

**IMPLEMENTASI CHATBOT OPENAI SEBAGAI CUSTOMER SERVICE  
PADA WHATSAPP DENGAN RETRIEVAL AUGMENTED GENERATION  
VIA LANGCHAIN**

**TUGAS AKHIR**



**FIKRY CATUR FARENZA  
311910029**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS MA CHUNG  
MALANG  
2025**

## PERNYATAAN ORISINALITAS TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fikry Catur Farenza  
NIM : 311910029  
Program Studi : Teknik Informatika  
Perguruan Tinggi : Universitas Ma Chung

Dengan ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Implementasi Chatbot OpenAI Sebagai Customer Service Pada WhatsApp Dengan Retrieval Augmented Generation Via LangChain”** adalah asli (orisinal) atau tidak plagiat dan benar hasil karya intelektual mandiri, dan belum pernah diterbitkan/dipublikasikan di manapun dan dalam bentuk apapun. Kecuali secara tertulis dikutip dalam skripsi ini dan disebutkan berupa sumber kutipan dan pada daftar pustaka.

Surat pernyataan ini saya buat dengan sebenar-benarnya tanpa adanya paksaan dari pihak mana pun. Apabila dikemudian hari terbukti kuat adanya penyimpangan dan kesalahan terhadap pernyataan ini, saya bersedia menerima konsekuensi dan sanksi apapun, sesuai dengan hukum dan peraturan yang telah berlaku di Universitas Ma Chung.

Malang, 9 Februari 2025



Fikry Catur Farenza  
NIM. 311910029



**LEMBAR PENGESAHAN**

**TUGAS AKHIR**

**IMPLEMENTASI CHATBOT OPENAI SEBAGAI CUSTOMER SERVICE  
PADA WHATSAPP DENGAN RETRIEVAL AUGMENTED GENERATION  
VIA LANGCHAIN**

Oleh:

**FIKRY CATUR FARENZA**

**NIM. 311910029**

dari:

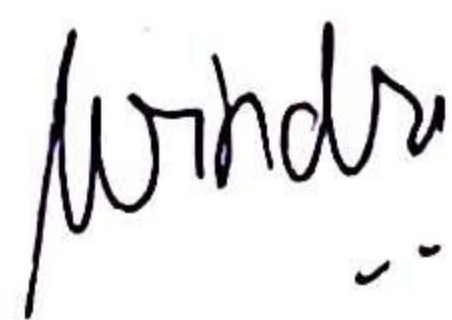
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI dan DESAIN**

**UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan dan berhak mendapatkan gelar Sarjana Komputer (S.Kom.)

**Dosen Pembimbing I,**



**Windra Swastika, Ph.D.**  
**NIP. 20070039**

**Dosen Pembimbing II,**



**Paulus Lucky T. I., S.Kom., MT.**  
**NIP. 20100005**

Dekan Fakultas Teknologi dan Desain,



**Prof. Dr. Eng. Romy Budhi Widodo**  
**NIP. 20070035**



## ABSTRAK

Perkembangan teknologi kecerdasan buatan (AI) telah membuka peluang baru dalam meningkatkan efisiensi dan kualitas layanan, termasuk di lingkungan perguruan tinggi. Dengan keterbatasan sumber daya manusia dalam menjawab pertanyaan mahasiswa, terutama terkait informasi "Student Guide", mendorong Universitas Ma Chung untuk mencari solusi inovatif dengan memanfaatkan teknologi AI tersebut. Penelitian ini bertujuan mengimplementasikan *chatbot* OpenAI sebagai layanan *customer service* otomatis di WhatsApp, aplikasi pesan instan yang sering dipakai oleh mahasiswa kampus. Metode *Retrieval Augmented Generation* (RAG) dipilih untuk mengatasi keterbatasan *chatbot* generatif yang memungkinkan pengambilan informasi dari dokumen PDF secara efisien. Implementasi melibatkan model OpenAI, *framework* LangChain, dan integrasi dengan WhatsApp API. Hasil survei menunjukkan 89,28% dari total respons yang didapat positif, dengan akurasi *chatbot* memberikan jawaban faktual mencapai 94,59% dan akurasi *retrieval* dokumen 97,30%. Meskipun demikian, masih ditemukannya *hallucination* pada 4,05% jawaban dan 2,7% respons *chatbot* gagal menemukan informasi dari pertanyaan.

**Kata kunci :** *Chatbot, LangChain, OpenAI, Retrieval Augmented Generation, WhatsApp Business API*

## ABSTRACT

Artificial intelligence (AI) has revolutionized numerous sectors, including higher education like universities, by offering innovative solutions to improve service efficiency and quality. This research addresses the challenge faced by Ma Chung University in efficiently handling student inquiries, particularly those concerning the "Student Guide," given the limitations of human resources. The study aims to implement an OpenAI-powered chatbot on WhatsApp, a widely-used messaging platform among students. The Retrieval Augmented Generation (RAG) method was employed to overcome the constraints of generative chatbots, enabling efficient information retrieval from PDF documents. The implementation involved the OpenAI model, the LangChain framework, and integration with the WhatsApp API. Survey results indicated that 89.28% of the responses were positive, with the chatbot achieving a 94.59% accuracy rate in providing factual answers and a 97.30% accuracy in document retrieval. Nevertheless, some hallucinations (4.05%) and instances where the chatbot failed to find relevant information (2.7%) were observed.

**Kata kunci :** *Chatbot, LangChain, OpenAI, Retrieval Augmented Generation, WhatsApp Business API*

## KATA PENGANTAR

Pertama-tama segala puji dan syukur ke hadirat Tuhan Yang Maha Esa atas hidayah dan karunia-Nya Tugas Akhir ini dapat terselesaikan dengan sebaik-baiknya. Laporan ini dibuat berdasarkan hasil dari berbagai macam kegiatan yang dilakukan penulis untuk menyelesaikan Tugas Akhir. Maka dari itu, pada kesempatan ini penulis ingin menyampaikan terima kasih banyak yang sebesar-besarnya kepada berbagai pihak yang berturut serta membantu proses pengerjaan laporan ini, antara lain:

1. Kedua orang tua dan keluarga terkasih yang selalu memberikan berbagai macam dukungan dan semangat dalam proses pembuatan laporan.
2. Bapak Dr.Eng. Romy Budhi, ST., MT. selaku Dekan dari Fakultas Teknologi dan Desain Universitas Ma Chung
3. Bapak Hendry Setiawan, ST., M.Kom. selaku Kepala Program Studi Teknik Informatika sekaligus sebagai ketua penguji Tugas Akhir.
4. Bapak Windra Swastika, S.Kom., MT., Ph.D. selaku salah satu pendamping Tugas Akhir saya.
5. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT. selaku Dosen Pembimbing Akademik dan begitu juga sebagai pembimbing Tugas Akhir.
6. Serta yang terakhir adalah teman-teman saya yang telah membantu dalam proses pengerjaan laporan secara langsung maupun tidak langsung.

Demi terciptanya Laporan Tugas Akhir yang lebih baik dan bermanfaat bagi kita semua, penulis akan selalu terbuka terhadap segala macam kritik dan saran yang ada. Demikian, semoga laporan Tugas Akhir ini dapat bermanfaat bagi yang membutuhkan.

Malang, 5 Februari 2025

Fikry Catur Farenza  
NIM. 311910029

## DAFTAR ISI

LEMBAR PENGESAHAN TUGAS AKHIR	Error! Bookmark not defined.
PERNYATAAN KEASLIAN TUGAS AKHIR	Error! Bookmark not defined.
ABSTRAK	Error! Bookmark not defined.
ABSTRACT	Error! Bookmark not defined.
KATA PENGANTAR	Error! Bookmark not defined.
DAFTAR ISI	Error! Bookmark not defined.
DAFTAR GAMBAR	Error! Bookmark not defined.
DAFTAR TABEL	Error! Bookmark not defined.
DAFTAR LAMPIRAN	Error! Bookmark not defined.
BAB I PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang	Error! Bookmark not defined.
1.2 Identifikasi Masalah	Error! Bookmark not defined.
1.3 Batasan Masalah	Error! Bookmark not defined.
1.4 Rumusan Masalah	Error! Bookmark not defined.
1.5 Tujuan	Error! Bookmark not defined.
1.6 Manfaat	Error! Bookmark not defined.
1.7 Luaran Penelitian	Error! Bookmark not defined.
1.8 Sistematika Penulisan	Error! Bookmark not defined.
BAB II TINJAUAN PUSTAKA	Error! Bookmark not defined.
2.1 Web Service	Error! Bookmark not defined.
2.2 REST API	Error! Bookmark not defined.
2.3 Prompt Engineering	Error! Bookmark not defined.
2.4 OpenAI API	Error! Bookmark not defined.
2.5 Token	Error! Bookmark not defined.
2.6 Semantik	Error! Bookmark not defined.
2.7 Embedding	Error! Bookmark not defined.
2.8 Vectorstore	Error! Bookmark not defined.
2.9 Retrieval Augmented Generation	Error! Bookmark not defined.
2.10 Python	Error! Bookmark not defined.
2.11 LangChain	Error! Bookmark not defined.

2.12	WhatsApp Business Platform	<b>Error! Bookmark not defined.</b>
2.12.1	WhatsApp Webhook	<b>Error! Bookmark not defined.</b>
2.13	Flask	<b>Error! Bookmark not defined.</b>
2.14	Penelitian Terdahulu	<b>Error! Bookmark not defined.</b>
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM</b>		<b>Error! Bookmark not defined.</b>
3.1	Tahapan Penelitian	<b>Error! Bookmark not defined.</b>
3.2	Analisis Kebutuhan	<b>Error! Bookmark not defined.</b>
3.3	Studi Pustaka	<b>Error! Bookmark not defined.</b>
3.4	Desain dan Perancangan Sistem	<b>Error! Bookmark not defined.</b>
3.4.1	Pembuatan Vectorstore dari Dokumen PDF	<b>Error! Bookmark not defined.</b>
3.4.2	Perancangan Metode Retrieval Augment Generation	<b>Error! Bookmark not defined.</b>
3.4.3	Perancangan Umpan Balik pada Chatbot	<b>Error! Bookmark not defined.</b>
3.4.4	Desain Tampilan Aplikasi Website	<b>Error! Bookmark not defined.</b>
3.5	Perancangan Pengujian	<b>Error! Bookmark not defined.</b>
3.5.1	Pengujian Kepuasan Pengguna	<b>Error! Bookmark not defined.</b>
3.5.2	Pengujian Kualitas Respons Chatbot	<b>Error! Bookmark not defined.</b>
<b>BAB IV HASIL DAN PEMBAHASAN</b>		<b>Error! Bookmark not defined.</b>
4.1	Implementasi Desain dan Fitur	<b>Error! Bookmark not defined.</b>
4.1.1	Sistem Generasi Vectorstore Database	<b>Error! Bookmark not defined.</b>
4.1.2	Implementasi Chat Bot pada Sistem	<b>Error! Bookmark not defined.</b>
4.1.3	Pengembangan Fitur Mengunggah Dokumen	<b>Error! Bookmark not defined.</b>
4.1.4	Integrasi Sistem dengan WhatsApp	<b>Error! Bookmark not defined.</b>
4.1.5	Sistem Percakapan dan Umpan Balik	<b>Error! Bookmark not defined.</b>



4.2	Peluncuran Aplikasi	<b>Error! Bookmark not defined.</b>
4.3	Hasil Pengembangan Aplikasi	<b>Error! Bookmark not defined.</b>
4.3.1	Aplikasi Web	<b>Error! Bookmark not defined.</b>
4.3.2	Aplikasi WhatsApp	<b>Error! Bookmark not defined.</b>
4.4	Pengujian Sistem Aplikasi	<b>Error! Bookmark not defined.</b>
4.5	Analisis Hasil Kinerja Chatbot	<b>Error! Bookmark not defined.</b>
4.5.1	Umpan Balik	<b>Error! Bookmark not defined.</b>
4.5.2	Penilaian Akurasi Jawaban Chatbot	<b>Error! Bookmark not defined.</b>
4.6	Analisis Kesalahan Chatbot dalam Merespons	<b>Error! Bookmark not defined.</b>
4.6.1	Kasus Pertanyaan Pimpinan dan Rektor Universitas	<b>Error! Bookmark not defined.</b>
4.6.2	Kasus Pertanyaan Umum dan Di Luar Konteks Dokumen	<b>Error! Bookmark not defined.</b>
4.6.3	Kasus Hallucination Pada Chatbot	<b>Error! Bookmark not defined.</b>
BAB V SIMPULAN DAN SARAN		<b>Error! Bookmark not defined.</b>
5.1	Kesimpulan	<b>Error! Bookmark not defined.</b>
5.2	Saran	<b>Error! Bookmark not defined.</b>
DAFTAR PUSTAKA		<b>Error! Bookmark not defined.</b>
LAMPIRAN		<b>Error! Bookmark not defined.</b>

## DAFTAR TABEL

Tabel 3.1 Daftar Biaya Kebutuhan Penelitian	<b>Error! Bookmark not defined.</b>
Tabel 3.2 Tabel Prompt untuk LLM	<b>Error! Bookmark not defined.</b>
Tabel 3.3 Format Data Umpan Balik yang Disimpan	<b>Error! Bookmark not defined.</b>
Tabel 4.1 Tabel Evaluasi Hasil Sistem Aplikasi	<b>Error! Bookmark not defined.</b>
Tabel 4.2 Tabel Data Umpan Balik Kurang dan Tidak Sesuai	<b>Error! Bookmark not defined.</b>
Tabel 4.3 Daftar Respons Tanya dengan Jawaban yang Salah	<b>Error! Bookmark not defined.</b>

## DAFTAR GAMBAR

- Gambar 2.1 Representasi Data Sebagai Vektor      **Error! Bookmark not defined.**
- Gambar 2.2 Pemanfaatan Model Embedding Dalam RAG **Error! Bookmark not defined.**
- Gambar 2.3 Contoh Pemanfaatan Vectorstore dalam Alur RAG dengan LLM  
**Error! Bookmark not defined.**
- Gambar 2.4 Diagram Alur RAG pada LLM      **Error! Bookmark not defined.**
- Gambar 2.5 LangChain Framework Overview      **Error! Bookmark not defined.**
- Gambar 2.6 Perbedaan Antar Aplikasi WhatsApp **Error! Bookmark not defined.**
- Gambar 2.7 Contoh Alur Implementasi WhatsApp API Cloud Webhook      **Error! Bookmark not defined.**
- Gambar 2.8 Contoh Notifikasi Pesan Masuk      **Error! Bookmark not defined.**
- Gambar 3.1 *Flowchart* Tahapan Penilitan      **Error! Bookmark not defined.**
- Gambar 3.2 Diagram Alur Sistem Aplikasi      **Error! Bookmark not defined.**
- Gambar 3.3 Alur Pengolahan Dokumen Menjadi *Vectorstore* **Error! Bookmark not defined.**
- Gambar 3.4 Gambar Kode Pemecahan Dokumen **Error! Bookmark not defined.**
- Gambar 3.5 Contoh Isi dari Sebuah Fragmen Dokumen **Error! Bookmark not defined.**
- Gambar 3.6 Alur Percakapan Aplikasi RAG dengan LangChain **Error! Bookmark not defined.**
- Gambar 3.7 Diagram Alur Umpan Balik Pada Chatbot **Error! Bookmark not defined.**
- Gambar 3.8 Desain Mockup Website Menu Unggah Dokumen **Error! Bookmark not defined.**
- Gambar 3.9 Desain Mockup Halaman Menu Chat **Error! Bookmark not defined.**
- Gambar 3.10 Contoh Survei Melalui WhatsApp      **Error! Bookmark not defined.**
- Gambar 3.11 Contoh Sebuah Data Respons Untuk Pengujian **Error! Bookmark not defined.**
- Gambar 4.1 Kode Mengolah Dokumen Menjadi Vectorstore **Error! Bookmark not defined.**

Gambar 4.2 Kode Simpan dan Memuat Vectorstore **Error! Bookmark not defined.**

Gambar 4.3 Hasil Vectorstore dari Dokumen **Error! Bookmark not defined.**

Gambar 4.4 Keluaran Pecahan Dokumen yang Didapat dari Vectorstore **Error! Bookmark not defined.**

Gambar 4.5 Kode Prompt Percakapan LangChain **Error! Bookmark not defined.**

Gambar 4.6 Contoh Percakapan dengan Chatbot **Error! Bookmark not defined.**

Gambar 4.7 Rekonstruksi Pertanyaan di Kiri dan Pencarian Dokumen ke Vectorstore di Kanan **Error! Bookmark not defined.**

Gambar 4.8 Generasi Jawaban dari Konteks dan Sejarah Percakapan **Error! Bookmark not defined.**

Gambar 4.9 Kode Form di File Bernama forms.py **Error! Bookmark not defined.**

Gambar 4.10 Contoh Kode Menampilkan dan Mengunggah Dokumen **Error! Bookmark not defined.**

Gambar 4.11 Contoh Hasil Pembuatan Akun WhatsApp **Error! Bookmark not defined.**

Gambar 4.12 Contoh Kode Verifikasi Webhook **Error! Bookmark not defined.**

Gambar 4.13 Verifikasi Webhook pada Dasbor Aplikasi **Error! Bookmark not defined.**

Gambar 4.14 Kode Contoh Payload Webhook Notifikasi Pesan Masuk **Error! Bookmark not defined.**

Gambar 4.15 Kode Mengirimkan Pesan WhatsApp **Error! Bookmark not defined.**

Gambar 4.16 Kode Model Class untuk Percakapan **Error! Bookmark not defined.**

Gambar 4.17 Contoh Kode Percakapan dengan Umpan Balik **Error! Bookmark not defined.**

Gambar 4.18 Hasil Deploy Sistem Aplikasi Percakapan dan Database **Error! Bookmark not defined.**

Gambar 4.19 Contoh Hasil Halaman Mengunggah Dokumen **Error! Bookmark not defined.**

Gambar 4.20 Hasil Halaman Chat pada Website **Error! Bookmark not defined.**

Gambar 4.21 Hasil Halaman Data pada Website **Error! Bookmark not defined.**



Gambar 4.22 Hasil Halaman Kredit pada Website **Error! Bookmark not defined.**

Gambar 4.23 Contoh Interaksi Pengguna Melalui WhatsApp**Error!   Bookmark not defined.**

Gambar 4.24 Contoh Sebuah Data Tanya Jawab **Error! Bookmark not defined.**

Gambar 4.25 Diagram Pie Chart Distribusi Umpan Balik dari Semua Tanya Jawab yang Menerima Umpan Balik Saja **Error! Bookmark not defined.**

Gambar 4.26 Contoh Data Tanya Jawab dengan Variabel Metrik Evaluasi **Error! Bookmark not defined.**

Gambar 4.27 Kualitas Chatbot Secara Keseluruhan**Error!   Bookmark not defined.**

Gambar 4.28 Histogram Kebenaran Jawaban **Error! Bookmark not defined.**

Gambar 4.29 Visualisasi Evaluasi Chatbot Menolak Untuk Menjawab **Error! Bookmark not defined.**

Gambar 4.30 Visualisasi Detail Akurasi Mendapatkan Fragmen Dokumen **Error! Bookmark not defined.**

Gambar 4.31 Visualisasi Evaluasi Halusinasi Chatbot**Error!   Bookmark not defined.**

Gambar 4.32 Tampilan Informasi Rektor Universitas Ma Chung dari Dokumen **Error! Bookmark not defined.**

Gambar 4.33 Tampilan Dokumen yang Sama dengan Jawaban Chatbot **Error! Bookmark not defined.**

## DAFTAR LAMPIRAN

Lampiran 1 Kode Pembuatan Vectorstore **Error! Bookmark not defined.**

Lampiran 2 Kode Form Pengunggahan Dokumen pada Halaman Website **Error! Bookmark not defined.**

Lampiran 3 Kode Percakapan Dengan Umpan Balik**Error! Bookmark not defined.**

Lampiran 4 Pecahan Dokumen Terdekat Kasus ID 24**Error! Bookmark not defined.**

Lampiran 5 Pecahan Dokumen Terdekat Kasus ID 23**Error! Bookmark not defined.**

Lampiran 6 Pecahan Dokumen Kasus Halusinasi **Error! Bookmark not defined.**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi dan komunikasi mengalami pertumbuhan yang sangat pesat dalam beberapa dekade terakhir, terutama di bidang kecerdasan buatan (AI). AI telah menjadi salah satu teknologi yang paling transformatif, dengan aplikasi yang semakin luas di berbagai sektor (Hostinger Tutorials, 2025). Kemampuan AI untuk memproses data secara cepat, memahami bahasa alami, dan memberikan solusi otomatis telah membuka peluang baru dalam meningkatkan efisiensi dan kualitas layanan. Salah satu bentuk implementasi AI yang sedang naik daun adalah AI model generatif dari bidang *Natural Language Processing*, yang mampu berinteraksi dengan pengguna secara *real-time* untuk memberikan informasi atau menyelesaikan masalah. Perkembangan ini mendorong banyak institusi, termasuk perguruan tinggi, untuk memanfaatkan teknologi chatbot guna meningkatkan layanan mereka (Isakova, 2024).

Di lingkungan perguruan tinggi, seperti Universitas Machung, proses penyampaian informasi kepada mahasiswa selain secara tatap muka, juga bisa melalui surel dan chat. Di mana, yang bisa disampaikan oleh staf sekretariat atau bagian admisi tergantung badan yang dituju. Namun, metode ini memiliki beberapa kelemahan, terutama dalam hal kapasitas sumber daya manusia. Jumlah mahasiswa yang bertanya tidak diimbangi dengan jumlah staf yang memadai. Staf yang tersedia juga memiliki tugas ganda, sehingga tidak mudah untuk fokus sepenuhnya dalam menjawab pertanyaan semua mahasiswa, terutama yang berkaitan dengan panduan akademik seperti "*Student Guide*". Padahal, informasi dalam panduan tersebut bersifat wajib dan harus dipahami oleh seluruh mahasiswa. Hal ini menimbulkan tantangan dalam memastikan bahwa setiap mahasiswa mendapatkan informasi yang akurat dan tepat waktu. Oleh karena itu, diperlukan solusi yang dapat mengotomatisasi proses penyampaian informasi ini.

WhatsApp, sebagai salah satu platform pesan instan paling populer di dunia, telah menjadi sarana komunikasi utama bagi banyak orang, termasuk mahasiswa dan staf di Universitas Machung. Menurut data terbaru, WhatsApp memiliki lebih

dari 2 miliar pengguna aktif di seluruh dunia (Statista, 2025). Keunggulan WhatsApp terletak pada kemudahan penggunaannya, fitur grup, dan kemampuan untuk mengirim berbagai jenis media. Baru-baru ini, Meta (perusahaan induk WhatsApp) telah mengintegrasikan fitur *Generative AI* ke dalam platform ini, membuka peluang baru untuk pengembangan layanan berbasis *chatbot*. Integrasi ini memungkinkan pengguna untuk berinteraksi dengan AI secara langsung melalui WhatsApp. Di mana, pengguna dapat mengobrol, mencari informasi, dan bahkan mengedit gambar pada Meta AI (Blog WhatsApp, 2024). Dengan memanfaatkan WhatsApp sebagai platform, diharapkan informasi dapat diakses dan dapat menjangkau lebih banyak mahasiswa secara efektif.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengimplementasikan *chatbot* OpenAI sebagai layanan *customer service* di WhatsApp. *Chatbot* ini diharapkan dapat membantu menjawab pertanyaan mahasiswa secara otomatis, khususnya yang berkaitan dengan informasi dalam buku "Student Guide". Dengan demikian, beban kerja staf dapat dikurangi, dan mahasiswa dapat memperoleh informasi yang dibutuhkan dengan lebih cepat dan akurat. Implementasi *chatbot* ini juga diharapkan dapat menjadi solusi inovatif untuk mengatasi keterbatasan metode konvensional.

Untuk mencapai tujuan tersebut, penelitian ini menggunakan metode *Retrieval Augmented Generation* (RAG) yang menggabungkan keunggulan *chatbot* generatif dan *chatbot* berbasis *retrieval*. Di mana, keterbatasan dari *chatbot* generatif yaitu, jumlah *input* yang terbatas, tidak efisien, dan berpotensi terjadinya fabrikasi jawaban. RAG memungkinkan *chatbot* untuk tidak hanya menghasilkan respons yang relevan secara kontekstual, tetapi juga mengambil informasi spesifik dari sumber data yang terstruktur secara efisien, seperti dari dokumen PDF (Lewis dkk., 2020). Pendekatan *hybrid* ini diharapkan dapat mengatasi keterbatasan *chatbot* generatif yang berpotensi menghasilkan respons kurang akurat atau tidak relevan. Sehingga dengan menggabungkan keunggulan dari *chatbot generative* dan *retrieval* ini, *chatbot* dapat memberikan jawaban dari dokumen eksternal secara presisi dan tetap sesuai dengan konteks pertanyaan.

Implementasi *chatbot* dalam penelitian ini melibatkan beberapa komponen utama, termasuk model OpenAI sebagai inti kecerdasan buatan, *framework*



LangChain untuk mengelola alur kerja RAG, dan integrasi dengan WhatsApp melalui API. LangChain dipilih karena kemampuannya dalam mengelola alur kerja yang kompleks dan menghubungkan berbagai komponen AI secara efisien. Evaluasi kinerja *chatbot* dilakukan melalui umpan balik pengguna dan metrik khusus yang dirancang untuk mengukur akurasi respons. Dengan demikian, penelitian ini tidak hanya fokus pada aspek teknis, tetapi juga pada pengalaman pengguna dan keakuratan *chatbot* dalam merespons pertanyaan di lapangan.

Diharapkan, implementasi *chatbot* ini dapat memberikan manfaat signifikan bagi Universitas Machung, seperti peningkatan efisiensi layanan, pengurangan beban kerja staf, dan peningkatan kepuasan mahasiswa. Hasil penelitian ini juga diharapkan dapat menjadi referensi bagi pengembangan layanan berbasis AI di institusi pendidikan lainnya. Dengan demikian, *chatbot* tidak hanya menjadi solusi sementara, tetapi juga langkah awal menuju transformasi digital yang lebih luas di lingkungan perguruan tinggi.

## **1.2 Identifikasi Masalah**

Tantangan masalah yang akan dihadapi pada penelitian ini adalah terinformasinya mahasiswa Universitas Ma Chung terkait informasi yang ada pada buku Student Guide. Namun, staf yang berjumlah terbatas dan memiliki tugas ganda, akan mengalami kesulitan dalam melayani semua pertanyaan mahasiswa terkait Student Guide. Di mana, penundaan dalam penyampaian informasi dan akses layanan yang kurang optimal, bisa berpotensi menurunkan kepuasan mahasiswa. Selain itu, adanya limitasi dari pemanfaatan *chatbot* generatif sebagai alternatif untuk mendapatkan informasi dari Student Guide seperti, berpotensi besar dalam memberikan jawaban yang salah (halusinasi), limit besaran *input*, kurang efisien, dan mahal.

## **1.3 Batasan Masalah**

Pengembangan aplikasi pada penelitian ini memiliki beberapa batasan, di antaranya yaitu sebagai berikut.

- a. Basis pengetahuan yang digunakan dalam penelitian ini terbatas pada dokumen PDF *Student Guide 2023*, yang diterbitkan oleh Universitas Ma Chung.

- b. Pembuatan teks dalam interaksi *chatbot* didukung oleh model GPT-4o yang disediakan oleh OpenAI.
- c. Proses *embedding* untuk pencarian semantik teks menggunakan model text-embedding-ada-002, yang juga merupakan produk dari OpenAI.
- d. Informasi yang diproses dan dianalisis oleh sistem *chatbot* bersumber dari berkas dengan format PDF.
- e. Sistem aplikasi yang dikembangkan tidak akan memproses konten berupa visual, seperti gambar, atau diagram dari dokumen PDF.
- f. Implementasi dalam pengembangan sistem *chatbot* akan memanfaatkan *framework* LangChain beserta dengan fasilitasnya yang telah disediakan.
- g. Penilaian terhadap kepuasan pengguna atas respons yang diberikan oleh sistem *chatbot* dilakukan melalui survei.
- h. Akurasi respons *chatbot* dalam memberikan informasi yang bersumber dari dokumen dievaluasi menggunakan metrik yang dirancang khusus.

#### **1.4 Rumusan Masalah**

Berdasarkan identifikasi masalah yang telah disampaikan, rumusan masalah dalam penelitian ini adalah bagaimana mengintegrasikan *chatbot* pada WhatsApp guna menggantikan staf Universitas Ma Chung dalam melayani pertanyaan-pertanyaan mahasiswa terkait *Student Guide*?

#### **1.5 Tujuan**

Adapun tujuan yang ingin dicapai dari rumusan masalah yang disampaikan pada penelitian ini adalah untuk menghasilkan aplikasi *chatbot* yang dapat menjawab selayaknya seperti staf Universitas Ma Chung dengan menggunakan metode *Retrieval Augmented Generation (RAG)* dalam merespons berdasarkan informasi dari buku *Student Guide*. Dengan pelayanannya yang juga dapat diakses melalui aplikasi WhatsApp.

#### **1.6 Manfaat**

Adapun Manfaat yang dapat diperoleh dalam pengerjaan tugas akhir ini adalah sebagai berikut.

- a. Bagi pembaca, manfaat yang didapatkan yaitu dapat memanfaatkan aplikasi pada tugas akhir ini untuk diterapkan pada organisasi, perusahaan, atau kebutuhan lainnya yang serupa. Serta dapat menjadi bahan referensi untuk belajar atau pengembangan lainnya yang berhubungan dengan topik tugas akhir ini.
- b. Bagi penulis, pengerjaan tugas akhir ini menjadi ajang memperdalam kemampuan dan memperluas ilmu dan wawasan terkait pemanfaatan AI.
- c. Bagi Universitas Ma Chung dan Prodi Teknik Informatika, tugas akhir ini bisa menjadi tambahan satu lagi koleksi kepustakaan. Serta dapat menjadi bekal untuk mempersiapkan mahasiswanya untuk dapat terjun di dunia masyarakat.

