



Penerapan Langchain Retriever Dengan Model Chat Openai Dalam Pengembangan Sistem Chatbot Hadis Berbasis Telegram

Niken Aisyah Maharani Herwanza ¹, Nazruddin Safaat Harahap ^{2*}, Febi Yanto ³, dan Fitri Insani ⁴

¹ Universitas Islam Negeri Sultan Syarif Kasim Riau; 12050123966@students.uin-suska.ac.id

² Universitas Islam Negeri Sultan Syarif Kasim Riau; nazruddin.safaat@uin-suska.ac.id

³ Universitas Islam Negeri Sultan Syarif Kasim Riau; febiyanto@uin-suska.ac.id

⁴ Universitas Islam Negeri Sultan Syarif Kasim Riau; fitri.insani@uin-suska.ac.id

* Korespondensi: nazruddin.safaat@uin-suska.ac.id

Sitasi: Herwanza, N. A. M.; Harahap, N. S.; Yanto, F; dan Insani, F. (2024). Penerapan Langchain Retriever Dengan Model Chat Openai Dalam Pengembangan Sistem Chatbot Hadis Berbasis Tele-gram. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 6(1), 70-83. <https://doi.org/10.35746/jtim.v6i1.514>

Diterima: 29-04-2024

Direvisi: 22-05-2024

Disetujui: 23-05-2024



Copyright: © 2024 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: In Islamic studies, the Hadiths of Prophet Muhammad (SAW) hold significant value as guides for behavior and faith. However, access to understanding Hadiths often presents challenges, especially for those who are not Hadith experts. The digitalization of Hadiths is still limited, making it time-consuming to find answers by sifting through the vast amount of available information. This research aims to create an efficient chatbot that provides answers related to Hadiths, including the original sources, quickly. The proposed solution is a technology-based approach through the development of a Hadith chatbot on Telegram, integrated with the LangChain Retriever and the GPT-4-1106-preview chat model from OpenAI. Using LangChain Retriever helps the chatbot find accurate answers by matching user questions with relevant Hadith databases, enhancing the accuracy of the chatbot's responses. The GPT-4-1106-preview chat model enables the chatbot to generate natural and context-appropriate responses, improving user interaction. The Rapid Application Development (RAD) method is applied in system development, through stages of Requirement Planning, User Design, Construction, and Cut-Over, including data analysis of Hadiths from the Nine Imam Hadith Books, totaling 62,169 Hadiths. The chatbot's performance evaluation uses the Scoring Evaluator framework with an average evaluation score of 0.97 and quality answer evaluation testing by five Hadith experts with an accuracy percentage of 90%. The Scoring Evaluator test results indicate that the responses are highly accurate and aligned with Hadith references, and the quality answer evaluation test on a Likert scale shows respondents strongly agree with the system's answers. This research contributes to laypersons wanting to learn Hadiths by utilizing the chatbot as an interactive and innovative learning medium. Further research can expand the focus to complex interpretations of Musykil al-Hadith and asbab al-wurud to address deeper questions about Hadith interpretation.

Keywords: Chatbot; Hadiths; Langchain; OpenAi; Retriever

Abstrak: Dalam kajian keislaman, hadis Nabi Muhammad SAW memiliki nilai penting sebagai panduan perilaku dan keimanan. Namun, akses terhadap pemahaman hadis sering menjadi tantangan, terutama bagi yang bukan ahli hadis. Digitalisasi hadis masih terbatas, sehingga mendapatkan jawaban atas pertanyaan seputar hadis memerlukan waktu lama dalam memilah banyaknya informasi yang tersedia. Penelitian ini bertujuan untuk menciptakan chatbot yang efisien dalam memberikan jawaban terkait hadis dengan menyertakan sumber asli hadis secara cepat. Penelitian ini mengusulkan solusi berbasis teknologi melalui pengembangan sistem chatbot hadis di Telegram, yang diintegrasikan dengan LangChain Retriever dan model GPT-4-1106-preview chat OpenAI. Dengan menggunakan LangChain Retriever membantu chatbot menemukan jawaban tepat dengan

mencocokkan pertanyaan pengguna dengan *database* hadis yang relevan untuk meningkatkan keakuratan respons *chatbot*. Model *GPT-4-1106-preview chat OpenAI* memungkinkan *chatbot* menghasilkan respons yang natural dan sesuai konteks sehingga meningkatkan interaksi dengan pengguna. Metode *Rapid Application Development* (RAD) diterapkan dalam pengembangan sistem, melalui tahapan *Requirement Planning, User Design, Construction, dan Cut-Over*, mencakup analisis data hadis dari Kitab 9 Imam Hadis yang totalnya mencapai 62.169 hadis. Evaluasi kinerja *chatbot* ini menggunakan pengujian *framework Scoring Evaluator* dengan skor rata-rata evaluasi 0,97 dan pengujian evaluasi kualitas jawaban oleh 5 orang ahli hadis dengan persentase akurasi sebesar 90 %. Hasil evaluasi pada pengujian *Scoring Evaluator* menunjukkan rata-rata jawabannya benar-benar akurat dan sejalan dengan referensi hadis, pada pengujian evaluasi kualitas jawaban menurut interval penilaian pada *skala likert* menandakan responden sangat setuju terhadap jawaban sistem. Penelitian ini memberikan kontribusi terhadap orang awam yang ingin belajar hadis dengan memanfaatkan *chatbot* sebagai media pembelajaran yang interaktif dan inovatif. Pada penelitian lebih lanjut, fokus dapat diperluas untuk pemaknaan yang kompleks dari *Musykil al-Hadits* dan *asabul wurud* untuk menjangkau pertanyaan-pertanyaan yang lebih dalam terhadap pemaknaan hadis.

Kata kunci: Chatbot; Hadis; Langchain; OpenAi; Retriever

1. Pendahuluan

Dalam Islam, hadis-hadis Nabi Muhammad SAW memiliki peran yang sangat penting dalam membimbing perilaku dan keyakinan beragama. Hadis merupakan sumber hukum kedua dalam agama Islam setelah Al-Qur'an sebagai penjelas bagi firman Allah SWT yang masih bersifat umum [1]. Kitab 9 Imam Hadis sering digunakan sebagai sumber rujukan utama dalam kegiatan dakwah maupun pembelajaran [2] yang mencakup beragam hadis yang ada. Hal ini menciptakan kerumitan dalam mencari jawaban tentang hadis, terutama yang bukan ahli hadis secara cepat dan tepat.

Digitalisasi tentang hadis masih terbilang sedikit [3], banyak orang menggunakan situs website keislaman sebagai sumber referensi untuk mencari solusi terhadap berbagai masalah yang dihadapi [4] dan sangat tergantung pada *internet* atau media *online* [5]. Banyak pengguna yang mengandalkan mesin pencari untuk mendapatkan informasi, meskipun harus berurusan dengan banyaknya data tidak relevan [6]. Akibatnya, dalam mendapatkan jawaban atas pertanyaan mengenai hadis memerlukan penyaringan dan pencarian satu per satu yang memakan waktu lama. Selain itu, dalam era kecerdasan buatan (*Artificial Intelligence/AI*), kebutuhan akan solusi sistem tanya jawab yang lebih efisien dan cepat dalam mencari informasi semakin penting.

