IMPLEMENTASI CHATBOT MENGGUNAKAN FRAMEWORK LANGCHAIN BERBASIS LLM GPT (STUDI KASUS : PANDUAN AKADEMIK UNIVERSITAS TRUNOJOYO)

Muhammad Dimas Arya Muhajir, Novi Prastiti, Meidya Koeshardianto
Sistem Informasi, Universitas Trunojoyo
Jl. Raya Telang, Perumahan Telang Inda, Telang, Kec. Kamal, Kabupaten Bangkalan, Jawa Timur 69162
Aryamuhajir888@gmail.com

ABSTRAK

Pada saat ini, pelayanan akademik atau informasi di program studi Sistem Informasi Universitas Trunojoyo Madura (UTM) masih manual yaitu menggunakan fitur *chatting* seperti Whatsapp. Namun, pelayanan informasi manual tersebut belum berjalan dengan maksimal. Jumlah pengurus atau staff dan banyaknya pesan yang dikirim oleh mahsiswa tidak sebanding sehingga respon pengurus lambat atau tidak terbalas. Dari permasalahan yang ada, diperlukan sebuah *chatbot* yang dapat menjawab pertanyaan-pertanyaan mahasiswa secara instan dan otomatis. Tujuan dari penelitian ini adalah untuk mengembangkan sebuah *chatbot* dengan menggunakan *framework* Langchain dan *Large Language Models (LLM) GPT-3.5 Turbo*. Implementasi Langchain dan LLM membuat *chatbot* dapat merespon pertanyaan mahasiswa sebagaimana bahasa manusia. Aplikasi yang dihasilkan dari penelitian ini berupa aplikasi python berbasis web dan menggunakan tampilan streamlit *chat*. Dari penelitian ini diharapkan *chatbot* yang dibuat dapat meningkatkan pengalaman mahasiswa dalam mendapatkan layanan informasi atau akademik dan mendapatkan nilai *User Acceptance Test* (UAT) yang tinggi. Pengujian akurasi dilakukan sebanyak lima kali dengan pendekatan yang berbeda tiap pengujian. Hasil pengujian nilai akurasi yaitu sebesar 96,66%. Pengujian *usability* sistem menggunakan *User Acceptance Test* (UAT) mendapatkan presentase sebesar 82,79% dan dapat dikategorikan dalam kategori sangat kuat.

Kata kunci : *Chatbot, Langchain, LLM*

1. PENDAHULUAN

Dalam era transformasi digital, penggunaan teknologi kecerdasan buatan semakin penting dalam berbagai sektor. *Chatbot*, sebagai salah satu aplikasi kecerdasan buatan yang paling populer saat ini, telah menjadi salah satu solusi untuk mempermudah interaksi manusia dengan suatu sistem informasi. *Chatbot* adalah sebuah program komputer yang bertujuan untuk menjawab atau berinteraksi dengan manusia dalam percakapan[1].

Chatbot dapat membuat proses bisnis menjadi lebih efektif karena chatbot mampu memberikan respon secara instan dan selalu aktif setiap saat. Salah satu perkembangan terkini dalam pengembangan chatbot adalah penggunaan *Large Language Model* (LLM).

ChatGPT adalah aplikasi *chatbot* berbasis LLM yang paling populer saat ini. *Large Language Model* (LLM) adalah model bahasa yang sangat besar dan kompleks yang dapat digunakan untuk menganalisis dan menggenerasikan teks[2].

Penggunaan LLM membuat percakapan antara pengguna dan mesin lebih terasa seperti obrolan antar manusia. ChatGPT memiliki pertumbuhan popularitas yang sangat cepat. ChatGPT memiliki 1 juta pengguna hanya dalam waktu 5 hari dari peluncurannya pada 30 November, 2022[3].

Kelebihan utama dari *chatbot* berbasis LLM adalah besarnya informasi yang diketahui model sehingga dapat meningkatkan pengalaman pengguna[2].

Namun, terdapat beberapa kendala ketika ingin mengembangkan aplikasi yang mengimplementasikan LLM seperti *GPT-3.5 Turbo*, yaitu memiliki batasan informasi publik hingga 21 September, 2021[4].

Model tersebut juga dapat mengembalikan informasi yang tidak relevan jika tanpa konteks yang benar, skalabilitas yang tidak baik, data yang tidak berstruktur akan sulit untuk diproses, dan sulit untuk mengatur alur yang kompleks[5].

Permasalahan tersebut dapat diatasi dengan *framework* Langchain. Langchain adalah sebuah *open-source framework* yang dapat memungkinkan pembuatan aplikasi LLM dengan data *custom* dari dokumen pribadi, skalabilitas tinggi, dan mempercepat proses pengembangan. Langchain dapat digambarkan sebagai jembatan antara LLM dengan aplikasi[5].

Panduan akademik adalah sumber informasi kunci yang menyediakan panduan tentang program studi, jadwal perkuliahan, syarat kelulusan, dan informasi penting lainnya yang relevan dengan perjalanan akademik mahasiswa. Sayangnya, dalam konteks sejumlah program studi di Universitas Trunojoyo Madura, terutama Prodi Sistem Informasi, pelayanan informasi masih tergantung pada proses manual yang kurang efektif dalam mengatasi kebutuhan mahasiswa sehingga terkadang mahasiswa dapat mengalami kesulitan dalam menemukan informasi yang mereka butuhkan atau memahami panduan akademik yang telah disediakan oleh universitas. Seringkali mahasiswa menghadapi tantangan dalam mencari informasi ini karena

banyaknya dokumen, aturan, dan ketentuan yang tersebar di berbagai *platform* atau media fisik. Berdasarkan hasil wawancara dari Ketua Program Studi Sistem Informasi UTM, ada beberapa keluhan yang dirasakan oleh beliau, diantaranya yaitu terlalu banyak pesan dari mahasiswa setiap harinya sehingga sulit untuk menjawab semua pesan tersebut. Dampaknya terasa secara langsung ke mahasiswa karena proses akademik terhambat oleh situasi yang ada. Dosen dan staf UTM juga terdampak di sisi *resource* yang terbatas sehingga tidak bisa fokus ke perkerjaan yang lebih produktif.

