Artikel Tugas Akhir Achmad Maulana Rochman.pdf

by luthfanhakim07@gmail.com 1

Submission date: 10-Feb-2025 10:16PM (UTC-0800)
Submission ID: 2574415611
File name: Artikel_Tugas_Akhir_Achmad_Maulana_Rochman.pdf (327.49K)
Word count: 2488
Character count: 11802

AI Assistant-Based Chatbot Application in Solving Coding Script Issues Using the Ollama Large Language Models
[Aplikasi Chatbot Berbasis AI Assistant dalam Menyelesaikan Permasalahan Coding Script Menggunakan Model Large Language Models Ollama]

Achmad Maulana Rochman[1)], Sumarno*[2)], Suprianto[3)], Irwan Alnarus Kautsar[4)]

[1)]Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia
[2)]Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia
[3)]Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia
[4)]Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: sumarno@umsida.ac.id

Abstract. *The advancement of artificial intelligence (AI) technology has opened up significant opportunities for developing systems capable of understanding and generating natural language. One of its applications is AI-based chatbots that assist software developers in solving programming-related issues. This study aims to design and implement an AI-based chatbot application using Large Language Models (LLM) from Ollama with a Streamlit-based interface. The application enables users to interact with the chatbot to obtain solutions for debugging and code recommendations. The system development follows the Waterfall methodology, including requirements analysis, design, implementation, testing, and maintenance. The AI model runs locally using Ollama, while Streamlit serves as the user interface to facilitate access and interaction. Testing results indicate that the application provides accurate and relevant responses, though system response time still requires optimization. With this chatbot, software developers are expected to enhance their work efficiency in resolving technical issues quickly and accurately.*

Keywords - *Chatbot; Artificial Intelligence; Large Language Models; Ollama; Streamlit; Programming*

Abstrak. *Perkembangan teknologi kecerdasan buatan (AI) telah membuka peluang besar dalam pengembangan sistem yang mampu memahami dan menghasilkan bahasa alami. Salah satu penerapannya adalah chatbot berbasis AI yang dapat membantu pengembang perangkat lunak dalam menyelesaikan permasalahan pemrograman. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi chatbot berbasis AI menggunakan model Large Language Models (LLM) dari Ollama dengan antarmuka berbasis Streamlit. Aplikasi ini memungkinkan pengguna untuk berinteraksi dengan chatbot guna mendapatkan solusi terkait debugging, dan rekomendasi kode. Pengembangan sistem dilakukan menggunakan metode Waterfall, dengan tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Model AI dijalankan secara lokal menggunakan Ollama, sementara Streamlit digunakan sebagai antarmuka untuk memudahkan akses dan interaksi pengguna. Hasil pengujian menunjukkan bahwa aplikasi dapat memberikan respons yang akurat dan relevan, meskipun masih perlu dilakukan optimalisasi dalam waktu respons sistem. Dengan adanya chatbot ini, diharapkan pengembang perangkat lunak dapat meningkatkan efisiensi kerja dalam menyelesaikan permasalahan teknis secara cepat dan akurat.*

Kata Kunci - *Chatbot; Kecerdasan Buatan; Large Language Models; Ollama; Streamlit; Pemrograman*

I. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) semakin pesat dalam beberapa dekade terakhir, memberikan dampak besar pada berbagai sektor. Salah satu cabang AI yang mengalami kemajuan signifikan adalah pemrosesan bahasa alami (Natural Language Processing/NLP) . Model-model bahasa besar (Large Language Models/LLM) seperti GPT (Generative Pre-trained Transformer), mampu memproses dan menghasilkan teks dengan tingkat akurasi yang mendekati kecerdasan manusia [1]. Hal ini membuka peluang besar bagi pengembangan aplikasi yang memanfaatkan AI untuk memahami dan menghasilkan bahasa manusia dalam berbagai konteks, termasuk dalam aplikasi chat berbasis AI.

Aplikasi chatbot berbasis AI telah banyak diterapkan dalam berbagai sektor, seperti layanan pelanggan, pendidikan, hingga asisten virtual pada perangkat mobile dan dekstop [2]. Sebagian besar aplikasi ini berbasis web dan memanfaatkan konektivitas internet untuk mengakses model AI yang dijalankan di server pusat. Dengan penggunaan teknologi cloud, aplikasi ini dapat menawarkan fleksibilitas dalam pengembangan dan pembaruan sistem yang lebih cepat, memungkinkan pemanfaatan model bahasa besar seperti yang ditawarkan oleh platform Ollama,

yang dapat mengoptimalkan pengalaman pengguna dalam memberikan jawaban yang relevan dan akurat terkait pemrograman.