Perkembangan *chatbot* telah meningkat pesat dalam beberapa tahun terakhir dan telah menjadi populer karena kemampuannya untuk meningkatkan pengalaman pengguna [7]. *Chatbot* adalah agen percakapan otomatis yang dapat berinteraksi dengan pengguna melalui *platform* pesan instan [8]. Dalam mengatasi tantangan tersebut, pengembangan *chatbot* berbasis kecerdasan buatan menjadi langkah yang inovatif. *Chatbot* dapat digunakan sebagai antarmuka yang responsif untuk memberikan informasi hadis secara efisien [9], menyediakan jawaban yang relevan terkait hadis secara cepat. Dengan memanfaatkan teknologi AI, *chatbot* dapat memperbaiki efisiensi dalam menjawab pertanyaan dengan cepat, mengatasi keterbatasan pencarian yang memakan waktu. Selain itu, *chatbot* juga dapat dirancang untuk menyertakan sumber asli, memastikan bahwa informasi yang disediakan dapat diverifikasi untuk kebenarannya terhadap informasi yang diberikan.

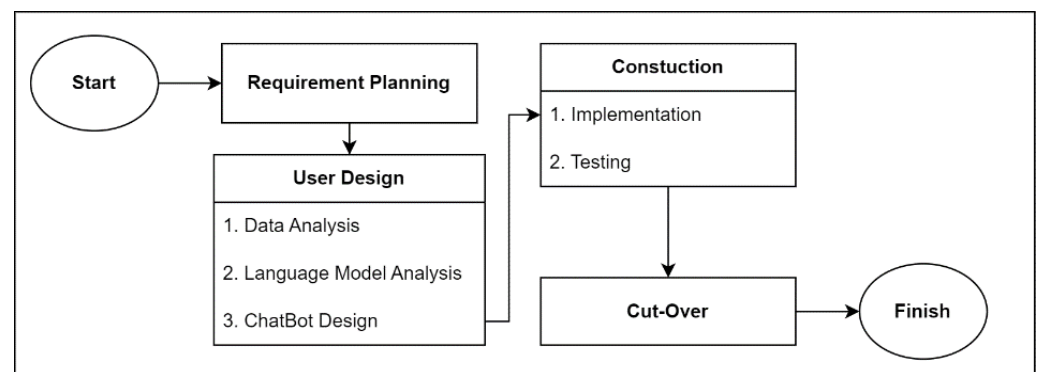
Penelitian mengenai *chatbot* dilakukan oleh Avisyah dkk [10], membahas pengembangan telegram bot menggunakan *OpenAI Model GPT-3* yang dapat menjawab dengan relevan secara cepat dan efisien. Penelitian lain yang membahas pembuatan *chatbot* dilakukan oleh Keivalya Pandya [11], pada penelitiannya mengembangkan *chatbot customer*

service menggunakan LLM khusus yakni *LangChain* untuk peningkatan retensi pelanggan, ekstraksi nilai pelanggan, dan citra merek. Penelitian lain oleh Arjun Pesaru dkk [12], penelitian ini membahas pengembangan *chatbot* PDF dengan *LangChain* dan LLMs Model untuk meningkatkan akurasi, kelancaran, dan relevansi respons. Pada penelitian yang dilakukan oleh Topsakal & Akinci [13], membahas pemanfaatan *Large Language Models* (LLMs) untuk memberikan jawaban terhadap pertanyaan yang terkait dokumen dengan *Retriever* untuk ekstraksi data dari berbagai jenis dokumen.

Dengan mengintegrasikan komponen dalam *LangChain* yakni *Retriever* [13] dan model *GPT-4-1106-preview chat OpenAI* dalam pengembangan sistem *chatbot* hadis berbasis Telegram, penelitian ini bertujuan untuk membuat *chatbot* yang dapat memberikan jawaban atas pertanyaan seputar hadis secara cepat dan efisien dengan menyertakan sumber asli hadis. Perbedaan signifikan dengan penelitian sebelumnya terletak pada penggabungan *LangChain Retriever* dan model *chat OpenAI*, yang menciptakan pendekatan komprehensif untuk meningkatkan responsivitas dan kecerdasan dalam menjawab pertanyaan berbasis bahasa alami serta menampilkan hadis yang relevan. Dengan demikian, diharapkan bahwa *chatbot* ini dapat menjadi alat yang berguna dalam memberikan bimbingan agama dan menjawab pertanyaan umat Islam dengan mudah.

2. Bahan dan Metode

Penelitian ini menggunakan metode RAD (*Rapid Application Development*) yakni pendekatan pengembangan perangkat lunak yang menekankan pembangunan sistem dalam periode waktu yang singkat [14]. Terdapat empat langkah utama dalam RAD (*Rapid Application Development*) berisi *Requirement Planning*, *User Design*, *Construction*, dan *Cut-Over* [15]. Metodologi Penelitian ini berisi penjelasan mengenai langkah-langkah yang akan dilaksanakan pada Gambar 1., berikut metodologi penelitian pada penelitian ini :



Gambar 1. Metodologi Penelitian

2.1 Requirement Planning

Mengidentifikasi masalah terkait pengembangan *Chatbot* yang dapat memberikan jawaban tentang hadis Islam untuk Pertanyaan dan Jawaban (*Question Answering*) secara interaktif serta menampilkan hadis yang relevan, terkait dengan *chatbot* telegram, penerapan *OpenAI Model*, penerapan *LangChain* dan objek data penelitian yakni hadis.

2.2 User Design

Proses *User Design* terkait dengan analisis dan desain sistem secara logis dan dikembangkan berdasarkan kebutuhan pengguna. Tahap ini memastikan bahwa sistem yang dikembangkan sesuai dengan alur dalam menyediakan informasi tentang hadis.

2.2.1 Data Analysis

Pengumpulan data hadis diambil dari sumber github yaitu <https://github.com/irsyadulibad/hadits-database>. Pada proses pengumpulan, dilakukan

pemilahan data seperti nomor hadis, nama kitab, bahasa Arab, dan terjemahan. Data hadis diekstrak dan disimpan dalam format txt untuk melakukan embedding menggunakan *OpenAI Embeddings* ke dalam *vectorstore chroma*. *OpenAI Embeddings* menjadi elemen penting dalam mengubah data hadis menjadi representasi vektor atau proses *embedding* [16], yang nantinya akan dikelola dalam *vectorstore chroma*. *Vectorstore* ini memberikan wadah yang efisien untuk penyimpanan [17] dan mengakses data hadis yang telah diolah.

OpenAI Embedding adalah representasi vektor padat yang dihasilkan oleh model-model OpenAI yang digunakan dalam arsitektur *bi-encoder* untuk melakukan pencarian vector [18]. *OpenAI Embedding* merujuk pada representasi vektor dari teks memungkinkan pengguna untuk membuat representasi teks berdasarkan model bahasa yang telah dilatih sebelumnya [19].

Chroma adalah *database* vektor yang sesuai dengan Indeks LLama menggunakan kerangka kerja *Large Language Model* (LLM) dalam menyediakan penyimpanan tanpa batas untuk penyematan yang dihasilkan oleh LLM [20]. *Chroma* menjadikan dokumen sebagai vektor dalam ruang multidimensi guna menangkap makna semantik dan hubungan antar dokumen [17].

2.2.2 Language Model Analysis

Model bahasa pada penelitian ini menggunakan *Large Language Model* (LLM) dalam menghasilkan teks baru serta memprediksi kata-kata yang dihasilkan secara otomatis. *Large Language Model* (LLM) yang digunakan berupa *Model GPT-4-1106-preview* dari *OpenAI* dalam penelitian ini.

