

**Implementasi Sistem Tanya Jawab (Question Answering)
Berbasis RAG (Retrieval-Augmented Generation) untuk
Otomatisasi Layanan Informasi di Perusahaan IT dan
Penyedia Pelatihan**

PROYEK AKHIR

Disusun Oleh:
Saidi Nur Pratama
3312311038

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Proyek Akhir yang berjudul “Implementasi Sistem Tanya Jawab (Question Answering) Berbasis RAG (Retrieval-Augmented Generation) untuk Otomatisasi Layanan Informasi di Perusahaan IT dan Penyedia Pelatihan” dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan program studi Teknik Informatika di Politeknik Negeri Batam.

Dalam proses penyusunan laporan ini, penulis mendapatkan banyak bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Allah SWT, yang telah senantiasa memberikan rahmat, nikmat, kesehatan, serta kemudahan kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan baik.
2. Kedua orang tua dan seluruh keluarga penulis, yang selalu memberikan doa, dukungan, serta motivasi selama penulis menempuh pendidikan.
3. Bapak Ir. Bambang Hendrawan, S.T., MSM., CIPMP., CISCIP., selaku Direktur Politeknik Negeri Batam.
4. Bapak Ahmad Hamim Thohari, S.S.T., M.T., selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Batam.
5. Ibu Yeni Rokhayati, S.Si., M.Sc., selaku Ketua Program Studi Teknik Informatika Politeknik Negeri Batam.
6. Ibu Mir'atul Khusna Mufida, S.ST., M.Cs selaku dosen pembimbing Proyek Akhir yang telah memberikan arahan, bimbingan, serta motivasi kepada penulis selama proses penyusunan Tugas Akhir ini.
7. Bapak Yusuf Rizky Nur C. S. Sn. M.A selaku wali dosen yang telah memberikan ilmu, arahan, dan masukan yang bermanfaat selama masa perkuliahan.

8. Seluruh dosen Program Studi Teknik Informatika Politeknik Negeri Batam, yang telah memberikan ilmu pengetahuan dan pengalaman selama masa perkuliahan.
9. Teman-teman seperjuangan yang selalu memberikan bantuan, dukungan, serta semangat selama proses penyusunan laporan proyek akhir ini.
10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu dalam penyusunan laporan proyek akhir ini.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan kedepannya. Akhir kata, penulis berharap laporan ini dapat memberikan manfaat bagi pembaca serta menjadi dasar dalam pengembangan sistem yang diusulkan.

DAFTAR ISI

Contents

KATA PENGANTAR	2
DAFTAR ISI.....	4
DAFTAR TABEL.....	7
DAFTAR GAMBAR	8
DAFTAR LAMPIRAN.....	9
ABSTRAK	10
BAB I PENDAHULUAN	11
1.1. Latar Belakang	11
1.2. Rumusan Masalah	11
1.3. Tujuan	12
1.4. Batasan Masalah	12
1.5. Manfaat	13
1.5.1. Manfaat Praktis	13
1.5.2. Manfaat Teoritis/Ilmiah	14
1.5.3. Manfaat Bagi PT Rawit Solusi Utama	14
1.5.4. Manfaat Bagi Pengguna	14
1.5.5. Manfaat Bagi Peneliti.....	15
1.5.6. Kontribusi terhadap Sustainable Development Goals (SDGs)	15
1.6 Kolaborasi dan Sinergi dalam Pelaksanaan Proyek	15
BAB II TINJAUAN PUSTAKA.....	17
2.1. Penelitian Terkait	17
2.2. Landasan Teori.....	20
2.2.1. Retrieval-Augmented Generation (RAG)	20
2.2.2. Vector Database (ChromaDB).....	20

2.2.3. Large Language Model (Ollama).....	20
2.2.4. LangChain.....	20
2.2.5. Profil Perusahaan IT dan Penyedia Pelatihan (Studi Kasus)	21
2.2.6. Python	21
2.2.7. Flask	21
2.2.8. MySQL.....	22
2.2.9. ChromaDB	22
2.2.10. Arsitektur Model AI.....	22
2.3. Metode Pengembangan Produk	25
BAB III ANALISIS DAN PERANCANGAN	28
3.1. Analisis Kebutuhan	28
3.1.1. Proses Bisnis Berjalan.....	28
3.1.2. Gambaran Umum Sistem	29
3.2. Metodologi Pengembangan AI	31
3.2.1. Pengumpulan Data (<i>Data Collection</i>).....	31
3.2.2. Pra-pemrosesan Data (<i>Preprocessing Data</i>).....	31
3.2.3. Penerapan Metode RAG (<i>Retrieval-Augmented Generation</i>).....	32
3.2.4. Integrasi dan Implementasi	32
3.2.5. Evaluasi Sistem	33
3.3. Perancangan Perangkat Lunak	34
3.3.1. Kebutuhan Fungsional & Non-Fungsional	34
3.3.2. Use Case Diagram.....	35
3.3.3. Skenario <i>Use Case</i>	36
3.3.3.1. Skenario Use Case Mengajukan Pertanyaan.....	36
3.3.3.2. Skenario Use Case Memperbarui Knowledge Base.....	37

3.3.3.3. Skenario Use Case Menyimpan Riwayat Percakapan	38
3.3.4. Activity diagram	39
3.3.5. Database Schema (ERD).....	39
3.3.6. Perancangan Antarmuka (<i>Wireframe</i>).....	40
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	43
4.1. Hasil Implementasi	43
4.1.1. Implementasi Arsitektur Sistem.....	43
4.1.2. Implementasi Halaman Utama Website.....	44
4.1.3. Implementasi Halaman Chatbot.....	45
4.1.4 Implementasi Dashboard Admin.....	46
4.1.5 Implementasi Manajemen Knowledge Base	47
4.1.6 Implementasi User Activity Log.....	48
4.2 Pengujian User Acceptance Testing (UAT).....	48
BAB V KESIMPULAN DAN SARAN.....	50
5.1 Kesimpulan	50
5.2 Saran.....	51
DAFTAR PUSTAKA	53
DAFTAR LAMPIRAN.....	54

DAFTAR TABEL

Tabel 2. 1. Perbandingan Dengan Penelitian Sebelumnya	17
Tabel 2. 2. Arsitektur RAG	25
Tabel 3. 1. Evaluasi Sistem	33
Tabel 3. 2. Use Case Mengajukan Pertanyaan	36
Tabel 3. 3. Use Case Memperbarui Knowledge Base	37
Tabel 3. 4. Use Case Riwayat Percakapan	38
Tabel 4. 1. Komponen Implementasi Sistem	43
Tabel 4. 2. Hasil Pengujian User Acceptance Testing	49

DAFTAR GAMBAR

Gambar 1. Diagram Pipeline RAG	24
Gambar 2. Waterfall Model	26
Gambar 3. Gambaran Umum Arsitektur Assistant Bot Berbasis RAG	30
Gambar 4. <i>UseCase</i>	35
Gambar 5. Activity diagram.....	39
Gambar 6. ER Diagram.....	39
Gambar 7. Wireframe Homepage	40
Gambar 8. Wireframe Percakapan	41
Gambar 9. <i>Wireframe Dashboard</i> Admin.....	41
Gambar 10. <i>Wireframe Upload Knowlage Base</i>	42
Gambar 11. Wireframe User List.....	42
Gambar 12. Halaman Utama Website.....	44
Gambar 13. Halaman Chatbot.....	45
Gambar 14. Dashboard Admin	46
Gambar 15. Manajemen Knowledge Base.....	47
Gambar 16. User Activity Log.....	48

DAFTAR LAMPIRAN

Lampiran A. Link Dokumentasi	54
Lampiran B. Link Github.....	54

ABSTRAK

Di era digital saat ini, kebutuhan terhadap layanan informasi yang cepat dan mudah diakses menjadi salah satu faktor penting dalam meningkatkan kualitas pelayanan perusahaan. PT Rawit Solusi Utama menghadapi kendala dalam penyampaian informasi terkait layanan perusahaan karena informasi masih tersebar pada berbagai sumber dan proses pencarian informasi membutuhkan waktu. Proyek akhir ini bertujuan untuk mengimplementasikan sistem Question Answering berbasis Retrieval-Augmented Generation (RAG) yang dapat membantu mengotomatisasi layanan informasi perusahaan. Sistem dikembangkan menggunakan LangChain sebagai framework orkestrasi, Ollama sebagai penyedia Large Language Model (LLM), Llama 3 sebagai model bahasa, serta ChromaDB sebagai vector database. Data dari dokumen dan sumber informasi perusahaan diproses melalui tahapan pemecahan dokumen menjadi beberapa bagian (chunking), embedding, dan penyimpanan ke dalam vector database. Ketika pengguna memberikan pertanyaan, sistem melakukan retrieval untuk menemukan informasi yang paling relevan dari knowledge base, kemudian informasi tersebut digunakan sebagai konteks bagi LLM untuk menghasilkan jawaban. Pengembangan sistem menggunakan metode Waterfall yang meliputi tahap analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Pengujian dilakukan menggunakan pendekatan black-box melalui User Acceptance Testing (UAT) untuk memastikan fungsi utama sistem berjalan sesuai kebutuhan pengguna. Hasil implementasi menunjukkan bahwa sistem dapat membantu pengguna memperoleh informasi perusahaan melalui antarmuka chatbot dengan memanfaatkan data yang tersedia pada knowledge base. Penerapan RAG juga memungkinkan jawaban chatbot lebih terarah karena proses generasi jawaban menggunakan konteks hasil pencarian dari knowledge base perusahaan.

Kata Kunci: Question Answering, RAG, LLM, LangChain, ChromaDB

BAB I PENDAHULUAN

1.1. Latar Belakang

Di era digital 4.0, kecepatan dan akurasi informasi menjadi kunci daya saing, terutama bagi perusahaan dengan model bisnis ganda seperti Perusahaan IT dan Penyedia Pelatihan. Keterbatasan waktu operasional dan tingginya volume pertanyaan yang berulang (*Frequently Asked Questions*) mengenai dua lini bisnis ini menimbulkan tantangan signifikan dalam efisiensi respons dan potensi *human error* pada pencatatan manual.

Oleh karena itu, diperlukan sebuah sistem cerdas yang tidak hanya memberikan jawaban cepat, tetapi juga akurat berdasarkan data internal perusahaan. Proyek ini mengusulkan solusi melalui Sistem Tanya Jawab (*Question Answering*) berbasis RAG (Retrieval-Augmented Generation). RAG memanfaatkan kemampuan *Large Language Model* (LLM) untuk menyusun jawaban secara alami, tetapi memfilter informasi tersebut agar hanya didasarkan pada dokumen spesifik Perusahaan IT dan Penyedia Pelatihan, sehingga mengatasi masalah halusinasi dan keterbatasan pengetahuan LLM murni.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, berikut ini merupakan rumusan masalah dalam penelitian :

1. Bagaimana merancang arsitektur sistem *Question Answering* berbasis *Retrieval-Augmented Generation* (RAG) yang mampu mengintegrasikan knowledge base dari Bidang Digital dan Bidang Academy dalam satu sistem?
2. Bagaimana mengimplementasikan sistem backend berbasis *Large Language Model* (LLM) dan Vector Database (Chroma) untuk menghasilkan respon informasi jasa dan kursus secara akurat dan *real-time*?