Penggunaan kecerdasan buatan, khususnya *chatbot* yang didukung oleh *Large Language Mode*l (LLM) *GPT-3.5* Turbo beserta *Framework* Langchain, menjadi solusi yang sangat relevan dan penting dalam menyediakan pelayanan informasi yang efisien. Ketika dalam percakapan, sering kali terdapat penulisan kata atau ejaan beserta tata bahasa yang berbeda tetapi memiliki maksud yang sama, hal tersebut bisa diatasi oleh model LLM karena telah dilatih untuk mempelajari data dengan volume yang sangat besar. Penggunaan LLM membuat percakapan antara pengguna dan mesin terasa seperti obrolan antar manusia. Inputan dari pengguna akan dianalisa dan dipahami oleh model LLM yang digunakan[6].

Pendekatan yang digunakan dalam penelitian ini ada *Knowladge Based*. *Dataset* yang dipakai disimpan di vektor *database*. Jawaban atau respon *chatbot* diperoleh dari hasil perhitungan kemiripan vektor inputan *user* dengan vektor di basis data vektor[7]. *Cosine Similarity* adalah suatu metrik jarak yang dapat digunkan untuk perhitungan kemiripan antara dua vektor pada studi kasus *chatbot*[8].

Perhitungan jarak vektor menggunakan *cosine similarity* dapat meningkatkan efisiensi pencarian informasi secara otomatis di basis data vektor. Oleh karena itu, basis data vektor Pinecone digunakan karena terdapat fitur *cosine* metrik yang dapat menghemat waktu dan biaya pengembangan *chatbot*.

2. TINJAUAN PUSTAKA

3.1. Penelitian Terdahulu

Adapun penelitian terdahulu yang dijadikan sebagai acuan pada penelitian ini. Telah banyak penelitian tentang *chatbot* berbasis LLM untuk penggunaan data atau kasus spesifik. *Chatbot* tanya jawab hadis berbabasis web dibangun menggunakan *framework* langchain dan model *GPT-3.5 Turbo* dari OpenAI memiliki presentase akurasi sebesar 89,4%[9].

WikiChat adalah *chatbot* berbasis LLM *GPT-3.5* dan *Colbert-v2* yang berfokus untuk mencari informasi yang bersumber dari Wikipedia[10]. Walaupun pengguna memberi pertanyaan terkait topik yang tidak populer, *chatbot* tersebut dapat memberi jawaban faktual dengan akurasi yang tinggi yaitu 82,4% dan 88,2% untuk kategori topik secara keseluruhan[10].

Kedua penelitian tersebut telah menunjukkan keberhasilan dan manfaat penggunaan *chatbot* berbasis LLM dalam berbagai sektor.

3.2. Chatbot

Chatbot adalah sebuah program komputer yang dirancang untuk berinteraksi dengan manusia melalui *chat* atau percakapan. *Chatbot* menggunakan kecerdasan buatan (AI) dan algoritma pemrosesan bahasa alami (NLP) untuk memahami, menganalisis, dan merespons pertanyaan atau perintah yang diberikan oleh pengguna[1].

Tujuan utama *chatbot* adalah memberikan solusi atau informasi kepada pengguna dengan cara yang mirip dengan percakapan manusia, menciptakan pengalaman yang lebih baik, dan mempercepat proses bisnis.

3.3. Large Language Model

Large Language Model (LLM) adalah model bahasa yang sangat besar dan kompleks yang dapat digunakan untuk menganalisis dan menggenerasikan teks. LLM berbeda dengan model bahasa lainnya karena ukurannya yang sangat besar dan kemampuannya untuk menganalisis teks dengan cara yang lebih baik dan akurat. LLM dapat dibuat untuk menganalisis teks dalam berbagai bahasa, termasuk bahasa Indonesia. Contoh aplikasi LLM yang paling umum adalah *chatbot*, sistem pengiriman pesan, dan asisten pribadi[11].

3.4. GPT-3-5 Turbo

GPT-3.5 Turbo adalah salah satu iterasi terbaru dari model bahasa yang dikembangkan oleh OpenAI. Model ini adalah model yang sangat besar dengan 20 miliar parameter dan dibangun berdasarkan pada arsitektur *Transformer*, yang telah menjadi standar dalam model bahasa terkini[12].

Transformer bekerja secara paralel yang berarti membaca kalimat secara keseluruhan. Oleh karena itu, arsitektur *transformer* memiliki jendela referensi yang jauh lebih besar daripada *Reccurence Neural Network* (RNN) atau *Long Short Term Memory* (LSTM) sehingga dapat menggunakan semua konteks kalimat ketika menggenarasikan sebuah teks baru[11].

Hal tersebut tentu menjadi salah satu keunggulan utama *GPT-3.5 Turbo* karena kemampuannya untuk memahami dan menghasilkan teks dengan tingkat akurasi yang sangat tinggi. Model ini juga multibahasa, memungkinkan penggunaan dalam berbagai bahasa. *GPT-3.5 Turbo* digunakan dalam berbagai aplikasi, seperti *chatbot* cerdas, penerjemahan otomatis, generasi konten, analisis teks, dan banyak lagi[4].