Large Language Models (LLM) adalah model bahasa yang dilatih pada kumpulan data dengan jumlah data yang besar untuk memproses dan memahami bahasa alami dengan pemahaman bahasa dan pengetahuan yang luas [21]. LLM memperoleh kemampuan memprediksi kata berikutnya dalam kalimat berdasarkan konteks kata sebelumnya, membantu model memahami tata bahasa, sintaksis, dan semantik Bahasa [22]. LLM menjalani pelatihan menggunakan dataset besar dan menggunakan algoritma yang kompleks untuk memahami serta menghasilkan bahasa alami yang digunakan [23].

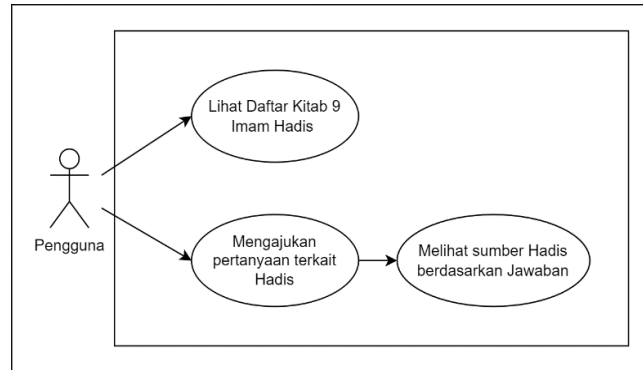
OpenAI ChatModel merupakan model yang dikembangkan oleh *OpenAI* untuk menghasilkan tanggapan dalam percakapan yang menyerupai respons manusia dengan pemahaman bahasa alami, dilatih melalui data teks berlimpah dari *internet* [24]. Model ini dapat memberikan respons yang mirip dengan tanggapan manusia terhadap perintah atau pertanyaan, membuatnya cocok untuk berbagai aplikasi percakapan [25]. *OpenAI ChatModel* atau *ChatGPT* merupakan bagian dari seri GPT (*Generative Pre-trained Transformer*), khususnya GPT-3.5. *Generative Pre-trained Transformer* (GPT) adalah model bahasa yang dikembangkan oleh *OpenAI* dan didasarkan pada arsitektur *Transformer* [26].

2.2.3 Chatbot Design

Perencanaan *chatbot* hadis mengintegrasikan *LangChain* dengan model *chat OpenAI* dalam upaya meningkatkan kemampuan responsifitas kinerja *chatbot* untuk memberikan jawaban yang relevan terkait dengan pertanyaan mengenai hadis dengan menampilkan sumber hadis yang relevan menggunakan *Retriever*. *Retriever* merupakan salah satu komponen dalam *LangChain* yang membantu dalam mencari dan mengambil data berdasarkan kesamaan *embedding* [13]. *Retriever* bekerja dengan menggunakan teknik pemampatan konteks berbasis pengambilan informasi [27]. Dengan adanya *Retriever*, query pengguna akan mencari hasil berdasarkan kemiripan *embedding* dan menghasilkan informasi yang relevan. *Langchain* adalah sebuah kerangka kerja yang memungkinkan pengguna untuk berinteraksi dengan dokumen dan membuat pencarian serta pengambilan informasi dari dokumen menjadi sangat mudah [28].

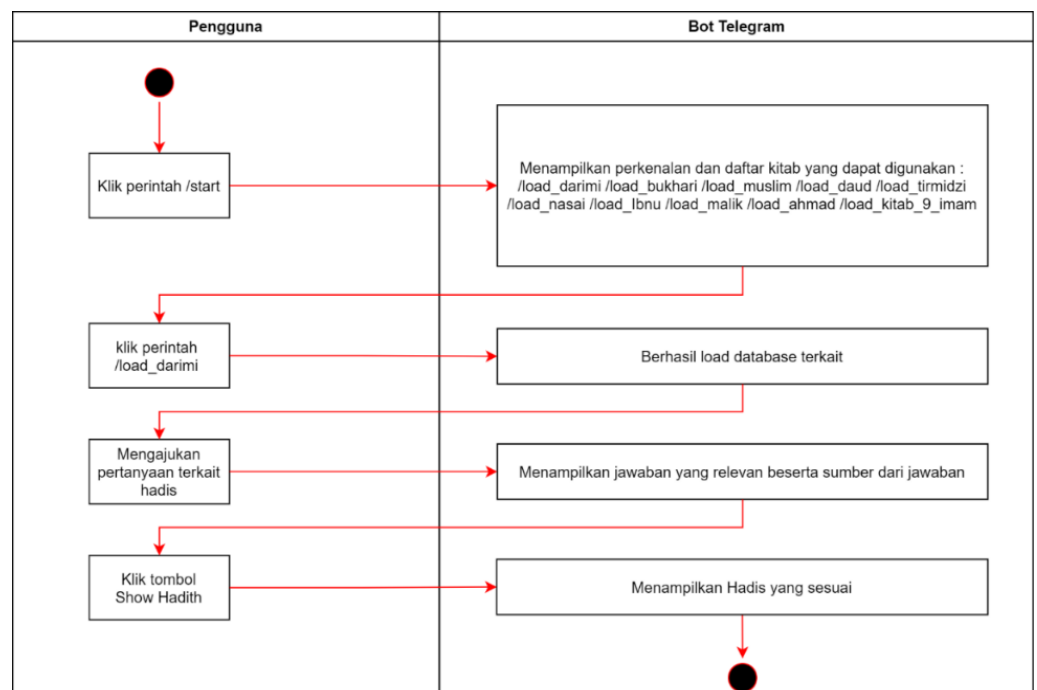
Perancangan bot melalui *Use Case Diagram* dan *Activity Diagram*, memberikan gambaran tentang fungsionalitas serta alur kerja yang diintegrasikan dalam pengembangan

bot berbasis data hadis Kitab 9 Imam. Pada *Use Case Diagram* yang tergambar dalam Gambar 2., terdapat satu aktor utama, yaitu pengguna. Pengguna dapat melakukan tiga fungsi, yaitu melihat daftar Kitab 9 Imam Hadis, mengajukan pertanyaan terkait hadis, dan melihat sumber hadis berdasarkan jawaban yang diberikan.



Gambar 2. Use Case Diagram Chatbot Hadis

Pada *Activity Diagram* tanya jawab hadis yang terdapat dalam Gambar 3., pengguna dapat memulai percakapan dengan bot Telegram melalui perintah /start. Setelahnya, bot akan merespons dengan pesan perkenalan dan daftar kitab yang dapat digunakan, termasuk perintah-perintah seperti /load_darimi, /load_bukhari, /load_muslim, /load_daud, /load_tirmidzi, /load_nasai, /load_ibnu, /load_malik, /load_ahmad, dan /load_kitab_9_imam. Pengguna kemudian dapat mengklik salah satu perintah daftar kitab untuk memuat database terkait, dan bot akan memberikan konfirmasi pesan berhasil load database terkait. Selanjutnya, pengguna dapat mengajukan pertanyaan terkait hadis, dan bot akan menampilkan jawaban relevan beserta sumbernya. Pengguna juga memiliki opsi untuk menampilkan hadis yang sesuai dengan jawaban melalui tombol "Show Hadis". Bot akan merespons dengan menampilkan hadis yang sesuai dengan permintaan pengguna.