3. Bagaimana mengevaluasi kinerja sistem RAG dalam memberikan jawaban yang relevan dan faktual berdasarkan dokumen internal Perusahaan IT dan Penyedia Pelatihan?

1.3. Tujuan

Sejalan dengan rumusan masalah di atas, tujuan yang ingin dicapai dari proyek akhir ini adalah:

1. Merancang arsitektur sistem *Question Answering* berbasis *Retrieval-Augmented Generation* (RAG) yang mampu mengintegrasikan *knowledge base* dari Bidang Digital dan Bidang Academy dalam satu sistem.
2. Mengimplementasikan sistem backend berbasis *Large Language Model* (LLM) dan *Vector Database* (Chroma) untuk menghasilkan respon informasi jasa dan kursus secara akurat dan *real-time*.
3. Melakukan evaluasi terhadap kinerja sistem RAG dalam memberikan jawaban yang relevan dan faktual berdasarkan dokumen internal Perusahaan IT dan Penyedia Pelatihan.

1.4. Batasan Masalah

Agar penelitian ini dapat lebih terarah dan fokus pada tujuan utama, maka ruang lingkup proyek ini dibatasi pada beberapa hal berikut:

1. Fungsi Utama: Sistem yang dikembangkan hanya berfokus pada layanan Tanya Jawab (*Question Answering*) untuk memberikan informasi (tidak termasuk fungsi transaksi, pendaftaran kursus *online*, atau *booking* jasa).
2. Cakupan Data: *Knowledge Base* yang digunakan terbatas pada dokumen yang memuat informasi bidang Digital (Jasa IT dan Digital Marketing) dan bidang Academy (Informasi Kursus dan Biaya) yang disediakan oleh perusahaan.
3. Teknologi AI: Implementasi berfokus pada arsitektur RAG (*Retrieval-Augmented Generation*) menggunakan *platform* Python dan LangChain, dengan LLM (*Large Language Model*) yang dijalankan melalui Ollama dan *Vector Database* Chroma.

4. Bahasa: Model LLM hanya diuji dan dioptimalkan untuk merespons pertanyaan dan memberikan jawaban dalam Bahasa Indonesia yang formal.
5. Antarmuka: *Output* sistem adalah API (*Application Programming Interface*) yang melayani pertanyaan dari *frontend* (yang dapat berupa *web interface* sederhana atau integrasi *chat*).
6. Pengujian: Pengujian utama (evaluasi) berfokus pada akurasi faktual (Grounding) jawaban yang dihasilkan oleh RAG dibandingkan dengan data sumber, bukan pada performa *User Interface* (UI) atau *load testing* pengguna massal.

1.5. Manfaat

Proyek akhir ini diharapkan dapat memberikan manfaat dari sisi praktis maupun teoritis, di antaranya adalah:

1.5.1. Manfaat Praktis

1. Otomatisasi Layanan: Menyediakan solusi chatbot berbasis LLM yang mampu merespons pertanyaan umum calon klien 24/7, mengurangi beban kerja tim Customer Service Perusahaan IT dan Penyedia Pelatihan.
2. Akurasi Informasi: Menjamin bahwa semua jawaban yang diberikan kepada pengguna adalah faktual dan terfilter (tidak halusinasi), karena terikat pada data internal perusahaan melalui arsitektur RAG.
3. Efisiensi Bisnis: Memudahkan proses diferensiasi informasi antara Perusahaan IT dan Penyedia Pelatihan dalam satu sistem, sehingga meningkatkan kecepatan respon dan kualitas layanan.

1.5.2. Manfaat Teoritis/Ilmiah

1. Studi Kasus RAG: Menjadi referensi dan studi kasus nyata dalam penerapan arsitektur RAG untuk mengatasi tantangan *halusinasi* LLM pada konteks bisnis jasa dan pendidikan di Indonesia.
2. Kompetensi LLM: Meningkatkan pemahaman dan kompetensi dalam merancang dan mengimplementasikan sistem AI end-to-end dengan integrasi antara Large Language Model (Ollama), Vector Embedding (HuggingFace), dan Vector Database (Chroma).
3. Dasar Pengembangan Lanjut: Dapat menjadi dasar untuk pengembangan sistem yang lebih kompleks di masa depan, seperti penambahan fitur multi-turn dialogue atau integrasi langsung dengan sistem CRM perusahaan.

1.5.3. Manfaat Bagi PT Rawit Solusi Utama

Sistem Question Answering berbasis Retrieval-Augmented Generation (RAG) yang dikembangkan mampu membantu PT Rawit Solusi Utama dalam menyediakan layanan informasi perusahaan secara lebih cepat, akurat, dan konsisten kepada pengguna. Informasi yang sebelumnya harus dijawab secara manual oleh staf kini dapat diberikan secara otomatis melalui chatbot sehingga meningkatkan efisiensi pelayanan serta mengurangi beban operasional perusahaan.

1.5.4. Manfaat Bagi Pengguna

Pengguna memperoleh kemudahan dalam mengakses informasi mengenai layanan Rawit Digital maupun Rawit Academy kapan saja melalui website tanpa harus menunggu respons dari admin. Jawaban yang diberikan bersumber dari knowledge base perusahaan sehingga lebih relevan dan sesuai dengan informasi resmi.

1.5.5. Manfaat Bagi Peneliti

Penelitian ini menjadi sarana penerapan ilmu yang diperoleh selama perkuliahan, khususnya pada bidang Artificial Intelligence, Natural Language Processing, Information Retrieval, serta pengembangan aplikasi berbasis Large Language Model (LLM). Selain itu, penelitian ini diharapkan menjadi referensi bagi penelitian selanjutnya yang mengembangkan sistem Question Answering berbasis Retrieval-Augmented Generation.

1.5.6. Kontribusi terhadap Sustainable Development Goals (SDGs)

Proyek akhir ini berkontribusi terhadap pencapaian Sustainable Development Goals (SDGs), khususnya SDG 9: Industry, Innovation and Infrastructure melalui penerapan teknologi kecerdasan buatan untuk meningkatkan efisiensi layanan informasi perusahaan. Implementasi sistem Question Answering berbasis Retrieval-Augmented Generation (RAG) mendukung pemanfaatan teknologi digital dalam meningkatkan kualitas layanan, otomatisasi proses penyampaian informasi, serta pengembangan infrastruktur teknologi yang inovatif dan berkelanjutan.

1.6 Kolaborasi dan Sinergi dalam Pelaksanaan Proyek

Pelaksanaan proyek ini melibatkan kolaborasi antara mahasiswa sebagai pengembang sistem dengan PT Rawit Solusi Utama sebagai mitra pengguna (stakeholder). PT Rawit Solusi Utama berperan sebagai penyedia kebutuhan sistem, sumber data, serta pemberi umpan balik selama proses pengembangan.

Kolaborasi dilakukan melalui beberapa tahapan, yaitu identifikasi kebutuhan pengguna, pengumpulan data layanan perusahaan, implementasi sistem, pengujian sistem, serta evaluasi hasil implementasi. Sinergi antara pengembang dan mitra perusahaan bertujuan agar sistem yang dibangun mampu menjawab kebutuhan nyata di lingkungan perusahaan.

Selain itu, proyek ini juga melatih kemampuan kerja sama tim, komunikasi, manajemen proyek, dan penyelesaian masalah secara kolaboratif yang merupakan bagian dari capaian pembelajaran lulusan Program Studi Teknik Informatika.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Berikut adalah beberapa kajian atau referensi tinjauan pustaka yang dilihat pada tabel 2.1.

Tabel 2. 1. Terkait Dengan Penelitian Sebelumnya

Peneliti/ Tahun	Judul/Topik Penelitian	Metode/Tekno logi	Kelebihan	Kekurangan /Celah
Lewis et al. (2020)	Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks	RAG, Dense Retrieval, LLM	Memperkenalkan konsep RAG yang menggabungkan retrieval dan generation untuk meningkatkan kualitas jawaban.	Fokus pada benchmark NLP umum dan belum diterapkan pada layanan informasi perusahaan.
Borgeau d et al. (2022)	Improving Language Models by Retrieving from Trillions of Tokens	Retrieval-based Language Model	Menunjukkan bahwa proses retrieval dapat meningkatkan kemampuan model dalam menjawab pertanyaan berbasis pengetahuan.	Membutuhkan sumber daya komputasi yang besar dan tidak membahas implementasi sistem layanan informasi.
Gao et al. (2024)	Retrieval-Augmented Generation for	Survey RAG, LLM	Memberikan tinjauan komprehensif	Bersifat konseptual dan tidak

	Large Language Models: A Survey		mengenai arsitektur, metode, dan tantangan RAG.	membahas implementasi sistem secara spesifik.
Huang & Huang (2024)	A Survey on Retrieval-Augmented Text Generation for Large Language Models	RAG, LLM	Menjelaskan berbagai pendekatan retrieval untuk meningkatkan kualitas generasi teks.	Tidak fokus pada penerapan RAG pada domain layanan informasi perusahaan.
Xu et al. (2024)	CRP-RAG: A Retrieval-Augmented Generation Framework for Logical Reasoning	RAG, Reasoning Framework	Meningkatkan kemampuan reasoning dalam proses retrieval dan generation.	Arsitektur relatif kompleks dan lebih berfokus pada logical reasoning dibanding layanan informasi.
Zhang et al. (2024)	Effectiveness of Retrieval-Augmented Generation-based LLM for Information Generation	RAG, LLM	Menunjukkan peningkatan akurasi jawaban dan penurunan hallucination dibanding LLM biasa.	Belum membahas integrasi dengan basis pengetahuan perusahaan secara lokal.

Zhang et al. (2024)	Performance Comparison of RAG and Fine-Tuned LLM	RAG vs Fine-Tuned LLM	Membuktikan bahwa RAG mampu menghasilkan jawaban yang lebih relevan dibanding model tanpa retrieval.	Fokus pada evaluasi model dan belum diterapkan pada sistem layanan informasi nyata.
Proyek Akhir Ini (2026)	Implementasi Sistem Tanya Jawab (Question Answering) Berbasis RAG untuk Otomatisasi Layanan Informasi di Perusahaan IT dan Penyedia Pelatihan	RAG, LangChain, Ollama, ChromaDB	Menghasilkan sistem Question Answering berbasis data internal yang akurat, kontekstual, dan dapat dijalankan secara lokal untuk menjaga keamanan data perusahaan.	Terbatas pada domain layanan informasi perusahaan dan sumber data yang tersedia pada Knowledge Base.

2.2. Landasan Teori

2.2.1. Retrieval-Augmented Generation (RAG)

RAG adalah arsitektur yang mengintegrasikan kemampuan pengambilan informasi dari sumber data eksternal dengan model bahasa besar (LLM). Tidak seperti LLM standar yang hanya mengandalkan data pelatihan statis, RAG memungkinkan model untuk mengakses data terbaru dan spesifik milik organisasi tanpa perlu melakukan proses retraining model.

2.2.2. Vector Database (ChromaDB)

Berbeda dengan database relasional (MySQL), *Vector Database* menyimpan data dalam bentuk vektor multidimensi. Dalam proyek ini, ChromaDB digunakan untuk menyimpan *embeddings* dari dokumen perusahaan, yang memungkinkan pencarian berdasarkan kesamaan makna (*semantic similarity*) melalui algoritma seperti *Cosine Similarity*.