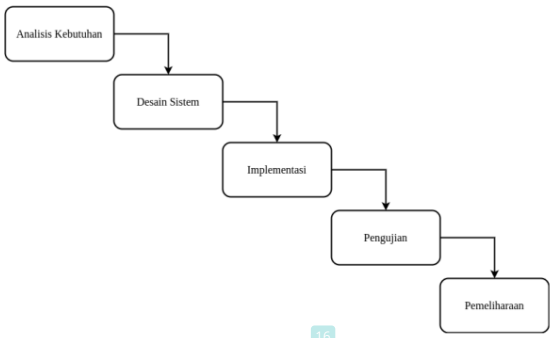
Ollama adalah salah satu platform yang menyediakan akses ke berbagai model bahasa besar yang dapat diintegrasikan dengan aplikasi lokal. Platform ini memungkinkan pengembang untuk membangun aplikasi dengan model AI yang dapat beroperasi tanpa bergantung pada server pusat, yang sangat penting dalam konteks aplikasi pengembangan perangkat lunak [3]. Penggunaan Ollama dalam aplikasi berbasis web menawarkan keuntungan dalam memanfaatkan model bahasa besar untuk tugas-tugas spesifik seperti pemrograman, debugging, atau pelatihan mandiri [4].

Untuk memudahkan pembuatan antarmuka pengguna, penelitian ini memanfaatkan Streamlit, sebuah framework open-source yang memungkinkan pengembang membangun aplikasi berbasis web interaktif dengan cepat dan mudah. Streamlit memungkinkan pembuatan aplikasi yang dapat diakses oleh pengguna tanpa tanpa memerlukan keterampilan desain frontend yang mendalam, sehingga dapat mempercepat proses pengembangan dan fokus pada logika aplikasi [5]. Dengan Streamlit, aplikasi berbasis web yang mengintegrasikan model AI dapat dengan mudah diakses dan digunakan oleh pengembang untuk mencari solusi cepat dalam debugging atau menjawab pertanyaan seputar bahasa pemrograman.

Berdasarkan kebutuhan tersebut, penelitian ini berfokus pada perancangan dan implementasi aplikasi chatbot berbasis AI menggunakan Streamlit dan memanfaatkan model bahasa besar Ollama. Aplikasi ini dirancang untuk membantu pengembang dalam menyelesaikan masalah terkait bahasa pemrograman atau kode, serta memberikan solusi cepat dalam proses debugging.

II. METODE

Metodologi penelitian berperan penting dalam memastikan proses pengembangan dan evaluasi aplikasi sesuai dengan tujuan yang telah ditetapkan. Penelitian ini menggunakan metode Waterfall untuk pengembangan aplikasi chatbot berbasis AI Assistant yang berfokus pada dukungan pengembang perangkat lunak. Metode ini dipilih karena pendekatan yang terstruktur seperti pada Gambar 1 [6].



Gambar 1. Diagram Proses Metode Waterfall

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

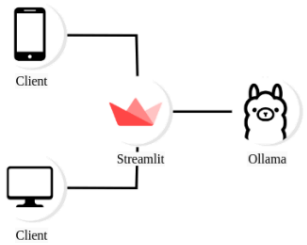
Pada tahap ini, dilakukan pengumpulan kebutuhan sistem berdasarkan studi literatur dan wawancara dengan calon pengguna, yaitu pengembang perangkat lunak. Analisis kebutuhan bertujuan untuk memahami fitur-fitur yang dibutuhkan, seperti kemampuan chatbot dalam menjawab pertanyaan teknis terkait debugging, rekomendasi kode, dan dokumentasi pemrograman.

B. Desain Sistem

Tahap ini melibatkan perancangan komprehensif untuk membangun sistem chatbot berbasis web yang menggunakan model Large Language Models (LLM) dari Ollama sebagai inti pengolahan bahasa alami dan framework Streamlit sebagai antarmuka pengguna. Desain sistem mencakup berbagai aspek berikut :

Arsitektur Sistem

Arsitektur sistem dirancang untuk mengintegrasikan teknologi utama yang digunakan, seperti virtual machine, Ollama untuk model AI, dan Streamlit sebagai framework frontend. Sistem dirancang agar dapat berjalan secara efisien dengan komponen-komponen yang saling terhubung.



