



## *Implementing a Qur'anic Tafsir Chatbot Through Chainlit with a Groq-Based Approach*

### **Implementasi Chatbot Tafsir Al-Qur'an Menggunakan Chainlit dengan Pendekatan Groq**

**Muhammad Rizky Maulana<sup>1</sup>,  
Nazruddin Safaat Harahap<sup>2\*</sup>, Okfalisa<sup>3</sup>, Yusra<sup>4</sup>**

<sup>1,2,3,4</sup>Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,  
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

E-Mail: <sup>1</sup>12150114879@students.uin-suska.ac.id, <sup>2</sup>nazruddin.safaat@uin-suska.ac.id,  
<sup>3</sup>okfalisa@gmail.com, <sup>4</sup>yusra@uin-suska.ac.id

Received Apr 03rd 2025; Revised Jun 16th 2025; Accepted Jun 20th 2025; Available Online Jun 25th 2025; Published Jun 25th 2025

Corresponding Author: Nazruddin Safaat Harahap

Copyright © 2025 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

#### **Abstract**

*Understanding the interpretation of the Qur'an is often a challenge in the world of education, especially for students, students, or the general public who do not have a background in interpretation or Arabic language skills. Limited access to easy-to-understand interpretations, as well as the absence of interactive learning media that can answer questions contextually, are obstacles in the process of learning Islam. This study aims to develop an artificial intelligence (AI)-based chatbot as a supporting media for learning responsive and adaptive interpretation of the Qur'an. The system is designed using the Chainlit framework as a web interface, supported by Groq to accelerate the inference process, and the integration of LangChain and Large Language Models (LLM) to understand the contents of the interpretation, especially the Jalalain Interpretation in Indonesian PDF format. The document used is a single representative digital file. The system process includes text extraction, text division (chunking), embedding formation, and vector-based semantic search. Evaluation using BERTScore produced an average precision value of 71.84%, recall of 78.11%, and F1-score of 74.80%, indicating the system's ability to provide semantically relevant answers. The results of this study contribute to providing efficient and contextual AI-based digital interpretation learning media, as well as being a potential solution to support a more interactive and modern Islamic education process.*

*Keyword: Artificial intelligence, Chainlit, Chatbot, Groq, Tafsir*

#### **Abstrak**

Pemahaman terhadap tafsir Al-Qur'an sering kali menjadi tantangan dalam dunia pendidikan, khususnya bagi siswa, mahasiswa, atau masyarakat umum yang tidak memiliki latar belakang ilmu tafsir maupun kemampuan bahasa Arab. Keterbatasan akses terhadap tafsir yang mudah dipahami, serta ketiadaan media pembelajaran yang interaktif dan mampu menjawab pertanyaan secara kontekstual, menjadi hambatan dalam proses pembelajaran keislaman. Penelitian ini bertujuan untuk mengembangkan chatbot berbasis kecerdasan buatan (AI) sebagai media pendukung pembelajaran tafsir Al-Qur'an yang responsif dan adaptif. Sistem dirancang menggunakan framework Chainlit sebagai antarmuka web, didukung oleh Groq untuk mempercepat proses inferensi, serta integrasi LangChain dan Large Language Models (LLM) untuk memahami isi tafsir, khususnya Tafsir Jalalain dalam format PDF berbahasa Indonesia. Dokumen yang digunakan berupa satu file digital utuh yang representatif. Proses sistem meliputi ekstraksi teks, pembagian teks (chunking), pembentukan embedding, dan pencarian semantik berbasis vektor. Evaluasi menggunakan BERTScore menghasilkan nilai rata-rata precision sebesar 71,84%, recall 78,11%, dan F1-score 74,80%, menunjukkan kemampuan sistem dalam memberikan jawaban yang baik secara semantik. Hasil penelitian ini berkontribusi dalam menyediakan media pembelajaran tafsir digital berbasis AI yang efisien dan kontekstual, serta menjadi solusi potensial untuk mendukung proses pendidikan Islam yang lebih interaktif dan modern.