2.2.3. Large Language Model (Ollama)

Ollama adalah alat yang memungkinkan pengoperasian LLM secara lokal (seperti Llama 3 atau Qwen). Penggunaan LLM lokal sangat krusial dalam layanan informasi Perusahaan IT dan Penyedia Pelatihan untuk menjaga kerahasiaan data internal perusahaan agar tidak dikirim ke server pihak ketiga (cloud).

2.2.4. LangChain

LangChain adalah *framework* yang digunakan untuk menghubungkan berbagai komponen AI. Dalam Assistant Bot, LangChain bertugas mengatur alur (*pipeline*) mulai dari pemuatan dokumen, pemotongan teks (*chunking*), hingga interaksi antara *retriever* dan LLM.

2.2.5. Profil Perusahaan IT dan Penyedia Pelatihan (Studi Kasus)

Perusahaan IT dan Penyedia Pelatihan merupakan penyedia solusi teknologi/layanan informasi. Saat ini, pemberian layanan informasi kepada klien masih dilakukan secara manual melalui staf admin. Kondisi ini menimbulkan hambatan berupa keterlambatan respon saat beban kerja meningkat atau di luar jam operasional. Assistant Bot hadir untuk mengotomatisasi alur informasi ini menggunakan basis data pengetahuan perusahaan yang sudah ada.

2.2.6. Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat *interpreted* dan memiliki filosofi desain yang berfokus pada keterbacaan kode. Dalam pengembangan *artificial intelligence* dan sistem RAG, Python menjadi standar industri karena memiliki ekosistem pustaka (*library*) yang sangat luas, seperti Pandas untuk manipulasi data dan LangChain untuk orkestrasi model bahasa. Pada proyek Assistant Bot, Python berfungsi sebagai perekat utama yang menghubungkan antara *vector database*, LLM, dan antarmuka API.

2.2.7. Flask

Flask adalah *micro-framework* berbasis Python yang digunakan untuk membangun aplikasi web. Berbeda dengan *framework* besar lainnya, Flask bersifat ringan dan memberikan fleksibilitas tinggi bagi pengembang untuk memilih *tool* atau *library* tambahan. Dalam sistem ini, Flask digunakan untuk membangun *backend* API yang menyediakan beberapa *endpoint* kritikal:

- POST /ask: Untuk memproses pertanyaan pengguna secara *real-time*.
- POST /update: Untuk memperbarui basis pengetahuan (*knowledge base*) baik melalui unggahan file maupun *web scraping*.
- GET /health: Untuk memantau status kesiapan sistem AI dan koneksi basis data.

2.2.8. MySQL

MySQL digunakan sebagai basis data relasional untuk menyimpan data operasional dan metadata sistem. Tabel yang digunakan meliputi:

- **users**, untuk menyimpan data pengguna sistem.
- **chat_histories**, untuk menyimpan riwayat percakapan antara pengguna dan chatbot.
- **kb_history**, untuk menyimpan informasi dokumen yang telah diunggah ke *Knowledge Base*.

Basis data MySQL berfungsi sebagai penyimpanan data transaksi dan pengelolaan informasi sistem.

2.2.9. ChromaDB

ChromaDB digunakan sebagai *vector database* yang mendukung proses RAG. Setelah dokumen diproses dan diubah menjadi *embedding*, data tersebut disimpan dalam bentuk vektor pada ChromaDB.

Fungsi utama ChromaDB meliputi:

- Menyimpan hasil *embedding* dokumen.
- Melakukan *vector similarity search* untuk menemukan informasi yang relevan.
- Mendukung proses *retrieval* sebelum jawaban dihasilkan oleh *Large Language Model* (LLM).

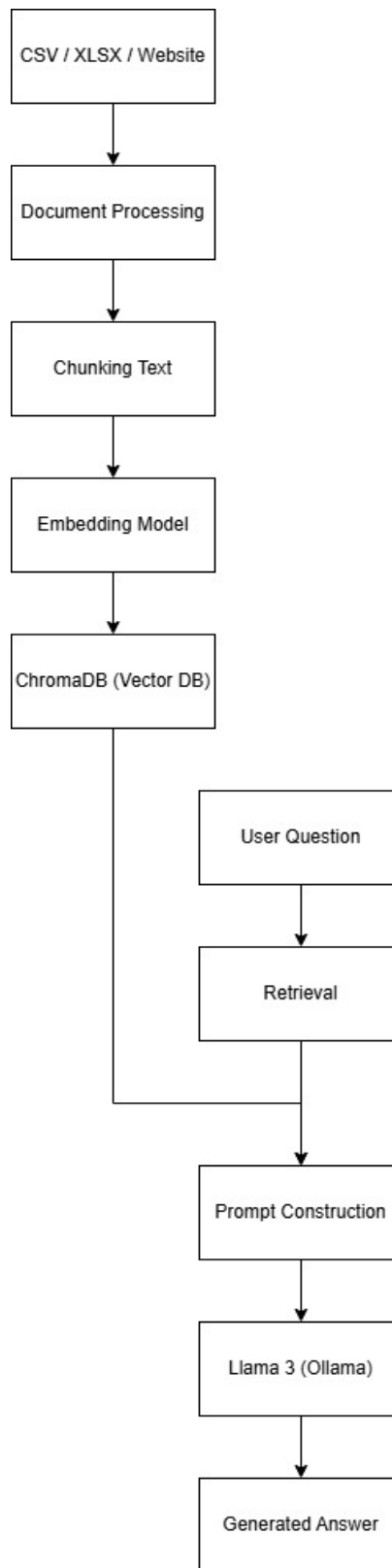
Dengan kombinasi MySQL dan ChromaDB, sistem dapat mengelola data operasional sekaligus menyediakan pencarian informasi berbasis semantik secara efisien.

2.2.10. Arsitektur Model AI

Sistem tanya jawab yang dikembangkan menggunakan pendekatan Retrieval-Augmented Generation (RAG). Arsitektur ini menggabungkan kemampuan pencarian informasi dari basis pengetahuan dengan kemampuan generatif dari Large Language Model (LLM) sehingga jawaban yang dihasilkan lebih relevan dan sesuai dengan data perusahaan.

Alur kerja sistem dimulai dari proses pengambilan data dari dokumen CSV, XLSX, maupun sumber Website yang telah ditentukan. Selanjutnya dokumen diproses menggunakan teknik chunking untuk membagi teks menjadi beberapa bagian yang lebih kecil agar mudah diolah. Setiap potongan teks kemudian dikonversi menjadi embedding vector menggunakan model embedding dan disimpan ke dalam ChromaDB sebagai basis data vektor.

Ketika pengguna mengajukan pertanyaan, sistem melakukan proses retrieval untuk mencari potongan dokumen yang paling relevan berdasarkan kemiripan vektor. Informasi yang ditemukan kemudian digabungkan dengan pertanyaan pengguna dalam proses prompt construction dan dikirim ke model Llama 3 yang dijalankan secara lokal menggunakan Ollama. Model akan menghasilkan jawaban berdasarkan konteks yang diberikan sehingga informasi yang disampaikan tetap mengacu pada data yang tersedia di knowledge base.



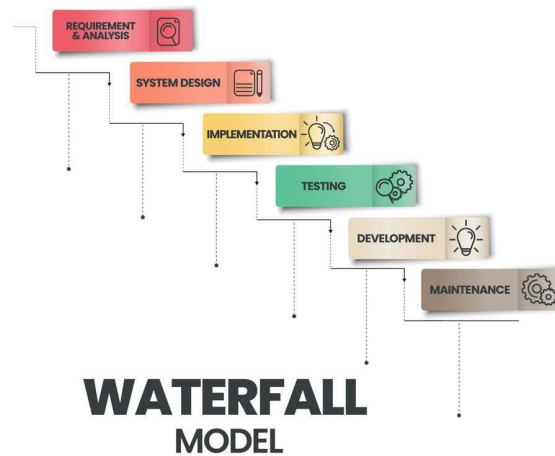
Gambar 1. Diagram Pipeline RAG

Tabel 2. 2. Arsitektur RAG

Komponen	Fungsi
Document Source	Sumber data berupa file CSV, XLSX, atau Website
Chunking	Membagi dokumen menjadi beberapa bagian kecil
Embedding	Mengubah teks menjadi representasi vektor
ChromaDB	Menyimpan embedding dan melakukan pencarian kemiripan
Retrieval	Mengambil dokumen yang paling relevan terhadap pertanyaan
Prompt Construction	Menggabungkan konteks hasil retrieval dengan pertanyaan pengguna
Llama 3 (Ollama)	Menghasilkan jawaban berdasarkan konteks yang diberikan
Response Generation	Menampilkan jawaban kepada pengguna

2.3. Metode Pengembangan Produk

Metode pengembangan perangkat lunak yang digunakan dalam pengembangan *Assistant Bot* adalah metode *Waterfall*. Menurut Bassil (2012), metode ini merupakan model sekuensial yang mengharuskan setiap tahapan diselesaikan sepenuhnya sebelum beralih ke tahapan berikutnya. Metode ini dipilih agar proses pengembangan sistem RAG yang kompleks, mulai dari integrasi LLM hingga basis data vektor, dapat terdokumentasi dengan terstruktur dan sistematis.



Gambar 2. Waterfall Model

1. *Requirement Analysis* (Analisis Kebutuhan)

Menurut Bassil (2012), tahap ini bertujuan untuk mengidentifikasi dan mendokumentasikan kebutuhan pengguna serta spesifikasi sistem secara rinci.

- Rencana Kegiatan: Melakukan observasi dan wawancara dengan staf operasional di Perusahaan IT dan Penyedia Pelatihan untuk mengidentifikasi data informasi layanan apa saja yang sering ditanyakan. Selain itu, dilakukan pengumpulan data internal berupa dokumen perusahaan (format .csv, .xlsx) dan daftar URL website yang akan dijadikan sebagai sumber pengetahuan (knowledge base) bagi bot.

2. *System Design* (Perancangan Sistem)

Tahap ini melibatkan perancangan arsitektur perangkat lunak, antarmuka pengguna, dan struktur data berdasarkan analisis kebutuhan sebelumnya.

- Rencana Kegiatan: Merancang arsitektur *Retrieval-Augmented Generation* (RAG) yang meliputi alur data ingestion, konfigurasi vector store menggunakan ChromaDB, dan skema database relasional MySQL untuk mencatat riwayat dokumen (kb_history). Peneliti juga merancang prompt

template yang akan digunakan untuk memandu LLM dalam menghasilkan jawaban.

3. *Implementation* (Implementasi)

Tahap implementasi adalah proses menerjemahkan desain ke dalam kode program yang dapat dieksekusi oleh mesin.

- Rencana Kegiatan: Melakukan pengkodean menggunakan bahasa pemrograman Python dengan *framework* Flask. Peneliti akan mengintegrasikan LangChain untuk manajemen alur RAG, HuggingFaceEmbeddings untuk pemrosesan teks, dan Ollama sebagai server lokal untuk menjalankan model bahasa besar (LLM) seperti Llama 3 atau Qwen.

4. *Integration & Testing* (Integrasi dan Pengujian)

Setelah unit-unit program selesai dibangun, sistem diintegrasikan dan diuji untuk memastikan tidak ada kesalahan fungsi (bug) dan sistem memenuhi kebutuhan awal.

- Rencana Kegiatan: Melakukan pengujian fungsional menggunakan metode Black Box Testing pada setiap endpoint API (/ask, /update). Selain itu, dilakukan pengujian akurasi jawaban bot untuk memastikan informasi yang diberikan sesuai dengan konteks dokumen yang diunggah dan tidak terjadi "halusinasi" pada AI.