Gambar 3. Activity Diagram Chatbot Hadis

2.3 Construction

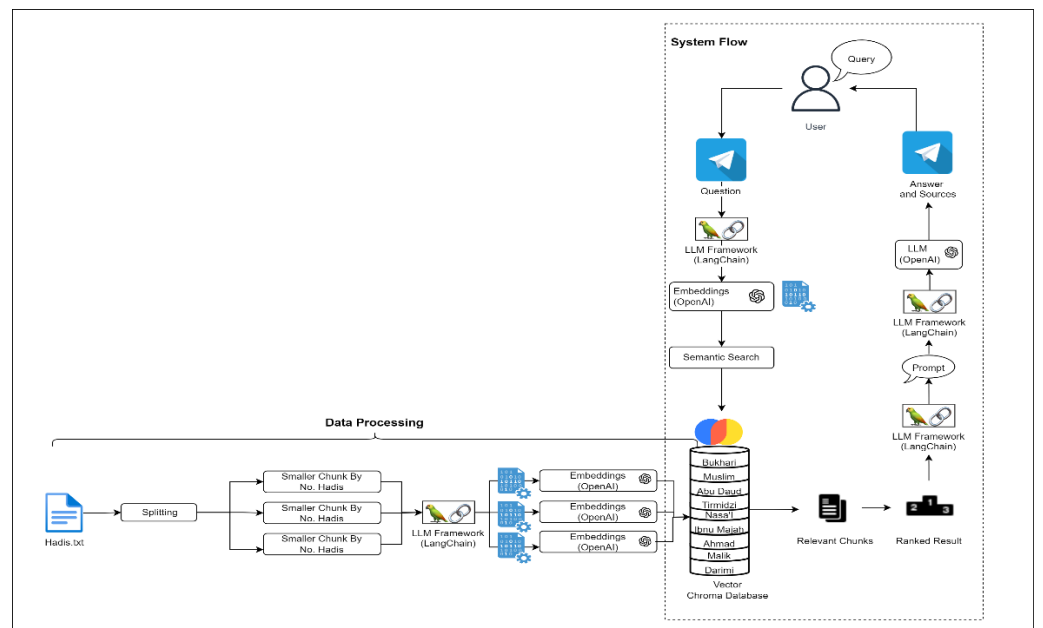
Tahap ini melibatkan proses implementasi dan pengujian sistem *chatbot* yang menerapkan desain sistem yang telah dikembangkan sebelumnya. Dalam fase implementasi, *Python* menjadi bahasa pemrograman utama yang akan diterapkan dalam *platform telegram*. Pengujian menggunakan *Scoring Evaluator* dari *langchain evaluation* untuk penilaian yang lebih terinci. *Scoring Evaluator* memungkinkan model bahasa untuk menilai prediksi model pada skala tertentu berdasarkan kriteria khusus yang kita tentukan. Pengujian kedua, evaluasi kualitas jawaban yang melibatkan ahli hadis dalam menilai kebenaran respons terhadap ilmu hadis, berikut nama responden ahli hadisnya yaitu Bapak Yani Hamdani S.SI.Lc, Bapak Agus Firdaus Chandra Lc., M.A, Bapak Ahmad Mas'ari, M.A, Ibuk Laila Sari Masyhur, M.A dan Bapak Arif Marsal Lc., M.A. Pengukuran respons menggunakan skala *likert* dengan skala 1-5 [29].

2.4 Cut-Over

Tahap *Cut-Over* pada proyek ini mencakup peralihan sistem dari *local server*, yang semula beroperasi pada komputer pribadi, ke *public server* agar dapat diakses oleh orang banyak.

3. Hasil

Pada bagian ini membahas analisis data yang diperoleh dalam penelitian ini dan hasil penelitian, dengan fokus pada *data processing* (pengolahan data) dan *system flow* (alur sistem) yang tergambar dalam Gambar 4.



Gambar 4. Kerangka Kerja Sistem

3.1 Data Processing (Pengolahan Data)

Data penelitian yang digunakan merupakan data hadis dalam bentuk txt yang mencakup Kitab 9 Imam Hadis atau kutubut tis'ah yakni Shahih Bukhari, Shahih Muslim, Sunan Abu Daud, Sunan Tirmidzi, Sunan Nasa'I, Sunan Ibnu Majah, Musnad Ahmad, Muwatha' Malik, dan Sunan Darimi.

Tabel 1. Data Kitab 9 Imam Hadis.

No	Imam Hadis	Kitab	Jumlah Hadis
1.	Shahih Bukhari	77	7008
2.	Shahih Muslim	56	5362
3.	Sunan Abu Daud	35	4590
4.	Sunan Tirmidzi	49	3891
5.	Sunan Nasa'i	51	5662
6.	Sunan Ibnu Majah	32	4332
7.	Musnad Ahmad	14	26363
8.	Muwatha' Imam Malik	32	1594
9.	Sunan Darimi	24	3367
Total			62169

Tabel 1. menunjukkan rincian data Kitab 9 Imam Hadis berupa jumlah kitab dan jumlah hadis dari masing-masing Imam Hadis. Total 62,169 hadis yang terhimpun. Proses pengumpulan data dalam satu hadis tertentu mencakup nomor hadis yang menandakan urutannya, nama kitab yang merujuk pada sumber hadis tersebut, bahasa Arab yang digunakan dalam teks aslinya, serta terjemahan hadis tersebut ke dalam bahasa Indonesia guna mempermudah pemahaman. Selanjutnya, dilakukan ekstraksi konten dari file txt agar dapat dibaca oleh sistem. Dalam konteks pengembangan perangkat lunak dengan menggunakan Python, langkah awal dalam ekstraksi teks seringkali mencakup membaca file teks. Pada saat membuka file tersebut, penting untuk menentukan encoding yang sesuai, seperti UTF-8, untuk memastikan bahwa karakter-karakter khusus atau non-ASCII seperti teks bahasa Arab dapat diinterpretasikan dengan benar oleh program. Proses ini memungkinkan informasi yang terkandung dalam teks, seperti teks hadis untuk diambil dan disiapkan dalam bentuk yang dapat dimanfaatkan oleh algoritma atau analisis data lebih lanjut.

3.1.1 Splitting

Proses selanjutnya merupakan pemisahan atau "splitting" hasil ekstraksi teks menjadi segmen-segmen (*Chunk*) kecil berdasarkan nomor hadis. Langkah ini penting untuk mengorganisir informasi secara terstruktur dan memungkinkan akses yang lebih efisien ke setiap hadis. Dalam pemisahan berdasarkan nomor hadis, setiap segmen atau bagian teks akan diidentifikasi sesuai dengan nomor hadis yang terkandung di dalamnya. Pemisahan berdasarkan nomor hadis menciptakan struktur yang jelas, memudahkan pengguna atau program untuk merujuk dan mengelola setiap hadis secara individual. Library *re* dengan fungsi *re.split* digunakan untuk mencapai pemisahan ini dengan memanfaatkan pola ekspresi reguler yang sesuai. Dalam konteks ini, penggunaan *metadata* menjadi relevan. *Metadata* digunakan untuk memberikan informasi tambahan terkait setiap segmen teks, seperti nomor hadis dan sumber atau koleksi hadis asal. Pada contoh Gambar 3., setiap segmen teks diidentifikasi dengan nomor hadis yang sesuai, dan *metadatas* mencatat sumber asal setiap segmen, seperti "1-Malik", "2-Malik", dan seterusnya.