Kata Kunci: Chainlit Chatbot, Groq, Kecerdasan Buatan, Tafsir



## 1. PENDAHULUAN

Al-Qur'an adalah kitab suci umat Islam yang berisi petunjuk hidup dan hukum-hukum Allah yang diturunkan untuk umat manusia. Bagi umat Muslim, memahami isi Al-Qur'an menjadi suatu kewajiban, karena dalam setiap ayatnya terdapat makna dan pelajaran yang menjadi pedoman hidup. Namun, bahasa Al-Qur'an yang menggunakan bahasa Arab klasik serta gaya bahasanya yang kaya akan makna menjadikan pemahaman terhadapnya tidak mudah, terutama bagi mereka yang tidak memiliki latar belakang pengetahuan bahasa Arab dan ilmu tafsir. Penelitian menunjukkan bahwa tafsir Al-Qur'an seperti karya Ibnu Katsir sangat membantu dalam menggali makna ayat-ayat Al-Qur'an. Namun, memahami tafsir ini juga memerlukan latar belakang pendidikan Islam yang mendalam [1]. Al-Qur'an adalah kalam Allah yang ditulis dalam bentuk mushaf, diturunkan kepada Nabi Muhammad SAW melalui Malaikat Jibril secara mutawatir baik lafadh maupun maknanya, dimulai dari al-Fatihah hingga an-Nas. Hal ini senada dengan salah satu penelitian yang menegaskan bahwa Al-Qur'an merupakan kalam Allah bersifat mu'jizat, membacanya adalah ibadah, dan berisi wahyu sebagai petunjuk dan Pelajaran [2] Tafsir Al-Qur'an, seperti Tafsir Al-Jalalain dan Tafsir Al-Azhar, memberikan panduan bagi umat Muslim dalam memahami wahyu Allah. Penelitian oleh Rizki et al [3], menunjukkan bahwa teknologi modern seperti *Large Language Models* (LLMs) dapat membantu menyederhanakan akses terhadap tafsir dengan tingkat akurasi sebesar 84.29%, menjadikannya solusi praktis untuk memahami tafsir secara lebih efisien. Kecerdasan buatan (*Artificial Intelligence/AI*) merupakan teknologi yang semakin berkembang pesat dan telah digunakan secara luas di berbagai bidang, termasuk pendidikan. Salah satu implementasi AI yang menarik perhatian adalah *chatbot*, yang mampu merespons pertanyaan secara kontekstual dan cepat. Namun, dalam konteks pendidikan Islam, masih terdapat tantangan signifikan dalam memahami tafsir Al-Qur'an secara personal dan mendalam, khususnya bagi pengguna yang tidak memiliki latar belakang bahasa Arab atau ilmu tafsir. Selain itu, media digital yang interaktif dan mampu menjawab pertanyaan secara real-time tentang tafsir masih sangat terbatas. Hal ini menunjukkan adanya kebutuhan untuk menghadirkan sistem berbasis teknologi yang dapat menyederhanakan akses pemahaman tafsir Al-Qur'an.

Dalam konteks aplikasi berbasis AI, penelitian oleh Topsakal et al. [4] dan Sreeram menunjukkan potensi *LangChain* dalam mendukung pengembangan aplikasi berbasis LLMs, seperti agen otonom dan *chatbot*, untuk eksplorasi lebih lanjut dalam aplikasi LLM. Framework *LangChain* digunakan untuk menciptakan sistem tanya jawab yang efektif, khususnya dalam mengekstrak informasi akurat dari dokumen PDF melalui aplikasi web interaktif. Penelitian terkait lainnya oleh Permadi et al. [5] juga mengungkapkan bahwa implementasi *chatbot* untuk Tafsir Al-Azhar dapat mencapai tingkat akurasi sebesar 83.71% melalui pengujian *User Acceptance Test* (UAT), menunjukkan potensinya dalam meningkatkan aksesibilitas dan pemahaman terhadap tafsir Al-Qur'an. Penelitian Mansurova [6] mengembangkan *chatbot* untuk domain blockchain yang menggabungkan LLM, *LangChain*, dan teknik pengayaan pengetahuan eksternal untuk meningkatkan keakuratan sistem dalam menjawab pertanyaan domain-spesifik dengan tingkat presisi yang tinggi. Sementara itu, Pesaru [7] mengimplementasikan *LangChain* dan LLM dalam asisten dokumen berbasis PDF yang memanfaatkan Pinecone sebagai basis data vektor, dengan penekanan pada pengelolaan dokumen yang efisien melalui pendekatan berbasis *embedding*.