Gambar 2. Arsitektur Sistem

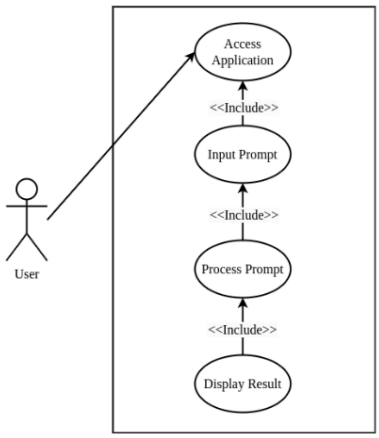
Desain Wireframe

Wireframe menggambarkan kerangka dasar antarmuka pengguna aplikasi, termasuk tata letak halaman input prompt, tombol eksekusi, dan area tampil hasil. Wireframe membantu dalam merancang alur interaksi pengguna yang intuitif [7].

Gambar 3. Desain Wireframe

Usecase Diagram

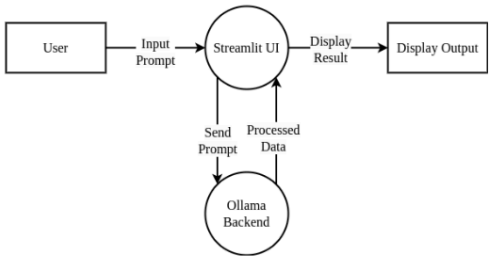
Diagram use case digunakan untuk memodelkan hubungan antara aktor (pengguna) dan sistem, serta menjelaskan fungsionalitas utama seperti memasukkan prompt, memvalidasi koneksi model, memproses input, dan menampilkan output [6].



Gambar 4. Usecase Diagram

Data Flow Diagram

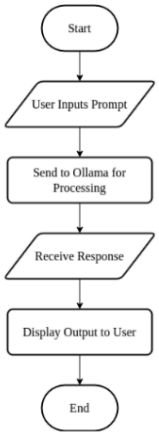
Diagram alur data menjelaskan bagaimana data mengalir di dalam sistem, dari pengguna ke backend, hingga model Ollama, dan kembali ke pengguna dalam bentuk hasil. Diagram ini mencakup level tinggi dan detail untuk menggambarkan hubungan antar komponen [8].



Gambar 5. Data Flow Diagram

Flowchart

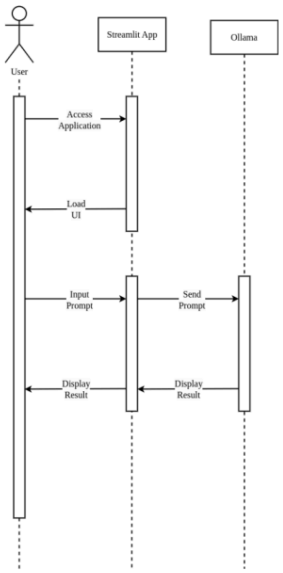
Flowchart menggambarkan alur logika proses dalam sistem, mulai dari input pengguna hingga keluaran yang dihasilkan oleh aplikasi. Flowchart ini memberikan pemahaman langkah demi langkah tentang bagaimana sistem bekerja [9].



Gambar 6. Flowchart

Sequence Diagram

Sequence diagram menunjukkan urutan komunikasi antara pengguna, antarmuka Streamlit, backend aplikasi, dan model Ollama. Diagram ini menjelaskan interaksi waktu nyata antara komponen dalam memproses sebuah prompt [10].