5. *Operation & Maintenance* (Operasi dan Pemeliharaan)

Tahap ini merupakan fase di mana perangkat lunak dioperasikan di lingkungan nyata dan dilakukan perbaikan jika ditemukan kendala setelah implementasi.

- Rencana Kegiatan: Mengimplementasikan *Assistant Bot* pada lingkungan operasional Perusahaan IT dan Penyedia Pelatihan. Pemeliharaan dilakukan dengan memantau kinerja vector database dan melakukan pembaharuan data jika terdapat perubahan informasi layanan atau kebijakan perusahaan di masa mendatang.

BAB III ANALISIS DAN PERANCANGAN

Bagian ini menjelaskan tahapan-tahapan yang dilakukan dalam pengembangan Assistant Bot, mulai dari proses pengumpulan data hingga evaluasi model. Metodologi ini disusun untuk memastikan sistem RAG (*Retrieval-Augmented Generation*) yang dibangun memiliki akurasi yang baik dan sesuai dengan kebutuhan operasional di Perusahaan IT dan Penyedia Pelatihan.

3.1. Analisis Kebutuhan

Bagian ini menjelaskan identifikasi kebutuhan sistem yang didasarkan pada analisis terhadap alur pemberian layanan informasi yang saat ini berlaku di Perusahaan IT dan Penyedia Pelatihan, serta solusi yang diusulkan melalui pengembangan Assistant Bot.

3.1.1. Proses Bisnis Berjalan

Saat ini, proses pemberian informasi mengenai layanan perusahaan di Perusahaan IT dan Penyedia Pelatihan masih dilakukan secara manual dan terpusat pada staf administrasi atau *customer service*. Alur kerja yang berjalan adalah sebagai berikut:

1. **Permintaan Informasi:** Calon klien atau mitra mengirimkan pertanyaan mengenai layanan melalui media komunikasi (WhatsApp, Email, atau telepon).
2. **Pencarian Data:** Staf admin menerima pertanyaan dan mencari informasi yang relevan di dalam dokumen internal seperti katalog layanan, daftar harga, atau dokumen prosedur (SOP) yang tersimpan dalam komputer kantor.
3. **Pemberian Jawaban:** Staf menyusun jawaban secara manual dan mengirimkannya kembali kepada penanya.

Permasalahan yang ditemukan:

- **Keterbatasan Waktu:** Respon hanya dapat diberikan pada jam kerja operasional. Pertanyaan yang masuk di malam hari atau hari libur akan mengalami keterlambatan respon (*delayed response*).

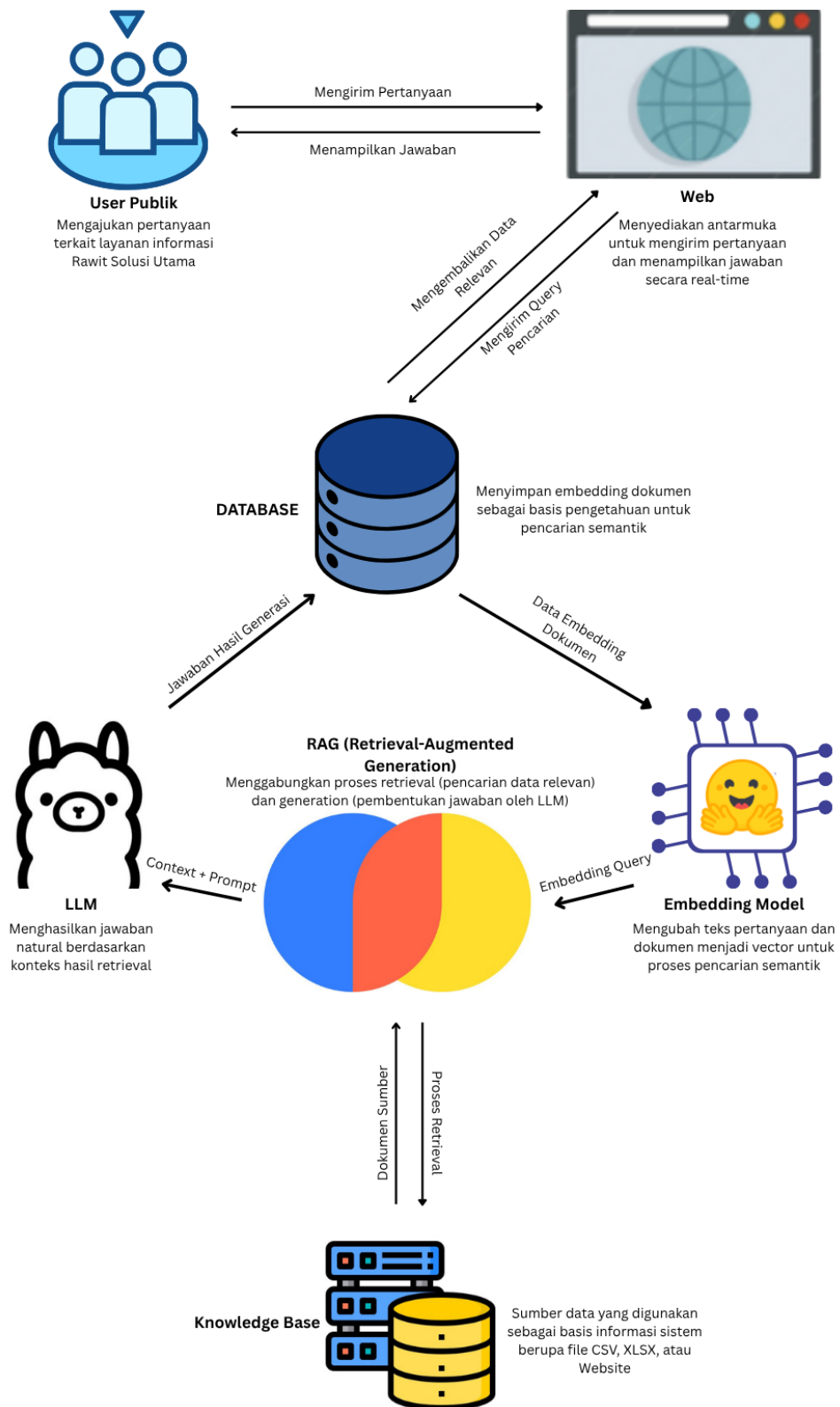
- Risiko Human Error: Penanganan dokumen manual yang banyak meningkatkan risiko kesalahan pemberian informasi atau ketidaksesuaian jawaban antar staf.
- Skalabilitas: Jika jumlah pertanyaan meningkat secara drastis dalam waktu bersamaan, staf admin akan kesulitan memberikan respon yang cepat dan akurat.

3.1.2. Gambaran Umum Sistem

Sebagai solusi atas permasalahan tersebut, diusulkan sebuah sistem tanya jawab otomatis bernama Assistant Bot yang mengimplementasikan metode *Retrieval-Augmented Generation* (RAG). Sistem ini dirancang untuk bekerja secara mandiri dengan cara "membaca" dan memahami dokumen internal perusahaan secara otomatis.

Gambaran umum arsitektur sistem yang diusulkan adalah sebagai berikut:

- Input Pengetahuan: Admin mengunggah dokumen (.csv, .xlsx) atau memasukkan URL website ke dalam sistem.
- Proses RAG: Sistem mengubah dokumen tersebut menjadi vector embeddings yang disimpan dalam Vector Database (ChromaDB).
- Interaksi Pengguna: Ketika pengguna bertanya, sistem tidak menjawab secara umum, melainkan melakukan pencarian informasi yang paling relevan dari database dokumen internal terlebih dahulu.
- Output: Jawaban dibangkitkan oleh model bahasa lokal (Ollama/Llama 3) sehingga jawaban yang dihasilkan bersifat natural, akurat, dan sesuai dengan data asli Perusahaan IT dan Penyedia Pelatihan.



Gambar 3. Gambaran Umum Arsitektur Assistant Bot Berbasis RAG

3.2. Metodologi Pengembangan AI

Sesuai dengan karakteristik proyek *Artificial Intelligence*, tahapan pengembangan difokuskan pada pengolahan data pengetahuan dan penerapan model bahasa. Berikut adalah rincian tahapannya:

3.2.1. Pengumpulan Data (*Data Collection*)

Pada tahap ini, dilakukan pengumpulan data yang akan menjadi basis pengetahuan bagi Assistant Bot. Data diperoleh melalui dua metode:

- **Data Dokumen:** Mengumpulkan dokumen internal Perusahaan IT dan Penyedia Pelatihan dalam format .csv dan .xlsx. Data ini mencakup daftar layanan teknis, FAQ, dan tabel harga yang bersifat terstruktur. Pengambilan data ini dilakukan melalui fungsi `load_documents_from_file` pada sistem.
- **Data Web (Web Scraping):** Mengambil informasi dinamis dari website resmi perusahaan. Menggunakan pustaka `WebBaseLoader` dari `LangChain`, sistem secara otomatis mengekstraksi teks dari daftar URL yang dikonfigurasi dalam `Config.SOURCE_URLS`. Hal ini memastikan bot tetap memiliki informasi terbaru yang dipublikasikan di web.

3.2.2. Pra-pemrosesan Data (*Preprocessing Data*)

Data mentah yang telah dikumpulkan harus diproses agar dapat dikelola secara optimal oleh model AI. Langkah-langkah yang dilakukan adalah:

- **Konversi Teks:** Mengubah data dari format `DataFrame` (CSV/Excel) atau HTML (Web) menjadi teks mentah (string).
- **Pemecahan Teks (Text Chunking):** Menggunakan `RecursiveCharacterTextSplitter` dengan parameter `Chunk Size 500` karakter. Pemilihan ukuran ini bertujuan agar informasi tidak terlalu panjang (menghindari limit token LLM) namun tetap mengandung makna yang utuh.
- **Pengaturan Overlap:** Ditetapkan `Chunk Overlap` sebesar 50 karakter. Penumpukan antar potongan teks ini berfungsi untuk menjaga

kesinambungan konteks informasi, sehingga tidak ada informasi yang terpotong secara paksa di tengah kalimat.

3.2.3. Penerapan Metode RAG (*Retrieval-Augmented Generation*)

Metode RAG diimplementasikan untuk memberikan jawaban yang akurat berdasarkan data internal perusahaan melalui dua alur utama:

- **Vektorisasi (Embedding):** Setiap potongan teks diubah menjadi representasi numerik menggunakan model HuggingFaceEmbeddings. Vektor-vektor ini kemudian disimpan dan diindeks ke dalam ChromaDB sebagai Vector Database.
- **Proses Retrieval:** Saat pengguna mengajukan pertanyaan, sistem akan mengubah pertanyaan tersebut menjadi vektor dan mencari potongan teks yang paling relevan di ChromaDB menggunakan algoritma kesamaan semantik.
- **Pembangkitan Jawaban (Generation):** Potongan teks relevan yang ditemukan dijadikan sebagai konteks bagi model Llama 3 (via Ollama) untuk menghasilkan jawaban yang natural namun tetap terbatas pada data resmi perusahaan.