Dalam penelitian Soygazi [8] mengeksplorasi kinerja LLM seperti *ChatGPT* dan *LangChain* dalam pendidikan matematika, terutama dalam memahami dan menjawab soal cerita, yang memberikan wawasan tentang penerapan teknologi canggih untuk mendukung pembelajaran. Selain itu, penelitian oleh Soygazi et al. [8] berfokus pada penggunaan teknik *Embedding-Based Retrieval* (EBR) dalam sistem pencarian Facebook, memberikan wawasan tentang pengaplikasian sistem pengambilan berbasis *embedding* yang relevan untuk pengembangan *chatbot*. Penelitian serupa oleh Mar'arif [9] menerapkan *chatbot* Telegram untuk menjawab pertanyaan fiqh kontemporer menggunakan *LangChain* dan LLM, dan menunjukkan keberhasilan dari sisi akurasi dan penerimaan pengguna.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan *chatbot* berbasis *Chainlit* sebagai antarmuka yang mampu menjawab pertanyaan pengguna mengenai tafsir Al-Qur'an khususnya pada Tafsir *Jalalain* dengan kontekstual dan akurat. Sistem ini akan mengintegrasikan *LangChain* dan *Groq* untuk mengoptimalkan pemrosesan LLM, dan ditujukan bagi Pelajar, Mahasiswa, serta Masyarakat umum sebagai media pembelajaran tafsir yang interaktif dan mudah diakses.

Penelitian ini berkontribusi dalam pengembangan *chatbot* tafsir Al-Qur'an yang berfokus pada Tafsir *Jalalain* dengan pendekatan baru melalui integrasi *Chainlit* sebagai antarmuka, *Groq* untuk percepatan inferensi, dan *Ollama* untuk pemrosesan lokal. Berbeda dari penelitian Permadi et al [5] dan Mar'arif et al [9] yang mengembangkan *chatbot* berbasis Telegram dan bergantung pada layanan *cloud*, namun dalam penelitian ini sistem dirancang sebagai aplikasi *web* interaktif dengan pengolahan dokumen tafsir PDF secara langsung. Selain itu, kualitas jawaban sistem ini dievaluasi menggunakan metrik semantik *BERTScore*, yang belum diterapkan dalam studi Rizki et al [3] maupun lainnya. Kombinasi teknologi ini merupakan inovasi yang belum banyak dieksplorasi, dan diharapkan dapat memperluas akses pembelajaran Tafsir *Jalalain* yang lebih efisien, *private*, dan kontekstual bagi pelajar maupun masyarakat umum.

## 2. METODOLOGI PENELITIAN

Metodologi penelitian ini mengacu pada pendekatan pengembangan sistem tanya-jawab berbasis dokumen PDF yang memanfaatkan teknologi *Large Language Models* (LLM) dan *framework LangChain*. Penelitian oleh Rizki *et al* [3] serta Permadi *et al* [5] menunjukkan efektivitas penggunaan *LangChain* dalam membangun sistem *chatbot* berbasis teks keislaman. Sementara itu, Mansurova *et al.* [6] dan Pesaru *et al* [7] mendemonstrasikan integrasi LLM dengan basis data vektor dan teknik *embedding* dalam domain *blockchain* dan manajemen dokumen. Penelitian ini memperluas pendekatan tersebut dengan menerapkan model inferensi lokal menggunakan *Ollama* dan akselerasi *Groq*, yang belum digunakan dalam studi terdahulu. Selain itu, sistem ini memproses dokumen tafsir yang bersumber dari file PDF secara langsung mengacu pada pendekatan seperti Medeiros *et al* [10] yang mengimplementasikan *chunking* dan *semantic retrieval* pada dokumen manual otomotif.

### 2.1. Permasalahan Ekstraksi pada Dokumen PDF dengan Struktur Kompleks dan Hasil Pindai

Masalah dalam penelitian ini diidentifikasi melalui studi literatur, observasi langsung, serta eksplorasi terhadap berbagai alat bantu ekstraksi data dari dokumen PDF seperti PyMuPDF, pdfminer.six, dan Tesseract OCR. Melalui proses eksplorasi dan pengujian, ditemukan sejumlah kendala teknis yang sering muncul saat pengguna mencoba mengambil informasi dari file PDF, khususnya yang berasal dari hasil pemindaian atau memiliki struktur tata letak yang kompleks. Salah satu permasalahan utama yang ditemukan adalah rendahnya akurasi dalam pengenalan teks pada file PDF hasil pemindaian. Dokumen jenis ini biasanya menyimpan teks dalam bentuk gambar, sehingga memerlukan proses *Optical Character Recognition* (OCR). Mengusulkan metode pra-pemrosesan gambar dan teknik *thresholding* adaptif yang dapat meningkatkan akurasi OCR secara signifikan dengan memfokuskan pada intensitas piksel utama dalam teks dan menghilangkan fitur yang tidak relevan. Selain itu, banyak dokumen PDF tidak dilengkapi dengan struktur semantik yang eksplisit, seperti heading, paragraf, atau tabel, sehingga menyulitkan sistem untuk memahami konteks isi dokumen. [11] mengembangkan *PubLayNet*, sebuah dataset besar untuk analisis tata letak dokumen, dengan mencocokkan representasi XML dan konten dari lebih dari satu juta artikel PDF yang tersedia secara publik di PubMed Central. Dataset ini memungkinkan pelatihan model deep learning yang lebih akurat dalam mengenali tata letak dokumen ilmiah. Masalah lainnya adalah keterbatasan alat ekstraksi dalam menangani PDF dengan struktur yang tidak standar, seperti tabel multi-kolom atau tata letak majalah. Tools seperti PyMuPDF dan pdfminer.six masih sering gagal mempertahankan urutan logis informasi dalam dokumen. Selain itu, format PDF yang tidak ramah mesin menjadi tantangan besar ketika digunakan bersama model berbasis AI seperti *ChatGPT* atau LLM lainnya, yang membutuhkan input teks yang bersih, terstruktur, dan semantik. Permasalahan-permasalahan ini menunjukkan bahwa proses pra-pemrosesan dokumen, rekonstruksi tata letak, serta penyusunan representasi semantik sangat diperlukan untuk memungkinkan sistem berbasis AI memahami dan merespons isi dokumen PDF secara kontekstual dan relevan.