3.5. Langchain

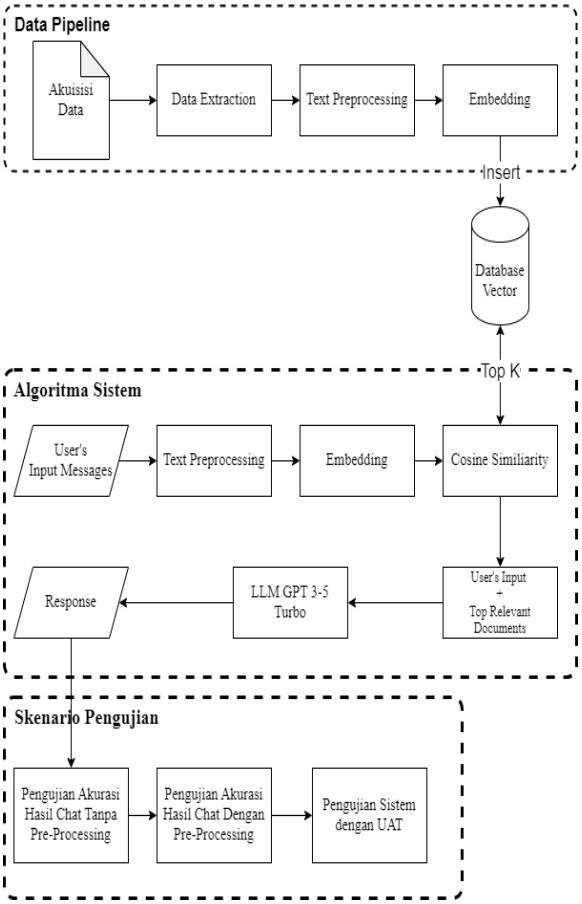
LangChain adalah sebuah *framework open-source* yang memungkinkan pengembang perangkat lunak yang bekerja dengan kecerdasan buatan (AI) dan *machine learning* untuk menggabungkan LLM dengan

komponen eksternal lainnya untuk mengembangkan aplikasi yang didukung oleh model bahasa besar atau LLM[5].

Tujuan dari LangChain adalah menghubungkan model bahasa besar seperti *GPT-3.5* dari OpenAI, dengan beragam sumber data eksternal untuk menciptakan dan mengambil manfaat dari aplikasi pemrosesan bahasa alami (NLP).

3. METODE PENELITIAN

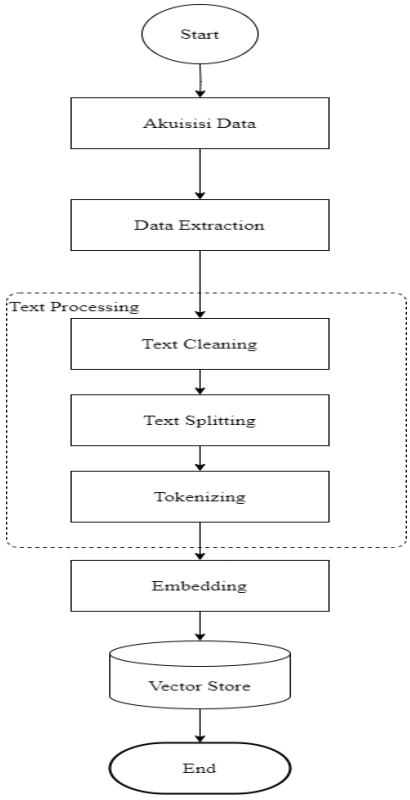
Tahapan ini disusun berdasarkan langkah-langkah yang terintegrasi *Framework* Langchain. Tahapan penelitian terdiri dari tiga tahapan inti yaitu Data Pipeline, Algoritma Sistem, dan Pengujian. Adapun detail tahapan penelitian yang akan dilakukan seperti pada Gambar 1.



Gambar 1. Diagram tahapan penelitian

3.1. Data Pipeline

Penelitian ini menggunakan data yang bersumber dari Buku Panduan Akademik Prodi Sistem Informasi Tahun 2023 dan edaran akademik terbaru. Adapun detail tahapan *data pipeline* yang akan dilakukan seperti pada Gambar 2. Data diperoleh dari laman resmi Fakultas Teknik Universitas Trunojoyo Madura dan diakses pada tanggal 10 Oktober, 2023. Buku Panduan Akademik Prodi Sistem Informasi Tahun 2023 memiliki 118 halaman dan 25825 kata-kata. Format buku yang didapatkan adalah *PDF*, maka diperlukan konversi atau ekstraksi teks supaya dapat diolah. PyPDF2 digunakan untuk ekstraksi PDF karena sangat cepat dalam proses ektraksi teks dan *outputnya* berdasarkan halaman beserta metadata-nya. Terdapat 18 data berupa data tabel pada buku tersebut. Tabel-tabel tersebut diubah menjadi sebuah narasi teks secara manual sehingga mempermudah teks dapat diproses dengan makna yang benar.



Gambar 2. *Flowchart* data pipeline hingga insert atau memasukkan data ke basis data vektor

Sebelum memasukkan data ke dalam aplikasi LLM, penting dilakukan *Text Preprocessing*. Adapun tahapan pada *text preprocessing* yang akan dilakukan dalam penilitian yaitu *Text Cleaning*, *Text Splitter*, dan *Tokenizing*. Di akhir pengujian, akan dilakukan perbandingan akurasi respon *chatbot* yang telah melalui *text preprocessing* dan tanpa melalui text *preprocessing*. Tahapan *cleaning* atau pembersihan data digunakan untuk menghapus karakter yang dianggap kurang penting atau tidak relevan dalam konteks analisis data seperti karakter-karakter khusus. *Whitespace*, baris kosong, dan standarisasi *bullet point* juga dilakukan pada tahapan ini. Manfaat membersihkan teks tersebut untuk memudahkan proses *embedding* yang lebih akurat.