### **1.7 Luaran Penelitian**

Luaran penelitian ini terdiri dari dua luaran utama, yaitu:

1. Sebuah chatbot yang diakses melalui aplikasi WhatsApp yang menyediakan informasi berdasarkan dokumen *Student Guide 2023* dalam format PDF.
2. Aplikasi untuk mengelola dokumen yang digunakan sebagai sumber informasi bagi *chatbot* dalam memberikan respons.
3. Publikasi ilmiah yang mendokumentasikan proses dan hasil penelitian dalam bentuk jurnal.

### **1.8 Sistematika Penulisan**

Laporan penelitian ini memiliki sistematika penulisan yang di mana dibagi menjadi 5 bab, yaitu sebagai berikut.

1. Bab 1 Pendahuluan

Pada bab pendahuluan terdapat pembahasan mengenai latar belakang tugas akhir, identifikasi masalah, batasan masalah, rumusan masalah, tujuan akhir, manfaat, luaran penelitian, dan sistematika penulisan.

2. Bab 2 Tinjauan Pustaka

Bab tinjauan pustaka berisikan mengenai uraian landasan teori dan alat-alat yang digunakan. Di antaranya seperti, OpenAI, Python, LangChain, Flask, WhatsApp, dan WA Automate Socket Client.

3. Bab 3 Analisis dan Perancangan Sistem

Penjelasan mengenai sistematika perancangan sistem, metode, dan cara pengujian dapat ditemui pada bab ini.

4. Bab 4 Hasil dan Pembahasan

Pembahasan hasil dari proyek yang telah dikerjakan akan dipaparkan secara detail pada bab 4. Pengujian untuk mengukur keberhasilan aplikasi juga ada pada bab ini.

5. Bab 5 Kesimpulan dan Saran

Bab 5 berisikan kesimpulan dari keseluruhan pengerjaan tugas akhir. Serta penjelasan mengenai apa saja yang dapat dikembangkan selanjutnya. Sehingga penelitian ini dapat dilanjutkan atau ditingkatkan oleh peneliti yang lain.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Web Service

*Web service* merupakan suatu perangkat lunak yang memiliki kemampuan untuk berinteraksi antar sistem melalui internet maupun intranet. Layanan web atau *Web service* bisa ditulis dengan bahasa pemrograman yang bermacam-macam, namun poin utamanya tetap bisa berkomunikasi atau bertransaksi informasi antar satu dengan yang lainnya (Cleo, 2024). *Web service* sendiri memiliki beberapa jenis yang berbeda-beda. Di mana dua jenis *web service* yang paling ditemui adalah sebagai berikut.

- a. REST, yang memberikan komunikasi dan konektivitas antar perangkat dan sebagian besar menggunakan HTTP sebagai protokol komunikasi.
- b. SOAP (*Simple Object Access Protocol*), adalah *web service* yang menggunakan XML (*Extensible Markup Language*) untuk bertransaksi informasi melalui HTTP atau SMTP (*Simple Mail Transfer Protocol*).

#### 2.2 REST API

*Application Programming Interface* atau yang sering dikenal dengan API adalah sebuah peraturan yang harus dipatuhi oleh aplikasi agar dapat saling berkomunikasi. Maka dari itu, pengembang aplikasi akan membuat sebuah API agar aplikasi lain dapat berinteraksi dengan aplikasi yang sedang dikembangkan. API bisa digambarkan sebagai penghubung atau jembatan antar aplikasi. Sedangkan *Representational State Transfer* atau REST merupakan arsitektur perangkat lunak yang mengatur API untuk bekerja dalam kondisi tertentu. *Web service* yang menerapkan arsitektur REST bisa disebut dengan RESTful web API. Sedangkan API menerapkan arsitektur REST juga sering disebut juga REST API.

Implementasi REST API juga dapat ditemui dengan pemanfaatan *Hypertext Transfer Protocol* (HTTP) dalam berkomunikasi. Sedangkan HTTP sendiri memiliki beberapa *method* untuk menginformasikan ke server dalam memilih perilaku yang diberikan (AWS, 2024). Di mana *method* yang paling umum digunakan yaitu, *GET*, *POST*, *PUT*, dan *DELETE*. Lalu, server juga akan

memberikan respon kepada *client*. Respon server yang menerapkan REST API akan memiliki 3 komponen utama. Komponen-komponen tersebut salah satunya adalah kode status. Contoh kode status yang diberikan ketika berhasil adalah 200. Lalu kode 404 yang berarti tidak ditemukan. Selain itu, komponen yang lainnya adalah *body* yang merupakan isi dari respons dan *header* yang biasanya berisi kumpulan *metadata* dari respons.

### 2.3 Prompt Engineering

*Prompt engineering* merupakan sebuah proses merekonstruksi kata-kata agar dapat dipahami dan diinterpretasikan oleh model (Diab dkk., 2022). Proses ini merupakan tahap penting dalam meningkatkan kualitas tanggapan yang dapat diberikan oleh model. Di mana, kata-kata yang telah dibuat tersebut bisa disebut dengan *prompt*. Dengan *prompt* yang telah dibuat tersebut akan menjadi sebuah deskripsi mengenai tugas yang akan dilakukan oleh AI. Sebuah *prompt* yang dibuat untuk model pembuatan teks, bisa menyerupai berbagai hal. Contohnya bisa berupa pertanyaan, perintah, atau pernyataan yang lebih panjang seperti menambahkan konteks, kumpulan instruksi, dan bahkan dapat menyertakan sejarah percakapan yang telah dilakukan.

Dengan adanya proses pembuatan *prompt* ini, *Large Language Model* (LLM) mendapatkan kemampuan untuk menyelesaikan tugas yang diberikan tanpa pelatihan lebih lanjut terhadap model (Wei dkk., 2022). Kemampuan yang dimiliki oleh LLM tersebut bisa disebut dengan *in-context learning*. *Prompt* yang dibuat sendiri tidak hanya terpaku oleh model generasi teks atau *text-to-text* saja. Proses pembuatan *prompt* dapat diberikan kepada berbagai macam LLM seperti model untuk generasi gambar (*text-to-image*) maupun video (*text-to-video*). Terdapat berbagai macam metode atau teknik yang dapat diterapkan untuk mendapatkan kualitas respon yang baik. Di mana, salah satu metode untuk menerapkan *prompt* kepada LLM yang digunakan dalam penelitian ini adalah *Retrieval Augmented Generation*.

## 2.4 OpenAI API

Didirikan pada Desember 2015 dengan misi utamanya adalah memastikan bahwa *Artificial General Intelligence* (AGI) dapat memberikan manfaat bagi seluruh umat manusia. OpenAI terkenal atas penciptaan model kecerdasan buatan canggih seperti seri GPT (*Generative Pre-trained Transformer*), yang telah merevolusi pemahaman dan generasi bahasa alami. Di antara berbagai macam contoh, ChatGPT dapat membuat lirik, menulis puisi, cerita fiksi, atau bahkan mencari *bug* pada kode pemrograman. Lalu, OpenAI GPT-4o yang juga memiliki semua kemampuan tersebut juga dapat diakses oleh publik untuk digunakan. Sehingga tidak heran jika ChatGPT menjadi salah satu aplikasi yang memiliki pertumbuhan pengguna tercepat setelah hari pertama peluncurannya (Megan Cerullo, 2023).

OpenAI secara aktif mempromosikan aksesibilitas terhadap model-model AI dan teknologi mereka, menjadikannya tersedia melalui API untuk pengembang, peneliti, dan bisnis yang ingin memanfaatkan kemampuan AI mereka dalam berbagai aplikasi. Di mana aksesibilitas ini yang akan menjadi fokus penting untuk memanfaatkan OpenAI pada penelitian ini. Dengan demikian, model yang telah disediakan oleh OpenAI dapat diintegrasikan dengan LangChain untuk pemanfaatan dengan tambahan fitur yang bisa diatur lebih lanjut. Penggunaan OpenAI API memberikan batasan terhadap pengaksesan model yang telah disediakan atau sebagai limitasi yang bisa disebut sebagai *rate limit*.

## 2.5 Token

Large Language Model akan membaca atau membuat teks per bagian yang disebut sebagai token. Bagian yang dimaksud bisa berbeda-beda. Contohnya seperti “ChatGPT is great!” yang akan berubah menjadi 6 bagian atau 6 token. Di mana, pembagian 6 token tersebut adalah untuk “chat”, “G”, “PT”, “is”, “great”, dan “!”. Jumlah token dapat bervariasi tergantung bahasa yang digunakan. Dalam bahasa tertentu terkadang sebuah kata bisa memiliki token lebih pendek dari sebuah karakter atau lebih panjang dari sebuah kata itu sendiri. Namun, berdasarkan (Advanced usage - OpenAI API, 2024) secara umum untuk teks berbahasa inggris, jumlah 1 token dapat mewakili 4 karakter atau 75% dari sebuah kata.

Jumlah token yang diproses akan memengaruhi beberapa hal. Aspek biaya merupakan salah satu pertimbangan utama, karena biaya penggunaan API dihitung berdasarkan jumlah token. Selain itu, waktu pemrosesan juga dipengaruhi oleh jumlah token. Contohnya seperti permintaan dengan token yang lebih banyak umumnya membutuhkan waktu yang lebih lama untuk diproses. Lalu, setiap model dari OpenAI akan selalu memiliki batasan jumlah token per permintaan. Misalnya, model gpt-3.5-turbo memiliki batasan 4097 token. Batasan ini diterapkan untuk mengurangi ketegangan infrastruktur. Permintaan yang sangat besar dan mendadak dapat membebani server secara signifikan, yang berpotensi mengakibatkan masalah kinerja seperti latensi yang tinggi atau bahkan kegagalan layanan.

## **2.6 Semantik**

Semantik didefinisikan sebagai kajian tentang makna yang disampaikan melalui bahasa, yang mencakup dari bagaimana kata, frasa, atau kalimat dalam memberikan makna. Kajian tersebut menelaah hubungan antara penanda (seperti kata dan simbol) dengan apa yang direpresentasikannya dalam realitas (Bagha, 2011). Hal tersebut meliputi berbagai teori makna, sebagai berikut.

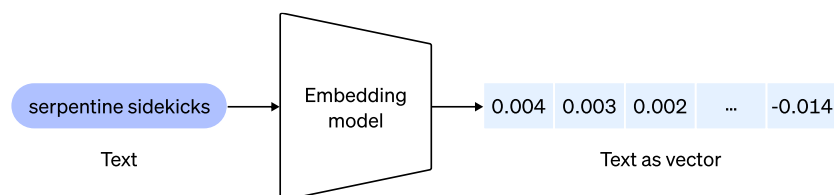
- Makna leksikal mengkaji makna yang melekat pada setiap kata (leksem) sebagaimana yang tercantum dalam kamus. Makna ini mencakup dua dimensi utama yaitu, denotasi (makna literal atau sebenarnya) dan konotasi (makna yang terkait dengan asosiasi, emosi, atau nilai budaya). Denotasi contohnya adalah seperti kata ‘mawar’ yang berarti sebuah nama dari bunga. Sedangkan konotasi dari kata ‘mawar’ bisa berarti romantis atau ekspresi cinta.
- Makna kontekstual merupakan interpretasi makna dalam semantik sangat dipengaruhi oleh konteks penggunaan bahasa. Faktor-faktor situasional, seperti konteks wacana, situasi sosial, dan pengetahuan bersama antara pembicara dan pendengar, dapat memodifikasi atau bahkan mengubah makna sebuah kata atau ujaran. Contohnya bisa seperti ‘suasana politik sedang panas’. Di mana panas memiliki makna situasi sedang tegang atau penuh konflik.

- Ambiguitas adalah kemampuan sebuah kata atau frasa untuk memiliki lebih dari satu interpretasi. Keberadaan ambiguitas ini dapat memengaruhi efektivitas komunikasi dan memerlukan pemahaman konteks yang mendalam untuk resolusinya. Contoh dari ambiguitas adalah contohnya ‘dia melihat orang dengan teropong’. Di mana, contoh tersebut memiliki dua makna. Arti pertama yaitu, dia menggunakan teropong untuk melihat orang. Atau orang yang dia lihat menggunakan teropong.

Dalam konteks pembuatan *embedding*, istilah semantik merujuk pada representasi makna dalam format numerik. Pemberian kode dengan semantik ini krusial bagi model *machine learning* untuk memahami dan memproses bahasa alami secara efektif. *Embedding* semantik dirancang untuk menangkap makna kata dengan menempatkannya dalam ruang vektor berdimensi tinggi, di mana kata-kata yang mirip maknanya terletak berdekatan. Pendekatan ini didasarkan pada Hipotesis Distribusional, yang menyatakan bahwa kata-kata dengan makna serupa cenderung muncul dalam konteks yang serupa. Oleh karena itu, *embedding* memungkinkan model untuk merefleksikan hubungan yang bernuansa antar kata, menjadikannya esensial untuk berbagai aplikasi Natural Language Processing (NLP).

## 2.7 Embedding

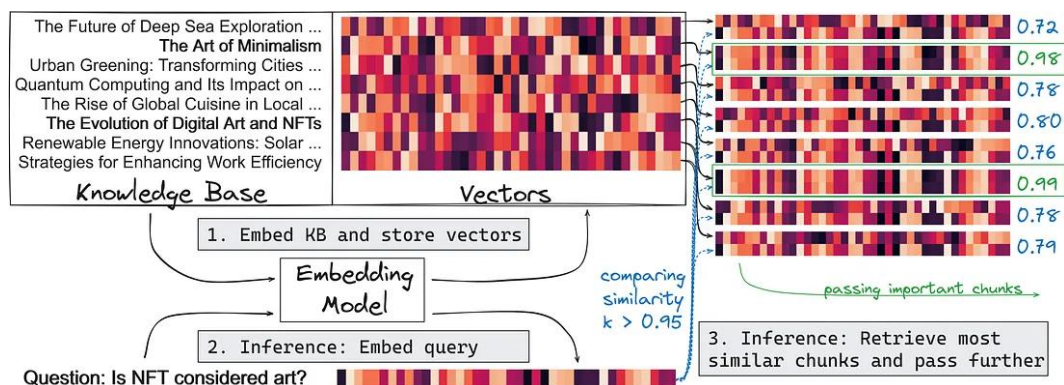
*Embedding* berperan sebagai komponen fundamental dalam *Retrieval Augmented Generation* (RAG) dengan mengkodekan dokumen atau data ke dalam ruang vektor semantik yang sama. Proses ini memungkinkan pencarian konten berdasarkan makna (semantik) yang lebih efektif dibandingkan pencarian berbasis kata kunci. Yang mana, akan meningkatkan kinerja *retrieval* dalam mendapatkan konten dengan makna terdekat. (*Embedding Techniques for Retrieval-Augmented Generation* | *Restackio*, t.t.).



Gambar 2.1 Representasi Data Sebagai Vektor

Sumber : <https://openai.com/index/new-embedding-models-and-api-updates/>

Representasi vektor yang dihasilkan oleh *embedding* memetakan fitur-fitur semantik dari data, di mana setiap dimensi vektor merepresentasikan aspek tertentu dari makna data tersebut. Hal ini memungkinkan sistem untuk mengukur kemiripan semantik antar data berdasarkan jarak antar vektor dalam ruang tersebut. Misalnya bisa dilihat pada Gambar 2.2, vektor yang berdekatan dalam ruang vektor semantik menunjukkan data yang memiliki makna yang serupa. Representasi data dalam bentuk vektor numerik memfasilitasi pencarian informasi yang relevan berdasarkan kemiripan semantik. Dalam ruang vektor berdimensi- $n$ , di mana  $n$  merepresentasikan jumlah fitur atau aspek yang diwakili, kedekatan antar vektor mencerminkan kemiripan makna antar data. Semakin dekat letak objek-objek tersebut dalam ruang vektor, semakin tinggi pula kemiripan semantik di antara data yang direpresentasikannya.



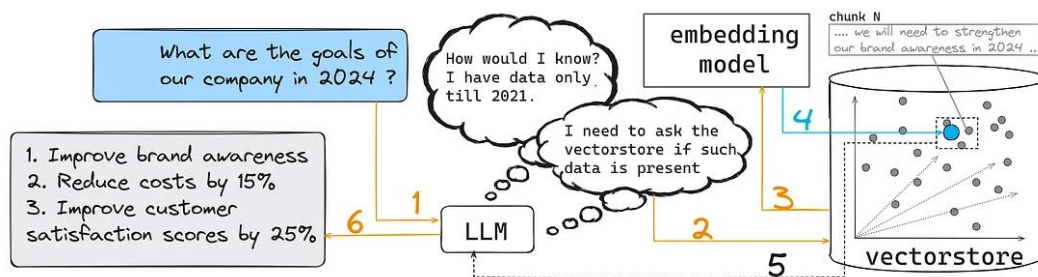
Gambar 2.2 Pemanfaatan Model Embedding Dalam RAG

Sumber : <https://pub.towardsai.net/langchain-101-part-3b-talking-to-documents-embeddings-and-vectorstores-c37d460f1519>

Proses representasi data menjadi vektor bisa dilakukan dengan memanfaatkan model *embedding* dari OpenAI. Salah satu model yang dapat mengubah kumpulan kata menjadi kumpulan vektor adalah text-embedding-3-small dari OpenAI. Kumpulan vektor atau yang bisa disebut dengan *embeddings*, akan disimpan pada *vector store* atau *vector database* sebagai hasil dari akhir dari proses pembuatan *embeddings*.

## 2.8 Vectorstore

*Vectorstore* merupakan tempat penyimpanan khusus yang dirancang untuk menyimpan, mengelola, dan mengambil *embedding* vektor berdimensi tinggi yang dihasilkan oleh model *machine learning* secara efisien (*Top Vector Stores: 9 Use Cases You Should Know*, 2024). Dengan data yang direpresentasikan berupa vektor, memungkinkan untuk dilakukannya teknik analisis lanjutan seperti *similarity search*, pengelompokan, dan klasifikasi. Kemampuan tersebut yang membuat *vectorstore* sangat berharga dalam aplikasi yang memanfaatkan NLP.



Gambar 2.3 Contoh Pemanfaatan Vectorstore dalam Alur RAG dengan LLM

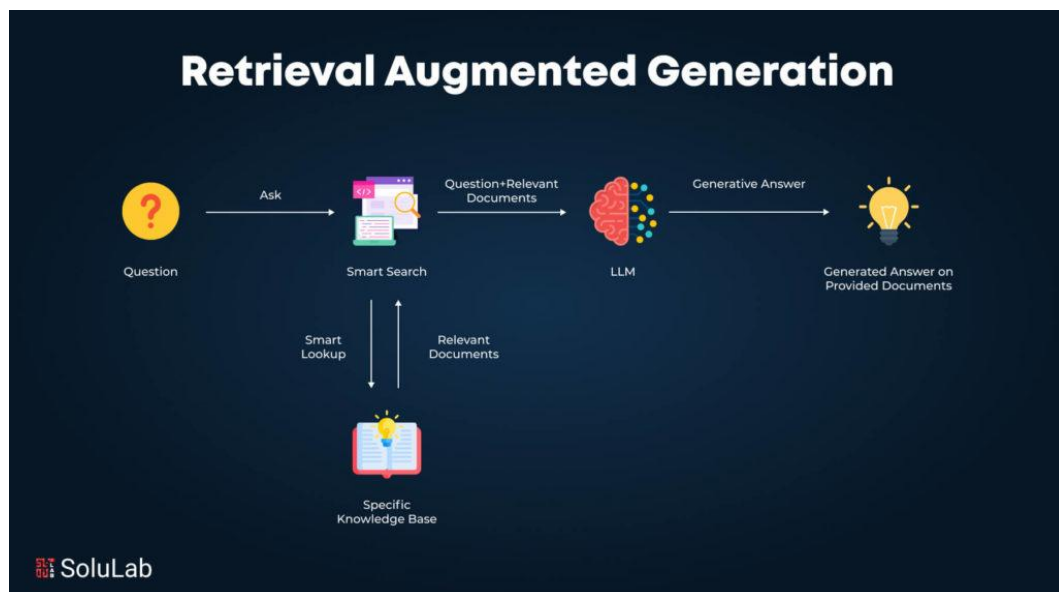
Sumber : <https://pub.towardsai.net/langchain-101-part-3b-talking-to-documents-embeddings-and-vectorstores-c37d460f1519>

Contohnya seperti memberikan kemampuan untuk mengurangi vektor, menambah vektor, atau bahkan menambahkan *metadata*. *Metadata* bisa berisikan macam-macam seperti nomor halaman atau nama sumber dokumen. Hal tersebut memungkinkan karena *vectorstore* memberikan indeks pada setiap vektor sebagai referensi sumber data berasal. Pencarian tetangga terdekat juga dilakukan oleh *vectorstore* yang dibantu dengan model *embedding* (Reznikov Ivan, 2024). Di mana, pencarian kedekatan data dengan pertanyaan pengguna bisa menggunakan algoritma seperti Euclidean distance atau Cosine similarity.

## 2.9 Retrieval Augmented Generation

*Retrieval Augmented Generation* (RAG) merupakan proses optimasi hasil keluaran dari LLM dengan pengetahuan di luar dari yang telah dilatih kepada model. LLM atau *Large Language Models* sendiri merupakan model komputasi yang dilatih dengan sejumlah data yang besar dan miliaran parameter agar memiliki

kemampuan dasar untuk menyelesaikan berbagai macam tugas seperti tanya jawab, mengartikan bahasa, atau membuat puisi. RAG memanfaatkan kemampuan LLM dengan memberikan pengetahuan dari luar yang lebih spesifik atau lebih terspesialisasi tanpa harus melatih ulang model. Pendekatan ini lebih efektif dalam meningkatkan kualitas keluaran yang diberikan LLM dan dapat berguna dalam berbagai macam kasus. Hal tersebut dikarenakan akan fleksibilitas yang didapatkan oleh pengguna LLM dalam pengembangan sistem berdasarkan LLM.



Gambar 2.4 Diagram Alur RAG pada LLM

Sumber : <https://www.solulab.com/what-is-retrieval-augmented-generation/>

Pada Gambar 2.4, terdapat beberapa proses yang dapat ditemui di RAG. Hal pertama dalam RAG adalah ketersediaannya sumber pengetahuan eksternal sebagai data. Data tersebut bisa tersimpan sebagai dokumen maupun *database*. Lalu, membuat proses untuk mencari informasi yang relevan dari masukkan yang diberikan. Proses ini biasanya berkaitan erat dengan pemanfaatan *vectorstore* untuk mendapatkan pecahan informasi yang relevan. Selanjutnya adalah memberikan *prompt* untuk mendapatkan keluaran yang diinginkan. Sehingga setiap kali pengguna mendapatkan pecahan dokumen yang relevan, LLM akan mengubah pecahan dokumen tersebut menjadi keluaran yang sesuai dengan *prompt* yang diberikan.



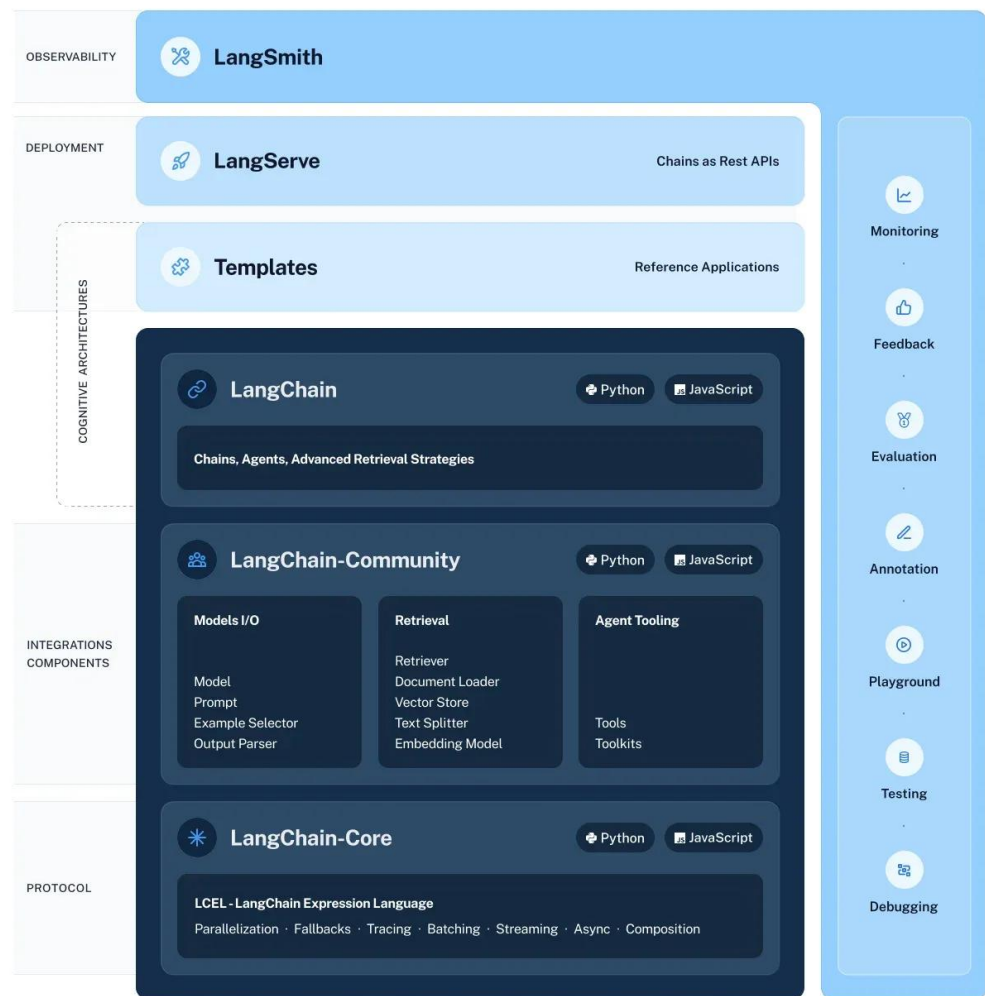
## 2.10 Python

Bahasa pemrograman Python terkenal dengan bahasanya yang mudah dibaca, sederhana, dan fleksibel. Salah satu yang menyebabkan mudah dibaca dan sederhananya adalah perintah yang kompleks bisa diringkas dengan hanya beberapa baris kode saja. Sedangkan untuk mendefinisikan sebuah blok kode, Python menggunakan *indent* atau spasi putih. Berbeda halnya dengan bahasa pemrograman seperti Java yang harus ditutup oleh titik koma. Menyebabkan Python tidak memiliki tanda baca yang berlebih pada setiap blok kodenya dan memudahkan pengembang untuk membaca kembali atau melakukan *debugging*.

Berkat kinerja yang cepat, Python sering kali digunakan untuk melakukan komputasi ilmiah. Hal tersebut juga didukung dengan sifatnya yang gratis, *multi-platform*, dan *library* yang luas membuat Python sangat sering digunakan untuk analisis atau manipulasi data, simulasi, pemodelan, dan banyak lagi. Di mana *library* utama yang sering digunakan untuk melakukan komputasi ilmiah di antaranya yaitu, NumPy, SciPy, dan Matplotlib. Meskipun demikian, Python juga dapat digunakan untuk melakukan pengembangan *website*. Di mana terdapat banyak sekali pilihan *framework* atau *library* yang bisa digunakan dalam pengembangan *website*. Contohnya seperti Flask, Django, Pyramid, dan masih banyak lagi. Maka dari sebab itu, bahasa pemrograman ini menjadi pilihan dasar pengembangan aplikasi pada penelitian ini.

## 2.11 LangChain

LangChain adalah sebuah *framework* yang didesain untuk mempermudah pengembangan aplikasi yang memanfaatkan *Large Language Model* (LLM). LangChain sendiri dirilis pada Oktober 2023 sebagai proyek *open source* oleh Harrison Chase. Sehingga dapat digunakan secara gratis oleh publik. Fitur utama yang menjadi titik pertimbangan penggunaan LangChain adalah pengembang dapat melakukan abstraksi terhadap model bahasa yang akan digunakan. Hal tersebut dimudahkan juga dengan komponen-komponen yang dapat digunakan secara modular. *Framework* ini juga dapat diintegrasikan dengan berbagai macam model. Contohnya seperti, Cohere, Hugging Face Hub, GPT4, dan masih banyak lagi.



Gambar 2.5 LangChain Framework Overview




Sumber: <https://bottopenguin.com/blogs/langchain-what-how-why-and-must-knows>

Arsitektur LangChain didasarkan pada konsep "Chain" yang memfasilitasi penggabungan berbagai komponen untuk membentuk alur sistem aplikasi. Framework ini menyediakan komponen-komponen inti yang dapat dikonfigurasi, meliputi *Prompts*, *Memory*, *Indexes*, dan *Chains*. Komponen *Memory* memungkinkan aplikasi untuk mempertahankan dan mengakses riwayat percakapan, sementara komponen *Indexes* mendukung integrasi dokumen eksternal sebagai konteks tambahan dalam pemrosesan bahasa alami. Sehingga dengan modularitas ini yang dapat memberikan fleksibilitas bagi peneliti untuk membangun aplikasi kompleks seperti *chatbot*, sistem tanya jawab, dan sistem

pemrosesan dokumen secara lebih efisien dibandingkan pengembangan sendiri dari awal.

## 2.12 WhatsApp Business Platform

WhatsApp Business Platform merupakan layanan yang dikembangkan oleh Meta (sebelumnya Facebook) untuk memfasilitasi komunikasi bisnis melalui WhatsApp secara terstruktur dan terotomatisasi. Platform ini menyediakan seperangkat alat dan antarmuka pemrograman aplikasi (API) yang memungkinkan bisnis untuk berinteraksi dengan pelanggan mereka melalui WhatsApp secara terprogram.