### 2.2 Pengumpulan Data

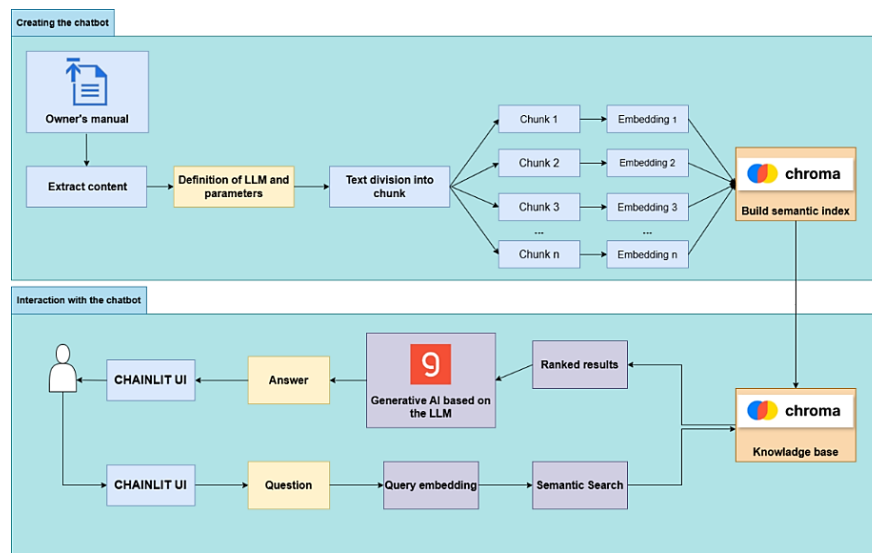
Dalam penelitian ini, proses pengumpulan data difokuskan pada pemanfaatan Tafsir *Jalalain* sebagai sumber utama penafsiran ayat-ayat Al-Qur'an. Tafsir *Jalalain*, karya Jalaluddin al-Mahalli dan Jalaluddin as-Suyuthi, dipilih karena karakteristiknya yang ringkas, mudah dipahami, dan memiliki struktur narasi yang konsisten, menjadikannya cocok untuk integrasi dalam sistem pencarian tematik berbasis teknologi. Teks Tafsir *Jalalain* tersedia secara daring melalui platform seperti Tafsirweb.com dan Quran.com, yang menyediakan antarmuka pencarian ayat yang spesifik serta kompatibel dengan perangkat lunak pemrosesan bahasa alami atau *Natural Language Processing* (NLP). Alasan utama pemilihan tafsir ini adalah karena penggunaan luasnya dalam pendidikan Islam serta ketersediaan format digital yang stabil dan bersih, yang mendukung kebutuhan teknis seperti tokenisasi dan ekstraksi makna. Studi menunjukkan bahwa tafsir berbasis web seperti Tafsir *Jalalain* tidak hanya memperluas akses publik terhadap tafsir klasik, tetapi juga memfasilitasi pengembangan studi tafsir berbasis teknologi informasi melalui struktur naratifnya yang sistematis dan terorganisir [12].