3.1.2 Embedding

Embedding Penelitian ini menggunakan *Embedding OpenAI* dengan model *text-embedding-3-large*. Sebelum teks dapat diubah menjadi representasi vektor atau *embedding* oleh model ini, dilakukan pemecahan setiap kata atau frasa dalam teks menjadi token-token yang sesuai. Model *text-embedding-3-large* melaksanakan proses tokenisasi secara internal, yang mencakup tahapan *encoding* dan *decoding*. Dalam tahap *encoding*, teks diubah menjadi serangkaian token *digital* yang dimengerti oleh model. Kemudian, proses tokenisasi mengurai kata-kata menjadi token kata. Pada tahap *decoding*, model memproses token-token tersebut kembali menjadi teks, sehingga memungkinkan pemahaman kontekstual dan semantik dari setiap kata atau frasa.

Setelah melalui proses tokenisasi beserta tahapannya, token-token yang dihasilkan menjadi input bagi model *text-embedding-3-large* dari *OpenAI*. Model ini kemudian menghasilkan *embedding* atau representasi vektor untuk setiap token, yang menggambarkan makna dan konteks dari kata atau frasa tersebut dengan akurat. *Langchain* digunakan untuk mengimplementasikan model *Embedding OpenAI Framework* ini menyediakan komponen yang dibutuhkan untuk mengintegrasikan model *text-embedding-3-large* dengan sumber data, memastikan bahwa proses pembuatan *embedding* teks dilakukan secara efisien dan terintegrasi.

3.1.3 Vectorstore

LangChain berfungsi sebagai penghubung model *text-embedding-3-large* dari *OpenAI* dengan sumber data (*database*). Setelah proses *embedding* oleh model *text-embedding-3-large* selesai dilakukan, vektor yang dihasilkan kemudian disimpan kembali dalam *database Chroma*. *Database* ini diorganisir untuk memfasilitasi penyimpanan dan pengambilan data *embedding*.

3.2 System Flow (Alur Sistem)

Alur sistem merupakan kelanjutan dari pengolahan data, dengan fokus pada pengolahan pertanyaan pengguna hingga menghasilkan jawaban yang dikeluarkan oleh *chatbot*. Ini melibatkan langkah-langkah terinci untuk memastikan respons yang akurat dan informatif dalam interaksi antara pengguna dan sistem *chatbot*.

3.2.1 Embedding Query

Dalam tahapan ini, pengguna menyampaikan pertanyaannya, yang kemudian ditambahkan ke dalam proses sistem. Proses ini mencakup tokenisasi didalam model *text-embedding-3-large*, di mana teks pertanyaan pengguna dipecah menjadi unit-unit lebih kecil, seperti kata-kata atau frasa. Selanjutnya, menggunakan model *text-embedding-3-large* dari *OpenAI*, teks yang telah di-tokenisasi diubah menjadi representasi numerik atau vektor (*embedding*). Model ini tidak hanya menyediakan *embedding*, tetapi juga mengintegrasikan proses tokenisasi dalam pembentukan vektor tersebut. Implementasi model *Embedding OpenAI* dengan sumber data digunakan *Langchain*. Representasi vektor atau *Embedding* yang dihasilkan dari model *text-embedding-3-large* memungkinkan sistem untuk melakukan pencarian semantik dan menemukan kesamaan makna antara pertanyaan pengguna dengan teks yang ada dalam ruang vektor.

Selain itu, untuk meningkatkan kinerja pencarian dan interaksi dengan pengguna, sistem ini menggunakan model *GPT-4-1106-preview* dalam proses penyempurnaan pertanyaan (*refined query*). Model ini memfasilitasi perbaikan pertanyaan dari pengguna, khususnya ketika terdapat riwayat percakapan. *Refined query* memungkinkan *chatbot* memberikan tanya jawab berkelanjutan sehingga membuat interaksi menjadi lebih hidup.

3.2.2 Semantic Search

Embedding OpenAI menghasilkan representasi vektor dari teks yang memungkinkan pencarian kesamaan vektor terhadap pertanyaan pengguna untuk menemukan teks yang serupa dalam ruang vektor. Metode ini menggunakan konsep jarak antara dua vektor untuk mengukur seberapa erat hubungan antara dua teks tersebut. Dalam konteks pencarian semantik di dalam *database chroma*, perhitungan kesamaan antara dua vektor dilakukan menggunakan *cosine similarity*. *Cosine similarity* mengukur kesamaan antara dua vektor berdasarkan sudut antara keduanya, bukan jarak *euclidean*. Oleh karena itu, dalam *database chroma*, ketika pengguna mengajukan pertanyaan, sistem terlebih dahulu mengubah *query* ke dalam representasi vektor menggunakan teknik *embedding*. Selanjutnya, sistem menggunakan algoritma *Approximate Nearest Neighbor Search (ANN)* yang ada di dalam *library Hierarchical Navigable Small World (HNSW)* untuk mencari dokumen yang paling mirip dengan *query* pengguna. Algoritma ANN ini bekerja dengan menghitung *cosine similarity* antara vektor *query* dan vektor dokumen yang ada di dalam *database*. Dengan

demikian, sistem dapat menentukan dokumen mana yang paling relevan dengan pertanyaan pengguna berdasarkan kedekatan semantiknya.

3.2.3 Ranked Result

Beberapa segmen hadis yang relevan dengan pertanyaan pengguna ditampilkan kemudian diurutkan berdasarkan skor kemiripan. *LangChain* mengarahkan *templat prompt* yang telah dibuat untuk memberikan instruksi kepada *GPT-4-1106-preview*, sebagai *Large Language Model* (LLM) yang ditugaskan untuk memproses pertanyaan yang diajukan pengguna. *GPT-4-1106-preview* akan mengolah prompt yang diberikan dan menghasilkan jawaban serta mencantumkan sumber informasi dari jawaban tersebut. LLM ini dipanggil dengan *LangChain* untuk melakukan tindakan berdasarkan *prompt* yang sudah disusun sebelumnya.

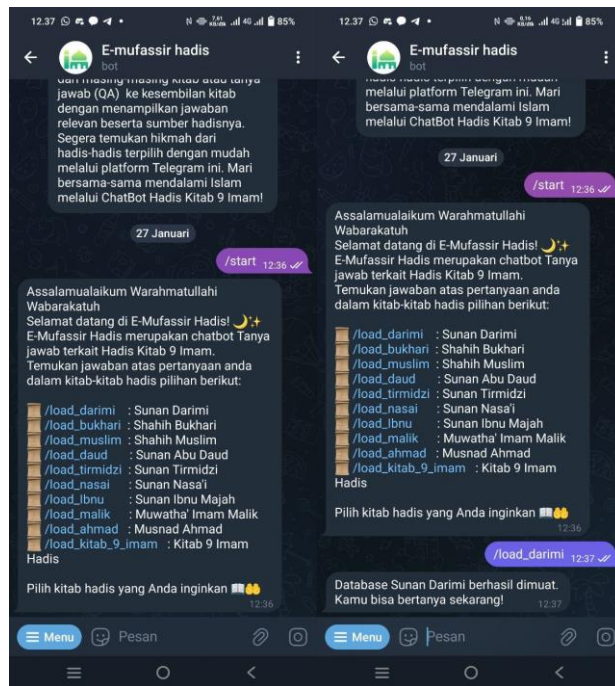
Dalam tahap hasil yang diurutkan berdasarkan rank (*Ranked Result*), *cosine similarity* digunakan untuk mengukur seberapa mirip segmen hadis tersebut dengan *query* yang diajukan pengguna. *Cosine similarity* menghitung kesamaan antara vektor pertanyaan pengguna dan vektor segmen (*chunk*) hadis, sehingga menghasilkan skor kemiripan untuk setiap segmen. Segmen-segmen tersebut kemudian diurutkan, skor yang lebih tinggi menunjukkan adanya kesamaan yang lebih besar antara segmen hadis dengan pertanyaan yang diajukan pengguna, sehingga segmen dengan skor kemiripan tertinggi akan ditampilkan lebih dulu. Berdasarkan skor kemiripan yang telah dihitung sebelumnya, LLM akan mengambil sejumlah dokumen teratas dari hasil peringkat. Dalam kasus ini, sistem telah ditetapkan untuk mengambil empat dokumen dengan skor kemiripan tertinggi, dokumen-dokumen ini kemudian akan diproses lebih lanjut oleh *GPT-4-1106-preview*. Model ini akan mengolah informasi dalam dokumen tersebut dan menghasilkan jawaban yang relevan dan informatif berdasarkan pertanyaan pengguna.