WhatsApp vs WhatsApp Business		
		
WhatsApp	WhatsApp Business	WhatsApp Business Platform/API
For individual users and personal conversations 🧑🧑	General business communication for SMBs 🧑🧑🧑🧑	Advanced solution for enterprise-sized companies and call centers 🏢📞
End-to-end encryption	End-to-end encryption	End-to-end encryption
Basic personal profile info (name, photo)	Business profiles (website, location, business hours)	Advanced chatbots
No automatic messages	Quick replies, away messages, and automatic greetings	Access to all automation features via the API (requires coding) or 3rd party vendors
No message analytics	Detailed message and response rate statistics	
Unlimited messages 🗨️	Unlimited messages 🗨️	1,000 free conversations per month 🗨️
Free ✅	Free ✅	Some services require payments 💰

Gambar 2.6 Perbedaan Antar Aplikasi WhatsApp

Sumber: <https://www.tidio.com/blog/whatsapp-business/>

Berbeda dengan aplikasi WhatsApp Business standar, Platform WhatsApp Business menawarkan kemampuan yang lebih luas dalam pengelolaan komunikasi massal, integrasi sistem, dan automasi pesan, yang sangat penting untuk operasi bisnis skala menengah hingga besar. WhatsApp Business Platform terdiri dari tiga API utama yang masing-masing memiliki fungsi dan karakteristik berbeda dalam mendukung operasional bisnis:

- Cloud API

Cloud API merupakan solusi yang dihosting langsung oleh Meta, menawarkan cara tercepat dan paling mudah untuk memulai integrasi WhatsApp Business. Layanan ini menghilangkan kebutuhan untuk mengelola infrastruktur server sendiri, memungkinkan bisnis fokus pada pengembangan fitur dan layanan pelanggan. Cloud API mendukung berbagai fitur seperti pengiriman pesan otomatis, manajemen template, dan integrasi webhook untuk notifikasi real-time.

- On-Premises API

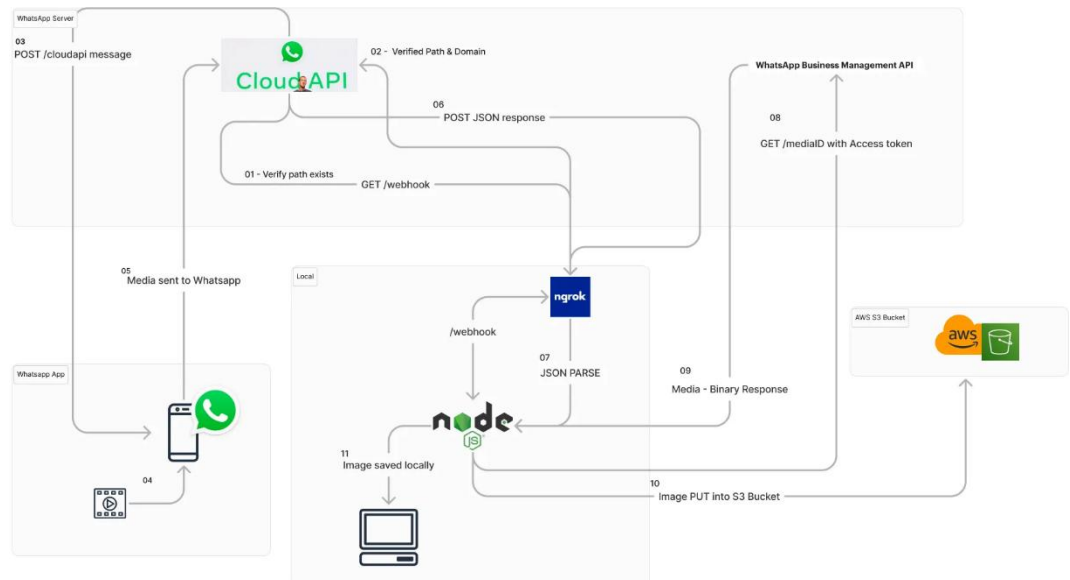
On-Premises API memberikan kontrol lebih besar kepada bisnis dengan memungkinkan hosting infrastruktur di server mereka sendiri. Solusi ini ideal untuk organisasi yang memiliki kebutuhan keamanan khusus atau regulasi data yang ketat. Meskipun memerlukan upaya lebih dalam hal pemeliharaan dan pengelolaan infrastruktur, On-Premises API menawarkan fleksibilitas maksimal dalam konfigurasi dan integrasi dengan sistem internal.

- Business Management API

Business Management API difokuskan pada pengelolaan aspek administratif dari WhatsApp Business, termasuk manajemen akun, pengaturan profil bisnis, dan pengelolaan kredensi API. API ini memungkinkan otomatisasi tugas-tugas administratif dan integrasi dengan sistem manajemen bisnis yang ada.

### 2.12.1 WhatsApp Webhook

Webhook untuk WhatsApp Business Platform adalah mekanisme komunikasi asinkron yang krusial bagi bisnis untuk menerima notifikasi *real-time* mengenai berbagai peristiwa yang terjadi dalam ekosistem WhatsApp. Mekanisme ini beroperasi berdasarkan prinsip *push notification*, di mana WhatsApp Platform secara otomatis mengirimkan data ke *endpoint* yang telah dikonfigurasi oleh bisnis setiap kali terjadi peristiwa yang relevan, seperti pesan masuk, status pengiriman pesan, atau perubahan status pengguna. Implementasi *webhook* ini esensial dalam pengembangan aplikasi yang responsif dan interaktif. Memungkinkan bisnis memberikan pengalaman pelanggan yang lebih baik melalui respons yang cepat, tepat, serta alur kerja dapat terotomatisasi.



Gambar 2.7 Contoh Alur Implementasi WhatsApp API Cloud Webhook

Sumber: <https://medium.com/@shreyas.sreedhar/downloading-media-using-whatsapps-cloud-api-webhooks-and-uploading-it-to-aws-s3-bucket-via-nodejs-07c5cbae896f>

Dari segi arsitektur teknis, *webhook* WhatsApp memanfaatkan protokol HTTPS untuk transmisi data terenkripsi dalam format JSON ke *endpoint* yang telah dikonfigurasi oleh bisnis. Setiap notifikasi *webhook* berisi informasi spesifik terkait peristiwa yang terjadi, termasuk metadata seperti *timestamp*, ID pesan, dan informasi pengirim. Integritas dan autentisitas komunikasi dijamin melalui mekanisme verifikasi tanda tangan *webhook* menggunakan kunci rahasia yang

dibagikan antara WhatsApp Platform dan server bisnis. Mekanisme ini memastikan bahwa setiap notifikasi yang diterima benar-benar berasal dari WhatsApp dan belum dimodifikasi, sehingga meminimalisir risiko manipulasi data.

WhatsApp Business Platform mengategorikan notifikasi *webhook* ke dalam beberapa tipe peristiwa utama, yang masing-masing membawa *payload* data yang berbeda. Kategori-kategori ini mencakup penerimaan pesan masuk (*incoming messages*), pembaruan status pengiriman pesan (*message status updates*), notifikasi perubahan status akun bisnis (*business account updates*), dan interaksi dengan templat pesan (*template interactions*). Setiap tipe peristiwa ini menyediakan informasi spesifik yang memungkinkan bisnis untuk memproses dan merespons notifikasi secara tepat sesuai dengan kebutuhan aplikasi mereka. Sebagai contoh, notifikasi *incoming messages* akan berisi teks pesan, pengirim, dan *timestamp*, sedangkan notifikasi *message status updates* akan memberikan informasi status pengiriman pesan (terkirim, diterima, dibaca). Dengan demikian, *webhook* memungkinkan integrasi yang mendalam dan responsif terhadap berbagai interaksi pengguna di WhatsApp.

#### Cloud API

For Cloud API, the object is always `whatsapp_business_account` but the `field` will be indicative of the type of information being sent.

#### On-Premises API

For On-Premises API, because your client is sending the Webhook instead of Meta, the object will be `contacts` and `messages`, `errors`, or `statuses` and `pricing`.

```
{
  "object": "whatsapp_business_account",
  "entry": [{
    "id": "WHATSAPP_BUSINESS_ACCOUNT_ID",
    "changes": [{
      "value": {
        "messaging_product": "whatsapp",
        "metadata": {
          "display_phone_number": PHONE_NUMBER,
          "phone_number_id": PHONE_NUMBER_ID
        },
        "contacts": [{
          "profile": {
            "name": "NAME"
          },
          "wa_id": PHONE_NUMBER
        }],
        "messages": [{
          "from": PHONE_NUMBER,
          "id": "wamid.ID",
          "timestamp": TIMESTAMP,
          "text": {
            "body": "MESSAGE_BODY"
          },
          "type": "text"
        }]
      },
      "field": "messages"
    }]
  }]
}
```

```
{
  "contacts": [{
    "profile": {
      "name": "NAME"
    },
    "wa_id": "WHATSAPP_BUSINESS_ACCOUNT_ID"
  }],
  "messages": [{
    "from": "PHONE_NUMBER",
    "id": "wamid.ID",
    "timestamp": "TIMESTAMP",
    "text": {
      "body": "MESSAGE_BODY"
    },
    "type": "text"
  }]
}
```

Gambar 2.8 Contoh Notifikasi Pesan Masuk

Sumber: <https://developers.facebook.com/docs/whatsapp/webhooks/>

WhatsApp Business Platform mengimplementasikan mekanisme *webhook* dengan fokus pada keandalan dan skalabilitas. Sistem *retry* otomatis dengan interval dan jumlah percobaan yang dapat dikonfigurasi memastikan pengiriman notifikasi meskipun terjadi gangguan sementara. Untuk menangani volume notifikasi yang tinggi, platform menerapkan batasan laju (*rate limiting*) dan *throttling* yang dapat dikonfigurasi, mencegah *overload* pada server bisnis dan menjaga latensi tetap rendah. Fitur monitoring dan *logging* juga tersedia, sehingga analisis performa *webhook* dan identifikasi potensi masalah dapat dilakukan.

### 2.13 Flask

Flask merupakan *framework* minimalis berbasis bahasa pemrograman python untuk pengembangan aplikasi website. *Framework* ini terkenal dengan keunggulannya yaitu ringan dan cocok untuk proyek aplikasi dengan skala kecil. Di mana, cocok dengan tujuan penelitian ini. Namun, dengan keringanannya tersebut, flask tidak memiliki berbagai macam fitur bawaan layaknya *framework full-stack* pada umumnya. Fitur bawaan yang tidak dimiliki flask seperti *authentication*, akun, perijinan akses, dan formulir validasi. Sehingga *library* eksternal dibutuhkan dalam mengimplementasikan fitur tersebut.

### 2.14 Penelitian Terdahulu

Sebelumnya telah ada penelitian yang berhubungan dengan pemanfaatan OpenAI oleh (NR Tejaswini dkk., 2023). Penelitian ini memanfaatkan OpenAI LLM dengan menggunakan LangChain untuk mencari data yang dibutuhkan pada dokumen PDF. Metode yang dilakukan adalah dengan memecah dokumen PDF menjadi beberapa bagian sebagai bahan yang nanti akan dipakai oleh LLM sebagai landasan pengetahuan awal. Sehingga penelitian menjadikan OpenAI dan hasil pemecahan dokumen PDF sebagai *back-end* aplikasi. Sedangkan untuk *Front-end* menggunakan Streamlit. Hasil dari penelitian ini pengguna dapat mengekstrak pengetahuan yang ada pada dokumen PDF dengan hasil yang sesuai dari dokumen aslinya. Serta aplikasi yang dibuat juga akan memberikan konfirmasi tidak tahu apabila mencari data yang tidak ada pada dokumen.

Pengembangan aplikasi terhadap penggunaan AI dengan dokumen PDF juga telah disediakan oleh aplikasi bernama (PDF.ai, 2023.). Aplikasi PDF.ai juga merupakan aplikasi untuk mencari data pada dokumen PDF dengan sistem *chat*. Tanggapan yang diberikan oleh AI pada aplikasi ini juga hampir sama dengan ChatGPT pada umumnya. Namun, jawaban yang diberikan juga disertai tombol untuk mengarahkan pengguna ke halaman yang sedang ditanyakan. Sehingga selain membaca tanggapan dari AI, pengguna juga bisa membaca sumbernya langsung yang digunakan AI untuk menjawab pertanyaan.

Penelitian yang dilakukan oleh (Alia dkk., 2024), berjudul "Implementation Open Artificial Intelligence ChattGPT Integrated With Whatsapp Bot,"



mengeksplorasi integrasi OpenAI dengan WhatsApp untuk mengembangkan *chatbot* yang mampu memberikan informasi kesehatan *real-time* dan panduan medis awal. Studi ini menyoroti potensi integrasi tersebut dalam meningkatkan aksesibilitas informasi kesehatan, efisiensi penyampaian informasi, dan literasi kesehatan di kalangan pengguna. Metodologi penelitian mencakup pengambilan kunci API dari ChatGPT, implementasi integrasi dengan WhatsApp menggunakan bahasa pemrograman JavaScript dan C++, serta validasi respons *chatbot* melalui konsultasi dengan para ahli. *Chatbot* yang dihasilkan didemonstrasikan mampu membantu pengguna dalam diagnosis penyakit awal, memberikan tips kesehatan, dan bahkan menjawab pertanyaan terkait pemrograman, menunjukkan fleksibilitas dan tingkat akurasi yang menjanjikan. Meskipun demikian, penelitian ini menekankan pentingnya validasi lebih lanjut untuk memastikan keandalan *chatbot* dalam mendiagnosis berbagai spektrum gejala penyakit.

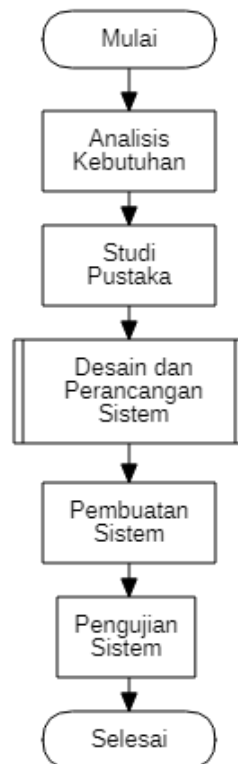
Penelitian (Pujiono dkk., 2024) mengkaji penerapan *Retrieval Augmented Generation* (RAG) dan *vector database* (Pinecone) pada *chatbot* untuk pengelolaan keuangan BLU, menggunakan regulasi (contohnya Peraturan Menteri Keuangan Nomor 129 Tahun 2020) yang diproses menjadi vektor. GPT-4, dengan *cosine similarity*, menunjukkan kinerja terbaik (skor rata-rata 0,404). Namun, penelitian ini juga mengidentifikasi beberapa tantangan, termasuk masalah konsistensi embedding data, di mana beberapa model memberikan jawaban yang sama untuk pertanyaan tertentu. Hal ini menunjukkan perlunya proses pembersihan data vektor yang lebih baik sebelum digunakan sebagai basis pengetahuan.

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Tahapan Penelitian

Tahapan penelitian yang diilustrasikan pada Gambar 3.1 mendokumentasikan proses pengembangan sistem aplikasi dalam penelitian ini. Sistem tersebut memanfaatkan metode RAG (Retrieval Augmented Generation) dengan model dari OpenAI yang sumber informasinya berasal dari dokumen.



Gambar 3.1 *Flowchart* Tahapan Penelitian

Sebagaimana diilustrasikan pada Gambar 3.1, tahapan awal yang harus dilalui sebelum memulai pengembangan sistem adalah tahap analisis kebutuhan. Pada tahapan ini, serangkaian kegiatan dilakukan, meliputi identifikasi permasalahan yang mendasari pengembangan sistem dan penetapan batasan masalah. Penetapan batasan masalah ini krusial untuk memastikan cakupan pengerjaan tetap terfokus dan tidak meluas di luar konteks penelitian. Detail mengenai identifikasi dan batasan masalah ini akan dijabarkan lebih lanjut pada bab

pendahuluan laporan ini. Setelah tahap analisis kebutuhan selesai, tahapan selanjutnya adalah studi pustaka. Pada tahap studi pustaka, dilakukan pencarian, pengumpulan, dan analisis referensi atau literatur ilmiah yang relevan dan berkaitan erat dengan topik penelitian yang sedang dikerjakan. Setelah landasan teoretis terbentuk, proses pengembangan dilanjutkan dengan tahapan desain dan perancangan sistem. Pada tahap ini, dirumuskan arsitektur, komponen, dan interaksi antar komponen sistem yang akan dibangun. Setelah bentuk dan struktur sistem yang akan dibuat dirumuskan secara detail, barulah masuk ke tahapan implementasi, yaitu tahapan pembuatan sistem secara aktual berdasarkan desain yang telah disepakati. Tahap akhir dari alur pengembangan ini adalah tahapan pengujian terhadap hasil implementasi sistem. Pada tahapan pengujian ini dilakukan evaluasi terhadap kinerja dan fungsionalitas sistem, yang pada akhirnya menentukan nilai keberhasilan sistem dalam mencapai tujuan yang telah ditetapkan.

### 3.2 Analisis Kebutuhan

Implementasi *chatbot* berbasis Retrieval Augmented Generation (RAG) pada WhatsApp memerlukan pemenuhan beberapa kebutuhan teknis dan fungsional. Pertama, metode RAG membutuhkan integrasi antara *Large Language Model* (LLM) dan model *embedding* untuk memungkinkan merespons berdasarkan informasi dari dokumen terstruktur seperti "Student Guide.pdf." Kedua, *chatbot* harus di-*hosting* secara *online* agar dapat diakses oleh WhatsApp Cloud, yang akan meneruskan pesan dari pengguna (mahasiswa) ke sistem *chatbot*. Dengan *chatbot* yang telah ter-*hosting* ini memungkinkan bahwa *chatbot* dapat beroperasi secara *real-time* dan tersedia 24/7 untuk melayani pertanyaan mahasiswa.

Dari sisi pengguna, sistem aplikasi ini dikategorikan menjadi dua kelompok yaitu, pengguna internal (admisi atau sekretariat) dan pengguna eksternal (mahasiswa). Pengguna internal dipersyaratkan memiliki perangkat dengan peramban web ( *web browser*) untuk mengakses antarmuka *website* sistem. Sementara itu, pengguna eksternal atau mahasiswa memerlukan akun WhatsApp sebagai media interaksi utama dengan sistem *chatbot*.

Dari sudut pandang pengembang, terdapat beberapa kebutuhan utama yang harus dipenuhi, yang umumnya berlaku dalam pengembangan sistem aplikasi.

Kebutuhan tersebut meliputi perangkat komputer yang memadai, koneksi internet yang stabil, dan perangkat lunak pengembangan yang relevan. Aspek biaya yang terkait dengan pemenuhan kebutuhan-kebutuhan yang telah disebutkan juga menjadi faktor penting yang dipertimbangkan dalam proses pengembangan.

Tabel 3.1 Daftar Biaya Kebutuhan Penelitian

Pengeluaran	Kebutuhan	Biaya Pengeluaran	Dolar ke Rupiah (Agustus 2024)
Kredit OpenAI	Model gpt-4o	\$ 5.00	Rp. 80,655.00
	Model text-embedding-ada-002		
Akun Render	Deploy aplikasi Flask	\$ 1.00	Rp. 16,131.00
	Database PostgreSQL		
Akun WhatsApp Business App	Nomor telepon kartu SIM	\$ 0.32	Rp. 5000,00
Total		\$ 6.32	Rp. 101.786.00

Berdasarkan rincian pengeluaran pada Tabel 3.1, biaya yang tertera merepresentasikan biaya minimum yang diperlukan agar sistem aplikasi dapat beroperasi secara fungsional. Perlu dicatat bahwa penggunaan kredit OpenAI bersifat berkelanjutan dan memerlukan pengisian ulang secara periodik sesuai dengan tingkat pemakaian. Penggunaan model ChatGPT dihitung berdasarkan satuan token, di mana jumlah token yang terpakai bervariasi tergantung pada jumlah kata yang dihasilkan oleh model dan bahasa yang digunakan. Semakin kompleks dan panjang respons yang dihasilkan, semakin banyak token yang dikonsumsi. Sebaliknya, biaya untuk akun Render dan akun WhatsApp bersifat *one-time payment* (sekali bayar), dengan catatan bahwa akun WhatsApp yang digunakan saat ini belum verifikasi sebagai akun bisnis. Penting untuk digarisbawahi bahwa semua kebutuhan yang tercantum pada Tabel 3.1 bersifat *scalable* dan dapat ditingkatkan (*upgrade*) sesuai dengan peningkatan kebutuhan operasional atau perkembangan organisasi maupun perusahaan di masa mendatang. Oleh karena itu, Tabel 3.1 menyajikan estimasi biaya minimum sebagai titik awal implementasi sistem.

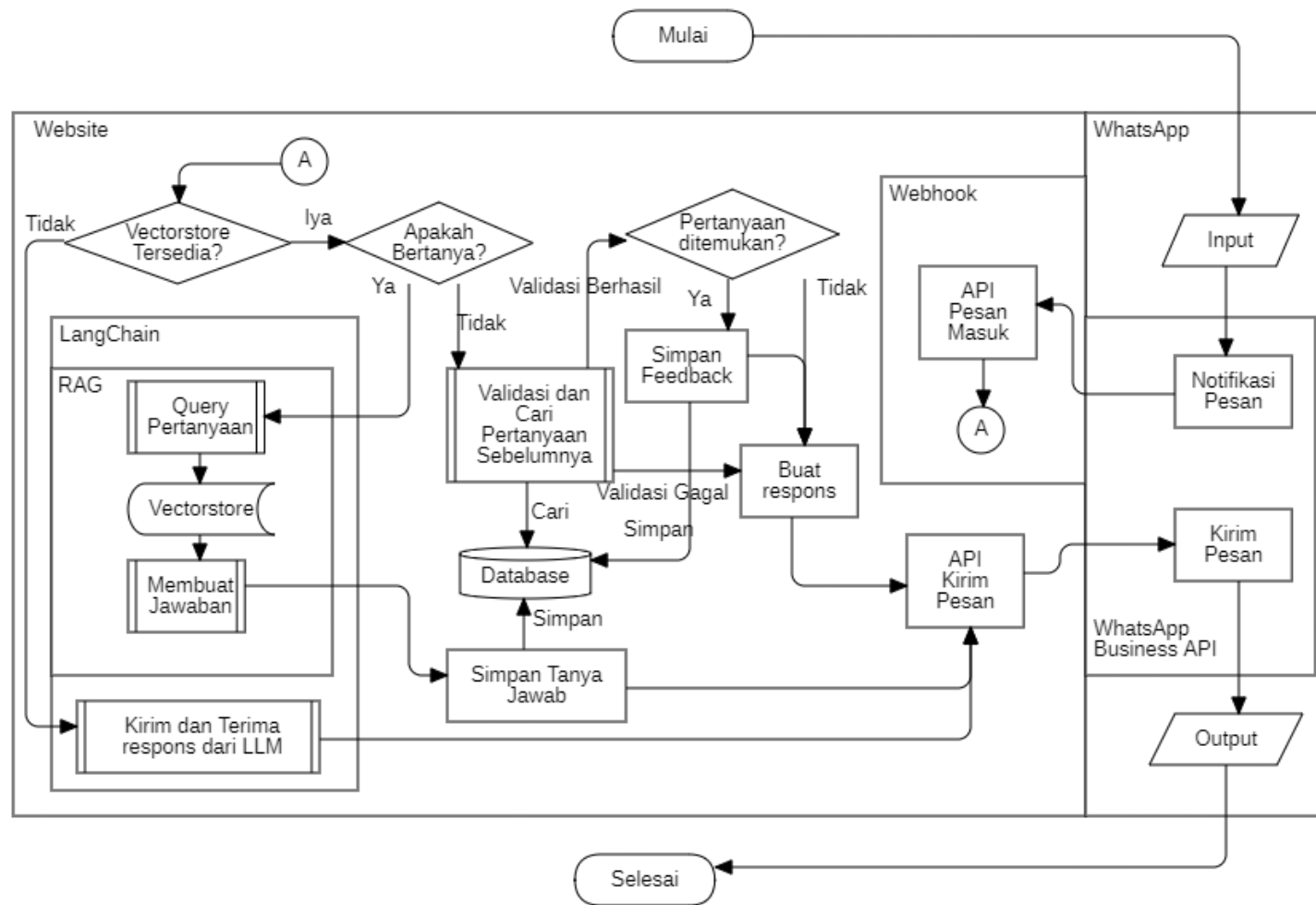
Pengembangan aplikasi ini akan diimplementasikan menggunakan bahasa pemrograman Python versi 3.11.9. Selain alokasi kredit untuk penggunaan ChatGPT, infrastruktur server yang memadai juga merupakan faktor krusial dalam keberhasilan penelitian ini. Dalam konteks ini, layanan *cloud hosting* seperti Render, yang mendukung Python versi 3.11.9 dan *libraries* penting seperti LangChain dan FAISS, dipilih sebagai platform *deployment*. Server ini akan berfungsi sebagai pusat pemrosesan, di mana dokumen PDF diunggah dan respons dari interaksi WhatsApp diolah. Nomor yang didaftarkan sebagai akun WhatsApp Business akan digunakan sebagai antarmuka untuk melakukan tanya jawab dengan sistem terkait dokumen PDF yang telah diunggah. Pemilihan platform dan *libraries* ini didasari pertimbangan kapabilitasnya dalam mendukung implementasi metode RAG (Retrieval Augmented Generation) yang diimplementasikan dalam penelitian ini.

### **3.3 Studi Pustaka**

Tahapan ini dilakukan penelusuran dan pembelajaran lebih lanjut mengenai metode–metode yang digunakan untuk meningkatkan keberhasilan sistem aplikasi. Proses inilah yang menjadi landasan dalam melakukan perancangan sistem aplikasi. Dengan kebutuhan yang tertera dan referensi–referensi yang mendukung akan mempermudah penulis dalam melakukan pengembangan sistem aplikasi.

### **3.4 Desain dan Perancangan Sistem**

Sistem aplikasi dirancang dengan dua antarmuka pengguna utama. Pertama, antarmuka web berfungsi sebagai tempat penyimpanan dan pengelolaan dokumen PDF yang menjadi basis pengetahuan *chatbot* serta *chatbot* itu sendiri. Kedua, antarmuka WhatsApp memungkinkan pengguna berinteraksi langsung dengan *chatbot* melalui percakapan teks. Di mana, diagram alur sistem dapat dilihat pada Gambar 3.2.

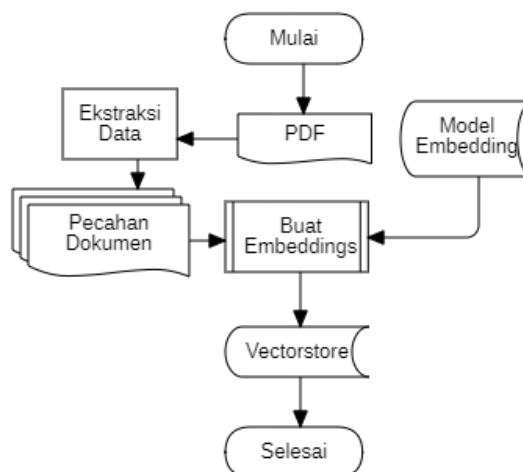


Gambar 3.2 Diagram Alur Sistem Aplikasi

Seperti yang diilustrasikan alur diagram pada Gambar 3.2, interaksi pengguna dengan sistem melalui WhatsApp difasilitasi oleh WhatsApp Business API. Pesan yang dikirimkan pengguna melalui WhatsApp pertama-tama diterima oleh API ini. Selanjutnya, agar sistem aplikasi dapat memproses pesan tersebut secara *real-time*, diimplementasikan mekanisme *webhook* yang terhubung ke API. *Webhook* ini memungkinkan sistem aplikasi untuk menerima notifikasi pesan masuk secara berkelanjutan selama server aplikasi berjalan. Berbeda dengan proses penerimaan pesan, pengiriman pesan balasan kepada pengguna juga tetap memanfaatkan WhatsApp Business API, namun tidak memerlukan konfigurasi *webhook*. Dengan demikian, WhatsApp Business API berperan sebagai gerbang utama untuk komunikasi dua arah antara sistem aplikasi dan platform WhatsApp, memastikan pertukaran data yang terstruktur dan terkelola dengan baik.

#### **3.4.1 Pembuatan Vectorstore dari Dokumen PDF**

Setelah data pesan masuk melalui *webhook*, sistem aplikasi akan menindaklanjutinya untuk menghasilkan respons jawaban yang relevan dengan pesan yang masuk. Diagram alur pada Gambar 3.2 menunjukkan 2 alur pemrosesan yang berbeda dari sebuah kondisi jika terpenuhi. Jika *vectorstore* tersedia, sistem menggunakan metode *Retrieval Augmentation Generation* (RAG) untuk menghasilkan respons berdasarkan pengetahuan dari dokumen. Sebaliknya, jika *vectorstore* belum tersedia, sistem akan langsung menggunakan model ChatGPT. Oleh karena itu, pengunggahan dokumen melalui antarmuka web oleh pengguna internal merupakan langkah krusial untuk mengaktifkan fungsionalitas RAG sebelum pengguna dari WhatsApp melakukan interaksi kepada *chatbot*.



Gambar 3.3 Alur Pengolahan Dokumen Menjadi *Vectorstore*

Sebagaimana diilustrasikan pada Gambar 3.3, teks dalam dokumen diekstraksi menjadi beberapa fragmen. Proses ekstraksi ini mengikuti aturan yang mengatur pemecahan data, yang detailnya dapat dilihat pada Gambar 3.4. Dua parameter utama yang digunakan adalah *size*, yang merepresentasikan jumlah karakter dalam setiap fragmen, dan *overlap*, yang merepresentasikan jumlah karakter yang tumpang tindih antar fragmen. Contohnya, *overlap* sebesar 1000 karakter berarti fragmen kedua akan memiliki 200 karakter yang sama dengan bagian akhir fragmen pertama. Tumpang tindih ini bertujuan untuk memfasilitasi pemahaman model percakapan terhadap fragmen yang mungkin berisi informasi penting yang terpotong.

```

1 from langchain_text_splitters import RecursiveCharacterTextSplitter
2 text_splitter = RecursiveCharacterTextSplitter(
3     chunk_size = 1000,
4     chunk_overlap = 200,
5     length_function = len,
6     is_separator_regex=False,
7 )
8 pages = text_splitter.split_documents(pdf.load())
  
```

Gambar 3.4 Gambar Kode Pemecahan Dokumen

Seperti yang ditunjukkan pada Gambar 3.4, hasil pemecahan dokumen berupa fragmen teks disimpan dalam sebuah daftar. Setiap fragmen teks ini memiliki metadata yang mencakup informasi sumber dokumen dan nomor halaman



tempat fragmen tersebut berasal. Hal tersebut memungkinkan pengguna nanti ketika menanyakan sumber informasi yang didapat seperti nomor halaman.