Fungsi *Text Splitter* dari *framework* LangChain yang cocok untuk digunakan penelitian ini yaitu *Recursive Character Text Splitter*. Fungsi ini mengambil daftar karakter. Kemudian mencoba membuat potongan berdasarkan pemisahan pada karakter pertama, tetapi jika ada potongan yang terlalu besar, maka bergerak ke karakter berikutnya, dan selama mungkin hingga ditambahkan *overlap*

(beberapa karakter terakhir akan muncul di potongan selanjutnya). Ukuran potongan yang dipakai adalah 500 dan 800 *token*.

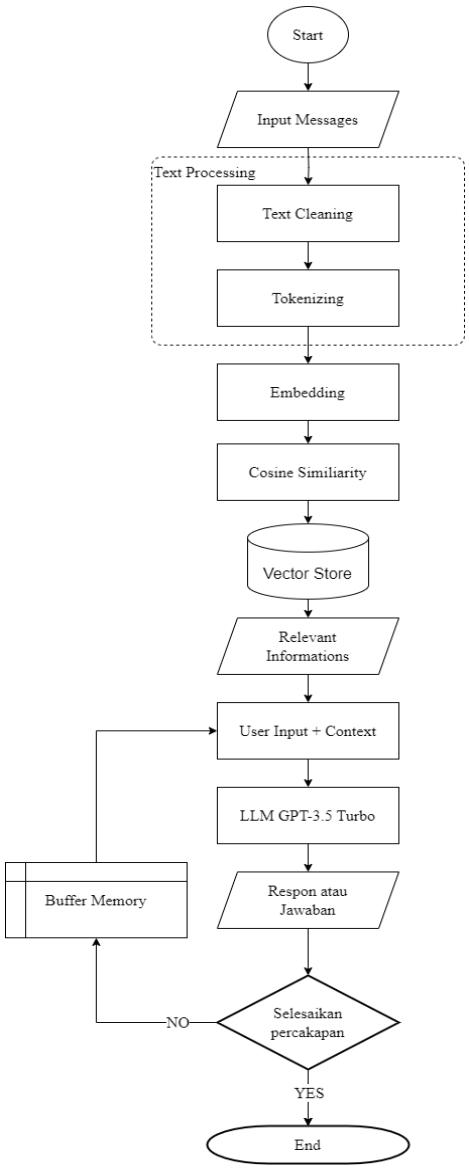
Hasil dari tahapan sebelumnya akan dilakukan *Tokenizing*. Pada tahap ini kata atau kalimat dipisahkan kata demi kata untuk meningkatkan pemrosesan dan penghitungan token yang akan digunakann oleh model *embedding*. Proses ini dilakukan dalam penggunaan model *embedding* OpenAI. Data yang telah diproses sebelumnya diubah menjadi vektor karena data akan disimpan di basis data vektor. Proses *text embedding* pada penelitian ini menggunakan API model *Text-embedding-3-small*. Hasil dari proses ini adalah berupa nilai vektor dengan dimensi sebanyak 1536 dan akan disimpan di basis data vektor, Pinecone.

3.2. Algoritma Sistem

Pendekatan yang digunakan dalam pengembangan sistem ini adalah *Knowledge Based* sehingga tidak melakukan *fine tuning* atau melatih ulang model yang digunakan. Informasi yang dibutuhkan akan disimpan di *vector database* dengan cara *embedding* menggunakan API *Embedding* model dari OpenAI. Ketika membuat index di Pinecone, terdapat pilihan jarak metrik untuk menentukan cara perhitungan kemiripan vektor. *Cosine Similarity* dipilih karena sesuai rekomendasi dari OpenAI dan cocok dengan model *Embedding* dari OpenAI, *Text-embedding-3-small*.

Pada algoritma yang digunakan, user akan melakukan input perintah atau pertanyaan ke sistem melalui *chatbox* atau *textbox* pada tampilan aplikasi. *Inputan* tersebut akan dilakukan teks *pre-processing* dengan menggunakan metode *Text Cleaning* untuk menghapus karakter khusus dan menghapus spasi berlebihan. Kemudian dilakukan *Tokenzing* untuk memecah teks menjadi unit-unit kecil seperti kata-kata. Setelah teks *pre-processing* selesai, dilakukan konversi teks ke vektor (*Embedding*) untuk dikirim ke basis data vektor dan dicarikan vektor dengan titik terdekat dengan vektor inputan user menggunakan fitur pencarian kemiripan *cosine* dari Pinecone. Jika ditemukan vektor dengan nilai kemiripan lima tertinggi (*top-k*), informasi yang ada di vektor tersebut akan dikirimkan ke sistem untuk diproses dengan *query* API LLM sebagai konteks.

Pertanyaan atau inputan awal user ditambah dengan konteks informasi yang didapatkan dari basis data, akan diolah menggunakan API model *gpt-3.5 Turbo* untuk menghasilkan dan menyusun jawaban atau informasi terkait panduan akademik Prodi Sistem Informasi yang akurat sesuai konteks inputan pengguna. Model *gpt-3.5 Turbo* harus diberi batasan dalam menjawab dan mengolah informasi (LLM *System Setting*). Batasan yang paling penting adalah *chatbot* hanya menjawab jika pertanyaan *user* seputar Sistem Informasi UTM, menjawab dengan Bahasa Indonesia, dan tidak boleh mengada-ada jika jawaban tidak ditemukan atau nilai *cosine similarity* sangat rendah. Respon yang telah dibuat oleh model akan ditampilkan di tampilan antarmuka streamlit *chat*. Dan respon juga akan disimpan sebagai riwayat percakapann di *buffer memory* menggunakan modul *ConversationBufferWindowMemory* dari LangChain. *Buffer Memory* hanya menyimpan K percakapan terakhir untuk menyimpan konteks dan tidak melebihi batasan *token gpt-3.5 Turbo*. Data yang telah disimpan di *buffer memory* akan dikembalikan ketika pengguna menginputkan pertanyaan lagi dan akan diolah dengan API *gpt-3.5 Turbo* bersama konteks yang sudah ada.