3.2.4. Integrasi dan Implementasi

Tahap akhir adalah mengintegrasikan komponen AI ke dalam aplikasi layanan informasi:

- **Flask API:** Bertugas sebagai orkestrator yang menghubungkan antarmuka pengguna dengan mesin AI. Flask menangani permintaan real-time melalui endpoint /ask untuk tanya jawab dan /update untuk pembaruan basis data.
- **Manajemen Riwayat (MySQL):** Melalui SQLAlchemy, sistem mencatat setiap aktivitas pembaruan pengetahuan ke dalam tabel kb_history. Integrasi ini memungkinkan admin memantau kapan data terakhir diperbarui dan berapa banyak informasi yang telah diproses oleh sistem.

3.2.5. Evaluasi Sistem

Evaluasi sistem dilakukan untuk mengukur kemampuan sistem *Question Answering* berbasis *Retrieval-Augmented Generation* (RAG) dalam memberikan jawaban yang relevan, faktual, dan sesuai dengan sumber informasi yang tersedia pada *Knowledge Base*. Tahap evaluasi dilakukan melalui beberapa aspek pengujian sebagai berikut.

- **Pengujian Grounding**

Pengujian *grounding* bertujuan memastikan bahwa jawaban yang dihasilkan sistem berasal dari informasi yang terdapat pada *Knowledge Base* serta meminimalkan terjadinya *hallucination*.

- **Pengujian Relevansi Jawaban**

Pengujian relevansi dilakukan untuk mengukur tingkat kesesuaian jawaban sistem terhadap pertanyaan yang diberikan oleh pengguna.

- **Pengujian Akurasi Faktual**

Pengujian akurasi faktual bertujuan memastikan bahwa informasi yang dihasilkan sistem sesuai dengan fakta dan data yang terdapat pada dokumen sumber.

- **Pengujian Endpoint API**

Pengujian *Application Programming Interface* (API) dilakukan untuk memastikan setiap endpoint dapat berfungsi dengan baik, menghasilkan respons yang sesuai, serta mampu menangani kesalahan masukan dengan benar.

Tabel 3. 1. Evaluasi Sistem

Aspek Pengujian	Tujuan	Metode
Grounding	Memastikan jawaban berasal dari sumber	Perbandingan dengan dokumen KB
Relevansi Jawaban	Mengukur kesesuaian jawaban dengan pertanyaan	Evaluasi manual

Akurasi Faktual	Memastikan informasi benar dan konsisten	Verifikasi terhadap dokumen sumber
Endpoint API	Memastikan API berjalan normal	Pengujian menggunakan Postman

3.3. Perancangan Perangkat Lunak

Tahap perancangan ini bertujuan untuk mendefinisikan arsitektur aplikasi, aliran data, dan antarmuka pengguna yang mengintegrasikan mesin AI ke dalam sistem layanan informasi Perusahaan IT dan Penyedia Pelatihan.

3.3.1. Kebutuhan Fungsional & Non-Fungsional

A. Kebutuhan Fungsional (Functional Requirement)

Kebutuhan fungsional mendefinisikan layanan atau fitur spesifik yang harus disediakan oleh sistem:

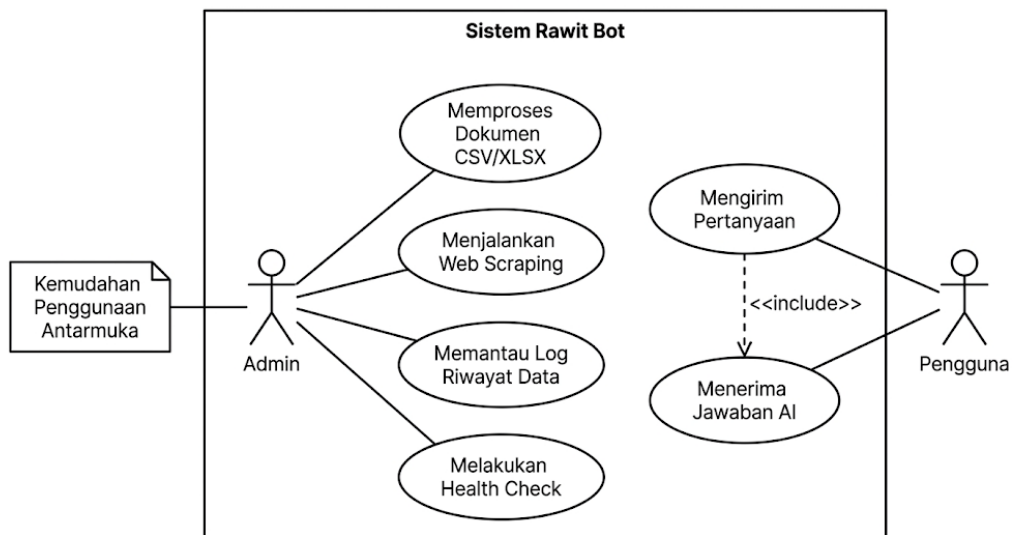
- FR-01 (Manajemen Pengetahuan): Sistem harus mampu mengunggah dan memproses dokumen dalam format .csv dan .xlsx untuk dijadikan basis data pengetahuan.
- FR-02 (Web Scraping): Sistem harus mampu menarik informasi secara otomatis dari URL website yang telah dikonfigurasi.
- FR-03 (Pencarian Semantik): Sistem harus mampu melakukan pencarian informasi yang relevan dari Vector Database berdasarkan pertanyaan pengguna.
- FR-04 (Tanya Jawab AI): Sistem harus mampu menghasilkan jawaban natural dalam Bahasa Indonesia berdasarkan konteks dokumen internal perusahaan.
- FR-05 (Logging): Sistem harus mencatat riwayat pembaruan basis pengetahuan ke dalam database relasional.

B. Kebutuhan Non-Fungsional (Non-Functional Requirement)

Kebutuhan ini berfokus pada kualitas dan batasan sistem:

- NFR-01 (Privasi Data): Seluruh proses pengolahan data dan inferensi AI harus dilakukan secara lokal (*Local LLM*) menggunakan Ollama untuk menjamin keamanan data internal Perusahaan IT dan Penyedia Pelatihan.
- NFR-02 (Ketersediaan): Sistem harus memiliki *Health Check* untuk memastikan koneksi ke basis data vektor dan model AI selalu dalam kondisi siap.
- NFR-03 (Usability): Antarmuka sistem harus responsif dan mudah digunakan oleh admin perusahaan tanpa memerlukan keahlian pemrograman AI.

3.3.2. Use Case Diagram



Gambar 4. *UseCase*

3.3.3. Skenario *Use Case*

3.3.3.1. Skenario Use Case Mengajukan Pertanyaan

Tabel 3. 2. Use Case Mengajukan Pertanyaan

Nama Use Case	Mengajukan Pertanyaan
Aktor	User Publik
Deskripsi	User mengajukan pertanyaan terkait layanan informasi perusahaan melalui antarmuka web.
Prasyarat	Sistem dan API dalam keadaan aktif.
Alur Utama	
Aktor	Sistem
1. User membuka halaman chatbot. 2. User mengetik pertanyaan. 3. User menekan tombol kirim.	4. Frontend mengirimkan pertanyaan ke API backend. 5. Sistem memulai proses RAG. 6. Sistem mengubah pertanyaan menjadi embedding. 7. Sistem melakukan pencarian data relevan pada ChromaDB. 8. Sistem menyusun prompt beserta konteks dokumen. 9. LLM menghasilkan jawaban. 10. Sistem menampilkan jawaban kepada user.
11. User menerima jawaban.	
Alur Alternatif	
Jika pertanyaan kosong, sistem menampilkan pesan kesalahan.	

3.3.3.2. Skenario Use Case Memperbarui Knowledge Base

Tabel 3. 3. Use Case Memperbarui Knowledge Base

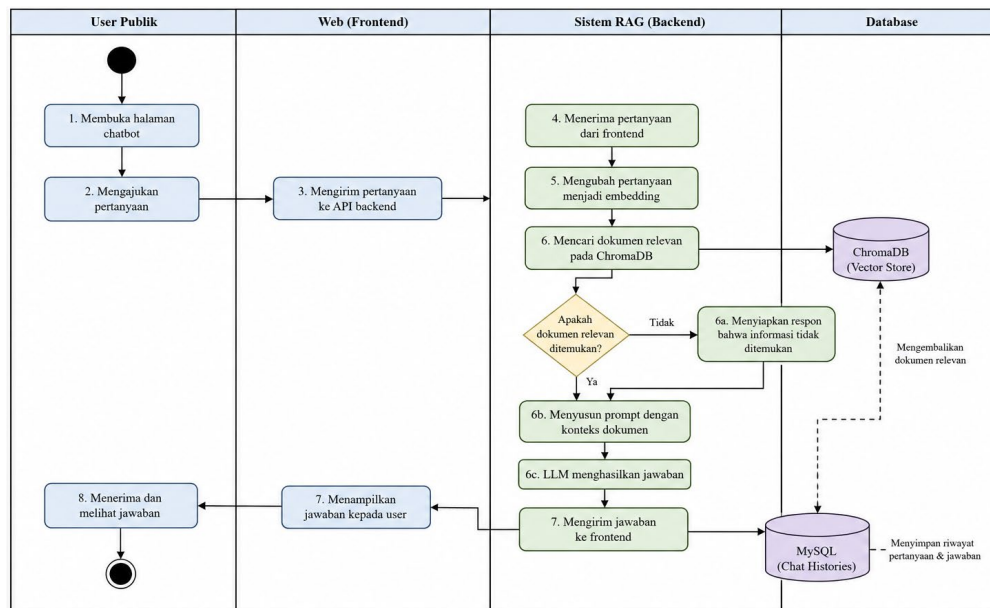
Nama Use Case	Memperbarui Knowledge Base
Aktor	Admin
Deskripsi	Admin memperbarui sumber informasi yang akan digunakan oleh sistem RAG.
Prasyarat	Admin memiliki dokumen sumber yang valid.
Alur Utama	
Aktor	Sistem
1. Admin memilih dokumen yang akan ditambahkan. 2. Admin mengunggah dokumen.	3. Sistem membaca dokumen sumber. 4. Sistem melakukan proses chunking. 5. Sistem membuat embedding dokumen. 6. Sistem menyimpan embedding ke ChromaDB. 7. Sistem mencatat riwayat pembaruan pada MySQL. 8. Sistem menampilkan notifikasi berhasil.
Alur Alternatif	
Jika format dokumen tidak sesuai, sistem menampilkan pesan kesalahan.	