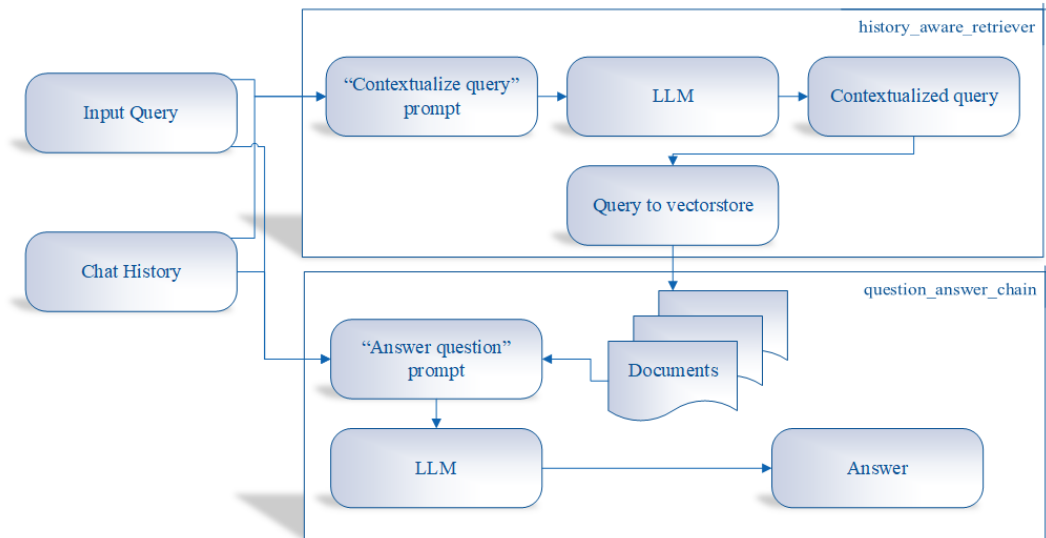
1	Document(metadata={
2	'source': '/content/drive/MyDrive/TA Chat
3	PDF/Student-Guide-UMC-2023.pdf', 'page': 15}, page_content=
4	'13 \n 8) Dalam setiap semester disajikan sejumlah
5	matakuliah yang \nmasing -masing mempunyai....'
6	)

Gambar 3.5 Contoh Isi dari Sebuah Fragmen Dokumen

Setelah fragmen teks dihasilkan, proses *embedding* menggunakan model *text-embedding-ada-002* akan mengekstraksi fitur semantik dari setiap fragmen. Setiap fragmen direpresentasikan sebagai vektor berdimensi tinggi, di mana setiap dimensi merepresentasikan fitur semantik tertentu. Vektor-vektor ini, yang disebut *embedding*, kemudian disimpan dalam *vectorstore*. Penelitian ini menggunakan FAISS sebagai implementasi *vectorstore*, yang menghasilkan dua berkas yaitu, satu berkas .pkl untuk metadata dan satu berkas .faiss untuk indeks vektor. Dengan terbentuknya *vectorstore*, pengguna dapat memanfaatkan metode RAG untuk pencarian informasi dari dokumen.

### 3.4.2 Perancangan Metode Retrieval Augment Generation

Layaknya layanan pelanggan pada umumnya, aplikasi ini diharapkan mampu melakukan percakapan interaktif dua arah. Oleh karena itu, kemampuan untuk mengingat konteks percakapan sebelumnya merupakan aspek krusial. Fungsionalitas ini dapat diimplementasikan dengan memanfaatkan *framework* LangChain.



Gambar 3.6 Alur Percakapan Aplikasi RAG dengan LangChain

Sumber : [https://python.langchain.com/v0.2/docs/tutorials/qa\\_chat\\_history/](https://python.langchain.com/v0.2/docs/tutorials/qa_chat_history/)

Gambar 3.6 mengilustrasikan alur pemrosesan *query* pertanyaan yang dijabarkan pada Gambar 3.2. Jadi, untuk mendapatkan dokumen yang diharapkan, sistem aplikasi akan membuat ulang pertanyaan yang sesuai dengan riwayat percakapan yang proses pada alur (Gambar 3.6) bernama “*Contextualized query*”. Dari proses yang telah dihasilkan tersebut akan digunakan untuk merumuskan *retriever* yang akan mencari fragmen dokumen dari *vectorstore* menggunakan model *Embedding* yang telah ditetapkan sebelumnya. Interaksi ini bertujuan untuk mendapatkan fragmen dokumen yang relevan dengan pertanyaan dan riwayat percakapan. Di mana, jumlah fragmen yang didapat dan diberikan kepada *Large Language Model* (LLM) pada penelitian ini adalah berjumlah 4. Kualitas jawaban sangat bergantung pada *prompt* yang diberikan, yang detailnya dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel Prompt untuk LLM

No.	Prompt	Fungsi
1	"Given a chat history and the latest user question which might reference context in the chat history, formulate a standalone question"	Memerintahkan model untuk membuat pertanyaan yang baik

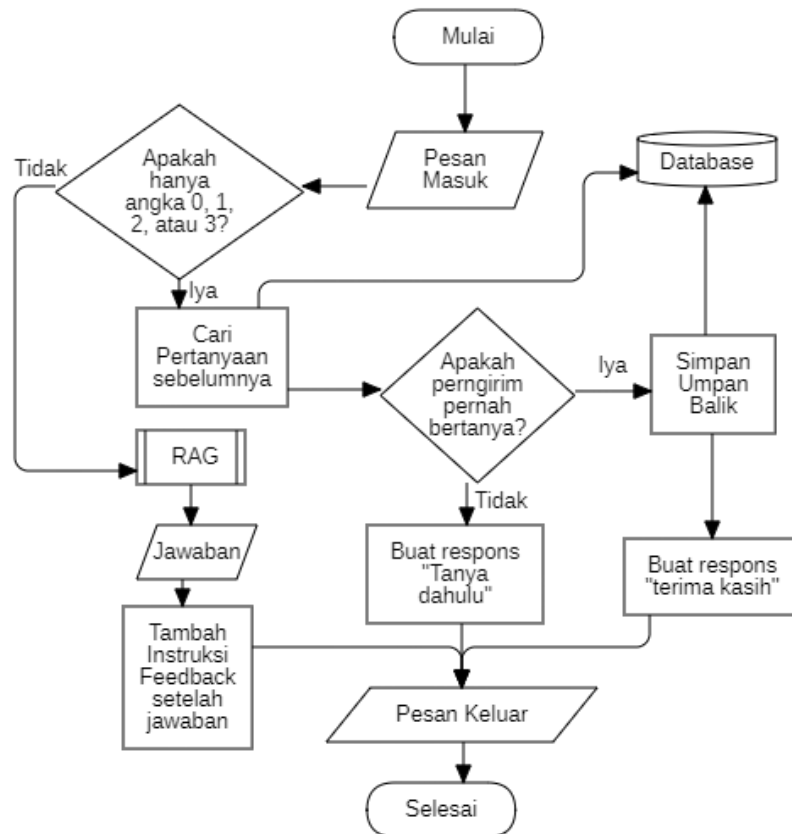
No.	Prompt	Fungsi
	which can be understood without the chat history. Do NOT answer the question, just reformulate it if needed and otherwise return it as is."	dari sejarah percakapan yang diberikan. Namun, pertanyaan tersebut harus dapat dipahami tanpa melihat sejarah percakapan.
2	"Your name is Quanta Quire. You are a bot created specifically for the Universitas Ma Chung research experiments. You are acting as a customer support that can help with whatever user need about information related. Use the following pieces of retrieved context to answer the question. If you don't know the answer, see the previous conversation to find the answer. Then, if you still don't know the answer, say that you don't know. Use three sentences maximum and keep the answer concise. response with Bahasa Indonesia. response as other language if i told you to do so. \n\n {context}"	Memberikan dasar informasi untuk identitas bot dan memerintahkan model untuk menjawab berdasarkan pecahan dokumen yang diberikan. Serta menyuruh model untuk menggunakan bahasa Indonesia dan menjawab dengan gaya preferensi yang diharapkan

Tabel 3.2 Tabel Prompt untuk LLM

*Prompt* yang dibuat adalah berdasarkan dari yang telah disediakan oleh (LangSmith, 2024). Namun, agar sesuai dengan tujuan penelitian, dilakukanlah sedikit modifikasi dari *prompt* yang telah diambil. Hasil *prompt* yang telah dimodifikasi bisa dilihat pada Tabel 3.2. Layaknya pelayanan pelanggan pada umumnya, diharapkan *chatbot* dapat merespons berdasarkan riwayat percakapan yang telah dilakukan. Seperti yang ditunjukkan pada Tabel 3.2, untuk menghasilkan *output* yang diharapkan, *prompt* pertama bertujuan untuk merumuskan pertanyaan yang lebih baik berdasarkan riwayat percakapan. *Output prompt* ini kemudian digunakan untuk mencari fragmen dokumen yang relevan dengan bantuan model *embedding*. Pemanfaatan GPT-4 yang kedua adalah untuk merekonstruksi jawaban yang mudah dipahami berdasarkan pertanyaan dan konteks yang diperoleh. Fragmen dokumen yang relevan diberikan melalui variabel *context* pada *prompt* kedua di Tabel 3.2.

### 3.4.3 Perancangan Umpan Balik pada Chatbot

Sebagaimana diilustrasikan dalam diagram alur pada Gambar 3.2, mekanisme umpan balik tersedia bagi pengguna setelah setiap respons yang diberikan oleh *chatbot*. Umpan balik ini bersifat opsional, memungkinkan pengguna untuk melanjutkan interaksi dengan mengirimkan pertanyaan berikutnya tanpa diwajibkan memberikan umpan balik terlebih dahulu.



Gambar 3.7 Diagram Alur Umpan Balik Pada Chatbot

Pada Gambar 3.2 telah dipaparkan bahwa adanya proses validasi umpan balik. Sedangkan detail proses validasi tersebut bisa dilihat pada Gambar 3.7. Validasi ini bertujuan untuk mengidentifikasi apakah pesan yang diterima dari pengguna merupakan umpan balik atau pertanyaan baru untuk *chatbot*. Proses validasi dilakukan dengan memeriksa format pesan. Sistem menginterpretasikan pesan yang hanya berisi angka 0, 1, 2, atau 3 sebagai umpan balik. Angka-angka ini diasumsikan merepresentasikan skala penilaian atau kategori umpan balik tertentu (misalnya, 0 untuk tidak sesuai, 3 untuk sangat sesuai). Pesan dengan format selain

angka-angka tersebut diinterpretasikan sebagai pertanyaan atau input baru untuk *chatbot*, bukan sebagai umpan balik.

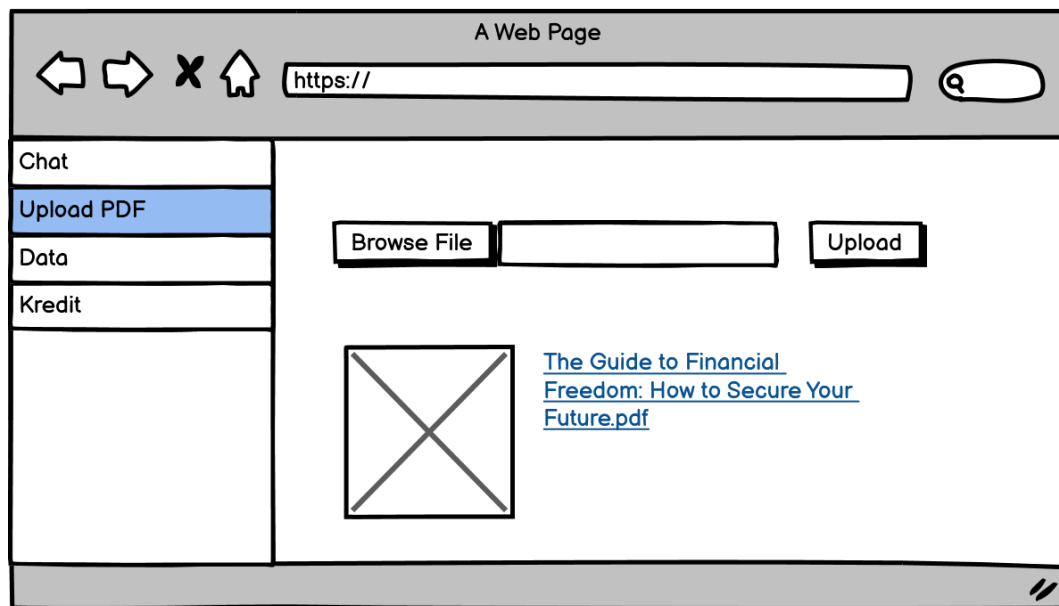
Setelah pesan diidentifikasi sebagai umpan balik, sistem akan melakukan pencarian di *database* untuk menemukan interaksi (pertanyaan) terakhir yang dilakukan oleh pengirim. Pencarian ini krusial untuk mengaitkan umpan balik dengan konteks pertanyaan yang relevan. Jika sistem tidak menemukan catatan interaksi sebelumnya dari pengirim yang bersangkutan, hal ini diinterpretasikan sebagai indikasi bahwa pengirim belum pernah berinteraksi dengan *chatbot* sebelumnya. Dalam kasus ini, umpan balik yang diterima tidak akan disimpan karena tidak dapat dikaitkan dengan pertanyaan secara spesifik.

Apabila pertanyaan terakhir dari pengirim berhasil ditemukan di *database*, sistem akan menyimpan umpan balik yang diterima dengan mengaitkannya dengan pasangan tanya jawab tersebut. Proses penyimpanan ini adalah dengan mengedit pasangan tanya jawab yang ditemukan dengan menambahkan angka umpan balik (0-3). Baik ketika umpan balik berhasil disimpan maupun ketika tidak berhasil disimpan (karena pengirim belum berinteraksi), sistem akan memberikan respons yang berbeda kepada pengirim. Respons ini berfungsi sebagai konfirmasi kepada pengguna mengenai status penerimaan umpan balik oleh sistem. Respons yang berbeda ini penting untuk transparansi dan memberikan kepastian kepada pengguna bahwa input mereka telah diproses, terlepas dari apakah input tersebut dapat dikaitkan dengan interaksi sebelumnya atau tidak.

Untuk memfasilitasi pemberian umpan balik yang benar, sistem menyediakan instruksi yang jelas kepada pengguna. Seperti yang ditunjukkan pada Gambar 3.7, setiap kali *chatbot* memberikan jawaban, sistem juga menyertakan instruksi tentang cara memberikan umpan balik. Instruksi ini bisa berupa penjelasan singkat tentang skala penilaian (0-3) dan bagaimana cara mengirimkannya. Penambahan instruksi ini bertujuan untuk memastikan pengguna memahami format dan cara memberikan umpan balik dengan benar. Selain itu, sistem juga menekankan bahwa pemberian umpan balik bersifat opsional. Pengguna dapat memilih untuk memberikan umpan balik atau melanjutkan interaksi dengan *chatbot* tanpa memberikan umpan balik terlebih dahulu. Hal ini memberikan fleksibilitas kepada pengguna dan tidak mengganggu alur interaksi utama.

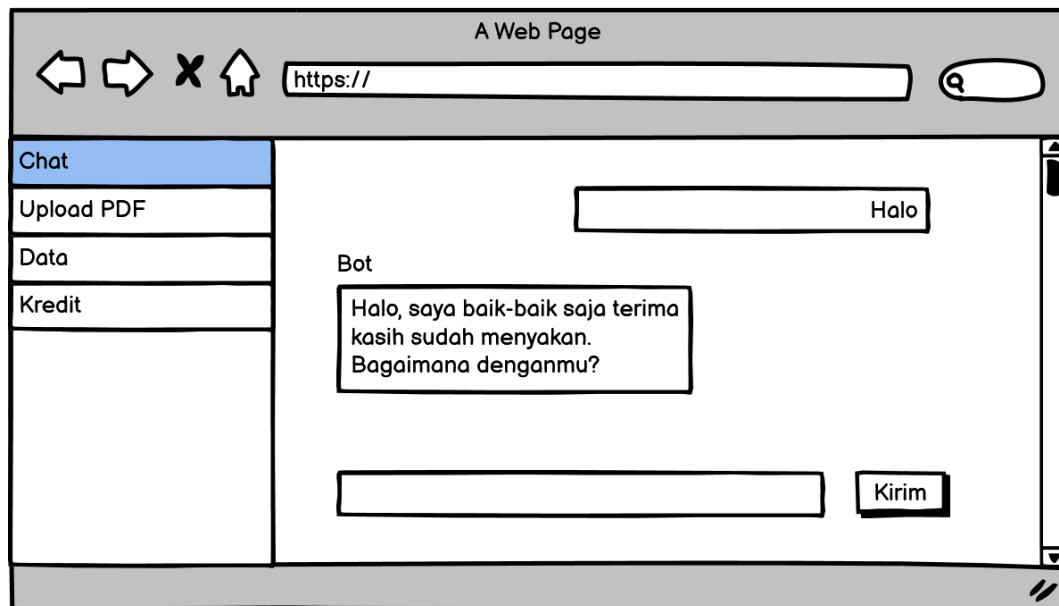
#### 3.4.4 Desain Tampilan Aplikasi Website

Untuk memastikan kemudahan penggunaan dan navigasi aplikasi *website*, desain yang *user-friendly* sangat diperlukan. Berdasarkan alur sistem aplikasi yang tertera pada Gambar 3.2, antarmuka *website* akan terdiri dari empat menu utama, yang masing-masing memiliki fungsi spesifik.



Gambar 3.8 Desain Mockup Website Menu Unggah Dokumen

Menu pertama yang krusial adalah halaman unggah dokumen. Halaman ini berfungsi sebagai tempat pengguna mengunggah dokumen yang diperlukan untuk pembuatan *vectorstore*. Tampilan *mockup* halaman ini dapat dilihat pada Gambar 3.8. Desain *mockup* tersebut menampilkan indikator status dokumen, yaitu tampilan informasi sekilas dokumen bahwa telah berhasil diunggah atau tampilan kosong sebagai indikasi belum adanya dokumen yang diunggah.



Gambar 3.9 Desain Mockup Halaman Menu Chat

Menu *chat* berfungsi sebagai alat pengujian respons *chatbot* secara internal, mensimulasikan interaksi yang akan terjadi melalui WhatsApp. Perlu ditekankan bahwa interaksi pada halaman *chat* ini tidak terhubung dengan WhatsApp Business API. Memungkinkan pengujian akses *chatbot* terhadap dokumen yang telah diunggah tanpa campur tangan dari WhatsApp. Perbedaan utama dengan interaksi melalui WhatsApp adalah pada data identifikasi *chat*. Di mana, *chat* melalui halaman ini akan ditandai dengan ID unik per sesi *browser*, sedangkan interaksi WhatsApp ditandai dengan asal nomor telepon pengguna. Hal ini memungkinkan pembedaan antara pengujian internal dan interaksi yang diasumsikan berasal dari mahasiswa atau eksternal. Sehingga mempermudah proses evaluasi *chatbot*.

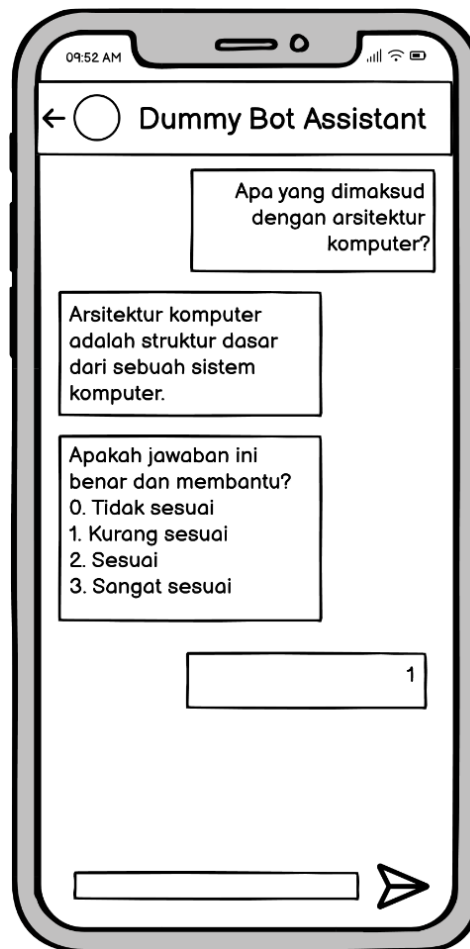
Selain menu *chat* dan unggah dokumen, antarmuka *website* juga dilengkapi dengan menu "Data" dan "Kredit" (lihat Gambar 3.9). Kedua menu ini bersifat informatif dan tidak memerlukan interaksi pengguna. Menu "Data" menyajikan rekam jejak lengkap interaksi tanya jawab yang telah terjadi dalam sistem. Fungsi ini memungkinkan tim internal untuk menganalisis dan memvalidasi respons *chatbot*, serta menghubungi pengguna jika diperlukan perbaikan. Dengan menu "Data" inilah peneliti dapat menganalisis setiap tanya jawab yang terjadi. Berbeda dengan menu "Data" yang dinamis, menu "Kredit" menampilkan daftar statis teknologi yang digunakan dalam pengembangan *website*.

### 3.5 Perancangan Pengujian

Penelitian ini menerapkan dua jenis pengujian. Pengujian kepuasan sistem aplikasi yang didapatkan dari survei. Serta pengujian kualitas chatbot dalam memberikan respons yang didapatkan dari metrik khusus.

#### 3.5.1 Pengujian Kepuasan Pengguna

Pengujian kepuasan pengguna pada sistem aplikasi akan menyerupai dengan OpenAI ChatGPT. Yaitu dengan memberikan pilihan umpan balik mengenai kualitas hasil jawaban yang diterima. Namun, pada penelitian ini sistem aplikasi hanya akan menerima umpan balik saja tanpa melakukan pengiriman ulang jawaban yang lebih baik layaknya ChatGPT pada umumnya. Sehingga pengguna hanya akan dapat mengirimkan umpan balik secara searah saja. Serta umpan balik juga akan ditawarkan setiap kali pengguna menerima jawaban dari sistem aplikasi.



Gambar 3.10 Contoh Survei Melalui WhatsApp



Pada Gambar 3.10 merupakan contoh umpan balik yang ditawarkan setiap kali sistem aplikasi memberikan jawaban. Pertanyaan umpan balik akan diberikan secara *hardcode* dan tidak menggunakan LLM sama sekali untuk mengolah kata – kata yang diberikan. Begitu juga dengan jawaban yang akan diterima. Di mana jawaban yang hanya diterima adalah hanya berupa angka dari 0 sampai 3 saja. Selainnya akan tidak dikenali dan dianggap memberikan pertanyaan lagi daripada menjawab umpan balik.

Data umpan balik yang terkumpul akan disimpan dalam basis data untuk dianalisis lebih lanjut oleh pengembang. Tujuan utama penyimpanan data ini adalah untuk mengevaluasi kualitas sistem aplikasi secara menyeluruh. Pengembang memerlukan akses terhadap pertanyaan yang diajukan pengguna dan respons yang diberikan sistem untuk mengkaji kesesuaian antara keduanya. Analisis ini krusial untuk mengukur tingkat kepuasan pengguna terhadap respons yang diterima dan mengevaluasi akurasi *chatbot* dalam memberikan jawaban yang relevan dan tepat. Struktur data yang akan disimpan dicontohkan pada Tabel 3.3.

Tabel 3.3 Format Data Umpan Balik yang Disimpan

timestamp	Pengguna	Pertanyaan	Jawaban	Kategori
2023-10-29 15:45:30	081xxxxxxxxx	Apa itu arsitektur komputer	Arsitektur komputer adalah studi tentang cara membuat komputer semakin rumit ....	3
2023-10-29 15:52:11	081xxxxxxxxx	Mengapa bumi bulat?	Bumi bulat karena gaya gravitasi.	2
2023-10-30 02:10:32	081xxxxxxxxx	Berapa ukuran huruf untuk laporan TA?	Format ukuran untuk laporan TA adalah berukuran 12	1

Umpan balik yang akan disimpan hanya umpan balik yang berkategori 0, 1, 2, dan 3 saja. Sedangkan data dari masukan pengguna yang disimpan adalah *timestamp*, pertanyaan yang ditanyakan, jawaban dari sistem aplikasi, dan kategori

yang diberikan. Dari struktur data tersebut, pengembang dapat melihat total jumlah masukan per kategori.

Sistem dirancang untuk dapat menyimpan data umpan balik pengguna secara terstruktur guna analisis dan evaluasi kinerja *chatbot*. Umpan balik yang disimpan berupa kategori dengan angka 0, 1, 2, dan 3. Kategori ini diasumsikan merepresentasikan skala penilaian atau tingkat kepuasan pengguna terhadap jawaban yang diberikan oleh sistem (misalnya, 0 untuk tidak sesuai dan 3 sesuai). Data yang disimpan untuk setiap umpan balik mencakup *timestamp* (waktu pengguna bertanya), pertanyaan yang diajukan pengguna, jawaban yang diberikan oleh sistem aplikasi, dan kategori umpan balik yang diberikan pengguna.

### 3.5.2 Pengujian Kualitas Respons Chatbot

Keakurasian *chatbot* dalam memberikan respons pada penelitian ini akan dinilai dengan sejumlah metrik khusus. Di mana, setiap respons yang diterima akan diberikan label tambahan untuk mengenali kualitas *chatbot* dari respons tersebut. Berikut adalah penjelasan mengenai setiap metrik yang digunakan.

- *document\_contain\_answer* merupakan kategori yang mendeteksi akan ketersediaan informasi pada dokumen dari pertanyaan pengguna.
- *retrieved\_document\_relevance* adalah untuk mendeteksi relevansi fragmen dokumen yang didapat dengan pertanyaan yang diajukan.
- *answer\_correctness* adalah ketepatan jawaban yang diberikan secara faktual.
- *refusal\_behavior* adalah untuk mengukur perilaku chatbot dalam menolak memberikan jawaban. Penolakan dapat dikategorikan sebagai penolakan yang benar (*justified refusal*) jika informasi yang diminta memang tidak tersedia di dokumen, atau penolakan yang salah (*unjustified refusal*) jika informasi sebenarnya tersedia tetapi *chatbot* gagal mengenalnya..
- *hallucination* adalah saat *chatbot* memberikan jawaban yang secara faktual salah atau fabrikasi jawaban dengan tingkat kepercayaan tinggi tanpa melihat tersedia atau tidak tersedianya informasi yang dicari pada dokumen.

Proses pemberian nilai terhadap setiap respons untuk evaluasi *chatbot*, dilakukan secara manual. Di mana peneliti akan mengevaluasi kejelasan dan

kesesuaian jawaban dari setiap respons satu per satu. Sehingga dengan metrik tambahan tersebut, diharapkan setiap respons dapat dilakukan analisis dan perhitungan akurasi dengan mudah.

```
1  [  
2    {  
3      "id": 32,  
4      "timestamp": "2024-08-05T12:28:31.021732",  
5      "user": "628xxxxx",  
6      "question": "bagaimana cara mendaftar UAS ?",  
7      "answer": "Maaf, saya tidak dapat membantu Anda.",  
8      "category": 1,  
9      "document_contains_answer": false,  
10     "retrieved_document_relevance": false,  
11     "answer_correctness": true,  
12     "refusal_behavior": true,  
13     "hallucination": false  
14   },  
15 ]
```

Gambar 3.11 Contoh Sebuah Data Respons Untuk Pengujian

Contoh respons data beserta kategori umpan baliknya dapat dilihat pada Gambar 3.11. Pada contoh tersebut, *chatbot* menolak untuk memberikan jawaban dengan justifikasi bahwa tidak ada fragmen dokumen yang relevan dengan pertanyaan yang diajukan. Penolakan ini dianggap tepat secara faktual karena informasi yang dicari memang tidak tersedia dalam dokumen. Dengan demikian, kualitas *chatbot* dapat dianalisis dan akurasi *chatbot* dalam memberikan respons dapat dikalkulasi secara lengkap.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Desain dan Fitur

Pengembangan aplikasi pada penelitian ini, terdiri dari beberapa sistem yang harus dibuat. Yaitu, sistem generasi *vectorstore database*, integrasi dengan WhatsApp, tampilan *website* untuk pengunggahan dokumen, dan *database* untuk menyimpan hasil evaluasi aplikasi.

##### 4.1.1 Sistem Generasi Vectorstore Database

Pengembangan percakapan dalam penelitian ini dimulai dengan membuat fungsi untuk mengolah dokumen PDF menjadi *vectorstore*. Dengan memanfaatkan LangChain, penggunaan fungsi *embedding* dapat dengan mudah diterapkan dengan kode berikut.

```
1 import os
2 from pypdf import PdfReader
3 from quanta_quire.helper import
4     delete_all_vectorstore, delete_all_pdfs
5 from langchain_openai import OpenAIEmbeddings
6 from langchain_community.document_loaders import PyPDFLoader
7 from langchain_text_splitters
8     import RecursiveCharacterTextSplitter
9 from langchain_community.vectorstores import FAISS
10
11 def splitter(split_size, split_overlap):
12     pdf= PyPDFLoader(os.path.join(current_app.config['UPLOAD_PATH'],
13     get_first_pdf_file()))
14     text_splitter = RecursiveCharacterTextSplitter(
15         chunk_size=split_size,
16         chunk_overlap=split_overlap,
17         length_function=len,
18         is_separator_regex=False,
19     )
20     return text_splitter.split_documents(pdf.load())
```

21	
22	<code>def create_faiss(chunks):</code>
23	<code>    faiss_index = FAISS.from_documents(chunks, OpenAIEmbeddings())</code>
24	<code>    return faiss_index</code>

Gambar 4.1 Kode Mengolah Dokumen Menjadi Vectorstore

Pada kode di atas, terdapat parameter besar *chunks* dan jumlah *overlap* pada fungsi *splitter*. Parameter besar *chunk* yang telah ditetapkan untuk membagi data dokumen menjadi beberapa bagian adalah sebanyak 1000 karakter. Sedangkan *overlap* sebanyak 200 karakter.