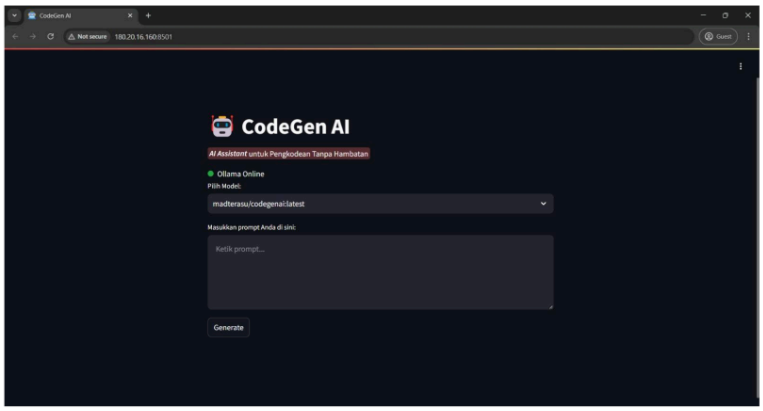
Gambar 7. Sequence Diagram

C. Implementasi

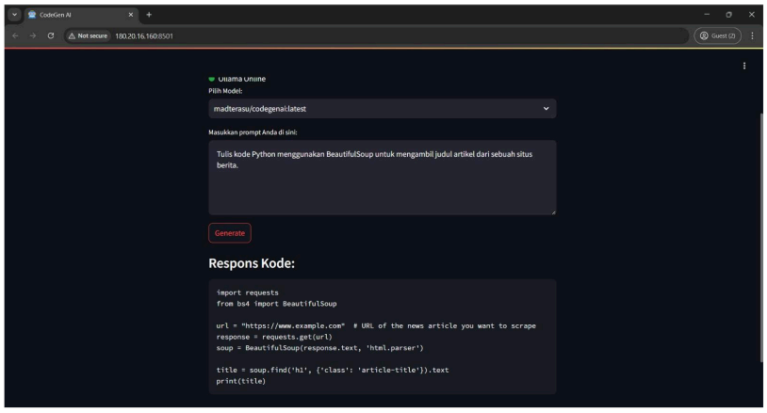
Pada tahap implementasi, sistem dikembangkan sesuai dengan desain yang telah dirancang. Model LLM Ollama diintegrasikan ke dalam aplikasi, sementara interface pengguna dibuat menggunakan Streamlit. Proses ini mencakup pengkodean, integrasi model, dan implementasi fitur yang sesuai dengan kebutuhan pengguna.

Halaman Utama

Pada halaman utama yang ditunjukkan pada Gambar 5, pengguna dapat memilih model LLM yang tersedia melalui dropdown, memasukkan prompt atau perintah dalam kotak input, dan menekan tombol Generate untuk mendapatkan respons dari model, ditunjukkan pada Gambar 6. Indikator status Ollama Online memastikan bahwa sistem dalam kondisi aktif dan siap digunakan. Dengan desain yang minimalis dan fokus pada fungsionalitas, halaman ini dirancang untuk memberikan pengalaman penggunaan yang sederhana dan efisien dalam menghasilkan kode secara otomatis.



Gambar 5. Halaman Utama



Gambar 6. Halaman Utama dengan Response

D. Pengujian

Tahap pengujian dilakukan untuk memastikan bahwa aplikasi berjalan sesuai dengan spesifikasi. Metode pengujian yang digunakan adalah Black box testing. Metode ini berupa pengujian sistem yang dilakukan dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian black box ini menitik beratkan pada spesifikasi fungsional dari perangkat lunak saja, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengetesan pada spesifikasi fungsional program [11]. Hasil dari pengujian yang dilakukan ditunjukkan pada Tabel 1.

Tabel 1. Hasil pengujian blackbox

No	Jenis Pengujian	Tujuan Pengujian	Deskripsi	Kriteria Keberhasilan	Hasil Pengujian
1.	Pengujian Input dan Output	Memastikan sistem menerima input dengan benar dan menghasilkan output yang sesuai.	Menguji apakah sistem dapat menerima input teks dari pengguna dan memberikan respons yang relevan dari model Ollama.	Respons sesuai dengan input yang diberikan, output relevan dengan konteks.	Berhasil
2.	Pengujian Alur Sistem	Pengujian Alur Sistem	Menguji alur dari pengguna memasukkan prompt, hingga aplikasi menampilkan respons.	Semua alur sistem berjalan lancar tanpa error.	Berhasil
3.	Pengujian Antarmuka Pengguna (UI)	Memastikan tampilan antarmuka sesuai dengan yang diharapkan dan mudah digunakan.	Menguji apakah antarmuka aplikasi Streamlit mudah dipahami dan user-friendly.	Antarmuka mudah digunakan dan sesuai dngan standart desain yang ditentukan.	Berhasil
4.	Pengujian Kinerja Respons	Memastikan aplikasi memberikan respons dalam waktu yang wajar.	Menguji waktu respons aplikasi dari saat input diberikan	Waktu respons tidak melebihi 5 detik dalam 90% kasus.	Gagal
5.	Pengujian Keamanan	Menguji apakah aplikasi aman terhadap potensi ancaman atau kesalahan input.	Menguji input yang salah atau berbahaya (misalnya input yang sangat panjang atau karakter khusus) untuk memastikan aplikasi tidak crash.	Aplikasi tetap aman dan tidak mengalami crash saat diberi input yang salah atau berbahaya.	Berhasil

E. Pemeliharaan

Setelah aplikasi diuji dan diimplementasikan, dilakukan tahap pemeliharaan untuk memperbaiki bug, meningkatkan fitur, atau menyesuaikan kebutuhan baru dari pengguna. Pemeliharaan juga mencakup pembaruan model LLM untuk meningkatkan akurasi respons.