3.3.3.3. Skenario Use Case Menyimpan Riwayat Percakapan

Tabel 3. 4. Use Case Riwayat Percakapan

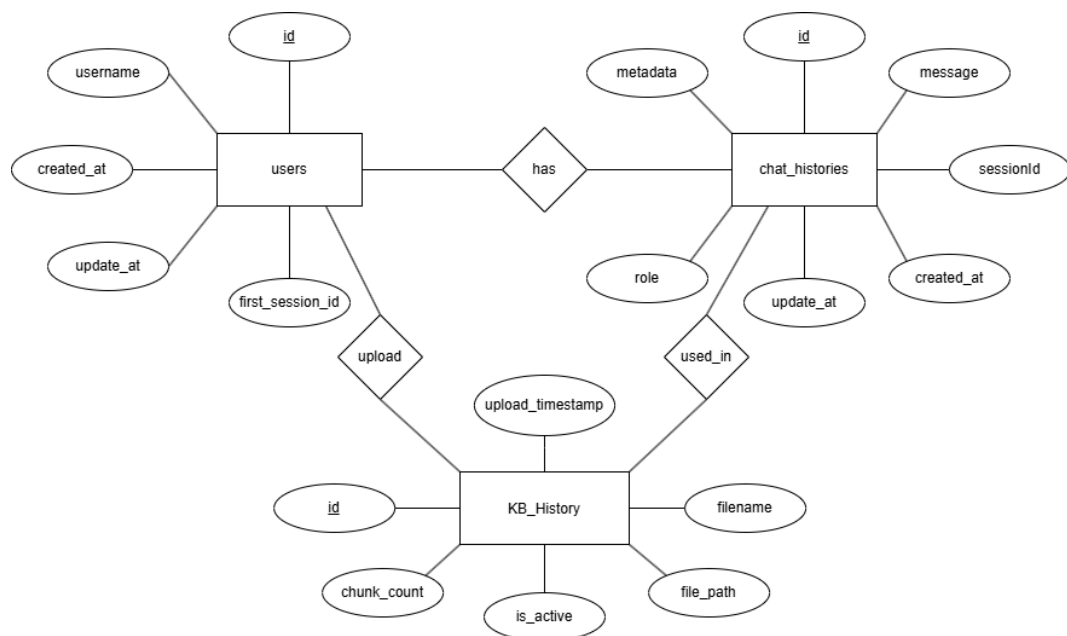
Nama Use Case	Menyimpan Riwayat Percakapan
Aktor	Sistem
Deskripsi	Sistem menyimpan pertanyaan dan jawaban pengguna ke database.
Prasyarat	Proses tanya jawab berhasil dilakukan.
Alur Utama	
Aktor	Sistem
	1. Sistem menerima pertanyaan pengguna. 2. Sistem menghasilkan jawaban. 3. Sistem menyimpan pertanyaan dan jawaban ke database MySQL. 4. Sistem menyimpan waktu percakapan.
Alur Alternatif	
Jika database gagal diakses, sistem tetap menampilkan jawaban tetapi riwayat tidak tersimpan.	

3.3.4. Activity diagram



Gambar 5. Activity diagram

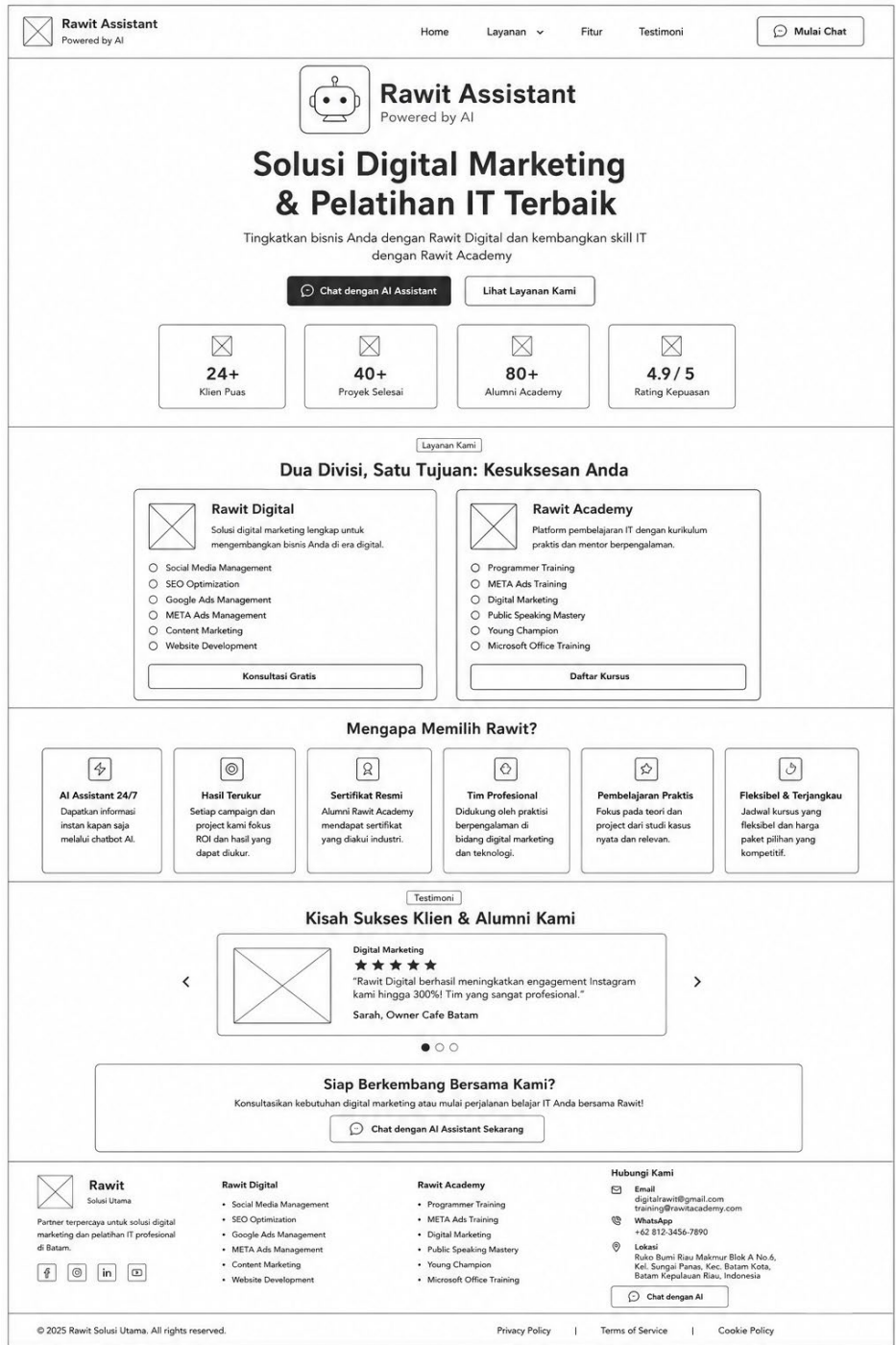
3.3.5. Database Schema (ERD)



Gambar 6. ER Diagram

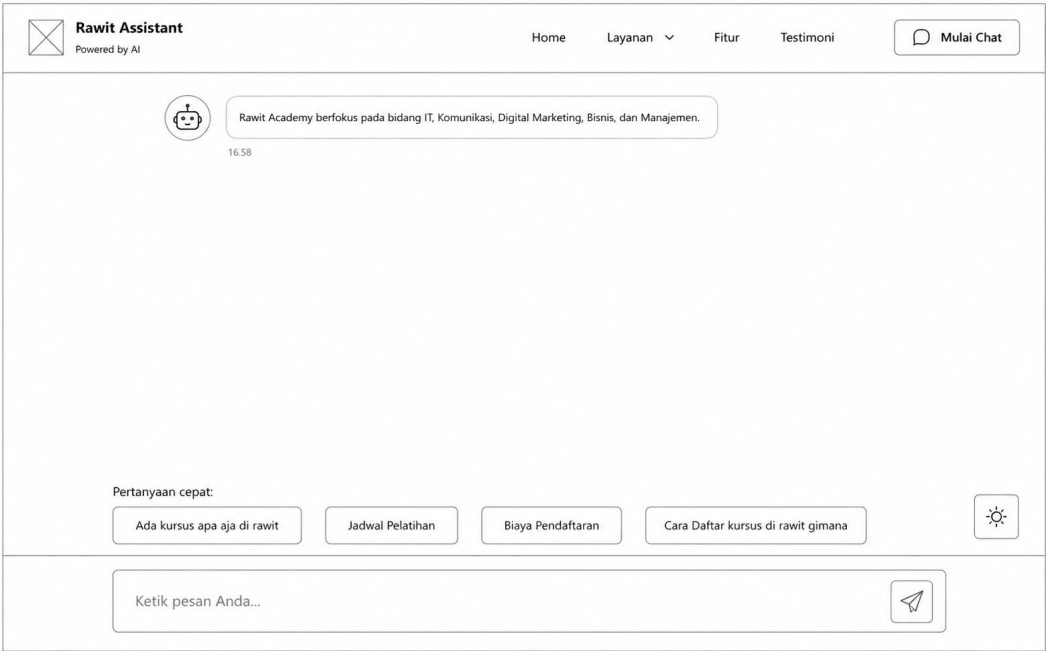
3.3.6. Perancangan Antarmuka (Wireframe)