Penyimpanan *vectorstore* yang diterapkan adalah dengan menyimpannya secara lokal. Jenis dan nama berkas akan berbeda tergantung dengan fungsi *embedding* yang digunakan. Karena fungsi *embedding* yang digunakan adalah FAISS, berkas yang disimpan ada 2 yaitu, `index.faiss` dan `index.pkl`. Dengan adanya kemampuan untuk dapat menyimpan *vectorstore* secara lokal, penggunaan token dari OpenAI dapat diminimalisir setiap kali aplikasi dijalankan.

1	<code>import os</code>
2	<code>from flask import current_app</code>
3	<code>from langchain_openai import OpenAIEmbeddings</code>
4	<code>from quanta_quire.helper import delete_all_vectorstore,</code>
5	<code>    delete_all_pdfs</code>
6	
7	<code>Def generate_vectorstore(chunks):</code>
8	<code>    try:</code>
9	<code>        delete_all_vectorstore()</code>
10	<code>        vectorstore = create_faiss(chunks)</code>
11	<code>        vectorstore.save_local(</code>
12	<code>            os.path.join(current_app.config['UPLOAD_PATH'],</code>
13	<code>                "vectorstore"))</code>
14	<code>    except Exception as e:</code>
15	<code>        current_app.logger.info(e)</code>
16	<code>        delete_all_pdfs()</code>
17	<code>        delete_all_vectorstore()</code>
18	

19	
20	def faiss_load_vectorstore():
21	return FAISS.load_local(
22	folder_path=os.path.join(current_app.config['UPLOAD_PATH'],
23	"vectorstore"),
24	embeddings=OpenAIEmbeddings(),
25	allow_dangerous_deserialization=True
26	)

Gambar 4.2 Kode Simpan dan Memuat Vectorstore

Setiap kali percakapan dimulai, aplikasi akan memanggil fungsi pada kode di atas untuk mencari pengetahuan yang relevan dengan percakapan. Serta apabila aplikasi mendeteksi perubahan pada dokumen, aplikasi hanya tinggal memanggil fungsi simpan saja untuk setiap pembuatan *vectorstore* baru.

```

Embedding dimensions: (412, 1536)
Mean value: -0.0007
Standard deviation: 0.0255
Min value: -0.7147
Max value: 0.2274

```

Gambar 4.3 Hasil Vectorstore dari Dokumen

Seperti yang sudah direncanakan pada perancangan sebelumnya, dengan jumlah 1000 karakter per fragmen, *overlap* 200 karakter, dan dari dokumen “Student Guide 2023”, telah menghasilkan hasil yang bisa terlihat pada Gambar 4.3. Di mana, dimensi *vector store* yang dihasilkan terdiri dari 412 pecahan dokumen, dengan setiap pecahan direpresentasikan oleh vektor 1536 dimensi (fitur). Nilai rata-rata (*mean*) dari seluruh *embedding* adalah -0,0007, mendekati nol, yang mengindikasikan normalisasi yang baik dan sesuai ekspektasi, mengingat *embedding* OpenAI dirancang untuk berpusat di sekitar nol. Rentang nilai minimum dan maksimum *embedding* adalah -0,7147 dan 0,2274, menunjukkan asimetri yang lazim dan mencerminkan karakteristik data teks. Di mana, nilai negatif memiliki rentang yang lebih besar dibandingkan nilai positif.

```

Search results for query: 'Batas minimal nilai toeic kelulusan berapa?

Score: 0.2467
Content: 172
  a. Setiap awal semester diadakan test TOEIC atau
  equivalensinya ;
  b. Setiap semester mahasiswa harus mampu menunjukkan
  performance kenaikan angka TOEIC sebesar 15 sampai
  dengan 20 point.
...
Metadata: {'source': '/home/fryctze/Documents/Project/Python/RESEARCH/'
-----

Score: 0.3091
Content: Service , USA, memberikan sertifikasi TOEIC.
Sebelum tes yang dilakukan oleh ITC, terlebih dahulu dilakukan tes
TOEIC ekuivalensi secara berkala di Universitas Ma Chung.
Syarat kelulusan adalah s...
Metadata: {'source': '/home/fryctze/Documents/Project/Python/RESEARCH/'
-----

Score: 0.3154
Content: Lulus dengan Sangat Memuaskan bila IPK: 3,5 1-3,75
Lulus dengan Memuaskan bila IPK: 3,00 -3,50

24. Evaluasi
Pemantauan dan evaluasi dilaksanakan dalam 2 tahap, yaitu: Evaluasi
Belajar Tahap ...
Metadata: {'source': '/home/fryctze/Documents/Project/Python/RESEARCH/'
-----

```

Gambar 4.4 Keluaran Pecahan Dokumen yang Didapat dari Vectorstore

Ilustrasi *query* terhadap *vector store* yang disajikan pada Gambar 4.4, telah memperlihatkan bahwa setiap pecahan dokumen memiliki akan skor sebagai representasi jarak kedekatan secara semantik dengan *query* yang telah diberikan. Dengan menggunakan model *text-embedding-ada-002*, diperoleh pecahan terdekat memiliki skor 0,2467. Fragmen dokumen yang terbaik ini kemudian akan diproses lebih lanjut oleh *chatbot* untuk menghasilkan jawaban yang relevan terhadap pertanyaan pengguna.

#### 4.1.2 Implementasi Chat Bot pada Sistem

Implementasi *chatbot* dengan LangChain melibatkan penggunaan fungsi *retriever* sebagai komponen kunci. Fungsi ini menerima *prompt* pengguna sebagai

input, lalu merekonstruksi atau memformulasikan ulang *prompt* tersebut menjadi pertanyaan yang lebih terstruktur dan spesifik. Pertanyaan yang telah direkonstruksi ini kemudian digunakan untuk melakukan pencarian dan pengambilan dokumen yang paling relevan dari basis pengetahuan (*knowledge base*).

```

1  def history_aware_retriever(retriever):
2      contextualize_q_prompt = ChatPromptTemplate.from_messages([
3          ("system", "Prompt pertama"),
4          MessagesPlaceholder("chat_history"),
5          ("human", "{input}"),
6      ])
7      return create_history_aware_retriever(
8          llm, retriever, contextualize_q_prompt
9      )
10 def rag_chain(retriever):
11     qa_prompt = ChatPromptTemplate.from_messages([
12         ("system", "prompt kedua {context}"),
13         MessagesPlaceholder("chat_history"),
14         ("human", "{input}"),
15     ])
16     question_answer_chain = create_stuff_documents_chain(
17         llm, qa_prompt)
18     return create_retrieval_chain(
19         history_aware_retriever(retriever), question_answer_chain)

```

Gambar 4.5 Kode Prompt Percakapan LangChain

Pada Gambar 4.5 terdapat fungsi *history\_aware\_retriever* untuk mendapatkan pecahan dokumen dengan melihat pertanyaan dan *history* percakapan. Di mana, sesuai dengan alur perancangan pada Gambar 3.6, terdapat beberapa langkah yang harus dilalui sebelum memperoleh dokumen yang diinginkan. Semua langkah tersebut dibungkus menjadi satu fungsi yang bisa terlihat pada baris kode 8 pada Gambar 4.5. Sehingga dokumen yang didapat memiliki relevansi tinggi dengan *history* percakapan tanpa disebutkan secara eksplisit oleh pengguna pada pertanyaan yang diberikan. Sedangkan fungsi '*create\_retrieval\_chain*' digunakan sebagai mengolah tanggapan yang mudah dicerna oleh manusia dari pecahan dokumen yang didapat. Untuk menjalankan



proses percakapan, digunakannya fungsi *RunnableWithMessageHistory* yang telah disediakan oleh LangChain. Di mana, keluaran dari *history\_aware\_retriever* akan digunakan kembali sebagai salah satu paramater dari. Fungsi tersebut juga sekaligus manajemen sejarah percakapan yang dimasukkan sebagai salah satu parameternya. Karena fungsi tersebut adalah *runnable*, maka dapat dijalankan dengan hanya memanggil *invoke*. Di mana, contoh kode keseluruhan dapat dilihat pada lampiran nomor 1.

```
=== CURRENT CHAT HISTORY ===
human: bagaimana cara mengambil cuti?
ai: Untuk mengambil cuti akademik, mahasiswa
harus mengisi formulir permohonan cuti
akademik dan berkonsultasi dengan Dosen
Penasehat Akademik. Setelah itu, mahasiswa
harus mendapatkan persetujuan dari Kepala
Program Studi dan menyampaikan permohonan
cuti kepada Dekan. Jika semua syarat telah
terpenuhi, Dekan akan menetapkan cuti
tersebut.
human: apa ada batasnya?
ai: Ya, batas pengajuan permohonan cuti akademik
sesuai dengan kalender akademik. Mahasiswa
tidak dapat mengajukan cuti di dalam semester
berjalan jika sudah melakukan KRS dan
mengikuti perkuliahan.
```

-----

Gambar 4.6 Contoh Percakapan dengan Chatbot

Hasil implementasi *chatbot* pada sistem disajikan pada Gambar 4.6. Terlihat bahwa *chatbot* berhasil memberikan jawaban yang relevan berdasarkan *vectorstore* yang telah dibangun sebelumnya. Gambar 4.6 juga mengilustrasikan kemampuan *chatbot* dalam merespons pertanyaan lanjutan, yang mengindikasikan bahwa sistem menyimpan riwayat interaksi pengguna. Detail alur kerja *chatbot* dijelaskan lebih lanjut pada Gambar 4.7 dan Gambar 4.8.

```

=== LLM PROMPT ===
System: Given a chat history and the latest user question which might reference context in the chat history, formulate a standalone question which can be understood without the chat history. Do NOT answer the question, just reformulate it if needed and otherwise return it as is.
Human: bagaimana cara mengambil cuti?
AI: Untuk mengambil cuti akademik, mahasiswa harus mengisi formulir permohonan cuti akademik dan berkonsultasi dengan Dosen Penasehat Akademik. Setelah itu, mahasiswa harus mendapatkan persetujuan dari Kepala Program Studi dan menyampaikan permohonan cuti kepada Dekan. Jika semua syarat telah terpenuhi, Dekan akan menetapkan cuti tersebut.
Human: apa ada batasnya?
WARNING:langchain_core.callbacks.manager:Error

=== RETRIEVER START ===
Original Query: Apakah ada batasan dalam pengajuan cuti akademik bagi mahasiswa?
WARNING:langchain_core.callbacks.manager:Error in Detail
WARNING:langchain_core.callbacks.manager:Error in Detail
WARNING:langchain_core.callbacks.manager:Error in Detail
WARNING:langchain_core.callbacks.manager:Error in Detail
WARNING:langchain_core.callbacks.manager:Error in Detail

=== RETRIEVED DOCUMENTS ===

Document 1:
Content: 34
c. Telah mengikuti program pendidikan sekurang -kurangnya 1 semester.
2. Tata cara cuti akademik adalah sebagai berikut:
a. Mahasiswa mengisi formulir permohonan cuti akademik ;
b. Mahas...
Metadata: {'source': '/home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta_quire/storage/Student-Guide-UMC-2023.pdf', 'page': 36}

Document 2:
Content: Mahasiswa cuti akademik ialah mahasiswa yang mengajukan surat permintaan cuti akademik selama periode tertentu dan disetujui dekan.

Mahasiswa non a ktif ialah mahasiswa yang tidak melakukan...
Metadata: {'source': '/home/fryctze/Documents/Project/P

```

Gambar 4.7 Rekonstruksi Pertanyaan di Kiri dan Pencarian Dokumen ke Vectorstore di Kanan

Seperti pada Gambar 4.5, terdapat fungsi *history\_aware\_retriever* yang mana kinerja fungsi tersebut terlihat pada Gambar 4.7 sisi kiri. Fungsi tersebut akan berjalan terlebih dahulu dengan menyertakan riwayat interaksi sebelumnya antara pengguna dan *chatbot*. Dengan memanfaatkan *prompt* awal yang diberikan pengguna dan riwayat percakapan yang ada, model *chat* GPT-4o akan memanfaatkan informasi tersebut untuk menghasilkan pertanyaan baru yang lebih mudah dipahami oleh model *embedding text-embedding-ada-002*. Pertanyaan yang direformulasi ini kemudian digunakan kembali untuk mencari pecahan dokumen dalam *vectorstore*, dengan tujuan menemukan pecahan dokumen yang memiliki makna semantik paling relevan dengan maksud pertanyaan pengguna yang sebenarnya. Proses ini memungkinkan *chatbot* untuk memberikan jawaban yang lebih akurat dan kontekstual, dalam situasi mendapatkan pertanyaan lanjutan atau pertanyaan yang ambigu.

```

=== LLM PROMPT ===
System: Your name is Quanta Quire. You are a
bot created specifically for the Universitas
Ma Chung research experiments. You are acting
as a customer support that can help with
whatever user need about information related.
Use the following pieces of retrieved context
to answer the question. If you don't know the
answer, see the previous conversation to find
the answer. Then, if you still don't know the
answer, say that you don't know. Use three
sentences maximum and keep the answer
concise. response with Bahasa Indonesia.
response as other language if i told you to
do so.

34
c. Telah mengikuti program pendidikan
sekurang -kurangnya
1 semester.

2. Tata cara cuti akademik adalah sebagai
berikut:
a. Mahasiswa mengisi formulir permohonan cuti
akademik ;
b. Mahasiswa berkonsultasi dengan Dosen
Penasehat
Akademik;

namun kemudian
mengajukan cuti dan disetujui oleh Dekan,
maka biaya kuliah
yang telah dibayarkan tidak dapat
dikembalikan baik sebagian
maupun keseluruhan . Biaya yang telah
dibayarkan dianggap
Human: bagaimana cara mengambil cuti?
AI: Untuk mengambil cuti akademik, mahasiswa
harus mengisi formulir permohonan cuti,
berkonsultasi dengan Dosen Penasehat
Akademik, meminta persetujuan Kepala Program
Studi, dan menyampaikan permohonan kepada
Dekan. Jika semua syarat telah terpenuhi,
Dekan akan menetapkan cuti. Mahasiswa yang
disetujui cutinya juga harus melunasi
administrasi keuangan sesuai peraturan yang
berlaku.
Human: apa ada batasnya?
WARNING:langchain_core.callbacks.manager:Error

=== LLM RESPONSE ===
Ya, ada batasnya. Cuti akademik dapat diambil
maksimal dua semester berturut-turut atau
empat semester tidak berturut-turut selama
tidak melebihi batas studi 14 semester. Batas
akhir pengajuan permohonan cuti akademik
sesuai dengan kalender akademik.

```

Gambar 4.8 Generasi Jawaban dari Konteks dan Sejarah Percakapan

Pecahan dokumen yang diperoleh kemudian diintegrasikan ke dalam *prompt* untuk memberikan konteks kepada *chatbot* dalam menghasilkan jawaban yang relevan. Ilustrasi proses pembentukan jawaban disajikan pada Gambar 4.8. Sebagaimana ditunjukkan pada gambar tersebut, *prompt* kedua berisi pertanyaan asli, bukan hasil rekonstruksi. Perlu ditekankan bahwa rekonstruksi pertanyaan pada Gambar 4.7 semata-mata bertujuan untuk memfasilitasi model *embedding* dalam mengidentifikasi pecahan dokumen yang relevan. Dengan demikian, *prompt* kedua menggunakan pertanyaan asli dengan tambahan sebuah pecahan dokumen sebagai konteks percakapan. Memungkinkan respons jawaban yang diberikan oleh *chatbot* lebih kontekstual dan sesuai dengan maksud pertanyaan.

#### 4.1.3 Pengembangan Fitur Mengunggah Dokumen

Situs web yang dikembangkan bertujuan sebagai tempat untuk mengunggah dokumen berformat PDF dan mengolah jawaban yang akan dikirimkan ke pengguna. Pengembangannya pada penelitian ini akan di-*deploy* pada server

render.com. Dengan syarat, aplikasi telah terunggah di penyedia Git yang salah satunya adalah GitHub.

```
1 from flask_wtf import FlaskForm
2 from flask_wtf.file import FileField, FileRequired, FileAllowed
3 from wtforms import SubmitField
4
5 class UploadForm(FlaskForm):
6     document = FileField('Upload Dokumen',
7         validators=[FileRequired(), FileAllowed(['pdf'])])
8     submit = SubmitField()
```

Gambar 4.9 Kode Form di File Bernama forms.py

Pengunggahan berkas dokumen pada penelitian ini menggunakan *library* python bernama flask\_wtf dan wtform. Pada Gambar 4.9 memperlihatkan bahwa dengan menggunakan *library* tersebut, pengembang dapat memberikan validasi. Di mana, validasi tersebut untuk mencegah pengguna mengunggah berkas selain berformat .pdf atau mengunggah tanpa memberikan berkas apapun.

```
1 @blueprint.route("/document", methods=['GET', 'POST'])
2 def document():
3     form = UploadForm()
4     if form.validate_on_submit():
5         # Save file and create vectorstore
6         return redirect(url_for('website.document'))
7     return render_template("menu/document.html", **context)
```

Gambar 4.10 Contoh Kode Menampilkan dan Mengunggah Dokumen

Pengunggahan dokumen dapat diimplementasikan dengan hanya satu rute saja. Di mana rute tersebut harus dapat menerima *method* GET dan POST sekaligus. Terlihat pada Gambar 4.10 bahwa pembuatan *vectorstore* akan dijalankan ketika pengguna mengunggah dokumen. Apabila tidak sedang mengunggah, akan diberikan tampilan halaman saja dengan tambahan informasi dokumen yang sudah pernah diunggah. Tentu pada tampilan halaman akan diberikan kondisi *if* untuk menampilkan informasi dokumen apabila telah tersedia dokumen sebelumnya. Seperti yang terlihat pada contoh kode halaman pengunggahan dokumen di

lampiran nomor 2, pesan dari gagalnya validasi juga dapat ditampilkan pada halaman. Dengan adanya pesan validasi, dapat menginformasikan kepada pengguna akan kesalahan pengunggahan yang telah dilakukan.

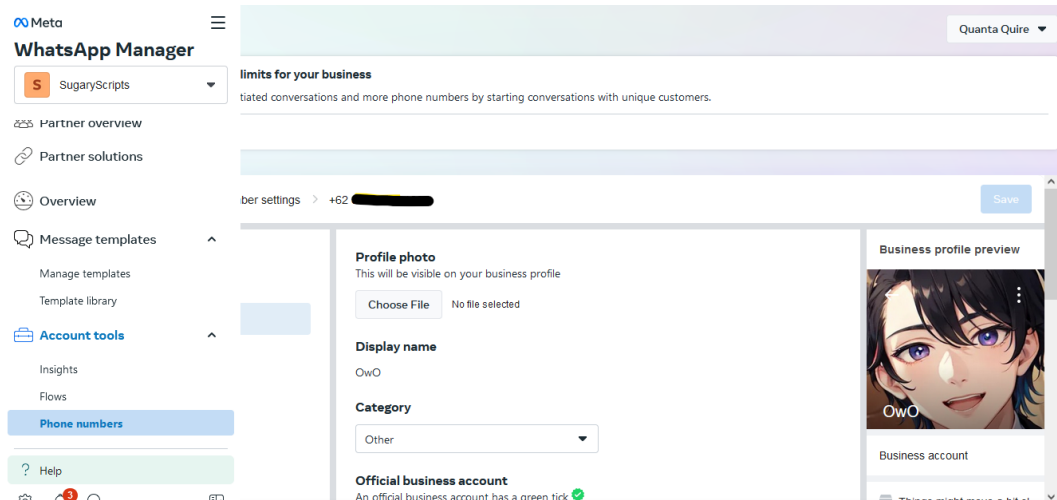
#### **4.1.4 Integrasi Sistem dengan WhatsApp**

Integrasi WhatsApp ke dalam sistem secara garis besar dibutuhkannya beberapa hal yang perlu dilewati terlebih dahulu. Dimulai dengan pembuatan akun WhatsApp, pembuatan *webhook*, sampai verifikasinya pada aplikasi untuk dapat berkomunikasi dengan WhatsApp. Namun, agar *webhook* berhasil terintegrasi dengan baik, sistem harus sudah di-*deploy* pada server. Karena Meta akan mengirimkan verifikasi kepada sistem setiap pengembang melakukan konfigurasi *webhook*. Di mana, verifikasi tidak akan terkirim apabila sistem hanya bisa diakses melalui jaringan lokal saja.

##### **4.1.4.1 Pembuatan Akun WhatsApp**

Sesuai yang ada pada Bab 3, untuk dapat mengintegrasikan WhatsApp ke sistem, pada penelitian ini membutuhkan minimal sebuah nomor telepon. Dan nomor telepon tersebut harus belum terdaftar pada akun WhatsApp pada WhatsApp Business Platform. Di mana langkah-langkah untuk membuat akun WhatsApp adalah sebagai berikut.

1. Membuat akun Meta Business Portfolio
2. Membuat aplikasi berjenis bisnis dari akun Meta Business Portfolio yang telah dibuat
3. Menambahkan nomor telepon pada menu WhatsApp Manager di Meta Business Portfolio



Gambar 4.11 Contoh Hasil Pembuatan Akun WhatsApp

Pada Gambar 4.11 merupakan contoh apabila akun WhatsApp telah berhasil dibuat. Nama aplikasi bisnis beserta dengan nama akun Meta Business Portfolio juga dapat dilihat pada Gambar 4.11 Di mana pada penelitian ini, nama akun Meta Business Portfolio yang dibuat adalah SugaryScripts serta aplikasi bisnisnya bernama Quanta Quire. Nomor yang sudah terdaftar tidak bisa dibuka dengan aplikasi WhatsApp pada umumnya. Jadi, perubahan informasi akun WhatsApp seperti foto profil atau deskripsi hanya bisa diubah melalui WhatsApp Manager seperti pada Gambar 4.11.

#### 4.1.4.2 Penyediaan Rute Webhook

Setiap kali setiap kali pengembang melakukan konfigurasi *webhook*, Meta akan mengirimkan verifikasi dengan *method* GET ke rute *webhook* aplikasi yang telah disediakan. Namun, pesan yang dikirim oleh pengguna WhatsApp kepada nomor yang telah terpasang, juga akan dikirimkan melalui rute yang sama dengan *method* POST. Maka dari itu, *webhook* dapat diimplementasikan dengan hanya membuat sebuah rute yang akan berperilaku berbeda berdasarkan *method* yang diterima.

1	@blueprint.route("/webhook", methods=["POST", "GET"])
2	def webhook():
3	if request.method == "GET":
4	return verify(request)
5	

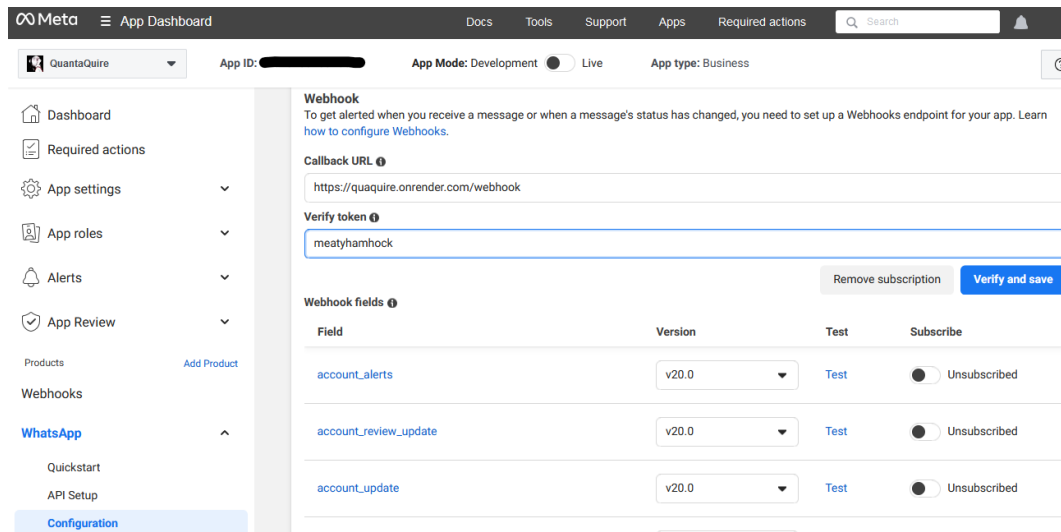
6	<code>elif request.method == "POST":</code>
7	<code>    return handle_message(request)</code>
8	<code>def verify(request):</code>
9	<code>    mode = request.args.get("hub.mode")</code>
10	<code>    token = request.args.get("hub.verify_token")</code>
11	<code>    challenge = request.args.get("hub.challenge")</code>
12	<code>    if mode and token:</code>
13	<code>        if (mode == "subscribe" and</code>
14	<code>            token == current_app.config['VERIFY_TOKEN']):</code>
15	<code>            return challenge, 200</code>
16	<code>        else:</code>
17	<code>            # return Gagal 403</code>
18	<code>    else:</code>
19	<code>        # return Parameter tidak lengkap 400</code>

Gambar 4.12 Contoh Kode Verifikasi Webhook

Pada sistem, telah tersedia sebuah variabel bernama *VERIFY\_TOKEN* yang dapat menyimpan teks apapun dengan tipe data *string*. Pada penelitian ini, variabel tersebut disimpan pada berkas bernama *.env* sebagai salah satu bentuk keamanan. Pada saat menyambungkan *webhook* pada dasbor aplikasi di Meta, isi teks variabel tersebut harus sama dengan teks yang akan diberikan. Berdasarkan dokumentasi yang telah disediakan oleh Meta (Getting Started - Meta Webhooks, 2024), informasi verifikasi dikirimkan melalui URL parameter. Maka dari itu, fungsi *verify* akan mengekstrak data dari parameter dan memberikan tanggapan dengan kode status sebagai konfirmasi atas keberhasilan melakukan verifikasi.

#### 4.1.4.3 Verifikasi Webhook

Setelah rute *webhook* dan teks untuk verifikasi telah tersedia pada sistem, verifikasi bisa dilakukan secara langsung pada dasbor aplikasi di Meta. Namun, tentu keberhasilan verifikasi hanya bisa diperoleh apabila sistem telah dapat diakses oleh Meta melalui internet.



Gambar 4.13 Verifikasi Webhook pada Dasbor Aplikasi

Pada Gambar 4.13 merupakan cara verifikasi *webhook* pada dasbor aplikasi di Meta. Teks yang berada pada bidang *verify token* harus sama dengan teks yang telah disiapkan pada sistem. Apabila berhasil melakukan verifikasi, pengguna dapat *subscribe* ke objek yang telah disediakan dengan klik tombol *switch* di kolom *Subscribe* pada Gambar 4.13. Sedangkan untuk penelitian ini, objek yang diperlukan hanya *message*. Sehingga dengan berlangganan objek tersebut, sistem dapat menerima setiap pesan yang dikirimkan oleh pengguna kepada nomor yang telah dipasang pada Bab 4.1.4.1. Di mana, dapat diolah lebih lanjut oleh fungsi *handle\_message* pada Gambar 4.12 untuk percakapan pada sistem.

#### 4.1.4.4 Pengiriman Pesan WhatsApp

Agar ekstraksi data berhasil, diperlukannya informasi mengenai bentuk data yang akan didapat dari WhatsApp API. Sehingga dengan mengetahui struktur data yang didapat, pengembang dapat membedah informasi yang didapat untuk diolah lebih lanjut. Di mana, contoh muatan notifikasi pesan yang diterima *webhook* dapat terlihat pada Gambar 4.10.

1	{ "object": "whatsapp_business_account",
2	"entry": [{
3	"id": "<WHATSAPP_BUSINESS_ACCOUNT_ID>",
4	"changes": [{
5	"value": {
6	"messaging_product": "whatsapp",



```

7      "metadata": {
8          "display_phone_number":
9      "<BUSINESS_DISPLAY_PHONE_NUMBER>",
10         "phone_number_id": "<BUSINESS_PHONE_NUMBER_ID>"
11     }, "contacts": [{
12         "profile": {
13             "name": "<WHATSAPP_USER_NAME>"
14         }, "wa_id": "<WHATSAPP_USER_ID>"
15     }],
16     "messages": [{
17         "from": "<WHATSAPP_USER_PHONE_NUMBER>",
18         "id": "<WHATSAPP_MESSAGE_ID>",
19         "timestamp": "<WEBHOOK_SENT_TIMESTAMP>",
20         "text": {
21             "body": "<MESSAGE_BODY_TEXT>"
22         }, "type": "text"
23     }]
24     }, "field": "messages"
25     }]]]}

```

Gambar 4.14 Kode Contoh Payload Webhook Notifikasi Pesan Masuk

Sumber : <https://developers.facebook.com/docs/whatsapp/cloud-api/webhooks/payload-examples#text-messages>

Seperti pada Gambar 4.14 terdapat beberapa informasi yang diperoleh ketika mendapatkan pesan masuk melalui *webhook*. Sedangkan pada penelitian ini, informasi yang akan digunakan hanya nomor telepon dan isi pesan yang diterima. Seperti yang terlihat pada Gambar 4.12, pesan yang diterima akan diproses oleh fungsi *handle\_message* yang detail kodenya terdapat pada Gambar 4.15.

```

1  def handle_message(body):
2      message = body["entry"][0]["changes"][0]["value"]["messages"][0]
3      if message["type"] == "text":
4          message_body = message["text"]["body"]
5          response, ask_feedback = chat(message_body, message["from"])
6          send_whatsapp_message(body, response)
7          if ask_feedback is not None:
8              send_whatsapp_message(body, ask_feedback)
9  def send_whatsapp_message(body, message):
10     value = body["entry"][0]["changes"][0]["value"]

```

```

11 phone_number_id = value["metadata"]["phone_number_id"]
12 from_number = value["messages"][0]["from"]
13 headers = {"Authorization":
14 f"Bearer{current_app.config['WHATSAPP_TOKEN']}",
15     "Content-Type": "application/json",}
16 url = "https://graph.facebook.com/v20.0/" + phone_number_id +
17 "/messages"
18 data = {
19     "messaging_product": "whatsapp",
20     "recipient_type": "individual",
21     "to": from_number,
22     "type": "text",
23     "text": {"body": message},
24 }
25 response = requests.post(url, json=data, headers=headers)
26 response.raise_for_status()

```

Gambar 4.15 Kode Mengirimkan Pesan WhatsApp

Sebagaimana dari panduan API dari Meta, pengiriman pesan akan membutuhkan sebuah token sebagai *authorization* pesan. Di mana, token tersebut dapat disimpan pada berkas bernama `.env` sebagai keamanan. Token bisa didapatkan di dasbor aplikasi pada menu WhatsApp – API Setup. Token tersebut bersifat sementara yang akan kadaluwarsa dalam waktu kurang dari 24 jam. Pada Gambar 4.11 juga terlihat bahwa sistem akan mengirimkan pertanyaan umpan balik jika variabel `ask_feedback` tidak kosong. Di mana, proses detail fungsi *chat* dapat dilihat pada Bab 4.1.5.