Selain Tafsir *Jalalain*, data pendukung juga dikumpulkan dari jurnal akademik yang diperoleh melalui database seperti *Google Scholar*, *JSTOR*, dan *ResearchGate*. Pencarian literatur dilakukan dengan kata kunci seperti "tafsir *Jalalain* digital", "tafsir klasik dan teknologi", serta "digitalisasi tafsir Al-Qur'an". Jurnal-jurnal yang digunakan merupakan publikasi terbaru (minimal sejak 2019) untuk memastikan relevansi konteks dan metodologi terhadap perkembangan teknologi dan isu-isu kontemporer dalam studi tafsir.

### 2.3 Analisa Sistem

Pada tahap ini, analisis sistem dilakukan untuk memahami kebutuhan sistem dan bagaimana elemen-elemen yang ada dapat mendukung tujuan penelitian. Proses analisis meliputi identifikasi fitur utama seperti pengolahan data PDF, integrasi tafsir Al-Qur'an, dan kemampuan respons interaktif. Alur sistem dirancang untuk memastikan efisiensi dalam membaca data dari file PDF dan menghasilkan jawaban berbasis tafsir menggunakan teknologi seperti *Chainlit*, *Ollama*, dan *Groq*. Analisis sistem ini mengacu pada prinsip-prinsip

rekayasa perangkat lunak yang menekankan pada identifikasi kebutuhan fungsional dan non-fungsional sistem secara menyeluruh [13]. Selain itu, penilaian juga mencakup potensi permasalahan teknis, seperti kompleksitas format file PDF, serta tantangan dalam mengoptimalkan pemrosesan pertanyaan berbasis teks menggunakan model bahasa besar (LLM) [14]. Kajian literatur oleh Safaat [15] mendukung pentingnya pengembangan sistem *chatbot* berbasis LLM, dengan menunjukkan peningkatan signifikan jumlah publikasi *chatbot* sejak tahun 2020. Studi tersebut juga mengidentifikasi tren riset terkini, termasuk pergeseran dari model berbasis aturan ke pendekatan generatif berbasis AI, serta potensi penerapan *chatbot* dalam konteks pendidikan dan keislaman.



**Gambar 1.** Alur Perancangan *Chatbot* Tafsir

Alur pada Gambar 1 menggunakan pendekatan dari penelitian sebelumnya [10] yang mencakup proses mengimpor file, mengekstrak konten, membagi teks menjadi potongan-potongan (*chunks*), mengubahnya menjadi *embedding*, membangun indeks AI semantik sebagai basis pengetahuan, kemudian melakukan pencarian terhadap basis tersebut untuk menemukan potongan yang paling relevan, dan akhirnya menghasilkan jawaban dengan bantuan model bahasa.

*Groq* menggunakan pendekatan perangkat lunak yang fleksibel dan efisien dalam mempercepat beban kerja deep learning, menjadikannya ideal untuk aplikasi inferensi berskala besar seperti *chatbot* tafsir [16]. *Ollama* merupakan sebuah *framework open-source* yang memungkinkan pengguna menjalankan LLMs secara lokal di perangkat mereka, tanpa perlu mengandalkan layanan cloud. Meskipun belum ada publikasi akademik khusus mengenai *Ollama*, pendekatan yang diusung sejalan dengan tren riset terbaru yang mendorong optimalisasi inferensi LLM secara lokal. [17] dalam penelitiannya menekankan pentingnya sistem local-cloud offloading untuk menjalankan LLM secara efisien dalam lingkungan multi-modal dan multi-tugas, guna meningkatkan kinerja sekaligus menjaga privasi data pengguna, membuktikan bahwa inferensi lokal dapat dicapai secara efektif. *Ollama* menghadirkan keunggulan serupa, seperti privasi data yang lebih tinggi karena tidak memerlukan pengiriman data ke server eksternal, kemudahan integrasi melalui antarmuka API dan CLI yang sederhana, serta kompatibilitas dengan beragam model LLM open-source seperti LLaMA, Mistral, dan Gemma. Dengan demikian, *Ollama* menjadi solusi praktis dan efisien dalam penerapan sistem LLM lokal, khususnya untuk aplikasi yang menuntut keamanan data dan respons real-time tanpa ketergantungan *cloud*.