IV. KESIMPULAN

Penelitian ini telah berhasil mengembangkan aplikasi chatbot berbasis AI Assistant menggunakan model Large Language Models (LLM) dari Ollama dengan antarmuka berbasis Streamlit. Aplikasi ini dirancang untuk membantu pengembang perangkat lunak dalam menyelesaikan permasalahan pemrograman, seperti debugging dan rekomendasi kode, melalui interaksi berbasis teks yang intuitif dan efisien.

Berdasarkan hasil pengujian, chatbot mampu memberikan respons yang akurat dan relevan terhadap pertanyaan pengguna, meskipun terdapat kendala pada waktu respons yang masih perlu dioptimalkan. Metode Waterfall yang digunakan dalam pengembangan memastikan bahwa setiap tahap, mulai dari analisis kebutuhan hingga pemeliharaan, dilakukan secara sistematis. Selain itu, penerapan model AI secara lokal dengan Ollama memungkinkan sistem berjalan tanpa ketergantungan pada server eksternal, sehingga meningkatkan efisiensi dan aksesibilitas bagi pengguna.

Dengan adanya aplikasi ini, diharapkan pengembang perangkat lunak dapat lebih mudah dalam mencari solusi terhadap permasalahan pemrograman secara cepat dan akurat. Untuk pengembangan lebih lanjut, disarankan melakukan optimasi waktu respons, penambahan fitur lanjutan, serta integrasi dengan platform lain guna meningkatkan fungsionalitas dan pengalaman pengguna.

Artikel Tugas Akhir Achmad Maulana Rochman.pdf

ORIGINALITY REPORT

17%	17%	10%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.radenintan.ac.id Internet Source	3%
2	e-theses.iaincurup.ac.id Internet Source	2%
3	eprints.walisongo.ac.id Internet Source	2%
4	repository.uin-suska.ac.id Internet Source	1%
5	repository.uinsaizu.ac.id Internet Source	1%
6	jht.politala.ac.id Internet Source	1%
7	jurnal.unma.ac.id Internet Source	1%
8	www.jurnal.una.ac.id Internet Source	1%
9	journal.institutpendidikan.ac.id Internet Source	1%
10	eprints.upj.ac.id Internet Source	1%
11	Submitted to UIN Raden Intan Lampung Student Paper	1%
12	jrsi.sie.telkomuniversity.ac.id Internet Source	1%

13	Kery Utami, Desta Sandya Prasvita, Yuni Widiastiwi. "Pengembangan Sistem Manajemen Bank Sampah berbasis Web untuk mewujudkan keberhasilan Ekonomi Sirkular di Masyarakat", The Indonesian Journal of Computer Science, 2023 Publication	<1 %
14	Puput Meilawati, Jaeni Jaeni. "Analisis Faktor – Faktor Yang Mempengaruhi Pengelolaan Dana Desa (Studi Kasus Pada Desa Di Kecamatan Sayung)", Journal of Economic, Bussines and Accounting (COSTING), 2023 Publication	<1 %
15	repository.upstegal.ac.id Internet Source	<1 %
16	e-journals.unmul.ac.id Internet Source	<1 %
17	ejurnal.unisri.ac.id Internet Source	<1 %
18	pt.scribd.com Internet Source	<1 %
19	digilib.its.ac.id Internet Source	<1 %
20	ejournal-binainsani.ac.id Internet Source	<1 %
21	journal.student.uny.ac.id Internet Source	<1 %
22	Arba Adhy Pamungkas, Cecep Nurul Alam, Aldy Rialdy Atmadja, Roby Juliansyah. "Integrasi Kamus Multibahasa pada Feed Forward Neural Network dan IndoBERT dalam Pengembangan Chatbot Mobile", Edumatic: Jurnal Pendidikan Informatika, 2024	<1 %

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		