan terstruktur.
b. Panduan interaktif step-by-step selama proses pendaftaran.
c. Fitur pelacakan status pendaftaran secara real-time.
d. Akses layanan sistem selama 24 jam (24/7).

2. Staf Administrasi YPI Al-Azhar

Staf administrasi sebagai pengelola sistem memiliki kebutuhan sebagai berikut.

a. Pengurangan beban layanan pertanyaan berulang.
b. Sistem yang membantu meminimalkan kesalahan input data pendaftaran..
c. Dashboard terpadu untuk monitoring status pendaftaran.
d. Kemudahan pembaruan informasi tanpa perubahan kode program.

3.3.3 Analisis Kebutuhan Funsgional

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 di YPI Al-Azhar.

Sistem chatbot harus mampu:

1. Menyediakan layanan informasi terpadu berbasis knowledge base internal.
2. Mengakses dan memproses data internal menggunakan pendekatan RAG.
3. Memandu proses pendaftaran siswa baru secara interaktif.
4. Mengelola pengisian dan validasi formulir pendaftaran.
5. Menerima dan memvalidasi dokumen pendaftaran.
6. Menampilkan status pendaftaran secara real-time.
7. Mendukung pembaruan knowledge base secara adaptif tanpa retraining LLM.

3.3.4 Kebutuhan Non Fungsional

Kebutuhan non-fungsional berkaitan dengan kualitas sistem chatbot yang mendukung keberhasilan implementasi sistem pada Tahap Implementasi Prototype dan Tahap Pengujian Sistem. Kebutuhan non-fungsional meliputi:

1. Keamanan data pengguna.
2. Kinerja dan waktu respons sistem.
3. Skalabilitas sistem.
4. Ketersediaan layanan 24/7.
5. Kemudahan penggunaan.
6. Kemudahan pemeliharaan sistem.

3.3.5 Analisis Kebutuhan Teknis

Analisis kebutuhan teknis dilakukan untuk mendukung Tahap Perancangan Sistem dan Tahap Implementasi Prototype sebagaimana ditunjukkan pada diagram penelitian. Analisis ini mencakup penentuan teknologi, arsitektur, dan infrastruktur sistem.

Kebutuhan teknis sistem meliputi:

1. Large Language Model

   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.

Komponen	Konfigurasi	Justifikasi Berbasis Literatur
Model Chatbot	GPT-4o-mini	GPT-4o-mini mencapai skor 82% pada benchmark MMLU untuk *textual intelligence*, mengungguli Gemini Flash (77,9%) dan Claude Haiku (73,8%) (OpenAI, 2024). Model ini juga menempati peringkat ke-4 pada LMSYS Chatbot Arena dengan skor 1275 dari 131 model (Kili Technology, 2024). Dalam konteks chatbot berbasis RAG,

		GPT-4o-mini dengan *Advanced RAG* menghasilkan respons yang lebih akurat dibandingkan pendekatan *Graph RAG* (arXiv:2512.00991, 2024).
Temperature	0, 0.7	Pada mode informational, pengaturan *temperature* = 0 menghasilkan output yang deterministik dan konsisten, yang penting dalam sistem RAG karena akurasi dan keandalan jawaban menjadi prioritas utama (Databricks, 2023). Pengaturan ini juga mengurangi variabilitas respons dan mendukung *reproducibility* pada lingkungan produksi. Pada mode transaksional, *temperature* = 0,7 memberikan keseimbangan antara kreativitas dan konsistensi, sehingga respons terdengar lebih natural namun tetap mengikuti alur proses pendaftaran yang terstruktur.
Max Tokens	1024, 500	Pada mode informational, batas maksimum 1024 token dinilai cukup untuk menghasilkan jawaban yang informatif dalam konteks chatbot tanya jawab pendidikan, sekaligus mempertimbangkan efisiensi biaya dan waktu respons. Literatur menyebutkan bahwa chatbot interaktif idealnya memberikan respons dalam waktu 3–5 detik untuk menjaga pengalaman pengguna (MDPI Electronics, 2025). Pada mode transaksional, batas 500 token dipilih karena respons umumnya berupa instruksi singkat, konfirmasi, atau