*Chainlit* adalah *framework open-source* berbasis Python yang memudahkan pengembangan *chatbot* dengan LLMs, mendukung antarmuka interaktif, integrasi mudah, visualisasi alur, dan multi-modalitas. Keunggulannya terletak pada kemudahan penggunaan dan fleksibilitas. Studi oleh Dasanayaka et al. membahas penerapan antarmuka percakapan interaktif dalam sistem AI berbasis LLM, yang sejalan dengan fitur *Chainlit* [18]. Penggunaan *Groq*, *Chainlit*, dan *Ollama* secara bersamaan memungkinkan optimalisasi sistem *chatbot* berbasis LLMs dengan pendekatan yang saling melengkapi. *Groq* menyediakan akselerasi hardware khusus yang mempercepat inferensi model AI secara signifikan dibandingkan CPU/GPU konvensional. *Chainlit* berfungsi sebagai *framework* pengembangan aplikasi *chatbot* yang modular dan interaktif, memudahkan integrasi dan visualisasi percakapan, yang sulit dicapai jika hanya mengandalkan dua komponen lainnya [19]. Sementara itu, *Ollama* berfokus pada manajemen dan penyajian model bahasa secara lokal dengan keamanan dan privasi yang lebih terjaga, mengoptimalkan kinerja model dan pengalaman pengguna [18].

Jika hanya menggunakan satu atau dua komponen saja, sistem akan kehilangan keunggulan tertentu. Misalnya, tanpa *Groq*, kecepatan pemrosesan bisa terhambat; tanpa *Chainlit*, interaksi pengguna dan pengembangan aplikasi menjadi kurang efisien; tanpa *Ollama*, manajemen model lokal dan privasi kurang

optimal. Oleh karena itu, kombinasi ketiganya memberikan sinergi yang meningkatkan performa, pengembangan, dan keamanan aplikasi *chatbot* secara keseluruhan.

Penggunaan representasi vektor (*embedding*) dalam sistem ini memungkinkan transformasi teks tafsir menjadi format numerik yang dapat diproses oleh model AI. Metode ini memungkinkan sistem untuk memahami konteks dan makna dari teks, sehingga dapat mencocokkan pertanyaan pengguna dengan bagian tafsir yang relevan. Dalam survei oleh Ziyaden[20] *embedding* vektor terbukti efektif dalam menangkap informasi sintaksis dan semantik, serta berguna sebagai fitur tambahan dalam berbagai tugas NLP [20].

## 2.4 Implementasi

Tahap implementasi merupakan proses penerapan sistem ke dalam lingkungan operasional yang sesungguhnya. Dalam penelitian ini, implementasi dilakukan melalui platform berbasis web untuk membangun *chatbot* yang dapat merespons pertanyaan berdasarkan isi dokumen PDF. Pada tahap ini, sistem dirancang untuk mengutamakan pertanyaan yang bersifat spesifik, guna memastikan hasil inferensi yang relevan dan kontekstual. Semakin terfokus pertanyaan yang diajukan, semakin tinggi tingkat kesesuaian jawaban yang dihasilkan oleh model bahasa yang digunakan. Model LLM yang digunakan dalam sistem ini dijalankan melalui layanan inferensi *Groq* untuk memastikan respons yang cepat dan efisien.

### 1. Pengembangan *Chatbot*

Pengembangan *chatbot* dilakukan menggunakan *Chainlit*. *Chainlit* menawarkan sejumlah keuntungan dan fitur yang membuatnya menjadi platform yang menarik untuk pengembangan dan implementasi *chatbot* berbasis web. Selain itu, *Chainlit* memudahkan integrasi dengan berbagai model bahasa (LLM) dan menyediakan antarmuka yang interaktif untuk berkomunikasi dengan pengguna secara langsung melalui browser. Dengan pendekatan ini, pengguna dapat mengakses *chatbot* tanpa perlu menginstal aplikasi tambahan, cukup melalui peramban *web* [21].

### 2. Pengembangan API

Dalam mendukung pengembangan *chatbot*, digunakan konfigurasi *Application Programming Interface* (API) yang memungkinkan sistem untuk berkomunikasi dengan model bahasa besar (LLM) melalui layanan pihak ketiga. Penelitian ini memanfaatkan *LangChain* sebagai kerangka kerja untuk mengelola interaksi dengan model, serta *Groq* sebagai penyedia layanan pemrosesan LLM dengan performa tinggi. Beberapa variabel lingkungan dikonfigurasi untuk mendukung proses integrasi dengan layanan *LangChain* dan *Groq*.

## 2.5 Pengujian

Pengujian ini bertujuan untuk menentukan kelayakan penggunaan sistem *chatbot* yang telah dikembangkan. Evaluasi dilakukan dengan membandingkan jawaban yang dihasilkan oleh sistem dengan referensi data yang tersedia, menggunakan *BERTScore* sebagai metrik utama untuk mengukur kualitas jawaban. *BERTScore* menghitung kesamaan semantik antara teks yang dihasilkan oleh model dengan teks referensi, dengan memanfaatkan *embedding* kontekstual dari model bahasa pra-latih seperti BERT. Pendekatan ini memungkinkan pengukuran yang lebih akurat terhadap kesamaan makna, bahkan ketika struktur kalimat atau pilihan kata berbeda.