3.3.6.1. Wireframe Homepage



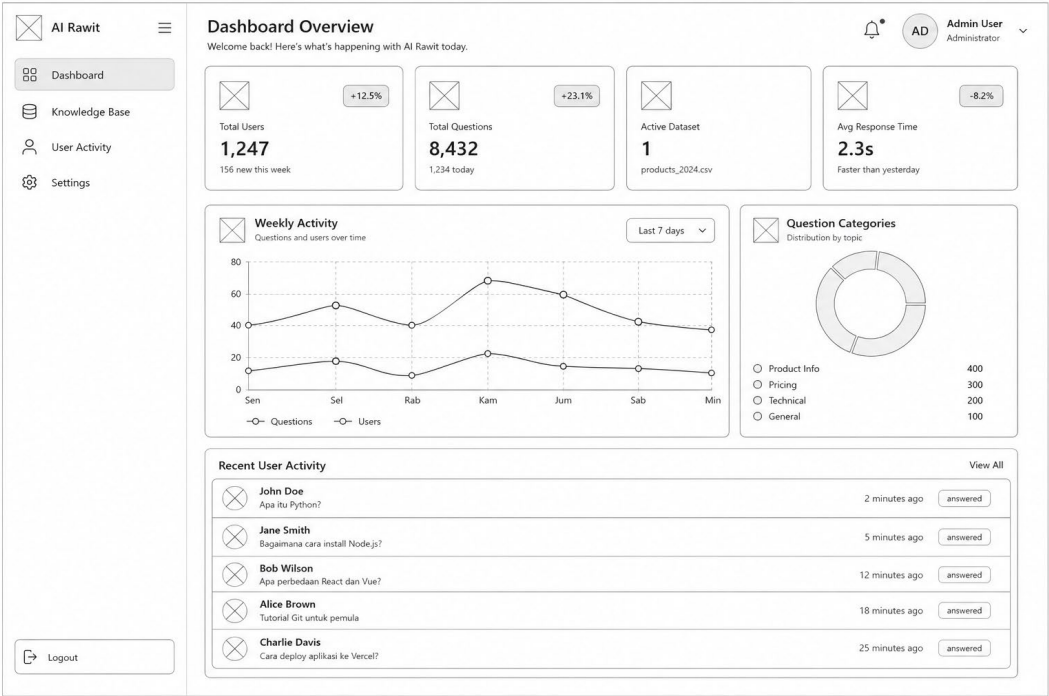
Gambar 7. Wireframe Homepage

3.3.6.2. Wireframe Percakapan



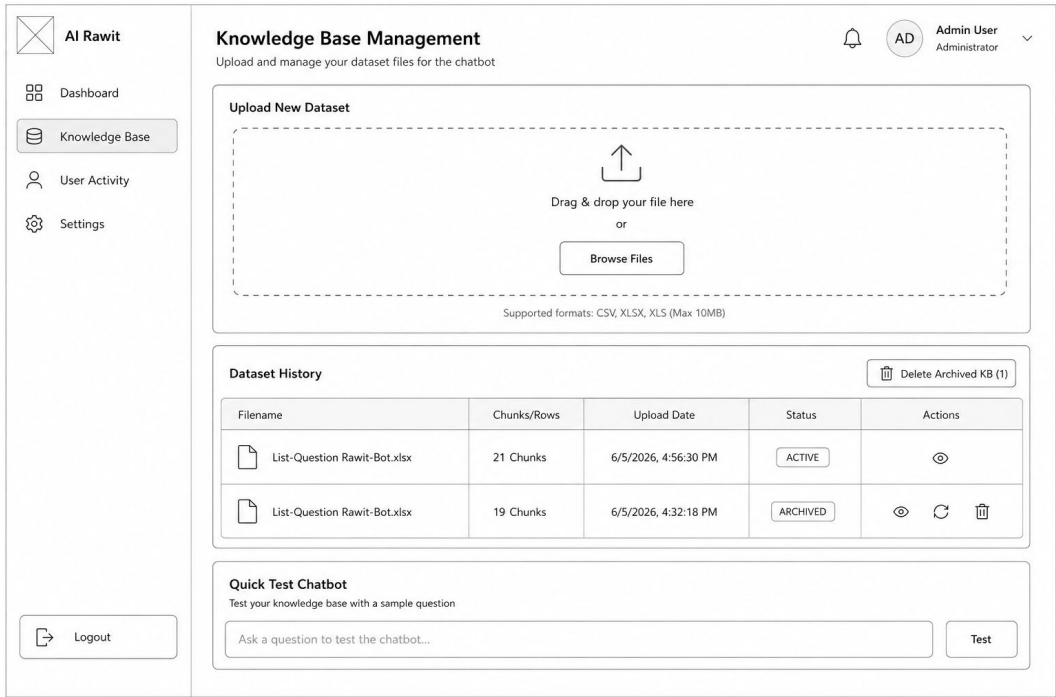
Gambar 8. Wireframe Percakapan

3.3.6.3. Wireframe Dashboard Admin



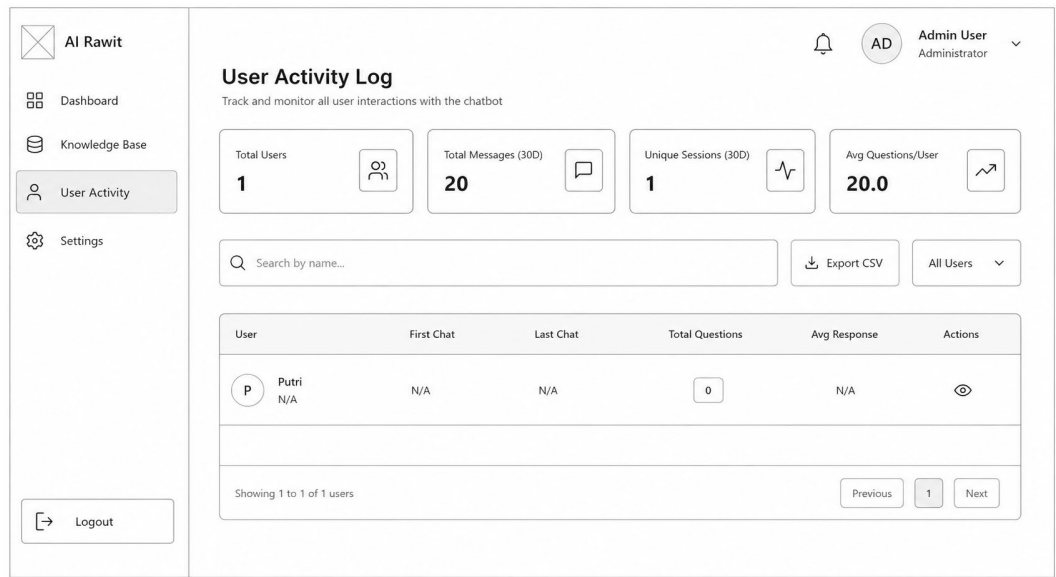
Gambar 9. Wireframe Dashboard Admin

3.3.6.4. Wireframe Upload Knowledge Base



Gambar 10. Wireframe Upload Knowledge Base

3.3.6.5. Wireframe User List



Gambar 11. Wireframe User List

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Hasil Implementasi

4.1.1. Implementasi Arsitektur Sistem

Sistem RawitBot diimplementasikan menggunakan pendekatan Retrieval-Augmented Generation (RAG) yang menggabungkan beberapa komponen utama, yaitu frontend berbasis web, backend API, model embedding, vector database ChromaDB, knowledge base, dan Large Language Model (LLM) Ollama.

Alur kerja sistem dimulai ketika pengguna mengajukan pertanyaan melalui antarmuka web. Pertanyaan tersebut dikirim ke backend untuk diproses menggunakan mekanisme Retrieval-Augmented Generation. Sistem akan melakukan pencarian semantik terhadap knowledge base yang tersimpan pada ChromaDB menggunakan embedding model. Hasil retrieval kemudian diberikan kepada Large Language Model untuk menghasilkan jawaban yang kontekstual dan akurat sebelum ditampilkan kembali kepada pengguna secara otomatis berdasarkan informasi yang tersedia pada knowledge base.

Tabel 4. 1. Komponen Implementasi Sistem

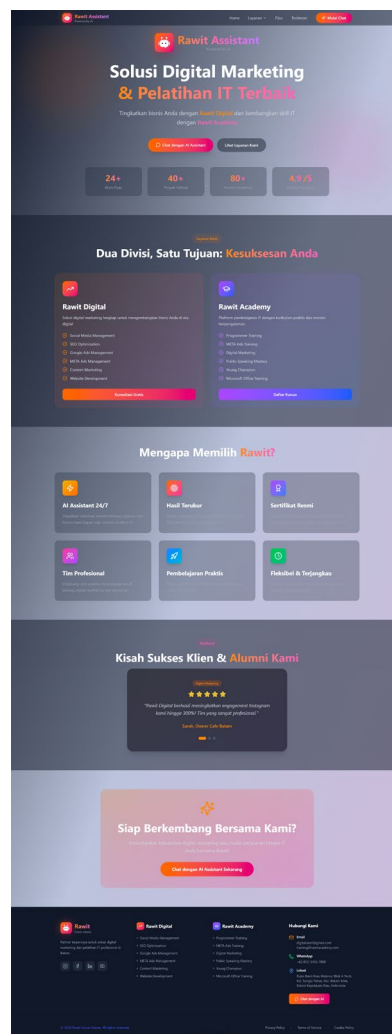
Komponen	Teknologi
Frontend	HTML, CSS, JavaScript
Backend	Python, FastAPI
Framework AI	LangChain
Large Language Model	Ollama
Vector Database	ChromaDB
Embedding Model	HuggingFace Sentence Transformer
Knowledge Base	Dataset FAQ Rawit (.xlsx)

4.1.2. Implementasi Halaman Utama Website

Halaman utama website berfungsi sebagai media informasi perusahaan dan pintu masuk pengguna untuk mengakses chatbot Rawit Assistant.

Fitur yang tersedia pada halaman utama meliputi:

1. Informasi layanan Rawit Digital.
2. Informasi layanan Rawit Academy.
3. Informasi keunggulan perusahaan.
4. Testimoni pengguna.
5. Tombol akses menuju chatbot.



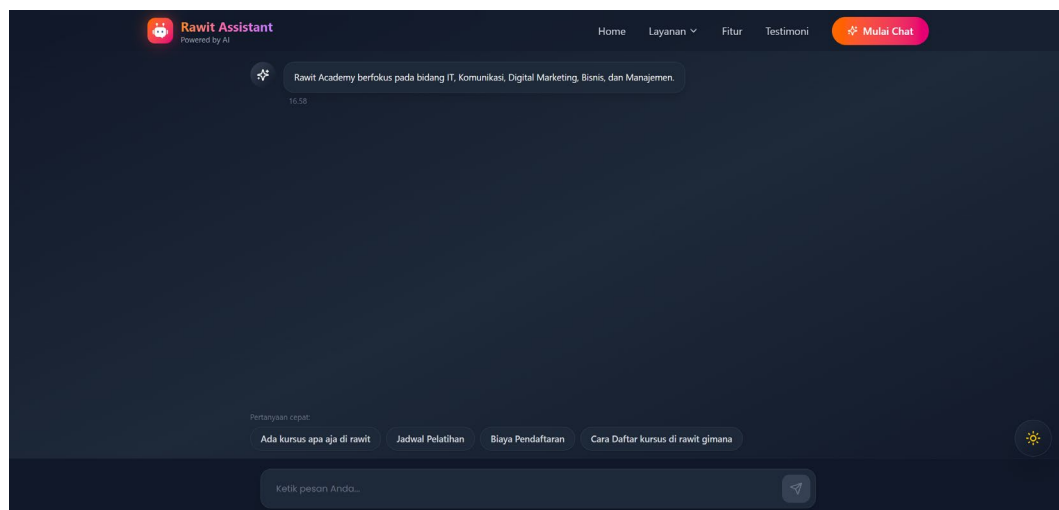
Gambar 12. Halaman Utama Website

4.1.3. Implementasi Halaman Chatbot

Halaman chatbot digunakan pengguna untuk berinteraksi langsung dengan sistem.

Fitur yang tersedia antara lain:

1. Input pertanyaan pengguna.
2. Menampilkan jawaban chatbot.
3. Menampilkan quick question.
4. Menampilkan percakapan secara otomatis berdasarkan informasi yang tersedia pada knowledge base.



Gambar 13. Halaman Chatbot

4.1.4 Implementasi Dashboard Admin

Dashboard admin digunakan untuk memonitor aktivitas sistem.

Fitur yang tersedia meliputi:

1. Total pengguna.
2. Total pertanyaan.
3. Dataset aktif.
4. Statistik performa chatbot.
5. Aktivitas pengguna.



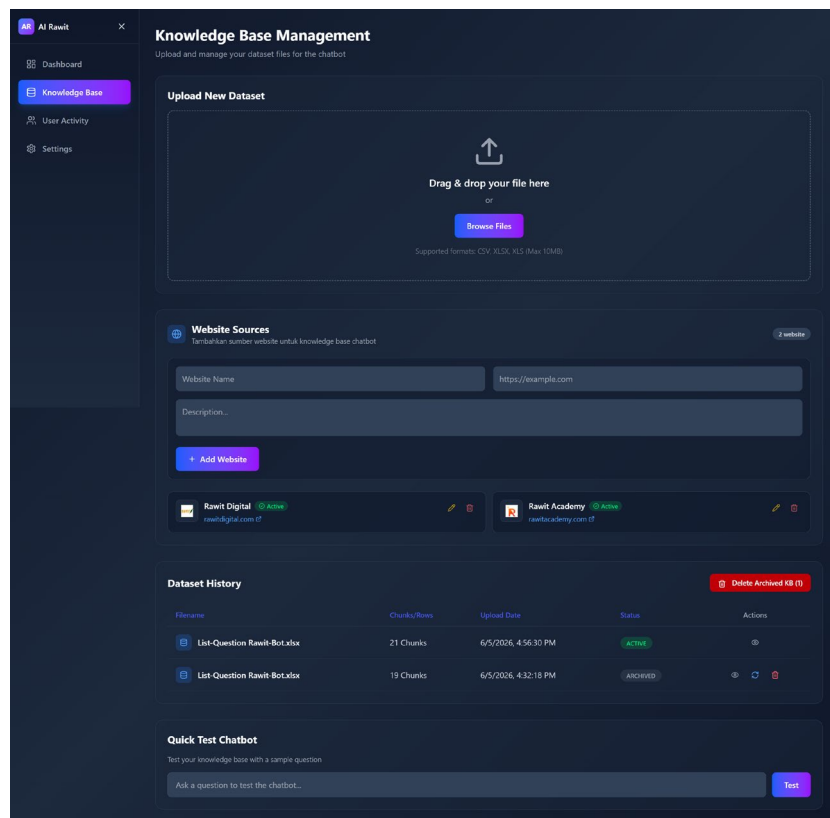
Gambar 14. Dashboard Admin

4.1.5 Implementasi Manajemen Knowledge Base

Halaman Knowledge Base digunakan admin untuk mengelola dataset chatbot.