Gambar 4. *Flowchart* sistem chatbot panduan akademik Prodi Sistem Informasi

3.3. Skenario Pengujian

Tahap pengujian dilakukan untuk mengetahui seberapa akurat jawaban dan tingkat kepuasan pengguna *chatbot*. Pengujian-pengujian yang dilakukan yaitu pengujian akurasi jawaban dengan dan tanpa *text preprocessing*, dan *user acceptance test*. Akurasi akan ditentukan dari hasil pengujian dengan 6

pertanyaan valid seputar Prodi Sistem Informasi. Setiap pertanyaan memiliki 5 variasi pertanyaan yang inti pokoknya sama, dengan mengharapkan respon jawaban yang sesuai dengan pertanyaan yang diajukan. Pengujian akan dilakukan langsung di aplikasi *chatbot* dengan menginputkan pertanyaan-pertanyaan yang telah disiapkan. Pada penelitian ini daftar pertanyaan yang diujikan didapat dari kepala program studi Sistem Informasi dan buku panduan akademik.. Hasil pengujian ini ditentukan dengan membagi jumlah jawaban yang sesuai dengan total jawaban dan di kalikan 100[1].

Berikut cara perhitungannya:

$$Akurasi = \frac{Jumlah\ Jawaban\ Sesuai}{Total\ Pertanyaan} \times 100\% \quad (1)$$

Pengujian pada sistem dilakukan menggunakan *User Acceptance Test* (UAT). Uji Coba Kelayakan Pengguna (UAT) dilakukan untuk memastikan bahwa perangkat lunak yang telah dikembangan sesuai dengan kebutuhan yang ada. Keluaran dari UAT adalah menentukan apakah perangkat lunak layak digunakan atau tidak oleh pengguna[1].

Pengujian UAT merupakan salah satu tahapan penting dalam siklus pengembangan perangkat lunak karena langsung melibatkan pengguna akhir. UAT sangat diperlukan karena dapat mengetahui perspektif pengguna akhir, validasi kesesuain dengan kebutuhan bisnis, mengidentifikasi masalah sistem, dan meningkatkan kualitas dan kepuasan pengguna. Setelah mendapat data dari responden maka dapat dihitung presentase UAT dengan persamaan berikut ini :

Berdasarkan perhitungan yang menyatakan nilai tertinggi tersebut dapat dicari presentase sebagaimana berikut:

$$Presentase = \frac{Jumlah\ skor\ total}{Nilai\ tertinggi}\ x\ 100\% \quad (2)$$

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengembangan

Antarmuka aplikasi *chatbot* dibangun menggunakan *framework* Streamlit. Pada halaman utama terdapat area *Chatbox*, digunakan untuk menampilkan riwayat atau urutan percakapan antara pengguna dan LLM. Pengguna dapat memasukkan pertanyaan atau *query* di kolom input di bagian bawah halaman. Untuk menambahkan informasi baru terkait program studi, dapat menggunakan kolom *upload* dokumen di bagian kiri halaman. Terdapat juga *radio button* untuk memilih menggunakan *preprocessing* atau tidak, *radio button* untuk memilih ukuran *parameter* (*chunks*, *overlap*), dan *slider* untuk menentukan ukuran dokumen yang dikembalikan atau *top-k*.

4.2. Hasil Pengujian dan Analisa

Pengujian akurasi *chatbot* dilakukan langsung menggunakan antarmuka yang sudah ada. Pertanyaan yang diuji terdapat 6 pertanyaan, dengan masing-masing pertanyaan memiliki 5 variasi pertanyaan. Variasi tersebut digunakan untuk menguji pertanyaan dengan kalimat yang mirip atau memiliki maksud yang sama dengan harapan jawaban yang dihasilkan konsisten. Pada pengujian ini pertanyaannya yang digunakan adalah pertanyaan-pertanyaan seputar program Studi Sistem Informasi dan Universitas Trunojoyo Madura yang didapatkan dari kepala program studi, pertanyaan yang sering ditanyakan oleh mahasiswa ke staff, dan pertanyaan spesifik untuk menguji kapabilitas chatbot.. Semua pertanyaan yang diujikan sudah benar-benar memiliki jawaban yang bersumber dari informasi di basisdata vektor.

Masing-masing pertanyaan memiliki tujuan pengujian yang berbeda. Pertanyaan nomor satu menguji pemrosesan logika dengan data yang sederhana. Pertanyaan nomor dua menguji pencarian pada kata yang sering muncul di dokumen seperti SKS. Pertanyaan nomor tiga menguji kemampuan *retrieval* dengan data yang berkelanjutan dan panjang. Pertanyaan nomor empat menguji pencarian informasi di dokumen yang berbeda. Pertanyaan nomor lima menguji kemampuan memproses data berupa tabel. Dan pertanyaan nomor lima menguji dengan informasi yang tidak terstruktur. Pengujian akurasi dilakukan empat kali yaitu menggunakan parameter *chunks*, *overlap*, *top-k* berbeda di tiap pengujian dengan *text preprocessing* dan tanpa *text preprocessing*. Hasil masing-masing pengujian menunjukkan pendekatan yang memiliki nilai akurasi yang lebih tinggi dan selanjutnya dapat dianalisa. Pengujian dilakukan 4 kali dan hasilnya pada Tabel 1