#### 4.1.5 Sistem Percakapan dan Umpan Balik

Penilaian pengguna terhadap tanggapan yang diterima sangat penting untuk disimpan sebagai pengujian kualitas sistem. Di mana, percakapan beserta dengan nilai dari percakapan tersebut akan disimpan menjadi satu baris data setiap tanggapan. Hal tersebut dapat dicapai dengan contoh tabel *database* seperti pada Gambar 4.12.

```

1 class ChatLog(db.Model):
2     id = db.Column(db.Integer, primary_key=True)
3     timestamp = db.Column(db.DateTime, default=datetime.utcnow)

```

4	<code>user = db.Column(db.String)</code>
5	<code>question = db.Column(db.Text)</code>
6	<code>answer = db.Column(db.Text)</code>
7	<code>point = db.Column(db.Integer, default=None)</code>

Gambar 4.16 Kode Model Class untuk Percakapan

Karena *database* yang digunakan adalah PostgreSQL, pembuatan *database* beserta dengan *table*-nya bisa dilakukan dengan menjalankan perintah *db.create\_all* dari *shell*. Di mana, *db* yang dimaksud merupakan sebuah variabel yang diinisiasi dari SQLAlchemy. Sehingga, *class* ChatLog pada Gambar 4.16 dapat digunakan setiap kali pengembang ingin membaca maupun menulis data percakapan dari *database*.

1	<code>def chat(session_id, message):</code>
2	<code>    ask_feedback = ("Apakah jawaban saya sudah tepat?....")</code>
3	<code>    try: # Check if the message is a feedback number</code>
4	<code>        number = int(message)</code>
5	<code>        if 0 &lt;= number &lt;= 3: # Save the feedback</code>
6	<code>            if session_id not in current_app.chats:</code>
7	<code>                return get_random_response("feedback_first"), None</code>
8	<code>            save_chat_feedback(session_id=session_id, point=message)</code>
9	<code>            return get_random_response("feedback"), None</code>
10	<code>    except ValueError:</code>
11	<code>        pass</code>
12	<code>    save_chat_log(session_id, message)</code>
13	<code>    return qna(session_id, message), ask_feedback</code>
14	<code>def qna(session_id, message):</code>
15	<code>    try:</code>
16	<code>        vectorstore_path =</code>
17	<code>os.path.join(current_app.config['UPLOAD_PATH'], 'vectorstore')</code>
18	<code>        if os.path.exists(vectorstore_path) and</code>
19	<code>os.path.isdir(vectorstore_path):</code>
20	<code>            retriever = faiss_load_vectorstore().as_retriever()</code>
21	<code>            return rag_chat(retriever, session_id, message)</code>
22	<code>        else:</code>
23	<code>            return basic_chat(session_id, message)</code>
24	<code>    except Exception as e:</code>
25	<code>        return "Silahkan coba lagi nanti."</code>

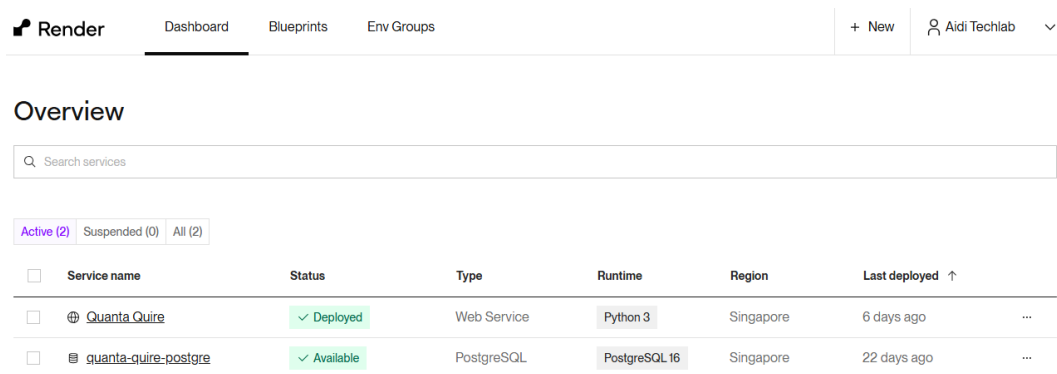
Gambar 4.17 Contoh Kode Percakapan dengan Umpan Balik

Secara garis besar, umpan balik akan disimpan pada percakapan yang terakhir kali diterima. Apabila pengguna tidak memberikan umpan balik pada percakapan sebelumnya, sistem akan langsung menyimpan percakapan yang baru tanpa memasukkan umpan balik ke percakapan sebelumnya. Hal tersebut dapat dicek dengan melakukan *parse* variabel ke tipe data *integer*. Kegagalan terhadap proses *parsing* merupakan tanda bahwa data variabel yang diterima adalah teks. Meskipun proses *parsing* berhasil, sistem akan mengecek angka yang masuk merupakan angka yang sesuai dengan instruksi pemberian umpan balik atau tidak.

Sedangkan untuk tanggapan dari AI dapat dilihat pada Gambar 4.17. Yang mana, sistem akan melakukan pengecekan ada atau tidaknya *vectorstore* yang tersimpan. Lalu, sistem akan memanfaatkan fungsi yang ada pada lampiran nomor 1 dengan *vectorstore* yang telah ditemukan. Sedangkan kode secara keseluruhan untuk sistem percakapan dan umpan balik dapat dilihat pada lampiran nomor 3.

## 4.2 Peluncuran Aplikasi

Pada penelitian ini, server yang digunakan untuk *deploy* sistem aplikasi beserta dengan *database* PostgreSQL merupakan Render. Namun, sistem aplikasi harus diunggah di penyedia Git seperti GitHub atau GitLab terlebih dahulu. Meskipun pembuatan akun Render berbayar, *hosting web service* maupun *database* PostgreSQL dapat digunakan secara gratis.



The screenshot shows the Render dashboard with the 'Overview' tab selected. It displays a table of services. The first service is 'Quanta Quire', a Web Service using Python 3 runtime, deployed 6 days ago. The second service is 'quanta-quire-postgre', a PostgreSQL database using PostgreSQL 16 runtime, available for 22 days.

Service name	Status	Type	Runtime	Region	Last deployed ↑
Quanta Quire	✓ Deployed	Web Service	Python 3	Singapore	6 days ago
quanta-quire-postgre	✓ Available	PostgreSQL	PostgreSQL 16	Singapore	22 days ago

Gambar 4.18 Hasil Deploy Sistem Aplikasi Percakapan dan Database

Pada Gambar 4.18 merupakan contoh apabila sistem aplikasi beserta dengan *database*-nya berhasil di-*deploy*. Hal yang perlu diperhatikan ketika melakukan

*deploy* sistem aplikasi adalah konfigurasi data yang ada pada berkas *.env* atau *environment variables*. Pada umumnya *.env* merupakan berkas yang berada pada daftar *.gitignore*, yang berarti tidak akan terunggah pada Git Provider. Oleh karena itu, diperlukan untuk menyalin konfigurasi yang ada pada *.env* ke tempat *.env* yang telah disediakan oleh Render pada penelitian ini. Selain itu, karena token WhatsApp yang digunakan adalah bersifat sementara, diperlukannya juga untuk selalu mengubah *.env* token WhatsApp yang ada pada Render dengan token dari dasbor aplikasi pada Meta. Selain itu, verifikasi *webhook* pada Bab 4.1.4.3 bisa dilakukan mengingat sistem telah dapat diakses melalui internet. Dengan berhasilnya pembuatan PostgreSQL pada Render, pembuatan *database* beserta tabelnya juga bisa dilakukan melalui *shell* dari flask.

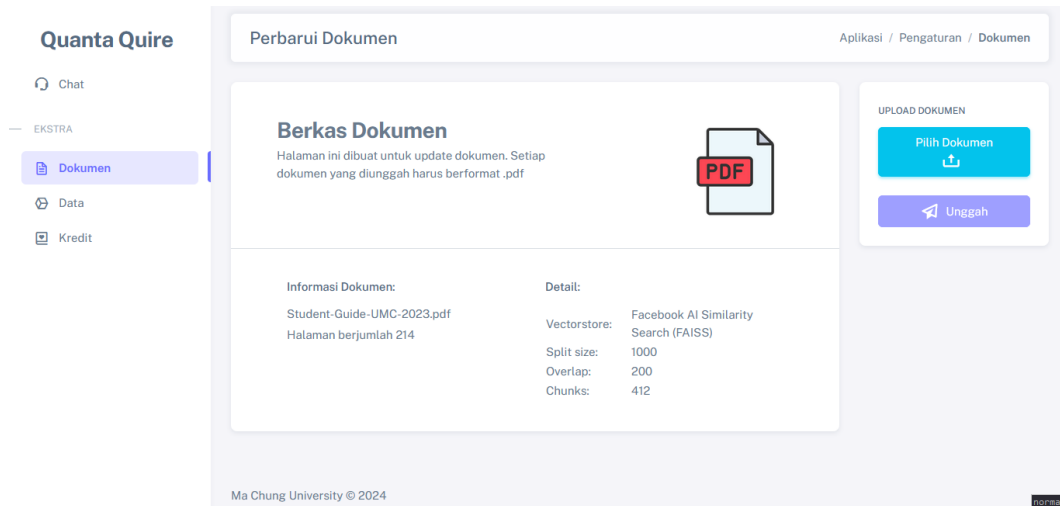
### **4.3 Hasil Pengembangan Aplikasi**

Sistem aplikasi ini mengimplementasikan dua antarmuka pengguna, yaitu aplikasi WhatsApp dan aplikasi berbasis web. Di mana, keduanya dirancang untuk melayani target pengguna yang berbeda juga. Aplikasi WhatsApp ditujukan untuk pengguna umum yang berinteraksi langsung dengan sistem, memfasilitasi pencarian informasi yang bisa berupa tanya jawab kepada *chatbot*. Sementara itu, antarmuka berbasis web dikembangkan khusus untuk pihak internal organisasi atau staf perusahaan. Antarmuka web ini menyediakan fungsi memantau dan mengelola dokumen sebagai sumber pengetahuan yang diambil oleh *chatbot*, serta memungkinkan staf untuk memantau kinerja sistem dan menganalisis data dengan melihat semua data obrolan yang masuk. Pemisahan antarmuka ini bertujuan untuk mengoptimalkan pengalaman pengguna bagi kedua kelompok target, dengan menyediakan fitur dan fungsionalitas yang relevan dengan kebutuhan masing-masing.

#### **4.3.1 Aplikasi Web**

Untuk memfasilitasi monitoring, analisis data, dan manajemen sistem secara menyeluruh, dikembangkanlah sebuah aplikasi berbasis web. Bagian ini akan menguraikan detail hasil aplikasi web, termasuk fitur-fitur yang mendukung visualisasi data berupa tabel, mengganti dokumen, dan interaksi dengan chatbot

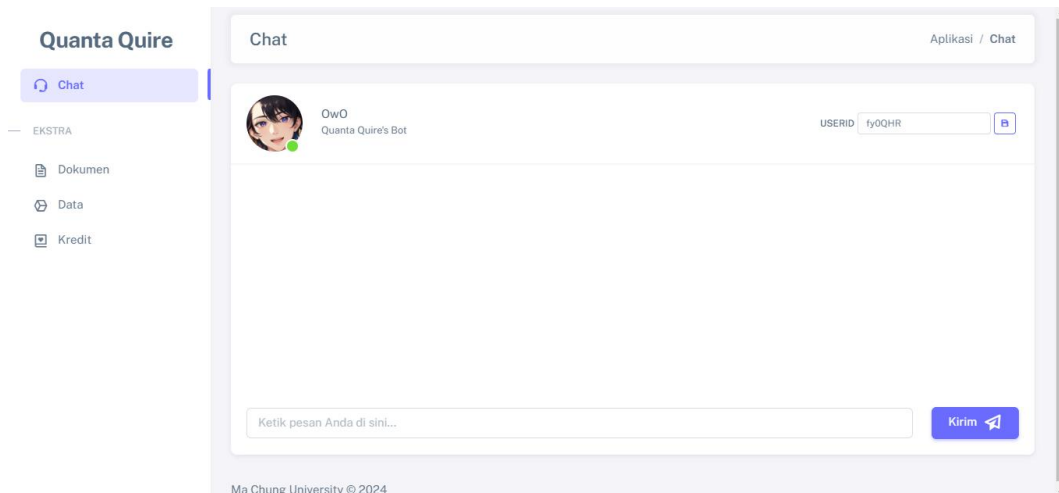
sebagai gambaran perilaku *chatbot* ketika mengobrol dengan mahasiswa nanti, yang krusial bagi staf internal.



Gambar 4.19 Contoh Hasil Halaman Mengunggah Dokumen

Gambar 4.19 memperlihatkan antarmuka web yang digunakan staf internal untuk mengunggah dokumen. Sistem secara otomatis memperbarui *vectorstore* setiap kali dokumen baru diunggah atau terjadi pergantian dokumen. Informasi detail mengenai fragmentasi dokumen ke dalam vektor seperti banyak karakter dan hasil total fargmen dokumen akan ditampilkan kepada pengguna setelah proses pembuatan *vectorstore* selesai. Jika tidak ada dokumen yang terdeteksi, sistem akan menampilkan pesan informatif yang menyatakan bahwa dokumen belum diunggah. Sehingga pengguna dapat mengambil tindakan yang diperlukan agar *chatbot* berfungsi dengan semestinya.

Antarmuka web yang diilustrasikan pada Gambar 4.19 dilengkapi dengan menu navigasi vertikal di sisi kiri, yang menyediakan akses langsung ke berbagai bagian aplikasi. Indikator halaman aktif yang ditampilkan di sisi kanan atas antarmuka memperkuat orientasi pengguna dengan secara jelas menunjukkan halaman yang sedang diakses. Kombinasi ini memungkinkan pengguna untuk menavigasi aplikasi dengan cepat dan efisien, serta meminimalkan potensi kebingungan dan meningkatkan *user experience*.



Gambar 4.20 Hasil Halaman Chat pada Website

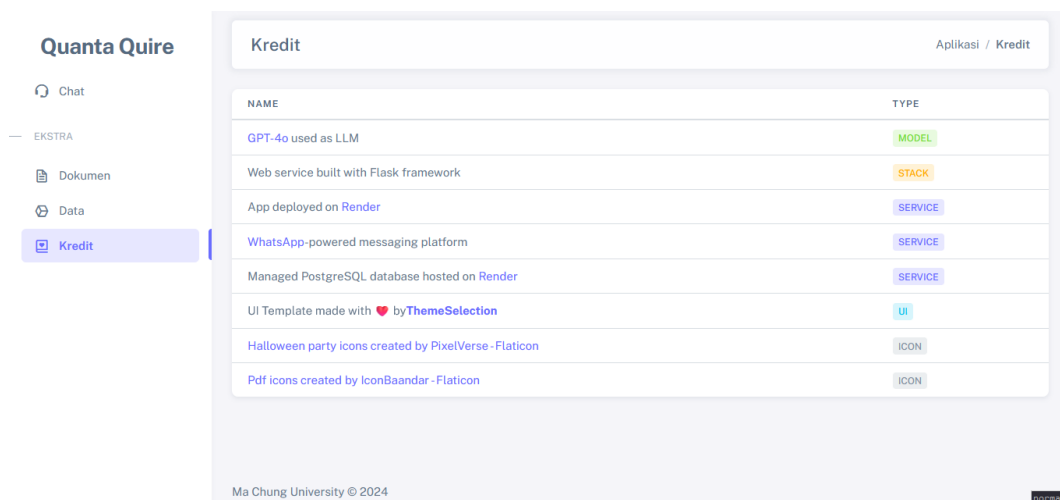
Aplikasi web juga menyediakan fitur untuk berinteraksi langsung dengan *chatbot*, memungkinkan pengembang untuk menguji dan memvalidasi perilaku *chatbot* dalam simulasi interaksi pengguna. Untuk membedakan sesi pengujian pada web dengan interaksi melalui aplikasi WhatsApp, sistem menggunakan pengidentifikasi bernama *userid* yang ditampilkan di sisi kanan atas halaman. Penggunaan *userid* ini memungkinkan identifikasi sesi pengujian tanpa memerlukan informasi pribadi seperti nomor telepon. Sistem akan secara otomatis menghasilkan *userid* baru dengan acak jika tidak ditemukan *userid* yang tersimpan pada peramban.

ID	TIMESTAMP	USER	QUESTION	ANSWER	CATEGORY
1	2024-08-03 08:34:19.084652	62[REDACTED]	hi	Halo! Bagaimana saya bisa membantu Anda hari ini?	None
2	2024-08-05 11:44:59.589291	62[REDACTED]	hi	Halo! Bagaimana saya bisa membantu Anda hari ini?	None
3	2024-08-05 12:02:02.166027	62[REDACTED]	halo	Halo! Ada yang bisa saya bantu?	None
4	2024-08-05 12:02:48.943797	62[REDACTED]	berapa minimal bimbingan untuk seminar proposal	Minimal bimbingan untuk seminar proposal tidak disebutkan secara spesifik dalam konteks yang ada. Jika Anda membutuhkan informasi lebih detail, sebaiknya Anda menghubungi Ketua Program Studi atau dosen pembimbing Anda.	3
5	2024-08-05 12:09:03.976097	62[REDACTED]	test	Halo! Bagaimana saya bisa membantu Anda hari ini?	None

Gambar 4.21 Hasil Halaman Data pada Website

Data percakapan yang tersimpan dalam *database* dapat diakses melalui menu data pada antarmuka web. Sebagaimana telah dijelaskan pada Bab 3.4.3,

pemberian umpan balik oleh pengguna bersifat opsional. Di mana, ketidakhadiran umpan balik direpresentasikan sebagai nilai 'None' pada data percakapan. Dengan adanya fitur ini, staf dapat meninjau setiap percakapan yang terjadi pada sistem. Pada Gambar 4.21 telah terlihat bahwa selain pertanyaan dan jawaban, nomor telepon juga akan disimpan. Nomor telepon pengguna disimpan untuk memfasilitasi tindak lanjut jika terdeteksi kesalahan informasi yang diberikan oleh *chatbot*. Sehingga staf dapat menghubungi pengguna dan memberikan klarifikasi atau bantuan teknis yang diperlukan.



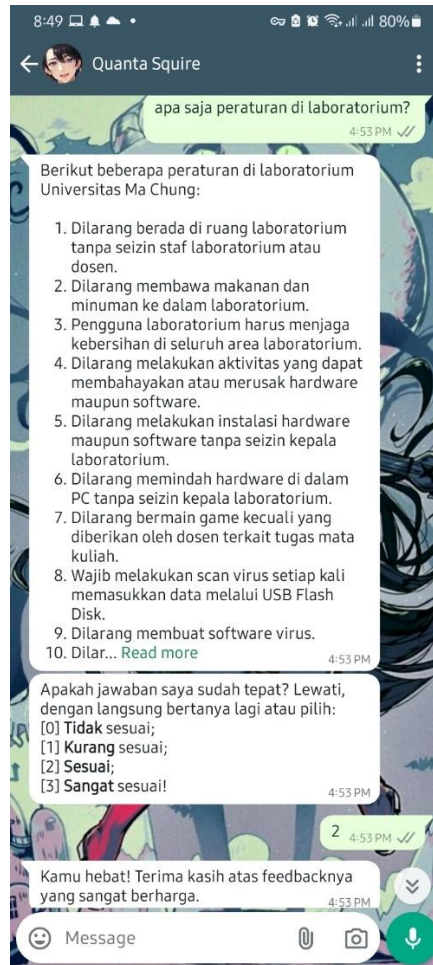
Gambar 4.22 Hasil Halaman Kredit pada Website

Fitur terakhir pada aplikasi web adalah kredit. Halaman kredit menampilkan daftar kredit untuk memberikan referensi terhadap kumpulan aset yang digunakan dalam pengembangan. Tampilan seperti nama dokumen atau *library* vectorstore yang digunakan juga dapat ditambahkan sebagai informasi tambahan. Setiap referensi memiliki label kategori sendiri untuk memberikan informasi kepada pengguna tujuan memanfaatkan alat atau aset yang dirujuk.

#### 4.3.2 Aplikasi WhatsApp

Mempertimbangkan popularitas dan kemudahan akses WhatsApp, aplikasi ini dipilih sebagai antarmuka bagi pengguna umum untuk berinteraksi dengan sistem. Bagian ini akan membahas hasil dari integrasi sistem dengan aplikasi WhatsApp, termasuk integrasinya dengan chatbot serta fitur umpan balik sebagai salah satu penentu keberhasilan chatbot dalam memberikan jawaban yang sesuai.





Gambar 4.23 Contoh Interaksi Pengguna Melalui WhatsApp

Untuk pengguna umum yang akan berinteraksi dengan sistem melalui WhatsApp dapat terlihat pada Gambar 4.23. Pengguna dapat bertanya secara langsung dengan akun chatbot dari nomor telepon yang sebelumnya telah ditentukan sebagai media chatbot dalam berinteraksi. Sistem dirancang untuk memberikan respons terhadap setiap pertanyaan pengguna. Setelah memberikan respons, sistem akan memberikan opsi bagi pengguna untuk memberikan umpan balik. Umpan balik ini bersifat opsional dan bertujuan untuk mengevaluasi kualitas respons *chatbot*. Pengguna dapat melanjutkan interaksi dengan mengajukan pertanyaan baru tanpa memberikan umpan balik, namun data umpan balik yang telah maupun belum diberikan sebelumnya tidak dapat dimodifikasi setelah pertanyaan baru diajukan. Karena sistem akan otomatis mengenali perubahan umpan balik ketika pesan bukan bertipe pertanyaan. Jadi, pengguna dapat langsung memodifikasi umpan balik dengan memberikan angka umpan balik terbaru

selayaknya memberikan umpan balik pada pertama kalinya. Sehingga perintah tambahan tidak diperlukan oleh pengguna dalam memodifikasi umpan baliknya.

#### 4.4 Pengujian Sistem Aplikasi

Sebagai validasi keberhasilan implementasi dan fungsionalitas sistem yang diusulkan, berikut telah disajikan visualisasi dari fitur-fitur yang telah diintegrasikan. Validasi setiap fitur dilakukan dengan harapan tidak menemui *bug* atau fitur yang tidak bekerja sesuai dengan harapan. Serta perbaikan dapat ditindak lanjuti ketika menemui hal yang tidak diharapkan pada sistem.

Tabel 4.1 Tabel Evaluasi Hasil Sistem Aplikasi

Aplikasi	Fitur	Berfungsi
Website	Dokumen	Unggah Dokumen
		Mengganti Dokumen
		Indikator Keberadaan Dokumen
	Data	Menampilkan Semua Data
		Menyimpan data obrolan
		Menyimpan nomor setiap obrolan
		Menyimpan userid dari obrolan <i>website</i>
	Chat	Mengobrol
		Mengganti userid
		Memberikan umpan balik
		Modifikasi umpan balik
	Kredit	Menampilkan Halaman Kredit
WhatsApp	Chat	Mengobrol
		Memberikan umpan balik
		Modifikasi umpan balik

Evaluasi terhadap sistem aplikasi, yang dirangkum dalam Tabel 4.1, menunjukkan bahwa seluruh fitur telah sesuai secara fungsionalitas dengan desain awal. Meskipun demikian, pengujian mengungkapkan adanya latensi yang perlu

diperhatikan. Analisis lebih lanjut mengindikasikan bahwa latensi ini disebabkan oleh karakteristik layanan *hosting* gratis yang digunakan. Layanan Render menyebutkan bahwa layanan *hosting* bertipe gratis ini mengimplementasikan mekanisme *server idling*, di mana *server* akan dinonaktifkan sementara setelah periode tidak aktif selama 15 menit (Deploy for Free – Render Docs, 2024). Proses aktivasi ulang *server* ini mengakibatkan penundaan respons aplikasi, dengan rentang waktu antara 30 detik hingga lima menit. Dampak penundaan ini juga dirasakan oleh pengguna WhatsApp, yang mengalami keterlambatan dalam menerima respons dari *chatbot*.

#### 4.5 Analisis Hasil Kinerja Chatbot

Untuk mengevaluasi hasil dari implementasi metode RAG dengan chatbot, survei telah dilakukan terhadap mahasiswa, dosen, dan staf Universitas Ma Chung. Survei ini bertujuan untuk mengukur tingkat keberhasilan dan akurasi chatbot dalam mengakses informasi dari dokumen. Informasi yang diberikan kepada *chatbot* dalam sistem ini bersumber dari dokumen berformatkan PDF berjudul Student Guide 2023.

1	[
2	{
3	"id": 109,
4	"timestamp": "2024-08-07T07:29:52.364169",
5	"user": "62xxxx",
6	"question": "Toeic score untuk kelulusan?",
7	"answer": "Untuk kelulusan, setiap mahasiswa harus
8	mencapai skor TOEIC minimal 500.",
9	"category": 3,
10	},...
	]

Gambar 4.24 Contoh Sebuah Data Tanya Jawab

Peneliti melakukan pembersihan terhadap data percakapan yang terkumpul, yang berjumlah 110 pasangan tanya jawab. Pembersihan ini bertujuan untuk meningkatkan kualitas data yang digunakan dalam evaluasi performa *chatbot*

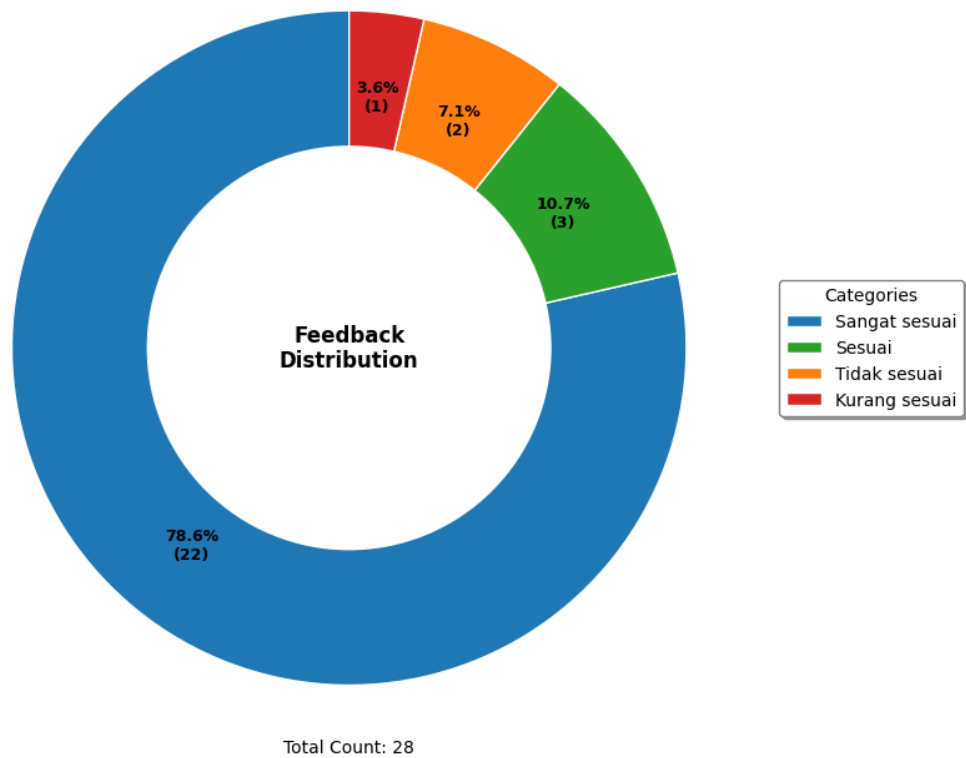
berbasis RAG. Kriteria penghapusan data didasarkan pada relevansi terhadap konteks survei. Data yang dianggap tidak relevan dan dihapus meliputi,

- Data tidak lengkap seperti sapaan hai atau halo
- Percakapan di luar konteks survei khususnya dalam lingkup Universitas Ma Chung seperti pertanyaan menyelesaikan matematika atau tentang bursa.

Setelah proses pembersihan, 74 data tanya jawab dinyatakan valid dan siap digunakan untuk tahapan evaluasi. Data ini akan menjadi dasar untuk mengukur performa chatbot dalam merespons pertanyaan berdasarkan Student Guide 2023, menganalisis efektivitas implementasi RAG, serta mengidentifikasi area perbaikan pada chatbot.

#### **4.5.1 Umpan Balik**

Berdasarkan rancangan pengujian yang menetapkan umpan balik sebagai opsional, tidak semua percakapan menghasilkan umpan balik. Untuk memberikan gambaran yang komprehensif, akan ditampilkan dua jenis visualisasi: yang pertama, distribusi kategori umpan balik dari seluruh percakapan untuk menunjukkan proporsi percakapan yang menerima umpan balik; dan yang kedua, distribusi kategori umpan balik dari percakapan yang benar-benar menerima umpan balik untuk menganalisis kecenderungan umpan balik yang diberikan.