Fitur yang tersedia:

1. Upload dataset.
2. Menampilkan riwayat dataset.
3. Mengaktifkan dataset.
4. Mengarsipkan dataset.
5. Menghapus dataset.
6. Quick test chatbot.



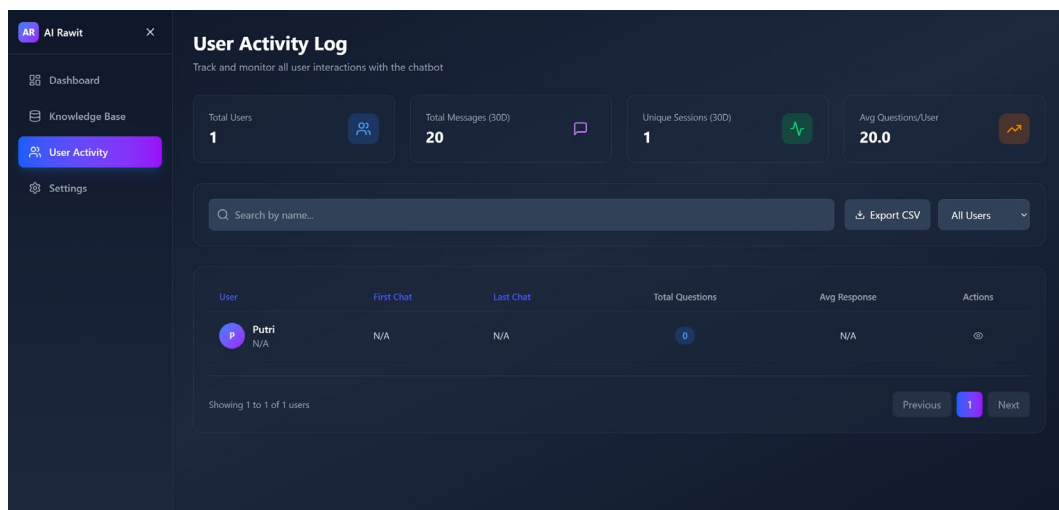
Gambar 15. Manajemen Knowledge Base

4.1.6 Implementasi User Activity Log

Halaman User Activity Log digunakan untuk memonitor interaksi pengguna terhadap chatbot.

Informasi yang ditampilkan:

1. Nama pengguna.
2. Total pertanyaan.
3. Waktu interaksi.
4. Riwayat percakapan.
5. Export data aktivitas pengguna.



Gambar 16. User Activity Log

4.2 Pengujian User Acceptance Testing (UAT)

Pengujian User Acceptance Testing dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan pengguna. Pengujian dilakukan menggunakan metode skenario pengujian dengan pendekatan black-box testing.

Tabel 4. 2. Hasil Pengujian User Acceptance Testing

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Membuka website	Website berhasil ditampilkan	Sesuai	Berhasil
2	Menekan tombol Mulai Chat	Halaman chatbot terbuka	Sesuai	Berhasil
3	Mengirim pertanyaan yang tersedia pada knowledge base	Chatbot memberikan jawaban yang relevan	Sesuai	Berhasil
4	Mengirim pertanyaan yang tidak tersedia	Sistem menampilkan pesan informasi tidak tersedia	Sesuai	Berhasil
5	Mengunggah dataset baru	Dataset berhasil tersimpan	Sesuai	Berhasil
6	Mengaktifkan dataset	Dataset menjadi aktif	Sesuai	Berhasil
7	Melihat aktivitas pengguna	Data aktivitas berhasil ditampilkan	Sesuai	Berhasil

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat diperoleh beberapa kesimpulan sebagai berikut.

1. **Sistem Question Answering berbasis Retrieval-Augmented Generation (RAG) untuk layanan informasi PT Rawit Solusi Utama berhasil dibangun dan diimplementasikan.** Sistem dikembangkan dengan memanfaatkan beberapa teknologi utama, yaitu LangChain sebagai framework orkestrasi, Ollama sebagai penyedia Large Language Model (LLM), dan ChromaDB sebagai vector database untuk penyimpanan embedding data. Integrasi ketiga komponen tersebut memungkinkan sistem bekerja secara terstruktur dalam memproses pertanyaan pengguna dan menghasilkan jawaban yang relevan.
2. **Penerapan metode Retrieval-Augmented Generation (RAG) mampu meningkatkan kualitas jawaban yang diberikan sistem.** Berbeda dengan chatbot konvensional yang hanya mengandalkan kemampuan generatif model bahasa, pendekatan RAG memanfaatkan proses pencarian informasi terlebih dahulu dari knowledge base internal perusahaan sebelum menghasilkan jawaban. Dengan demikian, sistem dapat memberikan jawaban yang lebih kontekstual, akurat, dan sesuai dengan data yang dimiliki oleh PT Rawit Solusi Utama.
3. **Integrasi antara knowledge base, embedding model, vector database, dan Large Language Model berhasil mendukung proses pencarian informasi secara efektif.** Dokumen yang dimiliki perusahaan terlebih dahulu diproses menjadi embedding dan disimpan ke dalam ChromaDB sehingga memungkinkan proses pencarian semantik dilakukan dengan cepat. Hasil pencarian tersebut kemudian digunakan sebagai konteks bagi LLM untuk membentuk jawaban yang lebih relevan terhadap pertanyaan pengguna.

4. **Sistem berhasil menyediakan layanan informasi perusahaan secara otomatis berdasarkan informasi yang tersedia pada knowledge base melalui antarmuka web yang mudah digunakan.** Pengguna umum dapat mengakses chatbot tanpa memerlukan kemampuan teknis khusus. Proses interaksi yang sederhana, yaitu memasukkan pertanyaan dan menerima jawaban secara langsung, memberikan kemudahan bagi pengguna dalam memperoleh informasi mengenai layanan, program pelatihan, maupun informasi lain yang tersedia pada PT Rawit Solusi Utama.
5. **Berdasarkan hasil User Acceptance Testing (UAT), seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna.** Fitur-fitur seperti pengelolaan knowledge base, pengiriman pertanyaan, proses pencarian informasi, hingga penyajian jawaban dapat berfungsi dengan baik. Hal ini menunjukkan bahwa sistem telah memenuhi kebutuhan fungsional yang telah ditetapkan pada tahap analisis dan perancangan sistem.
6. **Implementasi sistem memberikan manfaat bagi PT Rawit Solusi Utama dalam meningkatkan efisiensi penyampaian informasi perusahaan.** Dengan adanya chatbot berbasis RAG, proses pelayanan informasi yang sebelumnya dilakukan secara manual dapat diotomatisasi sehingga mengurangi beban administrasi, mempercepat proses pelayanan, serta meningkatkan konsistensi informasi yang diterima oleh pengguna.
7. **Penelitian ini membuktikan bahwa teknologi RAG memiliki potensi yang besar untuk diimplementasikan pada layanan informasi perusahaan.** Pendekatan ini tidak hanya mampu meningkatkan kualitas jawaban, tetapi juga dapat dikembangkan lebih lanjut menjadi sistem informasi cerdas yang adaptif terhadap kebutuhan organisasi di masa mendatang.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan sebagai bahan pengembangan sistem pada penelitian selanjutnya.

1. **Menambahkan mekanisme pembaruan knowledge base secara otomatis.** Saat ini proses penambahan dan pembaruan data masih dilakukan secara manual oleh administrator. Pada penelitian berikutnya, sistem dapat dikembangkan agar mampu melakukan sinkronisasi data secara otomatis dari sumber data perusahaan sehingga informasi yang tersedia selalu diperbarui tanpa memerlukan intervensi yang besar dari pengguna.
2. **Menambahkan fitur autentikasi dan manajemen hak akses pengguna.** Pengembangan fitur login diperlukan agar sistem dapat membedakan peran pengguna, seperti administrator dan pengguna umum. Dengan adanya manajemen hak akses, keamanan data perusahaan dapat ditingkatkan dan proses pengelolaan sistem menjadi lebih terkontrol.
3. **Mengembangkan kemampuan chatbot agar mendukung multi-bahasa.** Saat ini sistem masih berfokus pada penggunaan Bahasa Indonesia. Pada pengembangan selanjutnya, chatbot dapat ditingkatkan agar mampu melayani pengguna dalam berbagai bahasa, seperti Bahasa Inggris maupun bahasa lainnya, sehingga cakupan pengguna menjadi lebih luas.
4. **Menambahkan fitur analitik untuk memantau aktivitas pengguna dan pola pertanyaan.** Informasi mengenai pertanyaan yang paling sering diajukan, tingkat keberhasilan jawaban, dan aktivitas pengguna dapat dimanfaatkan oleh perusahaan untuk mengevaluasi kebutuhan informasi pengguna dan meningkatkan kualitas layanan yang diberikan.
5. **Mengintegrasikan sistem dengan platform komunikasi lainnya.** Pengembangan integrasi dengan media komunikasi seperti WhatsApp, Telegram, maupun platform pesan instan lainnya dapat meningkatkan aksesibilitas sistem sehingga pengguna tidak hanya terbatas menggunakan antarmuka web.
6. **Meningkatkan kemampuan sistem dalam menangani informasi yang lebih kompleks dan dinamis.** Pengembangan lanjutan dapat dilakukan dengan memperluas sumber data, menambahkan mekanisme validasi informasi, serta mengoptimalkan model bahasa yang digunakan agar sistem semakin adaptif terhadap perubahan kebutuhan informasi perusahaan.

DAFTAR PUSTAKA

- Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *International Journal of Engineering & Technology (iJET)*, 2(5), 2049–3444.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for knowledge-intensive NLP tasks*. *Advances in Neural Information Processing Systems*, 33, 9459–9474.
- Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J. B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., et al. (2022). *Improving language models by retrieving from trillions of tokens*. *Nature*, 601(7893), 308–314.
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Wang, Y., Sun, J., Guo, M., Wang, H., & Wang, H. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey*. *ACM Computing Surveys*.
- Huang, Y., & Huang, J. (2024). *A Survey on Retrieval-Augmented Text Generation for Large Language Models*. *Information Fusion*, 104, 102332.
- Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., et al. (2026). *Retrieval-Augmented Generation for AI-Generated Content: A Survey*. *Data Science and Engineering*.
- Xu, K., Zhang, K., Li, J., Huang, W., & Wang, Y. (2024). *CRP-RAG: A Retrieval-Augmented Generation Framework for Logical Reasoning*. *Electronics*, 13(17).
- Zhang, et al. (2024). Effectiveness of Retrieval-Augmented Generation-based LLM. *Automation in Construction*.
- Zhang, et al. (2024). Performance Comparison of RAG and Fine-tuned LLM. *Automation in Construction*.

DAFTAR LAMPIRAN

Lampiran A. Dokumentasi
[Dokumentasi](#)

Lampiran B. Link Github
[SaiiNP23/Rawit-ChatBot](#)