Tabel 1. Hasil Pengujian

Hasil Uji Pertanyaan					
No.	**Pertanyaan**	**Kesimpulan dengan *Preprocess***		**Kesimpulan tanpa *Preprocess***	
		Pertama	**Kedua**	**Pertama**	**Kedua**
1.1	Berapa jumlah SKS yang bisa saya ambil jika IP saya 3.3	Sesuai	Sesuai	Sesuai	Sesuai
1.2	IP semester lalu saya adalah 3.3, sekarang saya bisa ambil berapa sks?	Sesuai	Sesuai	Sesuai	Sesuai
1.3	IP aku 3.3. semester ini bisa berapa sks ya?	Sesuai	Sesuai	Sesuai	Sesuai
1.4	Apakah saya bisa ambil 23 sks jika IP saya 3.3?	Sesuai	Sesuai	Sesuai	Sesuai
1.5	Apakah ip 3.3 bisa mengambil 23 sks?	Sesuai	Sesuai	Sesuai	Sesuai
2.1	Berapa sks yang harus saya punya untuk syarat kelulusan prodi sistem informasi?	Sesuai	Sesuai	Sesuai	Sesuai

Hasil Uji Pertanyaan					
No.	**Pertanyaan**	**Kesimpulan dengan *Preprocess***		**Kesimpulan tanpa *Preprocess***	
		Pertama	**Kedua**	**Pertama**	**Kedua**
2.2	Apakah ada minimum sks yang harus ditempuh untuk syarat kelulusan sistem informasi? Dan berapa?	Sesuai	Sesuai	Sesuai	Sesuai
2.3	Apakah saya bisa lulus jika total SKS yang saya tempuh kurang dari yang ditentukan?	Tidak Sesuai	Sesuai	Sesuai	Sesuai
2.4	Berapa jumlah SKS wajib yang harus saya selesaikan untuk lulus dari prodi Sistem Informasi?	Tidak Sesuai	Tidak Sesuai	Tidak Sesuai	Sesuai
2.5	Berapa sks yg diperlukan untuk kelulusan	Sesuai	Sesuai	Tidak Sesuai	Sesuai
3.1	Apa saja syarat dan langkah untuk mengikuti program KKN ?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
3.2	Syarat untuk KKN apa aja?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
3.3	Apakah dengan mahasiswa yang belum menempuh 84 sks bisa mengikuti kkn? apa syaratnya?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
3.4	Apa saja persyaratan KKN tambahan yang harus dipenuhi selain yang ada di KRS?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
3.5	Berapa minimum IPK untuk bisa mendaftar KKN? Apa syaratnya?	Sesuai	Sesuai	Sesuai	Sesuai
4.1	Apa itu program fast track? Apa syaratnya?	Sesuai	Sesuai	Sesuai	Sesuai
4.2	Saya angkatan 2019, bagaimana caranaya untuk ikut fast track?	Sesuai	Sesuai	Sesuai	Sesuai
4.3	Fast track itu apa?	Sesuai	Sesuai	Sesuai	Sesuai
4.4	Apa syarat program fast track?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
4.5	Beri saya info tentang fast track	Sesuai	Sesuai	Sesuai	Sesuai
5.1	Mata kuliah sosio teknlogi informasi berapa sks? prasyaratnya apa?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
5.2	Sosio tekno berapa sks	Sesuai	Sesuai	Sesuai	Sesuai
5.3	Saya ingin mengambil mata kuliah sosio teknologi informasi, apakah ada prasyaratnya? berapa sks mata kuliah itu?	Sesuai	Sesuai	Tidak Sesuai	Sesuai
5.4	Sosio teknologi informasi berapa sks dan apa prasyaratnya?	Tidak Sesuai	Sesuai	Tidak Sesuai	Sesuai
5.5	Apa prasyarat matakuliah sosio tekno	Sesuai	Sesuai	Tidak Sesuai	Sesuai
6.1	Mata kuliah algoritma pemrograman itu bahas apa saja? dan berapa sks nya? Praktikumnya bagaimana	Sesuai	Sesuai	Tidak Sesuai	Tidak Sesuai
6.2	Algoritma pemrograman berapa sks? Apa yang dibahas di mata kuliah tersebut?	Sesuai	Sesuai	Sesuai	Sesuai
6.3	Apakah alpro ada praktikum? dan berapa bobot praktikumnya	Tidak Sesuai	Tidak Sesuai	Tidak Sesuai	Tidak Sesuai
6.4	Berapa sks algoritma pemrograman? Praktikum ada?	Tidak Sesuai	Sesuai	Tidak Sesuai	Sesuai
6.5	Bobot algoritma pemrgorman berapa sks? Berapa sks praktikumnya?	Sesuai	Sesuai	Tidak Sesuai	Tidak Sesuai

Pengujian pertama dengan *preprocess* pada Tabel 1 dilakukan dengan data yang telah dilakukan *preprocessing* pada data *pipeline*. Data tabel diubah menjadi narasi teks secara manual untuk memudahkan proses *embedding*. Kemudian dihapusnya angka halaman dan *header* dokumen. Lalu dilakukan *cleaning* data seperti *whitespace*, baris kosong, menghapus karakter khusus, dan membuat standar akhir baris. *Bullet point* juga dibuat standar baru supaya sama semua. Terakhir dilakukan fungsi *strip()* untuk memastikan tidak ada *whitespace* atau baris berlebihan setelah proses *cleaning*.

Di pengujian pertama dengan *text preprocessing* pada Tabel 1 menggunakan *paramater chunks_size* = 500, *overlap* = 50, dan *top-k* = 2. Pengujian pertama mendapatkan akurasi sebesar 83,33%. Sedangkan di pengujian kedua dengan *text preprocessing* menggunakan *paramater chunks_size* = 800, *overlap* = 150, dan *top-k* = 3. Pengujian kedua mendapatkan peningkatan akurasi menejadi sebesar 93,33%.

Pengujian pertama tanpa *preprocess* pada Tabel 1 dilakukan dengan data yang tanpa dilakukan *preprocessing* pada data *pipeline*. Data tabel dibiarkan seperti aslinya untuk menguji nilai akurasi dengan pendekatan tanpa *preprocess*. Proses *text cleaning* dilewatkan. *Text splitting* dan *embedding* tetap dilakukan seperti pada data *pipeline*.