Gambar 4.25 Diagram Pie Chart Distribusi Umpan Balik dari Semua Tanya Jawab yang Menerima Umpan Balik Saja

Dari 9 responden dengan total 74 percakapan, didapatkan percakapan dengan umpan balik berjumlah 28. Visualisasi berupa *Pie Chart* (Gambar 4.25) memfokuskan pada distribusi kategori umpan balik dari setiap pasangan tanya jawab yang menerima umpan balik. Di mana, pasangan tanya jawab yang tidak menerima umpan balik akan dikecualikan pada visualisasi tersebut (Gambar 4.25). Dari 28 percakapan yang memberikan umpan balik, mayoritas (78.57%, 22 percakapan) memberikan umpan balik 'Sangat sesuai', menunjukkan bahwa sebagian besar jawaban *chatbot* dinilai sangat relevan dengan pertanyaan pengguna. Kategori 'Sesuai' mencakup 10.71% (3 percakapan), sedangkan kategori 'Tidak sesuai' dan 'Kurang sesuai' masing-masing mencakup 7.14% (2 percakapan) dan 3.57% (1 percakapan). Visualisasi ini memberikan analisis yang lebih detail mengenai distribusi umpan balik yang diberikan oleh pengguna, dengan mengabaikan proporsi percakapan yang tidak memberikan umpan balik.

Perlu dilakukan analisis lanjutan dengan meninjau langsung pertanyaan dan jawaban pada kategori 'tidak sesuai' dan 'kurang sesuai'. Analisis ini bertujuan untuk mengidentifikasi penyebab pengkategorian tersebut, yang mungkin disebabkan

oleh kurangnya akurasi *chatbot* atau kurangnya informasi yang relevan dalam dokumen Student Guide 2023 yang diharapkan oleh pengguna. Data lengkapnya dapat dilihat pada Tabel 4.2.

Tabel 4.2 Tabel Data Umpan Balik Kurang dan Tidak Sesuai

ID	Pertanyaan	Jawaban	Kategori
19	siapa itu Pak Wastam ?	Maaf, saya tidak memiliki informasi mengenai Pak Wastam di Universitas Ma Chung. Jika Anda memerlukan informasi lebih lanjut, Anda bisa menghubungi pihak administrasi universitas untuk mendapatkan keterangan yang lebih lengkap.	Tidak sesuai
105	Bagaimana cara menulis laporan PKL	<p>Untuk menulis laporan PKL (Praktik Kerja Lapangan), Anda perlu menyusun laporan sesuai dengan ketentuan yang ada. Berikut adalah struktur umum yang bisa diikuti:</p> <ol style="list-style-type: none"> <li>1. Latar belakang.</li> <li>2. Tujuan PKL.</li> <li>3. Sasaran hasil yang diharapkan.</li> <li>4. Organisasi pengelola.</li> <li>5. Pelaksanaan kegiatan.</li> <li>6. Pembiayaan.</li> <li>7. Hambatan.</li> <li>8. Upaya penanggulangan hambatan.</li> <li>9. Kesimpulan.</li> <li>10. Saran.</li> </ol> <p>Laporan ini kemudian disampaikan kepada pihak terkait, seperti Rektor dan Sponsor. Untuk informasi lebih</p>	Tidak sesuai

ID	Pertanyaan	Jawaban	Kategori
		lanjut, Anda dapat menghubungi Biro Kemahasiswaan, Pendidikan Karakter dan Kepemimpinan (BKA) Universitas Ma Chung.	
82	menurut panduan student guide, apa syarat agar bisa yudisium	Berdasarkan panduan, syarat untuk yudisium adalah: 1. Mengumpulkan $\geq 10$ kredit keaktifan mahasiswa. 2. Memiliki sertifikat HSK, sertifikat IT, atau UKBI. 3. Mempublikasikan karya ilmiah minimal di jurnal nasional tidak terakreditasi. 4. Memenuhi syarat administrasi dari Program Studi, Fakultas, dan Universitas. 5. Telah lulus secara akademis dan administratif.	Kurang sesuai

Analisis pada Tabel 4.2 mengidentifikasi tiga kasus jawaban yang dikategorikan sebagai 'tidak sesuai' (2 kasus) dan 'kurang sesuai' (1 kasus). Dari dua kasus 'tidak sesuai', satu kasus dengan ID 19 disebabkan oleh ekspektasi pengguna terhadap informasi yang *seharusnya* ada dalam Student Guide 2023, namun faktanya tidak tersedia. Kasus 'tidak sesuai' yang lainnya dengan ID 105 disebabkan oleh *chatbot* yang memberikan jawaban meskipun informasi yang relevan *tidak* terdapat dalam dokumen. Kemungkinan penyebabnya adalah representasi informasi yang kurang tepat saat diproses oleh *chatbot*, atau pada proses fragmentasi dokumen menjadi *chunks* yang kurang optimal, meskipun kemungkinan ini relatif kecil. Lalu pasangan tanya jawab yang terakhir dengan kategori 'kurang sesuai' disebabkan oleh jawaban *chatbot* yang sebenarnya tepat dan didukung oleh dokumen, namun pengguna tetap memberikan umpan balik 'kurang sesuai'. Hal tersebut bisa mengindikasikan bahwa pengguna mengharapkan informasi yang lebih detail dari jawaban yang telah didapat. Dengan demikian, dapat disimpulkan bahwa dua dari tiga kasus permasalahan berakar pada keterbatasan informasi dalam dokumen sumber, sedangkan satu kasus kemungkinan disebabkan oleh performa *chatbot*.

#### 4.5.2 Penilaian Akurasi Jawaban Chatbot

Untuk mendapatkan informasi lengkap mengenai akurasi chatbot, peneliti memfasilitasi perhitungan beberapa metrik untuk evaluasi *chatbot* dengan menambahkan variabel berjenis *boolean* pada setiap pasangan tanya jawab. Variabel ini merepresentasikan hasil evaluasi jawaban *chatbot* terhadap *ground truth* yang bersumber dari Student Guide 2023. Seperti yang sudah tertera pada Sub Bab Perancangan Pengujian, variabel metrik akan ditambahkan tersebut yaitu, *document\_contains\_answer*, *retrieved\_document\_relevance*, *answer\_correctness*, *refusal\_behavior*, dan *hallucination*

1	[
2	{
3	"id": 109,
4	"timestamp": "2024-08-07T07:29:52.364169",
5	"user": "62xxxx",
6	"question": "Toeic score untuk kelulusan?",
7	"answer": "Untuk kelulusan, setiap mahasiswa harus mencapai skor TOEIC minimal 500.",
8	"category": 3,
9	"document_contains_answer": true,
10	"retrieved_document_relevance": true,
11	"answer_correctness": true,
	"refusal_behavior": false,
	"hallucination": false
	},...
	]

Gambar 4.26 Contoh Data Tanya Jawab dengan Variabel Metrik Evaluasi

Gambar 4.26 merupakan salah satu hasil data lapangan yang telah ditambahkan metrik evaluasi. Dengan diberikannya metrik evaluasi tersebut untuk setiap respons yang didapat, kalkulasi akurasi dan analisis terhadap chatbot dapat dilakukan.



```

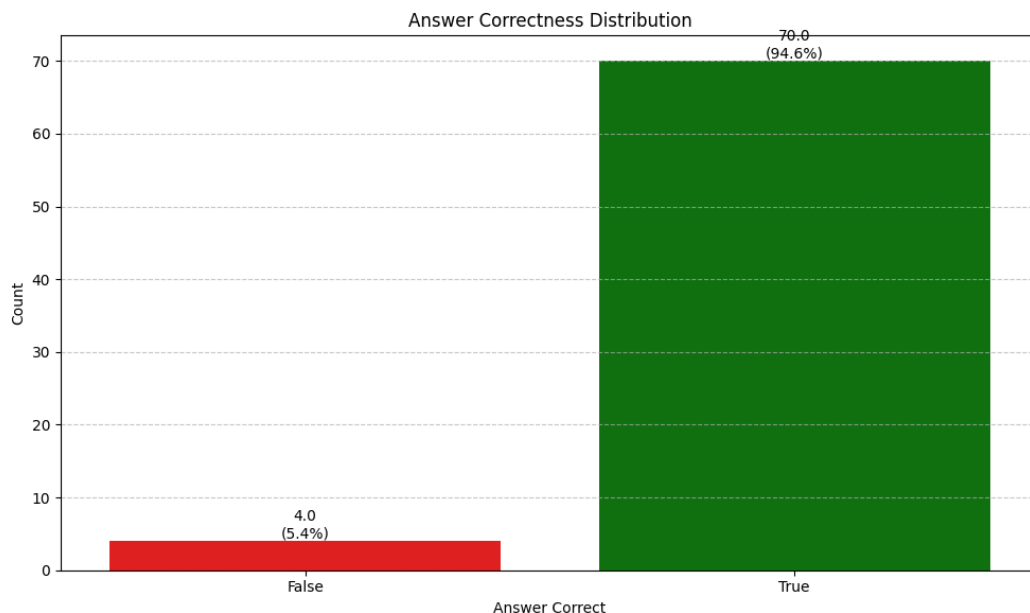
Total Responses: 74
Chatbot Performance Analysis Report
=====

Hallucinations: 3 (4.05%)
Total Refusals: 17 (22.97%)
  - Justified Refusals: 15 (88.24%)
  - Unjustified Refusals: 2 (11.76%)
Document Retrieval Accuracy: 72 (97.30%)
Correct Answers: 70 (94.59%)

```

Gambar 4.27 Kualitas Chatbot Secara Keseluruhan

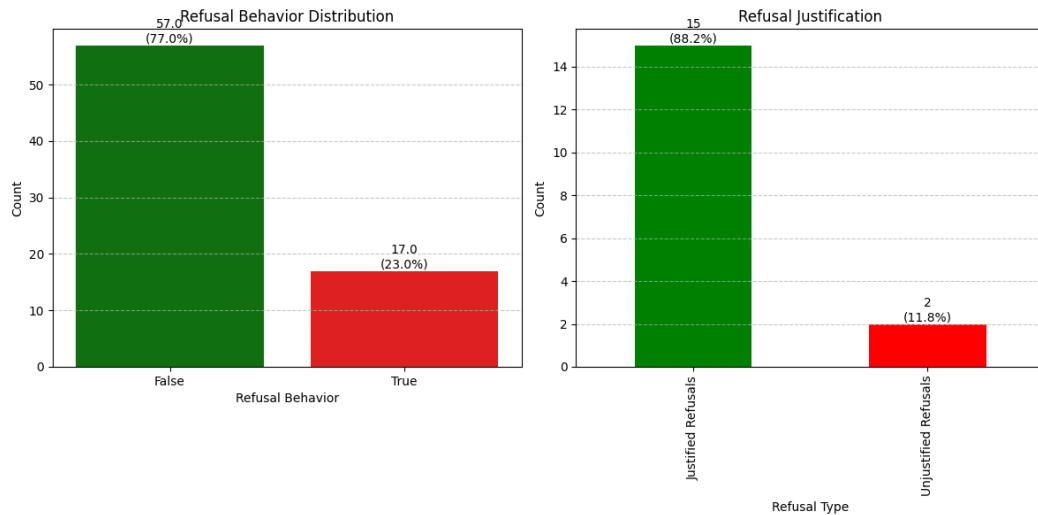
Kualitas *chatbot* secara keseluruhan, telah disajikan pada Gambar 4.27. Dari hasil tersebut, terlihat bahwa *chatbot* berhasil memberikan respons yang tepat dengan akurasi 94,59%. Hal tersebut didukung dengan tingginya keberhasilan yang dicapai *chatbot* dalam mendapatkan fragmen dokumen yang relevan terhadap pertanyaan, di mana mencapai 97,30%. *Chatbot* pun juga berhasil menolak menjawab pertanyaan yang tidak relevan dengan dokumen yang disediakan dengan persentase 88,24%. Meskipun demikian, fabrikasi jawaban oleh *chatbot* masih dapat ditemui dengan persentase 4,05%.



Gambar 4.28 Histogram Kebenaran Jawaban

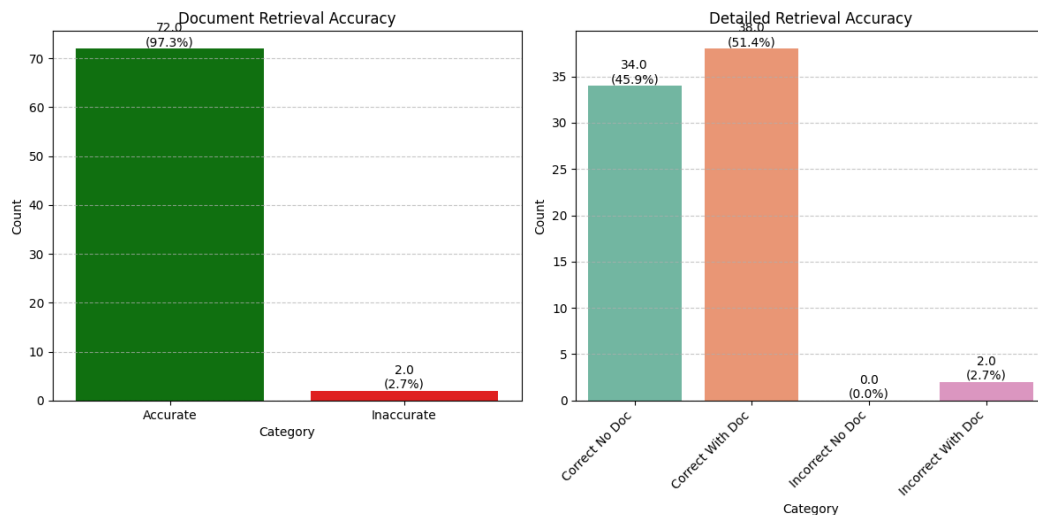
Berdasarkan histogram yang ditampilkan pada Gambar 4.28 terlihat bahwa 70 jawaban (94,59%) yang diberikan secara faktual benar. Sebaliknya, hanya 4 jawaban (5,4%) yang teridentifikasi salah. Proporsi jawaban benar yang tinggi ini

menunjukkan efektivitas *chatbot* dalam menyajikan informasi faktual, namun juga masih diperlukannya perbaikan lebih lanjut untuk meminimalisir memberikan jawaban tidak akurat.



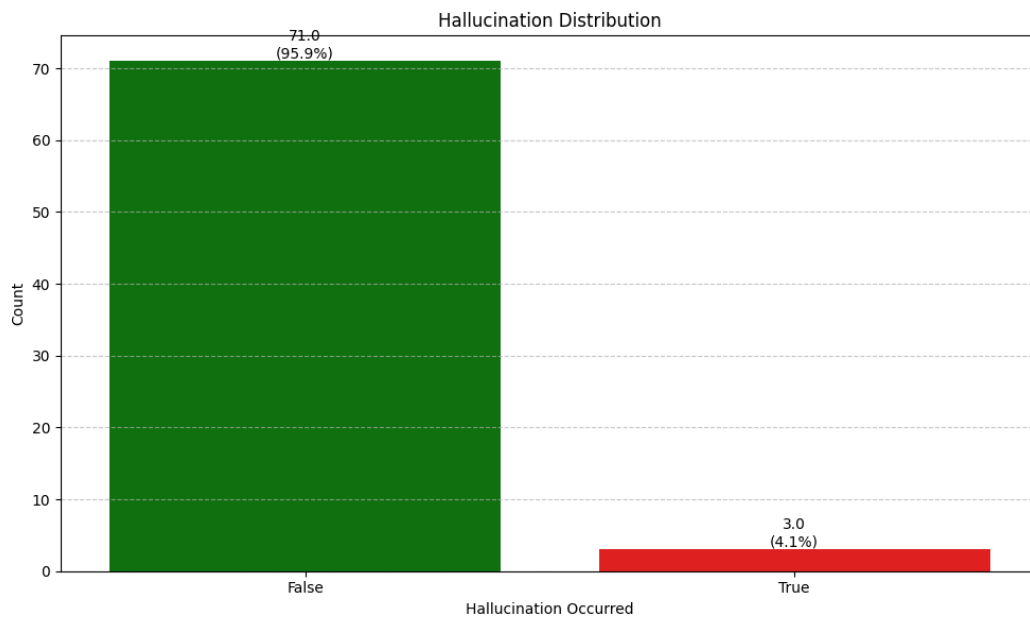
Gambar 4.29 Visualisasi Evaluasi Chatbot Menolak Untuk Menjawab

Hasil keakuratan chatbot dalam memberikan jawaban, juga bisa didapatkan dari chatbot secara benar memberikan jawaban dan juga menolak untuk menjawab. Di mana, visualisasi penolakan chatbot secara keseluruhan bisa terlihat pada Gambar 4.29. Dari total 17 penolakan (22,97%), sebagian besar, yaitu 15 penolakan (88,24%), dapat dibenarkan (*justified*). Penolakan yang dibenarkan ini mengindikasikan bahwa *chatbot* dengan tepat menolak menjawab pertanyaan karena informasi yang relevan memang tidak ditemukan dalam dokumen yang diakses. Sebaliknya, hanya 2 penolakan (11,76%) yang tidak dapat dibenarkan (*unjustified*), menunjukkan masih ditemuinya kegagalan oleh *chatbot* dalam mendapatkan fragmen dokumen yang dicari.



Gambar 4.30 Visualisasi Detail Akurasi Mendapatkan Fragmen Dokumen

Gambar 4.30 memperlihatkan analisis keakuratan chatbot dalam mendapatkan fragmen dokumen melalui dua grafik batang. Grafik pertama sebelah kiri, menunjukkan dominasi dokumen yang diambil secara akurat, dengan 70 dari 72 (97,3%) diklasifikasikan sebagai akurat dan hanya 2 (2,7%) tidak akurat. Grafik kedua sebelah kanan, merinci lebih lanjut hasil tersebut. Dari dokumen yang akurat, 38 (51,4%) berhasil diidentifikasi bahwa fragmen yang didapat sesuai dengan informasi yang disediakan pada dokumen (*Correct With Doc*), sementara 34 (45,9%) lainnya dikategorikan *Correct No Doc*, menunjukkan bahwa chatbot masih tetap memberikan jawaban yang diharapkan meskipun tanpa dokumen pendukung. Dua dokumen yang tidak akurat terbagi menjadi satu *Incorrect With Doc* dan satu *Incorrect No Doc*. Visualisasi ini menggambarkan kinerja tinggi sistem *retrieval* dalam menemukan dokumen yang relevan, sekaligus mengidentifikasi area yang membutuhkan perbaikan, terutama dalam penanganan dokumen yang tidak akurat. Di mana masih ditemui 2 (2,7%) jawaban yang berasal dari ketidakmampuan *chatbot* dalam mendapatkan fragmen dokumen yang relevan meskipun informasi yang dicari tersedia pada dokumen.



Gambar 4.31 Visualisasi Evaluasi Halusinasi Chatbot

Telah disajikan distribusi halusinasi chatbot pada Gambar 4.31. Telah disayangkan bahwa chatbot masih teridentifikasi memberikan fabrikasi jawaban atau informasi salah, yaitu berjumlah 3 jawaban (4,1%). Kehadiran *hallucination* meskipun dalam jumlah kecil, menekankan perlunya upaya berkelanjutan untuk meningkatkan keandalan dan akurasi *chatbot* dalam menyajikan informasi.

#### 4.6 Analisis Kesalahan Chatbot dalam Merespons

Meskipun metrik evaluasi secara umum menunjukkan kinerja yang baik, penting untuk menganalisis kasus di mana *chatbot* memberikan respons yang salah. Tabel 4.3 menyajikan contoh pasangan tanya jawab tersebut, yang berkontribusi pada penurunan nilai metrik evaluasi seperti akurasi, presisi, atau *recall*. Analisis ini bertujuan untuk mengidentifikasi pola kesalahan dan mendapatkan gambaran untuk rekomendasi perbaikan.

Tabel 4.3 Daftar Respons Tanya dengan Jawaban yang Salah

ID	Pertanyaan	Jawaban	Klasifikasi
23	Siapa rektor universitas sekarang ?	Saya tidak memiliki informasi terbaru mengenai siapa Rektor Universitas Ma Chung saat ini. Untuk informasi terkini, Anda dapat mengunjungi situs	Tidak menjawab, informasi ada di

ID	Pertanyaan	Jawaban	Klasifikasi
		web resmi Universitas Ma Chung atau menghubungi bagian administrasi universitas.	dokumen tapi tidak ada di fragmen, jawaban salah
24	siapa pimpinan universitas saat ini ?	Pimpinan Universitas Ma Chung saat ini adalah Dr. Chatief Kunjaya yang menjabat sebagai Rektor. Untuk informasi lebih lanjut, Anda dapat mengunjungi situs web resmi Universitas Ma Chung atau menghubungi pihak administrasi universitas.	Informasi ada di dokumen tapi tidak ada di fragmen, Jawaban salah halusinasi
105	Bagaimana cara menulis laporan PKL	Untuk menulis laporan PKL (Praktik Kerja Lapangan), Anda perlu menyusun laporan sesuai dengan ketentuan yang ada. Berikut adalah struktur umum yang bisa diikuti: <ol style="list-style-type: none"> <li>1. Latar belakang.</li> <li>2. Tujuan PKL.</li> <li>3. Sasaran hasil yang diharapkan.</li> <li>4. Organisasi pengelola.</li> <li>5. Pelaksanaan kegiatan.</li> <li>6. Pembiayaan.</li> <li>7. Hambatan.</li> <li>8. Upaya penanggulangan hambatan.</li> <li>9. Kesimpulan.</li> <li>10. Saran.</li> </ol>	Informasi tidak ada di dokumen, jawaban salah, halusinasi
ID	Pertanyaan	Jawaban	Klasifikasi
		Laporan ini kemudian disampaikan kepada pihak terkait, seperti Rektor dan Sponsor. Untuk informasi lebih lanjut, Anda dapat	

		menghubungi Biro Kemahasiswaan, Pendidikan Karakter dan Kepemimpinan (BKA) Universitas Ma Chung.	
30	bagaimana cara saya menjadi staff ?	<p>Berikut adalah langkah-langkah umum untuk menjadi staf di Universitas Ma Chung:</p> <ol style="list-style-type: none"> <li>1. <b>**Cek Lowongan**</b>: Kunjungi situs resmi Universitas Ma Chung di [machung.ac.id](https://machung.ac.id) dan periksa bagian "Karir" atau "Lowongan Kerja" untuk melihat posisi yang tersedia.</li> <li>2. <b>**Siapkan Dokumen**</b>: Siapkan dokumen-dokumen yang diperlukan seperti surat lamaran, CV, ijazah, transkrip nilai, sertifikat pendukung, dan dokumen lainnya yang relevan.</li> <li>3. <b>**Kirim Lamaran**</b>: Kirimkan lamaran Anda sesuai dengan petunjuk yang tertera pada lowongan pekerjaan. Lamaran bisa dikirimkan melalui email atau sistem rekrutmen online yang disediakan oleh universitas.</li> <li>4. <b>**Proses Seleksi**</b>: Jika lamaran Anda memenuhi kriteria, Anda akan diundang untuk mengikuti proses</li> </ol>	Informasi tidak ada di dokumen, jawaban benar
ID	Pertanyaan	Jawaban	Klasifikasi
		<p>seleksi yang meliputi tes tertulis, wawancara, dan tahap seleksi lainnya.</p> <ol style="list-style-type: none"> <li>5. <b>**Pengumuman**</b>: Tunggu pengumuman hasil seleksi. Jika Anda diterima, Anda akan mendapatkan informasi lebih lanjut mengenai langkah-langkah berikutnya untuk</li> </ol>	

		memulai pekerjaan sebagai staf di Universitas Ma Chung.	
		Pastikan untuk selalu memeriksa informasi terbaru di situs web resmi universitas atau menghubungi bagian SDM untuk informasi lebih lanjut.	
57	bagaimana jika saya tersesat dan tak bisa kembali ?	Jika Anda tersesat dan tidak bisa kembali, lakukan langkah-langkah berikut: 1. Tetap tenang dan coba cari petunjuk atau tanda-tanda yang dapat membantu Anda mengenali lokasi Anda. 2. Gunakan ponsel Anda untuk mengakses aplikasi peta atau layanan navigasi seperti Google Maps. 3. Hubungi teman, keluarga, atau layanan keamanan kampus untuk meminta bantuan.	Informasi tidak ada di dokumen, jawaban benar
59	bagaimana jika saya mencair gaji saya ?	Untuk mencairkan gaji Anda, Anda bisa mengikuti langkah-langkah berikut: 1. Pastikan dana gaji sudah masuk ke rekening Anda.	Informasi tidak ada di dokumen,
ID	Pertanyaan	Jawaban	Tipe
		2. Kunjungi ATM atau bank terdekat untuk menarik uang. 3. Jika Anda mengalami masalah, hubungi bagian keuangan perusahaan Anda untuk bantuan lebih lanjut.	jawaban benar
31	Bagaimana cara saya mengakses wifi di kampus ?	Untuk mengakses Wi-Fi di kampus Universitas Ma Chung, ikuti langkah-langkah berikut:\n\n1. **Dapatkan Akun**:\nPastikan Anda memiliki akun email Universitas Ma Chung yang diberikan kepada setiap mahasiswa.\n2. **Koneksi Wi-Fi**:\nPilih jaringan Wi-Fi yang tersedia di lingkungan kampus.\n3.	Informasi tidak ada di dokumen, jawaban salah, halusinasi

---

**\*\*Login\*\*:** Masukkan kredensial (username dan password) yang sama dengan akun email Universitas Ma Chung Anda. \n4. **\*\*Akses\*\*:** Setelah berhasil login, Anda dapat mengakses internet menggunakan jaringan Wi-Fi kampus.\n\nJika mengalami kendala, Anda bisa menghubungi Unit Pelaksana Teknis Sistem Informasi untuk bantuan lebih lanjut.

---

Tabel 4.3 Daftar Respons Tanya dengan Jawaban yang Salah

Tabel 4.3 menyajikan seluruh pasangan tanya jawab yang berkontribusi terhadap penurunan performa *chatbot*. Untuk mendapatkan analisis secara lengkap, dilakukannya analisis terhadap semua pasangan tersebut.

#### 4.6.1 Kasus Pertanyaan Pimpinan dan Rektor Universitas

Ditetapkan di : Malang  
Pada tanggal : 31 Mei 2023  
Rektor,

ttd

**Dr. Murpin Josua Sembiring, SE, MSi**

iii

Gambar 4.32 Tampilan Informasi Rektor Universitas Ma Chung dari Dokumen

Pada kasus pertama terdapat dua pertanyaan mirip (ID 23 dan ID 24), merupakan pertanyaan mengenai identitas pimpinan atau rektor Universitas Ma Chung yang seharusnya dapat dijawab menggunakan informasi pada halaman iii dari dokumen seperti pada Gambar 4.30 Lebih tepatnya pada tanda tangan dengan nama dan jabatan. Akan tetapi, *chatbot* memberikan jawaban yang diperoleh dari model ChatGPT yang bersumber dari internet, yang memang benar namun tetap salah karena sudah *outdated*.



Untuk menganalisis lebih lanjut penyebab kesalahan chatbot dalam menjawab pertanyaan terkait pimpinan universitas dan rektor, peneliti melakukan peninjauan terhadap fragmen dokumen yang relevan. Analisis fragmen dokumen ini bertujuan untuk mengidentifikasi apakah informasi yang dibutuhkan sebenarnya tersedia dalam dokumen sumber dan bagaimana chatbot memproses informasi tersebut. Fragmen dokumen yang berkaitan dengan pertanyaan tentang "pimpinan" dan "rektor" masing-masing disajikan pada Lampiran 4 dan Lampiran 5.

Berdasarkan fragmen dokumen yang didapatkan pada Lampiran 4, ditemukan bahwa *chatbot* tidak berhasil mendapatkan informasi pimpinan universitas dari dokumen. Meskipun informasi terkait seharusnya bisa, fungsi *embedding* tidak dapat menghubungkannya dengan konsep "pimpinan". Hal ini berakibat pada *chatbot* memberikan respons yang bersumber dari internet, yang meskipun faktual, tidak berasal dari dokumen yang seharusnya menjadi basis informasi. Temuan ini mengindikasikan adanya potensi masalah dalam kemampuan fungsi *embedding* untuk memahami sinonim atau hubungan semantik antara "pimpinan" dan "rektor universitas".

Analisis pada Lampiran 5 menunjukkan hasil serupa. Fragmen dokumen yang memuat tanda tangan dan jabatan rektor juga tidak terdeteksi oleh *chatbot*. Akibatnya, *chatbot* merespons dengan menyatakan bahwa informasi tersebut tidak tersedia. Hal ini memperkuat indikasi adanya kendala dalam pemrosesan informasi visual dan tata letak dokumen oleh *chatbot*, khususnya dalam menginterpretasi tanda tangan dan jabatan yang menyertainya.

Berdasarkan temuan analisis fragmen dokumen, dapat disimpulkan bahwa terdapat dua potensi kekurangan pada kasus pertanyaan ini:

- **Kesulitan dalam Pemahaman Semantik:** Fungsi *embedding* tampaknya kesulitan mengidentifikasi hubungan semantik antara kata "pimpinan" dan "rektor universitas". Hal ini menunjukkan adanya kekurangan pada model *embedding* yang digunakan dalam memahami variasi linguistik dan konteks penggunaan kata.
- **Kendala dalam Pemrosesan Tata Letak Dokumen:** Tata letak informasi, khususnya tanda tangan dan jabatan pada halaman iii, kurang optimal dipahami oleh fungsi *embedding*. Hal ini mengindikasikan perlunya

perbaikan pada kemampuan *chatbot* dalam memproses informasi visual dan tata letak dokumen, termasuk tata kelola pemberian informasi pada dokumen.