		permintaan data spesifik. Pengaturan ini mengoptimalkan biaya pemrosesan dan mempercepat waktu respons dalam dialog multi-turn pendaftaran.
Model Embeding	text-embedding-3-small	Model ini menunjukkan peningkatan signifikan dibandingkan *text-embedding-ada-002*, dengan skor MIRACL benchmark meningkat dari 31,4% menjadi 44,0% untuk *multilingual retrieval*, serta skor MTEB benchmark meningkat dari 61,0% menjadi 62,3% untuk tugas bahasa Inggris (OpenAI, 2024). Selain itu, model ini memiliki efisiensi biaya yang lebih tinggi, yaitu sekitar 5× lebih murah, dengan harga $0.00002 per 1.000 token.
Dimensi Model Embeding	1536	Dimensi 1536 dipilih karena mampu menghasilkan representasi vektor yang cukup detail untuk menangkap nuansa semantik teks secara efektif. Model embedding ini menerapkan pendekatan Matryoshka Representation Learning, yang memungkinkan *trade-off* antara dimensi vektor dan performa retrieval sesuai kebutuhan sistem (Pinecone, 2024).

2. Retrieval Augmented Generation

   Pendekatan RAG digunakan untuk mengombinasikan 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, pembuatan embedding, serta penyimpanan hasil embedding ke dalam *vector database*.

5. Arsitektur Microservices

   Sistem dirancang menggunakan arsitektur *microservices* yang terdiri atas beberapa modul utama. 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.

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.