3.3 Implementasi Sistem

Pada bagian Implementasi akan membahas tampilan dan desain sistem yang telah diimplementasikan dalam *platform* Telegram. Pembahasan akan mencakup gambaran singkat tentang interaksi pengguna dengan *chatbot* hadis yang telah dikembangkan secara efektif.

3.3.1 Tampilan Memulai Percakapan dan Berhasil Load Data

Pengguna dapat memulai percakapan dengan chatbot melalui perintah `/start`. Bot akan merespons dengan pesan perkenalan dan daftar kitab yang dapat diakses, seperti `/load_darimi`, `/load_bukhari`, `/load_muslim`, `/load_daud`, `/load_tirmidzi`, `/load_nasai`, `/load_ibnu`, `/load_malik`, `/load_ahmad`, dan `/load_kitab_9_imam`. Setelah pengguna memilih salah satu dari daftar kitab, yakni `/load_darimi`, bot memberikan konfirmasi bahwa "Database Sunan Darimi berhasil dimuat, Kamu bisa bertanya sekarang!" memungkinkan pengguna untuk mulai mengajukan pertanyaan. Tampilan interaksi ini dapat dilihat pada Gambar 5.



Gambar 5. Tampilan memulai Percakapan dan Berhasil Load Data

366

367

3.3.2 Tampilan Ketika Bertanya dan Menampilkan Hadis yang Sesuai

368

Pada tampilan interaksi selanjutnya, pengguna dapat mengajukan pertanyaan terkait hadis. Bot akan menampilkan jawaban yang relevan beserta sumbernya. Pengguna juga dapat melihat hadis yang sesuai dengan mengklik tombol "Show Hadis". Selanjutnya, bot akan menampilkan hadis yang sesuai dengan pertanyaan. Pengguna dapat melakukan pertanyaan berkelanjutan, dan bot akan memberikan sumber yang sesuai dengan pertanyaan tersebut. Pengguna juga dapat melihat hadis yang sesuai dengan pertanyaan berkelanjutan dengan mengklik tombol "Show Hadis". Tampilan interaksi ini dapat ditemukan pada Gambar 6.

369

370

371

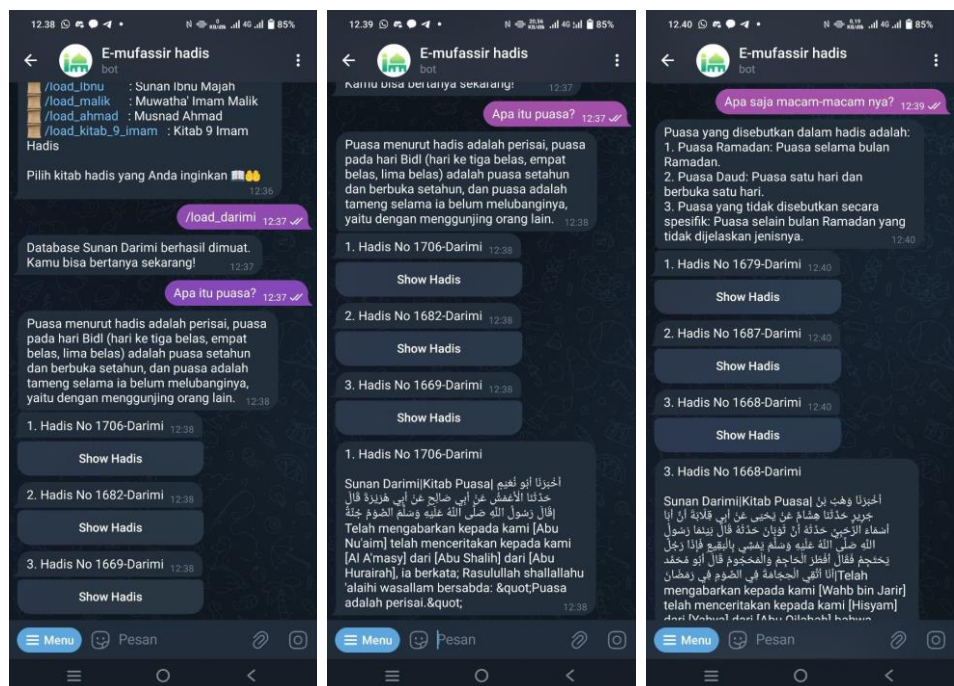
372

373

374

375

376



Gambar 6. Tampilan Ketika Bertanya dan Menampilkan Hadis yang Sesuai

377

378

4. Pembahasan

Scoring Evaluator dari langchain evaluation memungkinkan model bahasa untuk menilai kualitas prediksi model. Pada pengujian dengan Scoring Evaluator, skala yang ditetapkan antara 1 hingga 10, untuk menilai kinerja dan kebenaran jawaban sistem. Skala Scoring Evaluator [30] yang telah ditetapkan dipengujian ini sebagai berikut:

Score 1: The answer is completely unrelated to the reference.

Score 3: The answer has minor relevance but does not align with the reference.

Score 5: The answer has moderate relevance but contains inaccuracies.

Score 7: The answer aligns with the reference but has minor errors or omissions.

Score 10: The answer is completely accurate and aligns perfectly with the reference.

Dalam evaluasi chatbot ini, masing-masing jawaban telah dibuat berdasarkan 4 referensi hadis yang relevan, seperti contoh Gambar 7. mengambil 4 referensi hadis dari sebuah pertanyaan.