Di pengujian pertama tanpa *text preprocessing* pada Tabel 1 menggunakan *paramater chunks_size* = 500, *overlap* = 50, dan *top-k* = 2. Pengujian pertama mendapatkan akurasi sebesar 50%. Sedangkan di pengujian kedua tanpa *text preprocessing* menggunakan *paramater chunks_size* = 800, *overlap*

= 150, dan *top-k* = 3. Pengujian kedua mendapatkan peningkatan akurasi menejadi sebesar 90%.

4.3. Analisa Hasil Pengujian Akurasi

Setelah dilakukan pengujian akurasi dengan berbagai macam kondisi atau parameter pada Tabel 1, menunjukkan bahwa banyak faktor penentu tingkat akurasi *chatbot*. Pengujian pertama dengan *text preprocessing* (*chunks size* = 500, *overlap* = 50, *top-k* = 2) mendapatkan hasil akurasi sebesar 83,33%. Dari lima pertanyaan yang tidak bisa terjawab, penyebabnya yaitu jendela konteks yang terpotong atau murni kesalahan pencarian kemiripan vektor sehingga informasi yang diterima dari pinecone tidak sesuai dengan *query*. Jika pertanyaan nomor 5.4 dianalisa, pada pengujian pertama *retrieval* mendapatkan kode mata kuliah sosio teknologi informasi SI62006. Sedangkan pada pertanyaan 5.1, mendapatkan kode SI72001 yang mana memiliki informasi yang sesuai dengan konteks.

Pada pertanyaan nomor 5.4, pengujian pertama tidak mendapatkan jawaban sesuai karena informasi atau dokumen yang dibutuhkan terpotong oleh batasan *top-k*=2. Oleh karena itu, pengujian kedua (*chunks size* = 800, *overlap* = 150, *top-k* = 3) dengan *top-k*=3 dapat menjawab pertanyaan sesuai harapan. Walaupun pertanyaan nomor 5.1 dan 5.4 memiliki inti pertanyaan yang sama dan seharusnya memiliki jawaban yang serupa, namun pada pencarian kemiripan, nomor 5.1 memiliki *confidence score* 0,66 dan nomor 5.4 memiliki *confidence score* 0,59. Hal ini menunjukkan bahwa *embedding* model sangat sensitif terhadap penulisan atau ungkapan pada pertanyaan. Solusi untuk mengatasi hal tersebut adalah dengan menambah ukuran parameter seperti pada pengujian kedua.

Hasil akurasi pengujian tanpa *text preprocessing* jauh lebih rendah daripada dengan *text preprocessing*. Pengujian pertama tanpa *text preprocessing* mendapatkan akurasi sebesar 50% dan yang kedua mendapatkan akurasi sebesar 90%. Salah satu faktor yang menyebabkan rendahnya nilai akurasi adalah hasil *embedding* data berupa tabel yang tanpa diproses atau diubah menjadi narasi teks sebelumnya. Pertanyaan nomor 5 dan 6 adalah pertanyaan yang diujikan menggunakan informasi yang berupa tabel. Pengujian pertama tanpa *text preprocessing* hanya berhasil menjawab satu dari lima pertanyaan serupa. Namun, pada pengujian kedua (*chunks size* = 800, *overlap* = 150, *top-k* = 3) berhasil menjawab 5 dari 5 pertanyaan dengan sesuai harapan.

Presentase akurasi secara keseluruhan dari pengujian tanpa *text preprocess* pertama dan kedua meningkat sebesar 80% (dari 50% ke 90%). Hal ini menunjukkan bahwa dengan *text preprocessing* dan tanpa *text preprocessing* dapat memiliki selisih performa yang kecil yaitu 93,333% dan 90%. Oleh karena itu, selain *text preprocessing*, faktor yang mempengaruhi nilai akurasi *chatbot* adalah jumlah *chunks*, jumlah *overlap*, dan *top-k* yang digunakan. Jika *chunks* atau potongan data berukuran terlalu besar maka akan lebih sulit untuk mencari informasi relevan yang tepat. Sebaliknya, jika *chunks* terlalu kecil maka konteks akan terpotong sehingga sulit untuk memahami informasi yang panjang. Ukuran *overlap* yang besar meningkatkan akurasi *similiarty search* dan mengurangi hilangnya informasi. Namun, jika *overlap* terlalu besar maka berpotensi menimbulkan *noise* sehingga menurunkan akurasi pencarian.

Dikarenakan terdapat pertanyaan yang dapat dijawab pada pengujian dengan *preprocessing* tetapi tidak dijawab pada pengujian tanpa preprocessing dan sebaliknya, maka dibuat iterasi terbaru yang menggabungkan dua pendekatan tersebut menjadi satu *query*. *Combined Query* adalah menggabungkan hasil pencarian kemiripan dengan dan tanpa preprocessing untuk mendapatkan dokumen-dokumen dari dua pendakatan sekaligus sehingga LLM dapat menggenerasikan jawaban dengan konteks yang lebih luas. Pengujian *combined query* menggunakan parameter yang sama seperti pengujian kedua pada "*with*" dan "*without*" *preprocess*. Hasil pengujian menunjukkan bahwa kombinasi *query* tidak dapat menjawab pertanyaan hanya pada nomor 6.3 karena di pengujian dengan dan tanpa *preprocessing* memang sama-sama tidak dapat menjawab dengan sesuai. Nilai akurasi yang didapatkan adalah sebesar 96,66%. Walaupun nilai akurasi meningkat, *response time* dari *chatbot* mengalami penurunan karena *query* nya dua kali lipat dari pendekatan sebelum-sebelumnya.