Keunggulan *BERTScore* dalam evaluasi sistem *chatbot* telah dibuktikan dalam berbagai penelitian. Misalnya, dalam penelitian [22] *BERTScore* menunjukkan korelasi yang lebih tinggi dengan penilaian manusia dibandingkan metrik tradisional seperti BLEU dan ROUGE dalam tugas-tugas generasi teks. Selain itu, penelitian [23] menemukan bahwa metrik berbasis *embedding* seperti *BERTScore* memiliki korelasi yang lebih baik dengan penilaian subjektif manusia dibandingkan metrik berbasis n-gram, menjadikannya pilihan yang lebih andal untuk mengevaluasi kualitas jawaban dalam sistem *chatbot*.

### 1. *Precision*

*Precision* merupakan ukuran yang digunakan untuk mengevaluasi sejauh mana token-token dalam jawaban sistem (kandidat) sesuai atau relevan dengan token-token dalam kalimat referensi. Evaluasi ini dilakukan dengan mencocokkan setiap token dalam jawaban sistem terhadap token dalam referensi menggunakan pendekatan pairwise cosine similarity. Nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar informasi yang dihasilkan oleh sistem memang relevan dan sesuai dengan konteks referensi. Dengan demikian, *Precision* merepresentasikan tingkat ketepatan sistem dalam menghasilkan token yang benar [22].

### 2. *Recall*

*Recall* mengukur kemampuan sistem dalam menangkap seluruh informasi penting yang terdapat dalam kalimat referensi. Penilaian dilakukan dengan membandingkan token-token dari referensi terhadap token-token yang terdapat dalam hasil keluaran sistem. Metode pairwise cosine similarity digunakan

untuk mengidentifikasi tingkat kemiripan antar token. Nilai *Recall* yang tinggi menunjukkan bahwa sistem berhasil mengenali sebagian besar, atau bahkan seluruh, informasi yang penting dari referensi. Oleh karena itu, *Recall* mencerminkan tingkat kelengkapan hasil yang diberikan oleh sistem [24].

### 3. *F1-Score*

*F1-Score* merupakan metrik evaluasi yang menggabungkan nilai *Precision* dan *Recall* ke dalam satu nilai agregat untuk memberikan gambaran yang lebih seimbang mengenai performa sistem. Nilai ini diperoleh dengan menggunakan rata-rata harmonis dari *Precision* dan *Recall*, sehingga *F1-Score* hanya akan tinggi apabila kedua metrik tersebut juga memiliki nilai yang tinggi. Penggunaan *F1-Score* sangat penting dalam konteks evaluasi sistem berbasis bahasa alami, karena mampu menunjukkan keseimbangan antara ketepatan dan kelengkapan informasi yang dihasilkan [14].

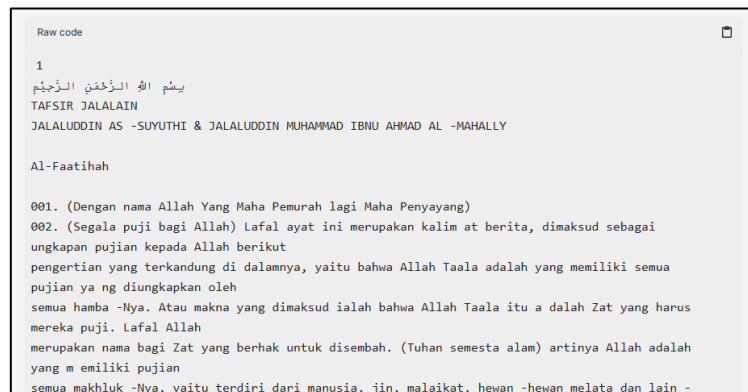
## 3. HASIL DAN PEMBAHASAN

Setelah model siap, implementasi *chatbot* tafsir diintegrasikan menggunakan *Chainlit*. Hasil implementasi dapat dilihat pada Gambar 1, di mana *chatbot* membentuk hubungan interaktif langsung antara pengguna dan sistem. Dengan memanfaatkan LLM yang dijalankan melalui *Ollama* serta diakselerasi oleh *Groq*, *chatbot* mampu merespons pertanyaan pengguna secara cepat dan relevan berdasarkan isi dokumen tafsir dalam format PDF. *Chainlit* berperan sebagai antarmuka interaktif yang memungkinkan pengguna menjelajahi isi tafsir secara intuitif dan mendalam. Integrasi ini memungkinkan pemrosesan dan pemahaman teks tafsir, khususnya dalam konteks Tafsir Jalalain, sehingga *chatbot* dapat memberikan respons yang personal, kontekstual, dan mudah dipahami.