Gambar 7. Contoh Referensi Hadis dari Pertanyaan

Disediakan 10 sampel contoh pertanyaan umum, jawaban yang dihasilkan sistem akan dinilai prediksi model/jawabannya pada skala Scoring Evaluator yang telah ditetapkan. Hasil evaluasi menggunakan sistem Scoring Evaluator dari langchain evaluation adalah sebagai berikut:

Tabel 2. Pengujian Scoring Evaluator

No	Pertanyaan	Evaluasi result
1.	Jika saya menemukan barang di jalan, apakah saya boleh mengambilnya ?	1,0
2.	Ketika saya sedang berpuasa, saya tidak sengaja meminum air karena saya lupa, bagaimana dengan puasa saya? apakah puasa saya akan diterima ?	1,0
3.	Saya ketiduran pada pagi hari sehingga ketinggalan shalat subuh, bagaimana dengan shalat subuh saya ?	1,0
4.	Saat sedang berpuasa saya muntah, bagaimana dengan puasa saya ?	1,0
5.	Pada saat shalat saya ingin membuang angin, namun karena sedang shalat saya menahannya, bagaimana dengan shalat saya ?	1,0
6.	Saya memberi makan orang yang berpuasa, seperti apa pahala yang saya dapatkan ?	1,0
7.	Ketika saya sedang shalat, kemudian ada orang yang mengucapkan salam kepada saya, apa yang harus saya lakukan ?	1,0
8.	Saya sedang melaksanakan sholat dan kemudian ada orang yang lewat didepan saya, apa yang harus saya lakukan ?	0,7

No	Pertanyaan	Evaluasi result
9.	Ketika saya sedang duduk, tiba-tiba ada anjing yang mendekat dan menjilati kaki saya sehingga air liur anjing tersebut mengenai kaki saya, bagaimana saya membersihkannya ?	1,0
10.	Apa Keutamaan bulan Ramadhan ?	1,0
Rata-rata		0,97

Evaluasi ini menggunakan parameter $normalize_by = 10$ untuk menstandarisasi skor. Dari pengujian yang telah dilakukan pada Tabel 2., sistem menilai prediksi model dengan tepat dalam 9 dari 10 kasus. Dalam kasus di mana satu prediksi menerima evaluasi result 0.7, ini menunjukkan bahwa prediksi tersebut memiliki *Score* sebesar 7 dari 10 berdasarkan kriteria yang ditetapkan yakni pada *Score* 7 Jawabannya sejalan dengan referensi tetapi memiliki kesalahan atau kelalaian kecil. Sementara itu, 9 kasus lainnya pada *Score* 10 yakni Jawabannya benar-benar akurat dan sejalan sepenuhnya dengan referensi. Evaluasi result ini berada dalam rentang nilai yang baik, dengan nilai rata-rata evaluasi keseluruhan yaitu 0.97 dari rentang 0 - 1.

Pengujian lain dilakukan dengan evaluasi kualitas jawaban, evaluasi ini melibatkan partisipasi dari 5 ahli hadis yang memberikan penilaian terhadap 10 pertanyaan beserta jawaban dan sumber seperti pada pengujian *Scoring Evaluator* sebelumnya. *Score* Sangat Setuju (SS) bernilai 5, *Score* Setuju (S) bernilai 4, *Score* Netral (N) bernilai 3, *Score* Tidak Setuju (TS) bernilai 2, dan *Score* Sangat Tidak Setuju (STS) bernilai 1. Hasil yang didapatkan dari masing-masing *Score* adalah sebagai berikut: *Score* SS atau 5 berjumlah 28, *Score* S atau 4 berjumlah 20, *Score* N atau 3 berjumlah 1, *Score* TS atau 2 berjumlah 1, dan *Score* STS atau 1 berjumlah 0.

$$Persentase\ Akurasi = \frac{Total\ Score}{X} \times 100\%$$

$$Persentase\ Akurasi = \frac{((5 \times 28)(4 \times 20)(3 \times 1)(2 \times 1)(1 \times 0))}{5 \times 10 \times 5} \times 100\%$$

$$= 90\%$$

Hasil persentase akurasi didapat dari Jumlah atau total *Score* untuk masing-masing *Score* dikalikan dengan nilai *Score* dari *Score* yang terkait. Setelah itu, hasil tersebut dikalikan dengan persentase 100%. Selanjutnya, dilakukan perhitungan persentase akurasi dengan membagi total *Score* tersebut dengan total *Score* maksimumnya (X). Proses ini menghasilkan persentase akurasi sebesar 90 %. Pada Skala *Likert* [29] interval penilaian pada rentang Indeks 0% – 19,99% merupakan Sangat Tidak Setuju, Indeks 20% – 39,99% merupakan Tidak Setuju, Indeks 40% – 59,99% merupakan Kurang Setuju, Indeks 60% – 79,99% merupakan Setuju, dan Indeks 80% – 100% merupakan Sangat Setuju. Pada penelitian ini akurasi sebesar 90 % merupakan rentang indeks 80% – 100%, dapat disimpulkan responden sangat setuju terhadap jawaban yang dihasilkan sistem. Pada Skala *Likert* ini, persepsi responden didasari dari penilaian ahli atau pakar hadis untuk menilai kualitas jawaban yang dihasilkan oleh sistem seputar hadis.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan *Langchain Retriever* dengan model *GPT-4-1106-preview* dari *OpenAI* dalam pengembangan sistem *chatbot* hadis berbasis Telegram telah berhasil dilakukan dengan baik. Penelitian ini memanfaatkan data dari Kitab 9 Imam Hadis yang terdiri dari total 62.169 hadis sebagai sumber pengetahuan bagi *chatbot*. Pada pengujian menggunakan *Scoring Evaluator* dari *langchain evaluation* juga menunjukkan hasil dengan *Score* rata-rata total yang diperoleh mencapai 0,97. *Score* ini mengindikasikan bahwa *chatbot* hadis berbasis

Telegram ini memiliki kemampuan yang sangat baik dalam memberikan jawaban yang relevan dan sesuai dengan pertanyaan yang diajukan. Selain itu, pada pengujian evaluasi kualitas jawaban dari *chatbot* berhasil memperoleh persentase akurasi sebesar 90 % berdasarkan penilaian ahli hadis. Nilai ini menunjukkan bahwa responden sangat setuju terhadap jawaban yang diberikan oleh *chatbot*. Dari 10 pertanyaan yang disediakan, keseluruhan pertanyaan merupakan pertanyaan yang umum dan sederhana yang sering diajukan oleh orang awam. Sistem ini telah berhasil memberikan jawaban yang memadai untuk pertanyaan-pertanyaan tersebut, memberikan kegunaan yang signifikan bagi pengguna awam dalam mencari pemahaman tentang hadis. Hal ini juga menunjukkan bahwa integrasi ini menghasilkan sistem *chatbot* yang efektif dan efisien dalam menjawab pertanyaan-pertanyaan terkait hadis secara umum. Dengan adanya *chatbot* ini, masyarakat dapat dengan mudah mengakses dan mempelajari hadis-hadis dari Kitab 9 Imam Hadis melalui *platform* Telegram yang mudah digunakan. Untuk pengembangan penelitian lebih lanjut, fokus dapat diperluas untuk mengatasi pemaknaan yang lebih kompleks dari *Musykil al-Hadits* dan *asababul wurud* dalam hadis dengan menambah kitab *syarah* untuk menjawab pertanyaan-pertanyaan yang lebih dalam dan kompleks serta menghindari munculnya kesalahpahaman terhadap pemaknaan hadis.

Ucapan Terima Kasih: Penulis mengucapkan terimakasih kepada teman-teman dan para dosen atas kontribusi untuk masukkan dan arahan dalam penulisan penelitian ini.