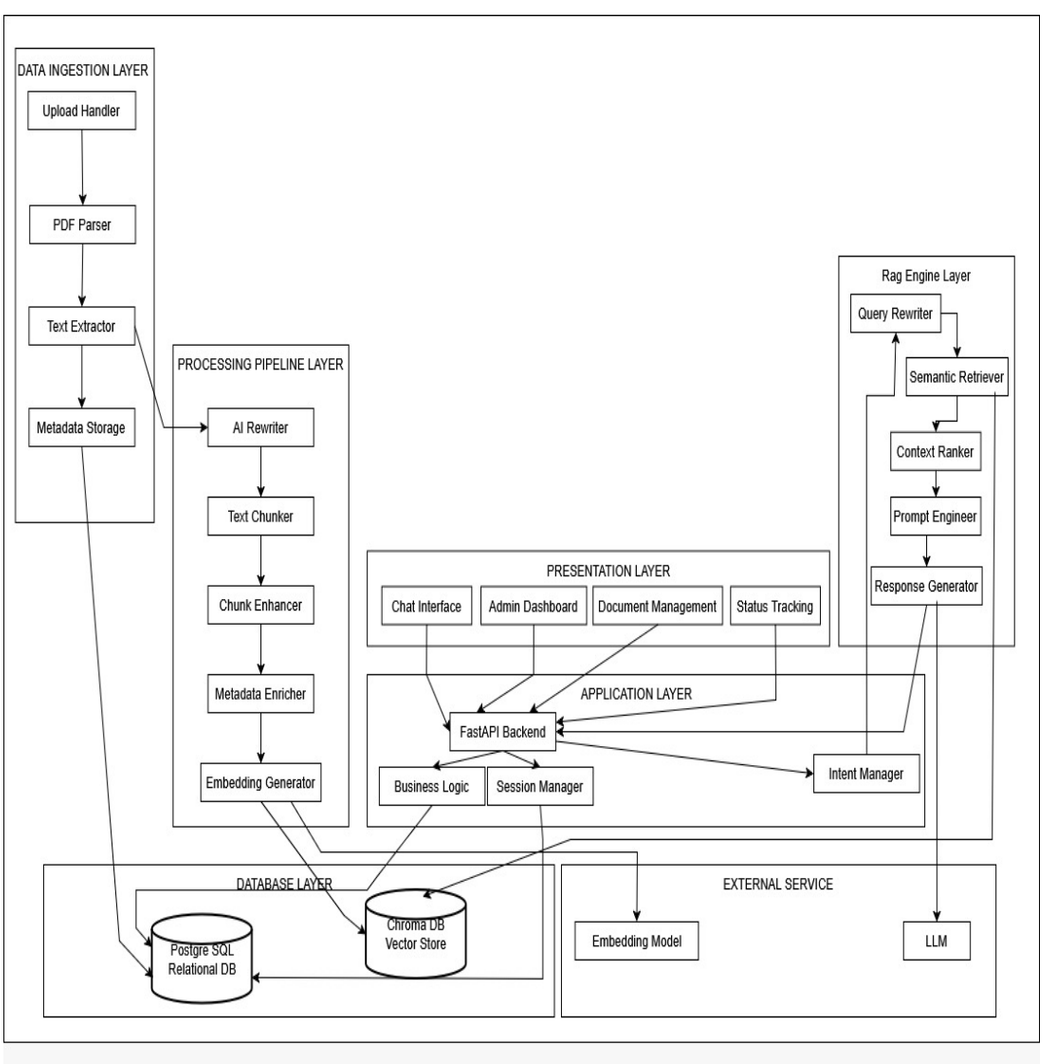
3.4 Perancangan Sistem

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 RAG, alur percakapan chatbot, basis data, antarmuka pemrograman aplikasi, serta antarmuka pengguna. Pada tahap ini, rancangan dibuat sebagai acuan utama dalam proses implementasi prototype.

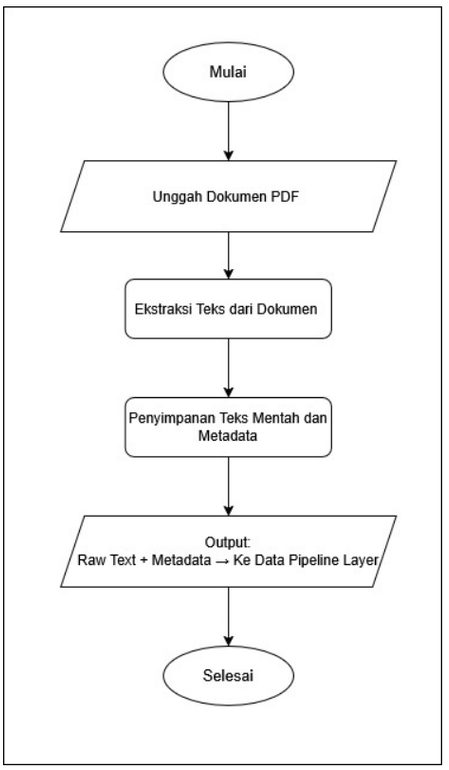
3.4.1 Arsitektur Sistem

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 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 untuk membentuk knowledge base yang lebih terstruktur dan mudah dipahami, pemecahan teks, 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 *query rewriting* berbasis konteks dan metadata, pencarian dokumen menggunakan *semantic retrieval* pada vector database yang diperkaya dengan penyaringan metadata, penyusunan dan perankingan konteks hasil pencarian, serta integrasi dengan model bahasa besar melalui teknik *prompt engineering* untuk menghasilkan jawaban yang relevan dan kontekstual.
Application Layer	FastAPI backend dengan REST API endpoints, business logic untuk intent classification (*informational vs transactional*), session management dan conversation history.
Presentation Layer	Chat interface untuk interaksi user-chatbot, document management interface, dan 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.