4.4. Hasil Pengujian Sistem dengan User Acceptance Test

Pengujian *User Acceptance Test* pada penelitian ini melibatkan ketua prodi Sistem Informasi, dosen UTM, dan 32 mahasiswa Sistem Informasi. Berikut hasil pengujian UAT:

$$\begin{aligned} &Jumlah\ skor\ dari\ responden\ yang\ menjawab\ SS\ = 87\ x\ 5 \\ &Jumlah\ skor\ dari\ responden\ yang\ menjawab\ S\ = 143\ x\ 4 \\ &Jumlah\ skor\ dari\ responden\ yang\ menjawab\ RR\ = 35\ x\ 3 \\ &Jumlah\ skor\ dari\ responden\ yang\ menjawab\ TS\ = 7\ x\ 2 \\ &Jumlah\ skor\ dari\ responden\ yang\ menjawab\ STS\ = 0\ x\ 5 \\ &\text{Jumlah skor total } = 1126 \end{aligned} \quad (3)$$

Hasil jawaban dari responden sebanyak 34 orang, kemudian dapat dihitung nilai tertinggi dan terendah seperti berikut:

$$\begin{aligned} &Nilai\ tertinggi = 34\ x\ 8\ pertanyaan\ x\ 5\ poin = 1360\ (SS) \\ &Nilai\ terendah = 34\ x\ 8\ pertanyaan\ x\ 1\ poin = 272\ (j\ STS) \end{aligned} \quad (4)$$

Berdasarkan perhitungan yang menyatakan nilai tertinggi tersebut dapat dicari presentase sebagaimana berikut:

$$Presentase = \frac{1126}{1360}\ x\ 100\% = 82{,}79\% \quad (5)$$

Nilai presentase yang didapatkan dari pengujian UAT yaitu sebesar 82,79%. Menurut pedoman tingkat *usability* pengujian UAT maka hasil pengujian termasuk dalam kategori sangat baik.

5. KESIMPULAN

Hasil pengujian akurasi menunjukkan bahwa pendekatan menggunakan *text preprocessing* memiliki nilai akurasi lebih tinggi daripada pendekatan tanpa menggunakan *text preprocessing* pada data *pipeline*. Performa chatbot juga ditentukan dari parameter yang digunakan. Pertanyaan yang tidak terjawab sesuai harapan kemungkinan disebabkan oleh terpotongnya informasi dari jendela konteks, hasil pemrosesan teks, jenis sumber data, dan rendahnya hasil pencarian kemiripan mengguna jarak matrik *cosine* di pinecone. Dari lima pengujian akurasi yang telah dilakukan menunjukkan bahwa nilai akurasi terbaik diperoleh dari penggabungan *query* dengan dan tanpa *preprocessing* yaitu sebesar 96,66%.

Namun, *combined query* membuat respon *chatbot* lebih lambat karena bekerja dua kali daripada pendekatan lainnya. Berdasarkan pengelolaan hasil kuisioner yang sudah dikumpulkan menunjukkan hasil dari kinerja sistem dan kepuasan pengguna mendapatkan nilai *User acceptance test* sebesar 82,79% dan dapat dikategorikan sangat kuat. Adapun saran untuk penelitian selanjutnya yaitu proses pengujian nilai akurasi dilakukan dengan lebih banyak data dan kombinasi parameter lagi untuk mencari titik terbaik pada studi kasus yang digunakan dan tahapan pemrosesan data dilakukan dengan pendekatan yang lebih baik. Terutama data berupa data tabel.

DAFTAR PUSTAKA

[1] Nuzul Hikmah, Dyah Ariyanti, dan Ferry Agus Pratama, "Implementasi Chatbot Sebagai Virtual Assistant di Universitas Panca Marga Probolinggo menggunakan Metode TF-IDF," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 4, no. 2, hlm. 133–148, Agu 2022,

[2] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, dan D. Yang, "Is ChatGPT a General-Purpose Natural Language Processing Task Solver?," Feb 2023,

[3] T. Wu *dkk.*, "A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, hlm. 1122–1136, Mei 2023,

[4] J. Ye *dkk.*, "A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models," Mar 2023.

[5] F. Soygazi dan D. Oguz, "An Analysis of Large Language Models and LangChain in Mathematics Education," 2023.

[6] S. J. Semnani, V. Z. Yao, H. C. Zhang, dan M. S. Lam, "WikiChat: Stopping the Hallucination of Large Language Model Chatbots by Few-Shot Grounding on Wikipedia," Mei 2023,

[7] Nurhapiza, N. S. Harahap, M. Fikry, dan M. Affandes, "Penerapan Chatbot pada Aplikasi Web Tanya Jawab Tentang Fiqih Jual Beli Islam Menggunakan LangChain," *Journal of Computer System and Informatics (JoSYC)*, vol. 5, no. 3, 2024, Diakses: 12 Januari 2025.

[8] K. Park, J. S. Hong, dan W. Kim, "A Methodology Combining Cosine Similarity with Classifier for Text Classification," vol. 34, no. 5, hlm. 396–411, Apr 2020,

[9] M. Irfan Syah, N. Safaat Harahap, Novriyanto, dan S. Sanjaya, "Penerapan Retrieval Augmented Generation Menggunakan Langchain dalam Dalam Pengembangan Sistem Tanya Jawab Hadis Berbasis Web," *ZONAsi*, vol. 6, no. 2, 2024, Diakses: 12 Januari 2025.

[10] S. J. Semnani, V. Z. Yao, H. C. Zhang, dan M. S. Lam, "WikiChat: A Few-Shot LLM-Based Chatbot Grounded with Wikipedia," Mei 2023,

[11] A. Tamkin, M. Brundage, J. Clark, dan D. Ganguli, "Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models," Feb 2021,

[12] M. Singh, J. Cambronero, S. Gulwani, V. Le, C. Negreanu, dan G. Verbruggen, "CodeFusion: A Pre-trained Diffusion Model for Code Generation," Okt 2023