Referensi

- [1] N. Yuslem, A. Ardiansyah, J. Juriono, U. K. Rambe, and I. F. A. Nasution, "Perkembangan Studi Hadis: Telaah dan Pemetaan Kajian Hadis Pada UIN Sumatera Utara," *AL QUDS : Jurnal Studi Alquran dan Hadis*, vol. 6, no. 3, Dec. 2022, doi: 10.29240/alquds.v6i3.3865.
- [2] A. Zubir Rosdi, S. Najihuddin Syed Hassan, N. Asiah Fasehah Muhamad, N. Izzatul Huda Mohamad Zainuzi, M. Shiham Mahfuz, and F. Pengajian Quran dan Sunnah, "Panduan Asas Kaedah Kenal Pasti Status Hadis: Kajian Diskriptif Penggunaan Ensiklopedia Hadis 9 Imam," *Journal of Hadith Studies*, vol. 8, no. 1, pp. 2550–1488, 2023, doi: 10.33102/johs.v8i1.225.
- [3] S. Muhammad Alfatih, Q. Saifuddin Zuhri, and M. Inayatul, "Digitalisasi Hadis Ala Pusat Kajian Hadis(PKH): Distribusi, Ciri, dan Kontribusi dalam Kajian hadis Indonesia," *Mashdar : Jurnal Studi Al-Quran dan Hadis*, vol. 3, no. 2, pp. 105–128, Mar. 2021, doi: 10.15548/mashdar.v3i2.2982.
- [4] W. Wasoni and M. I. Helmy, "Pemaknaan Hadis-Hadis Jihad dalam Website VOA ISLAM dan Relevansinya dengan Diskursus Islam Indonesia," *AL QUDS: Jurnal Studi Alquran dan Hadis*, vol. 6, no. 1, p. 343, May 2022, doi: 10.29240/alquds.v6i1.3401.
- [5] R. Fauzana, "Strategi Komunikasi Dakwah bil Qalam Komunitas Revowriter di Media Digital," *Idarotuna*, vol. 3, no. 3, p. 229, Feb. 2022, doi: 10.24014/idarotuna.v3i3.16440.
- [6] A. Abdi, S. Hasan, M. Arshi, S. M. Shamsuddin, and N. Idris, "A Question Answering System In Hadith Using Linguistic Knowledge," *Comput Speech Lang*, vol. 60, Mar. 2020, doi: 10.1016/j.csl.2019.101023.
- [7] Tatwadarshi P. Nagarhalli, Vinod Vaze, and N. K. Rana, "A Review of Current Trends in the Development of Chatbot Systems," *2020 6 th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2020, doi: 10.1109/ICACCS48705.2020.9074420
- [8] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: a Multimodal Low-Code Chatbot Development Framework," *IEEE Access*, vol. 8, pp. 15332–15346, 2020, doi: 10.1109/aCCESS.2020.2966919.
- [9] A. Hariansyah, E. Haerani, and M. Affandes, "Implementation of Telegram Chatbot as Information Service of Madani Hospital Pekanbaru," *Jurnal Ilmiah Merpati*, vol. 11, no. 3, 2023, doi: 10.24843/JIM.2023.v11.i03.p05.
- [10] G. F. Avisyah, I. J. Putra, and S. S. Hidayat, "Open Artificial Intelligence Analysis using ChatGPT Integrated with Telegram Bot," *Jurnal ELTIKOM*, vol. 7, no. 1, pp. 60–66, Jun. 2023, doi: 10.31961/eltikom.v7i1.724.
- [11] Keivalya Pandya and Prof. Dr. Mehfuza Holia, "Automating Customer Service using LangChain Building custom open-source GPT Chatbot for organizations," *arXiv preprint*, 2023, arXiv : 2310.05421
- [12] Arjun Pesaru, Taranveer Singh Gill, and Archit Reddy Tangella, "AI assistant for document management Using Lang Chain and Pinecone," *International Research Journal of Modernization in Engineering Technology and Science*, Jun. 2023, doi: 10.56726/irjmet42630.
- [13] O. Topsakal and T. C. Akinci, "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast," *All Sciences Proceedings*, 2023, [Online]. Available: <http://as-proceeding.com/>;Konya,Turkey<https://www.icaens.com/>

- [14] Z. Fahma, I. Gorby, C. Ramdani, and K. N. Meiah, "Website-Based Competence Certification Information System Using Rapid Application Development (Rad) Method," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 2, pp. 219–226, 2022, doi: 10.20884/1.jutif.2022.3.2.173. 496
497
498
- [15] M. M. Hossain, S. Krishna Pillai, S. E. Dansy, and A. A. Bilong, "Mr. Dr. Health-Assistant Chatbot," *International Journal of Artificial Intelligence*, vol. 8, no. 2, pp. 58–73, Dec. 2021, doi: 10.36079/lamintang.ijai-0802.301. 499
500
- [16] D. Sukma Hani and C. Indah Ratnasari, "Klasifikasi Masalah Pada Komunitas Marah-Marah di Twitter Menggunakan Long Short-Term Memory," *Jurnal Media Informatika Budidarma*, vol. 7, pp. 1829–1837, 2023, doi: 10.30865/mib.v7i4.6755. 501
502
- [17] S. K. Nigam, S. K. Mishra, A. K. Mishra, N. Shallum, and A. Bhattacharya, "Legal Question-Answering in the Indian Context: Efficacy, Challenges, and Potential of Modern AI Models," *arXiv preprint*, 2023, arXiv : 2309.14735 503
504
- [18] J. Lin, R. Pradeep, T. Teofili, and J. Xian, "Vector Search with OpenAI Embeddings: Lucene Is All You Need," *arXiv preprint*, Aug. 2023, arXiv : 2308.14963 505
506
- [19] N. Muennighoff, "SGPT: GPT Sentence Embeddings for Semantic Search," *arXiv preprint*, Feb. 2022, arXiv : 2202.08904 507
- [20] U. Gupta, "GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models," *arXiv preprint*, Sep. 2023, arXiv : arxiv.org/abs/2309.03079 508
509
- [21] J. Yang et al., "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond," *arXiv preprint*, Apr. 2023, arXiv : 2304.13712 510
511
- [22] Y. Cheng, J. Chen, Q. Huang, Z. Xing, X. Xu, and Q. Lu, "Prompt Sapper: A LLM-Empowered Production Tool for Building AI Chains," *arXiv preprint*, Jun. 2023, arXiv : 2306.12028 512
513
- [23] R. Tang, Y.-N. Chuang, and X. Hu, "The Science of Detecting LLM-Generated Texts," *arXiv preprint*, Feb. 2023, arXiv : 2303.07205 514
515
- [24] M. Ranjit, G. Ganapathy, R. Manuel, and T. Ganu, "Retrieval Augmented Chest X-Ray Report Generation using OpenAI GPT models," *arXiv preprint*, May 2023, arXiv : 2305.03660 516
517
- [25] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "GPT-Driver: Learning to Drive with GPT," *arXiv preprint*, Oct. 2023, arXiv : 2310.01415 518
519
- [26] M. Bommarito and D. M. Katz, "GPT Takes the Bar Exam," *arXiv preprint*, Dec. 2022, arXiv : 2212.14402 520
- [27] Y. Bai et al., "Longbench: A Bilingual, Multitask Bench-Mark For Long Context Understanding," *arXiv preprint*, 2023. arXiv : 2308.14508 521
522
- [28] Adith Sreeram and Pappuri Jithendra Sai, "An Effective Query System Using LLMs and LangChain," in *International Journal of Engineering Research & Technology (IJERT)*, Elsevier B.V., 2023, pp. 367–369. doi: 10.1016/j.procs.2016.02.023. 523
524
- [29] I. Budi Trisno, D. Febrian Elvianto, and U. Widya Kartika, "Aplikasi Sistem Informasi Manajemen Pengiriman Barang Pt. Gst," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 9, no. 5, 2022, doi: 10.25126/jtiik.202295390. 525
526
- [30] "Scoring Evaluator | 🦜🗣️ LangChain.", 2024. [Online]. Available: https://python.langchain.com/v0.1/docs/guides/productionization/evaluation/string/scoring_eval_chain/ 527
528
529
530