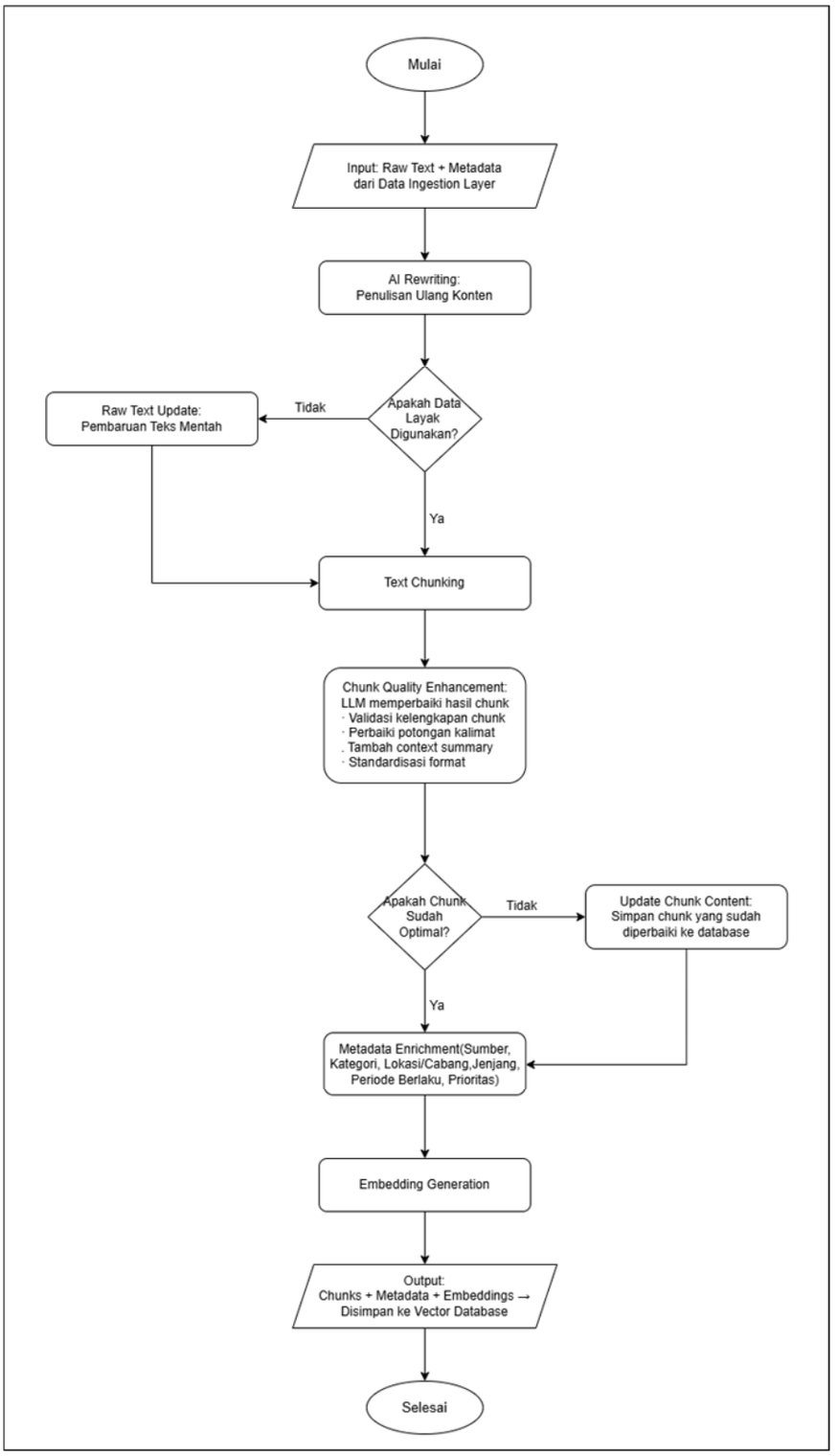
3. Penyimpanan Teks Mentah dan Metadata

   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

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

2. Pembaruan Teks Mentah

   Tahap ini dilakukan apabila hasil penulisan ulang pada tahap sebelumnya dinilai belum memenuhi standar kualitas. Sistem memperbarui teks mentah dalam basis data menggunakan versi hasil perbaikan dan validasi, sehingga menggantikan teks hasil ekstraksi awal dan memastikan konsistensi data sebelum diproses lebih lanjut.

3. Pemecahan Teks

   Pada tahap pemecahan teks, sistem menerapkan strategi *semantic chunking* untuk membagi dokumen berdasarkan perubahan makna dan konteks informasi. Pendekatan ini menghasilkan potongan teks yang tetap utuh secara semantik dan mendukung kualitas representasi basis pengetahuan pada tahap pemrosesan selanjutnya. Penerapan konfigurasi semantic chunking ini menghasilkan potongan teks yang lebih kontekstual dan terstruktur secara makna, serta mendukung kualitas representasi basis pengetahuan pada tahap pemrosesan selanjutnya.

4. Peningkatan Kualitas Chunk

   Setelah proses pemecahan teks, sistem melakukan tahap peningkatan kualitas chunk untuk mengevaluasi dan memastikan bahwa setiap potongan teks yang dihasilkan telah memenuhi standar kualitas yang ditetapkan. Pada tahap ini, *Large Language Model* digunakan untuk melakukan validasi terhadap kelengkapan informasi, koherensi makna, dan konsistensi format setiap chunk.

   Sistem mengidentifikasi chunk yang tidak dapat dipahami secara mandiri, terpotong di tengah kalimat, atau kehilangan konteks penting. Selain itu, sistem mengevaluasi kesesuaian struktur kalimat dan konsistensi istilah agar tidak menimbulkan ambiguitas pada tahap pemrosesan selanjutnya. Setiap chunk juga diperkaya dengan ringkasan konteks singkat untuk memperjelas posisi dan hubungan informasi dalam keseluruhan dokumen.