#### **4.6.2 Kasus Pertanyaan Umum dan Di Luar Konteks Dokumen**

Selain menganalisis performa *chatbot* terhadap pertanyaan yang jawabannya terdapat dalam dokumen, penelitian ini juga mengidentifikasi adanya pertanyaan yang diajukan pengguna di luar konteks dokumen yang disediakan. Pertanyaan-pertanyaan ini mencakup topik-topik umum seperti prosedur menjadi karyawan, cara pencairan gaji, dan tindakan yang perlu diambil apabila tersesat. Penting untuk dicatat bahwa informasi terkait pertanyaan-pertanyaan ini tidak tersedia dalam dokumen sumber yang menjadi basis pengetahuan *chatbot*. Hal ini memberikan kesempatan untuk mengamati bagaimana *chatbot* menangani situasi di mana informasi yang diminta tidak tersedia dalam dokumen.

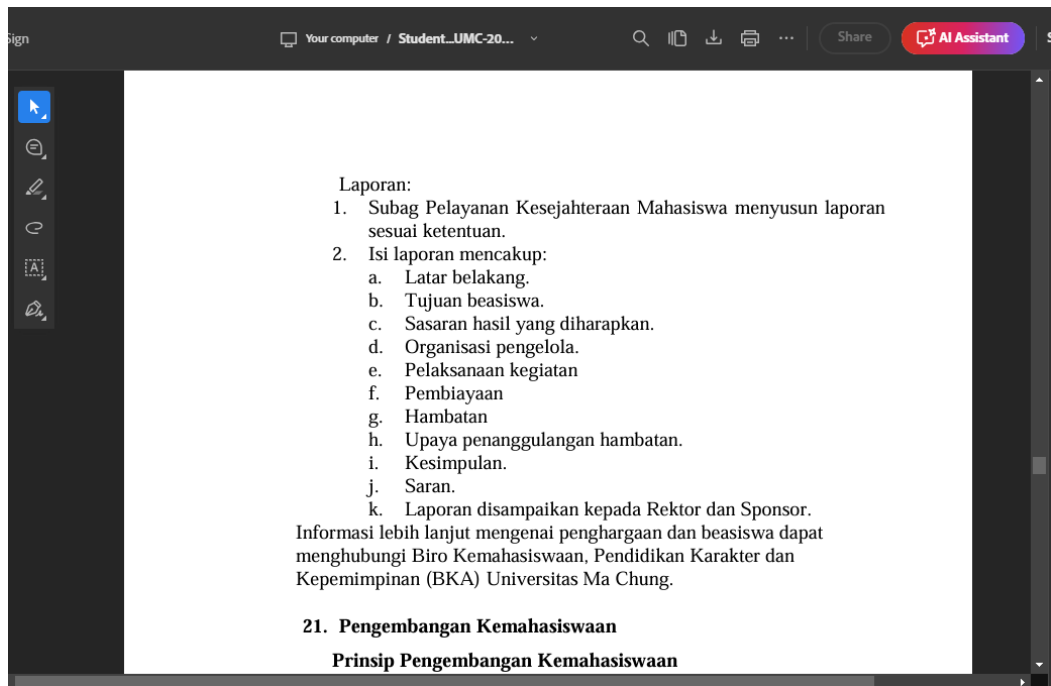
Menariknya, meskipun informasi yang ditanyakan tidak ada dalam dokumen, *chatbot* tetap bisa memberikan respons terhadap semua pertanyaan di luar konteks tersebut. *Chatbot* berhasil memberikan jawaban dengan mengandalkan informasi yang diambil dari internet. Meskipun pada beberapa kasus jawaban yang diberikan faktual dan bermanfaat bagi pengguna, pendekatan ini menimbulkan potensi masalah yang signifikan. Ketergantungan pada informasi dari internet membuka peluang bagi *chatbot* untuk menyajikan informasi yang kurang akurat, *outdated*, atau bahkan menyesatkan. Validitas dan reliabilitas sumber informasi dari internet tidak selalu terjamin, dan *chatbot* berpotensi menampilkan informasi yang bertentangan dengan kebijakan atau informasi resmi universitas yang seharusnya menjadi rujukan utama.

Temuan ini mengindikasikan adanya kebutuhan untuk merevisi *prompt* yang diberikan kepada *chatbot* dan memperkuat kontrol terhadap sumber informasi yang digunakannya. *Prompt* perlu dirancang sedemikian rupa sehingga *chatbot* dapat membedakan antara pertanyaan yang relevan dengan dokumen dan pertanyaan di luar konteks. Implementasi mekanisme kontrol sumber informasi yang lebih ketat, seperti membatasi akses *chatbot* hanya pada dokumen yang telah

diverifikasi, juga krusial untuk memastikan akurasi dan konsistensi informasi yang diberikan.

#### 4.6.3 Kasus Hallucination Pada Chatbot

Teridentifikasi adanya fenomena halusinasi pada *chatbot* berdasarkan analisis interaksi dengan ID 105. Dalam interaksi tersebut, pengguna mengajukan pertanyaan mengenai panduan penulisan laporan Praktik Kerja Lapangan (PKL). Meskipun dokumen "Student Guide 2023" tidak memuat informasi terkait penulisan laporan PKL, *chatbot* memberikan jawaban dengan tingkat kepercayaan yang tinggi. Hal ini mengindikasikan adanya fabrikasi informasi oleh *chatbot*, di mana informasi yang diberikan tidak bersumber dari data yang tersedia.



Gambar 4.33 Tampilan Dokumen yang Sama dengan Jawaban Chatbot

Verifikasi silang antara respons yang didapat dari *chatbot* (Tabel 4.3) dan dokumen *Student Guide 2023* (Gambar 4.31) menunjukkan adanya kesamaan informasi. Hal ini mengindikasikan bahwa model *embedding* bekerja dengan baik mengingat bagian ini yang paling mendekati dengan pertanyaan yang diajukan. Namun, analisis lebih lanjut pada Lampiran 6 mengungkapkan bahwa fragmen dokumen yang memiliki kemiripan semantik tertinggi dengan pertanyaan dimulai dengan kata "Laporan:". Fragmen ini, meskipun relevan secara umum, berada

dalam konteks penulisan laporan yang lebih luas, dan bukan secara spesifik laporan PKL. Oleh karena itu, diasumsikan bahwa model ChatGPT menggeneralisasi informasi ini dan berasumsi bahwa format atau susunan penulisan laporan tersebut identik dengan format laporan PKL. Potongan dokumen yang kurang spesifik ini menjadi salah satu potensi penyebab *chatbot* memberikan jawaban yang kurang tepat untuk konteks PKL.

## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Penelitian ini telah berhasil menghasilkan *chatbot* menggantikan staf Universitas Ma Chung dalam melayani mahasiswa yang bertanya dalam lingkup buku Student Guide. Hal tersebut dicapai dengan mengintegrasikan metode Retrieval Augmented Generation (RAG) pada *chatbot* berbasis model OpenAI melalui platform WhatsApp. Dengan memanfaatkan teknologi LangChain, sistem mampu merespons pertanyaan dengan melakukan pencarian informasi dari dokumen PDF dengan lebih efisien dan optimal. Hasil survei menunjukkan 25 dari 28 respons mendapatkan tanggapan positif terhadap relevansi jawaban yang diterima. Perhitungan evaluasi *chatbot* dalam memberikan jawaban faktual menggunakan metrik khusus, didapatkan akurasi mencapai 94.59%. Hal tersebut didukung dengan akurasi pencarian dokumen yang mencapai 97,30% menunjukkan kemampuan *chatbot* untuk memberikan jawaban yang akurat berdasarkan dokumen yang diterima. Meskipun demikian, ditemukan adanya *hallucination* pada 4,05% jawaban dan 2,7% respons, *chatbot* gagal menemukan informasi yang relevan, mengakibatkan penolakan jawaban. Implementasi ini mencakup pembuatan *vectorstore*, integrasi dengan WhatsApp Business API, dan penggunaan LangChain untuk menyederhanakan proses pencarian dan meningkatkan efisiensi sistem. Pendekatan ini membuktikan efektivitas teknologi RAG dalam meningkatkan aksesibilitas dan efisiensi layanan informasi, memberikan kontribusi besar bagi organisasi atau perusahaan yang ingin mengadopsi teknologi *chatbot* berbasis kecerdasan buatan.

#### **5.2 Saran**

Berdasarkan evaluasi dan analisis yang telah dilakukan, beberapa saran pengembangan diajukan untuk meningkatkan kinerja, cakupan, dan keandalan sistem *chatbot* untuk penelitian selanjutnya. Saran-saran ini difokuskan pada peningkatan kemampuan *chatbot* dalam memahami konteks dan nuansa percakapan, fakta informasi, pengurangan halusinasi, dan kualitas respons jawaban.

1. Berdasarkan berbagai masukan, disarankan agar sistem umpan balik tidak ditampilkan setiap kali pengguna menerima jawaban dari *chatbot*, untuk menghindari penumpukan pesan. Sebagai alternatif, tautan untuk memberikan umpan balik secara keseluruhan dapat disertakan di profil WhatsApp pengguna, sehingga instruksi umpan balik tidak mengganggu.
2. Kemampuan *chatbot* untuk mengenali dan memahami gambar pada dokumen diperlukan, terutama untuk menangani dokumen yang mencakup silsilah organisasi atau diagram.
3. Diperlukannya untuk memantau penggunaan token setiap kali adanya percakapan pada *chatbot* guna membantu manajemen pengeluaran.
4. Perlu diterapkan *prompt* yang lebih ketat agar *chatbot* hanya mengambil informasi dari dokumen. Atau jika informasi dari internet digunakan, jawaban harus mencantumkan sumbernya dengan jelas agar pengguna mengetahui asal informasi.
5. Diperlukan pengelolaan yang lebih baik terhadap basis pengetahuan dan proses pemecahan dokumen untuk mengurangi halusinasi. Sehingga penelitian selanjutnya dapat menata kembali pecahan yang telah dibuat atau bisa menata dokumen agar pecahan yang didapat memiliki informasi yang lengkap.
6. Pengurangan halusinasi juga bisa dengan mencoba menggunakan pecahan dokumen lebih dari satu sebagai konteks untuk *prompt* serta menggunakan model *embedding* lain selain text-embedding-ada-002.

## DAFTAR PUSTAKA

- 47 *AI statistics and trends for 2025: Latest insights and data*. (2025, Januari 10).  
Hostinger Tutorials. <https://www.hostinger.com/tutorials/ai-statistics>
- Advanced usage - OpenAI API*. (t.t.). Diambil 1 Agustus 2024, dari  
<https://platform.openai.com/docs/advanced-usage/managing-tokens>
- Alia, P. A., Prayogo, J. S., Kriswibowo, R., & Setyadi, A. T. (2024). Implementation  
Open Artificial Intelligence ChatGPT Integrated With Whatsapp Bot.  
*Advance Sustainable Science, Engineering and Technology*, 6(1).  
<https://doi.org/10.26877/asset.v6i1.17909>
- Bagha, K. N. (2011). A Short Introduction to Semantics. *Journal of Language  
Teaching and Research*, 2(6). <https://doi.org/10.4304/jltr.2.6.1411-1419>
- Deploy for Free – Render Docs*. (t.t.). Diambil 23 Desember 2024, dari  
<https://render.com/docs/free#free-web-services>
- Diab, M., Herrera, J., Chernow, B., & Mao, C. (2022). *Stable Diffusion Prompt  
Book*. <https://openart.ai/promptbook>.
- Embedding Techniques for Retrieval-Augmented Generation | Restackio*. (t.t.).  
Diambil 20 Desember 2024, dari <https://www.restack.io/p/retrieval-augmented-generation-answer-embedding-techniques-cat-ai>
- Getting Started - Meta Webhooks*. (t.t.). Diambil 1 Agustus 2024, dari  
<https://developers.facebook.com/docs/graph-api/webhooks/getting-started#verification-requests>
- Isakova, T. (2024, Januari 23). *AI Adoption by Industry: Top Use Cases Uncovered*.  
InData Labs. <https://indatalabs.com/blog/ai-adoption-by-industry>
- LangSmith*. (2024, Juni). <https://smith.langchain.com/hub/rbm/rag-prompt>
- 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, 2020-December*.  
<https://arxiv.org/abs/2005.11401v4>

- Megan Cerullo. (2023). *ChatGPT acquired 100 million active users faster than TikTok and Instagram*. CBS News. <https://www.cbsnews.com/news/chatgpt-chatbot-tiktok-ai-artificial-intelligence/>
- Mengobrol dengan Meta AI di WhatsApp*. (2024, September 25). Blog WhatsApp. [https://blog.whatsapp.com/talk-to-meta-ai-on-whatsapp?lang=id\\_ID](https://blog.whatsapp.com/talk-to-meta-ai-on-whatsapp?lang=id_ID)
- Most popular messaging apps 2024 | Statista*. (2025). Statista. <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>
- NR Tejaswini, Vidya S, & Dr. T Vijaya Kumar. (2023). LangChain-Powered Virtual Assistant for PDF Communication. *International Research Journal of Modernization in Engineering Technology and Science*. <https://doi.org/10.56726/irjmets43587>
- PDF.ai*. (t.t.). Diambil 15 September 2023, dari <https://pdf.ai/>
- Pujiono, I., Agtyaputra, I. M., & Ruldeviyani, Y. (2024). IMPLEMENTING RETRIEVAL-AUGMENTED GENERATION AND VECTOR DATABASES FOR CHATBOTS IN PUBLIC SERVICES AGENCIES CONTEXT. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 10(1), 216–223. <https://doi.org/10.33480/jitk.v10i1.5572>
- Reznikov Ivan. (2024). *LangChain 101: Part 3b. Talking to Documents: Embeddings and Vectorstores*. <https://pub.towardsai.net/langchain-101-part-3b-talking-to-documents-embeddings-and-vectorstores-c37d460f1519>
- Top Vector Stores: 9 Use Cases You Should Know*. (2024, Juni 16). TiDB. <https://www.pingcap.com/article/top-vector-stores-9-use-cases-you-should-know/>
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). *Emergent Abilities of Large Language Models*. <https://arxiv.org/abs/2206.07682v2>
- What Are Web Services? Easy to Learn Concepts with Examples | Cleo*. (t.t.). Diambil 1 Agustus 2024, dari <https://www.cleo.com/blog/knowledge-base-web-services>



*What is RESTful API? - RESTful API Explained - AWS.* (t.t.). Diambil 1 Agustus 2024, dari [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

## LAMPIRAN

### Lampiran 1 Kode Pembuatan Vectorstore

```
1  from langchain.chains import create_history_aware_retriever,
2  create_retrieval_chain
3  from langchain_core.prompts import ChatPromptTemplate,
4  MessagesPlaceholder
5  from langchain.chains.combine_documents import
6  create_stuff_documents_chain
7  from langchain_core.runnables.history import
8  RunnableWithMessageHistory
9  from quanta_quire.app import llm, get_rag_chat_log
10 def history_aware_retriever(retriever):
11     contextualize_q_system_prompt = ( "Prompt pertama" )
12     contextualize_q_prompt = ChatPromptTemplate.from_messages([
13         ("system", contextualize_q_system_prompt),
14         MessagesPlaceholder("chat_history"),
15         ("human", "{input}"),
16     ])
17     return create_history_aware_retriever(
18         llm, retriever, contextualize_q_prompt
19     )
20 def rag_chain(retriever):
21     system_prompt = ("prompt kedua {context}")
22     qa_prompt = ChatPromptTemplate.from_messages([
23         ("system", system_prompt),
24         MessagesPlaceholder("chat_history"),
25         ("human", "{input}"),
26     ])
27     question_answer_chain = create_stuff_documents_chain(
28         llm, qa_prompt)
29     return create_retrieval_chain(
30         history_aware_retriever(retriever),
31         question_answer_chain)
32 def chat_rag_chain(retriever):
33     return RunnableWithMessageHistory(
34         rag_chain(retriever),
```

```
35     get_rag_chat_log,  
36     input_messages_key="input",  
37     history_messages_key="chat_history",  
38     output_messages_key="answer",  
39 )  
30 def rag_chat(retriever, session_id, message):  
31     response = chat_rag_chain(retriever).invoke(  
32         {"input": message},  
33         config={"configurable": {"session_id": session_id}},  
34     )["answer"]  
35     return response
```

Lampiran 2 Kode Form Pengunggahan Dokumen pada Halaman Website

```
1 <form enctype="multipart/form-data" method="post">
2     {{ form.hidden_tag() }}
3     <div class="form-group">
4         {{ form.csrf_token }}
5         {{ form.document.label(class="form-label") }}
6         <p class="px-1 mb-0" id="file-name"></p>
7         {% if form.document.errors %}
8         <span class="text-danger">{{ form.document.errors[0] }}</span>
9         {% endif %}
10        <label class="btn btn-info" for="file" tabindex="0">
11            Pilih Dokumen
12            {{ form.document(hidden="", accept="application/pdf",
13 id="file") }}
14        </label>
15    </div>
16    <button id="upload_submit" class="btn btn-primary"
17 type="submit">
18        Unggah
19    </button>
20 </form>
```

### Lampiran 3 Kode Percakapan Dengan Umpan Balik

```

1  import os
2
3  from flask import current_app
4
5  from quanta_quire.app.chat_basic import basic_chat
6  from quanta_quire.app.chat_rag import rag_chat
7  from quanta_quire.app.message_utils import get_last_ai_message,
8  get_last_human_message
9  from quanta_quire.app.vectorstore import faiss_load_vectorstore
10 from quanta_quire.helper import get_random_response,
11 insert_chat_log
12
13 def chat(session_id, message):
14     ask_feedback = ("Apakah jawaban saya sudah tepat? ...")
15     try: # Check if the message is a feedback number
16         number = int(message)
17         if 0 <= number <= 3: # Save the feedback
18             if session_id not in current_app.chats:
19                 return get_random_response("feedback_first"), None
20             save_chat_feedback(session_id=session_id, point=message)
21             return get_random_response("feedback"), None
22     except ValueError:
23         pass
24     response = qna(session_id, message)
25     save_chat_log(session_id, message)
26     return response, ask_feedback
27
28 def save_chat_log(session_id, message):
29     ai = get_last_ai_message(current_app.chats, session_id)
30     insert_chat_log(user=session_id, question=message,
31                    answer=ai.content)
32
33 def save_chat_feedback(session_id, point):
34     ai = get_last_ai_message(current_app.chats, session_id)
35     question = get_last_human_message(current_app.chats, session_id)
36     insert_chat_log(user=session_id, point=point, feedback=True)

```

37	
38	<code>def qna(session_id, message):</code>
39	<code>try:</code>
40	<code>vectorstore_path =</code>
41	<code>os.path.join(current_app.config['UPLOAD_PATH'], 'vectorstore')</code>
42	<code>if os.path.exists(vectorstore_path) and</code>
43	<code>os.path.isdir(vectorstore_path):</code>
44	<code>retriever = faiss_load_vectorstore().as_retriever()</code>
45	<code>return rag_chat(retriever, session_id, message)</code>
46	<code>else:</code>
47	<code>return basic_chat(session_id, message)</code>
48	<code>except Exception as e:</code>
49	<code>response_message = "Silahkan coba lagi nanti."</code>
50	<code>return response_message</code>

Content #1

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 20

Content:

langkah awal berdirinya Universitas, dengan para pelopor adalah Soegeng Hendarto, Mochtar Riady, Teguh Kinarto, Hendro Sunjoto, Koentjoro Loekito, Effendy Sudargo, Agus Chandra, Hadi Widjojo, Nuryati Tanuwidjaya, Nehemja, Alex Lesmana Samudra, Evelyn Ad am, Hadi Surjono, Nagawidjaja Winoto , dan Soebroto Wirotomo.

Secara aklamasi dan dengan pernyataan kebulatan tekad alumni dari seluruh dunia, dalam Reuni Akbar peringatan hari Ulang Tahun ke - 60 SMA Ma Chung di Malang, pada 17 Juli 2005 diletakkan batu pertama pembangunan Universitas.

Dalam rangka memperlancar jalannya pengelolaan Universitas dalam jangka panjang maka dibentuk Yayasan Harapan Bangsa Sejahtera (YHBS) yang menaungi Universitas.

Alumni Ma Chung senior yaitu Prof. Dr. Yang Zhiling dan alm. Prof. Dr. Ling Bin memberikan banyak usulan sehubungan dengan pembangunan dan pengelolaan Universitas. Usulan beliau tersebut kemudian dijadikan pijakan pertama bagi perencanaan ( blue print )

Content #2

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 144

Content:

pimpinan Universitas dalam mengembangkan kegiatan kemahasiswaan.

7. Pemanfaatan sarana dan prasarana Universitas secara maksimal oleh organisasi -organisasi kemahasiswaan dalam melaksanakan kegiatan kemahasiswaan.

-----  
Content #3

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 57

Content:

55

Ma Chung atau berasal dari luar Universitas Ma Chung atau praktisi yang telah memenuhi syarat.

2. Setelah Ketua Program Studi menyetujui usulan dosen penguji, dilanjutkan kepada Dekan untuk di berikan SK penugasan.

#### 6. Penggantian Dosen Pembimbing

Dosen Pembimbing Utama atau Pendamping dapat diganti apabila tidak aktif memberikan bimbingan seperti tugas ke luar negeri atau di luar kampus Universitas Ma Chung, atau alasan lainnya yang menurut pertimbangan Ketua Program Studi akan berdampak negatif terhadap penyelesaian studi mahasiswa. Dosen yang bersangkutan diminta untuk mengajukan surat permohonan pengunduran diri. Atas dasar surat permohonan tersebut Ketua Program Studi akan menunjuk penggantinya . Apabila diperlukan, Ketua Program Studi akan berkonsultasi dengan berbagai pihak yang terkait. Segala konsekuensi administrasi dalam penggantian dosen pembimbing diatur oleh Ketua Program Studi sesuai dengan ketentuan yang berlaku.

#### 7. Rencana Studi

-----  
Content #4

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 184

Content:

bahwa kepemimpinan merupakan salah satu elemen penting yang sangat dibutuhkan pada berbagai aras dan berbagai fungsi. Berdasarkan hal ini, maka salah satu visi Universitas Ma Chung adalah menjadi salah satu Universitas di Indonesia yang memiliki Leadership Center (Pusat Pengembangan



Kepemimpinan) yang memiliki fungsi dalam pembinaan  
softskill terutama dari segi kepemimpinan dan manajerial, serta  
-----

Content #1

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 53

Content:

51

2. Pemberian penghargaan kepada lulusan terbaik tingkat Universitas ditetapkan oleh Rektor dengan pertimbangan Senat Akademik Universitas.

Content #2

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 20

Content:

langkah awal berdirinya Universitas, dengan para pelopor adalah Soegeng Hendarto, Mochtar Riady, Teguh Kinarto, Hendro Sunjoto, Koentjoro Loekito, Effendy Sudargo, Agus Chandra, Hadi Widjojo, Nuryati Tanuwidjaya, Nehemja, Alex Lesmana Samudra, Evelyn Ad am, Hadi Surjono, Nagawidjaja Winoto , dan Soebroto Wirotomo.

Secara aklamasi dan dengan pernyataan kebulatan tekad alumni dari seluruh dunia, dalam Reuni Akbar peringatan hari Ulang Tahun ke - 60 SMA Ma Chung di Malang, pada 17 Juli 2005 diletakkan batu pertama pembangunan Universitas.

Dalam rangka memperlancar jalannya pengelolaan Universitas dalam jangka panjang maka dibentuk Yayasan Harapan Bangsa Sejahtera (YHBS) yang menaungi Universitas.

Alumni Ma Chung senior yaitu Prof. Dr. Yang Zhiling dan alm. Prof. Dr. Ling Bin memberikan banyak usulan sehubungan dengan pembangunan dan pengelolaan Universitas. Usulan beliau tersebut kemudian dijadikan pijakan pertama bagi perencanaan ( blue print )

-----  
-----  
Content #3

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 193

Content:

191

Sarana dan Prasarana Universitas Ma Chung

1. Gerbang Utama Universitas Ma Chung

2. Gedung Rektorat ( Gedung Soegiharto – Pusat Layanan  
Administrasi Universitas )

3. Gedung Bakti Persada (Gedung Perkuliahan )

-----  
Content #4

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 4

Content:

ii

KEPUTUSAN REKTOR UNIVERSITAS MA CHUNG

No : 0020/MACHU NG/SK -REK/ V/2023

Tentang

PEDOMAN PENDIDIKAN UNIVERSITAS MA CHUNG

REKTOR UNIVERSITAS MA CHUNG

Menimbang :1.Bahw a untuk tertibnya proses pembelajaran di Universitas Ma Chung perlu pedoman di bidang pendidikan;

2.Bahwa untuk Pedoman Pendidikan Universit as Ma Chung di atas perlu ditetapkan dengan keputusan Rektor .

Mengingat : 1. Undang -Unda ng Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional ;

2. Undang -Undang Nomo r 12 tahun 2012 tentang Pendidikan Tinggi ;

3. Peraturan Menteri Pendidikan dan Kebud ayaan Nomor 3 tahun 2020 tentang Standar Nasional Pendidikan Tinggi

4. Peraturan Pemerintah Republik Indones ia Nomor 19 tahun 2005 tentang Standar Nasional Pendidikan;

5. Peraturan Pemerintah Republik Indonesia Nomor 3 7 tahun 2009 tentang Dosen ;

6. Peraturan Peme rintah Republik Indonesia Nomor 17 Tahun 2010 tentang Pengelolaan dan Penyelenggaraan

-----



Content #1

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 143

Content:

141

Laporan :

1. Subag Pelayanan Kesejahteraan Mahasiswa menyusun laporan sesuai ketentuan.

2. Isi laporan mencakup:

- a. Latar belakang.
- b. Tujuan beasiswa.
- c. Sasaran hasil yang diharapkan.
- d. Organisasi pengelola.
- e. Pelaksanaan kegiatan
- f. Pembiayaan
- g. Hambatan
- h. Upaya penanggula ngan hambatan.
- i. Kesimpulan.
- j. Saran .
- k. Laporan disampaikan kepada Rektor dan Sponsor .

Informasi lebih lanjut mengenai penghargaan dan beasiswa dapat menghubungi Biro Kemahasiswaan, Pendidikan Karakter dan Kepemimpinan (BKA) Universitas Ma Chung.

21. Pengembangan Kemahasiswaan

Prinsip Pengembangan Kemahasiswaan

Definisi pengembangan:

1. Pengembangan yang dimaksud adalah pengembangan kemampuan intelektual, afektif, psikomotorik, kepribadian dan spiritual sehingga mahasiswa dapat menjadi insan yang utuh dan seimba ng sesuai dengan visi dan misi Universitas.

2. Pengembangan kemampuan intelektual adalah

-----  
-----

Content #2

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 137

Content:

- Mengumumkan pemenang kepada seluruh peserta.
- d. Setiap UKM Kesenian memiliki aturan khusus untuk bidangnya masing-masing dalam penyelenggaraan event semacam ini.
- e. Kegiatan pameran atau pagelaran dapat dilaksanakan dalam berbagai kesempatan, yang dijadwalkan secara rutin ataupun insidental berdasarkan permintaan pihak lain.
- f. Pelaksanaan pagelaran atau pameran secara rinci dapat dilakukan sebagai berikut:
  - Membentuk kepanitiaan.
  - Mempersiapkan materi pagelaran atau pameran.
  - Menyebarkan informasi menggunakan poster, pamlet, spanduk, baliho dan internet.
  - Mencari izin pelaksanaan.
  - Mencari sponsor penunjang dana.
  - Melaksanakan kegiatan pagelaran/pameran.
- 7. Membuat evaluasi pelaksanaan dan membubarkan panitia.

-----  
-----

Content #3

Source: /home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 74

Content:

72

(5) Mahasiswa wajib melaporkan nilai tengah dan/ atau nilai akhir yang diperoleh dari mitra program MBKM kepada Koordinator Perguruan Tinggi untuk masing-masing Program MBKM, Koordinator MBKM Prodi, dan Kepala Program Studi.

7. Internal

- (6) Mahasiswa melakukan KRS sesuai program MBKM mandiri Prodi yang diikuti melalui MAC IS.
- (7) Mahasiswa melaksanakan Program MBKM mandiri Prodi sesuai jadwal dan prosedur yang ditentukan Prodi masing -masing
- (8) Mahasiswa wajib berkoordinasi dengan Koordinator MBKM Prodi
- (9) Mahasiswa wajib melaporkan perkembangan pelaksanaan Program MBKM Koordinator MBKM Prodi dan dosen pembimbing yang ditunjuk Kepala Program Studi
- (10) Mahasiswa wajib melaporkan penilaian yang diperoleh dari Mitra MBKM mandiri Prodi kepada Koordinator MBKM Prodi dan Kepala Program Studi.

#### 8. Jenis Kegiatan

Jenis kegiatan pembelajaran MBKM sesuai peraturan pemerintah ada 8 (delapan) , yakni magang/praktik kerja, pertukaran

-----  
-----

Content #4

Source:/home/fryctze/Documents/Project/Python/RESEARCH/TA-ChatPdf/quanta\_quire/storage/Student-Guide-UMC-2023.pdf

Page: 133

Content:

131

2) Cakupan penyampaian informasi , meliputi:

- a. Peraturan pertandingan ;
- b. Persyaratan peserta ;
- c. Jadwal pelaksanaan ;
- d. Penentuan juara.
- e. Hadiah.

6. Pelaksanaan Pertandingan/Perlombaan:

- 1) Peraturan yang digunakan dalam pertandingan/perlombaan sudah ditetapkan oleh cabang olahraga masing -masing.
- 2) Pelaksanaan pertandingan/perlombaan dilakukan melalui langkah -langkah berikut :
  - a. Menyebarkan informasi ke seluruh Program



Studi.

- b. Pendaftaran peserta.
- c. Melaksanakan technical meeting.
- d. Melaksanakan pertandingan/perlombaan.
- e. Mengumumkan pemenang.
- f. Memberikan hadiah kepada seluruh peserta pertandingan /perlombaan.

7. Pemberian Penghargaan dan Hadiah:

- 1) Maksud dan tujuan pemberian penghargaan dan hadiah adalah menghargai mahasiswa atas prestasinya dalam mengikuti kejuaraan atau perlombaan olahraga.
- 2) Bentuk penghargaan piagam dan/atau uang pembinaan .
- 3) Besarnya hadiah uang pembinaan ditetapkan berdasarkan:

-----