   Apabila hasil evaluasi menunjukkan bahwa kualitas chunk telah memenuhi standar yang ditetapkan, proses dilanjutkan ke tahap penambahan metadata dokumen. Namun, apabila kualitas chunk dinilai belum optimal, sistem melanjutkan ke tahap pembaruan konten chunk untuk dilakukan perbaikan lebih lanjut.

5. Pembaruan Konten Chunk

   Tahap pembaruan konten chunk hanya dilakukan apabila hasil evaluasi pada tahap peningkatan kualitas chunk menunjukkan bahwa kualitas chunk belum memenuhi standar yang ditetapkan. Pada tahap ini, sistem melakukan perbaikan terhadap chunk yang bermasalah berdasarkan hasil validasi sebelumnya, termasuk penyesuaian batas chunk, penambahan konteks yang diperlukan, serta penyempurnaan struktur dan format teks.

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, 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

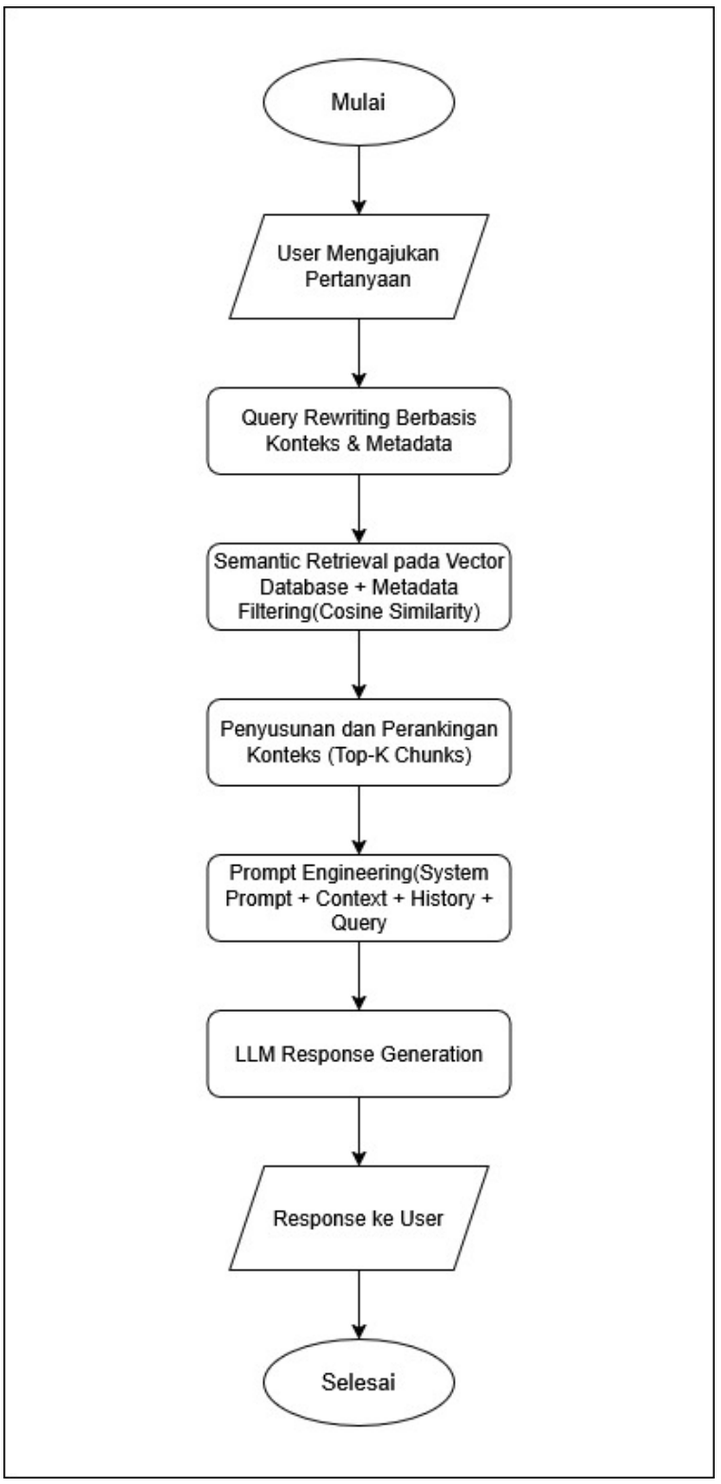
3. Output Infrastruktur Basis Data

   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 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.