### 3.1 Hasil Tahapan

#### 3.1.1 Ekstraksi Teks dari Dokumen PDF

Pada tahap ini, sistem melakukan ekstraksi isi teks dari dokumen PDF menggunakan pustaka pemrosesan dokumen (seperti PyPDF2). Tujuan dari proses ini adalah mengambil seluruh isi teks secara terstruktur agar dapat diolah lebih lanjut dalam tahapan berikutnya. Proses ekstraksi teks dari file PDF dapat ditunjukkan pada gambar 2.



```
Raw code
1
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
TAFSIR JALALAIN
JALALUDDIN AS -SUYUTHI & JALALUDDIN MUHAMMAD IBNU AHMAD AL -MAHALLY

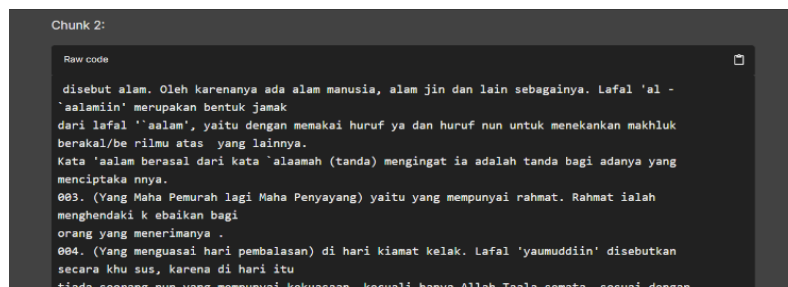
Al-Faatihah

001. (Dengan nama Allah Yang Maha Pemurah lagi Maha Penyayang)
```

Gambar 2. Proses Ekstraksi Teks dari File PDF

#### 3.1.2 Pembagian Teks ke dalam Chunk

Teks hasil ekstraksi dibagi menjadi beberapa bagian kecil (chunks) agar dapat diproses secara lebih efisien oleh model bahasa. Pembagian ini mempertimbangkan batas panjang token tertentu sesuai dengan kapasitas model LLM yang digunakan. Ilustrasi pembagian teks menjadi beberapa chunk dapat dilihat pada Gambar 3.

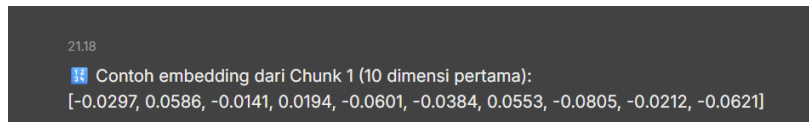


```
Chunk 2:
Raw code
disebut alam. Oleh karenanya ada alam manusia, alam jin dan lain sebagainya. Lafal 'al -
'alamiin' merupakan bentuk jamak
dari lafal 'alam', yaitu dengan memakai huruf ya dan huruf nun untuk menekankan makhluk
berakal/be rilmu atas yang lainnya.
Kata 'alam berasal dari kata 'alaamah (tanda) mengingat ia adalah tanda bagi adanya yang
menciptakanya.
003. (Yang Maha Pemurah lagi Maha Penyayang) yaitu yang mempunyai rahmat. Rahmat ialah
menghendaki k ebaikan bagi
orang yang menerimanya .
004. (Yang menguasai hari pembalasan) di hari kiamat kelak. Lafal 'yaumuddiin' disebutkan
secara khus, karena di hari itu
tiada seorang pun yang mempunyai kekuasaan, kecuali hanya Allah Taala semata, sesuai dengan
```

Gambar 3. Ilustrasi Pembagian Teks menjadi Beberapa Chunk

### 3.1.3 Pembentukan *Embedding*

Setiap chunk kemudian diubah menjadi bentuk vektor numerik menggunakan model *embedding*. *Embedding* ini merepresentasikan makna semantik dari setiap potongan teks dan akan digunakan untuk pencarian semantik. Visualisasi hasil transformasi chunk ke bentuk *embedding* ditunjukkan pada gambar 4.



**Gambar 4.** Visualisasi Hasil Transformasi Chunk ke Bentuk *Embedding*

### 3.1.4 Pembangunan Indeks Semantik

*Embedding* yang telah dibentuk disimpan dan diindeks menggunakan basis data vektor (dalam hal ini Chroma). Indeks ini berfungsi sebagai knowledge base yang dapat digunakan untuk melakukan pencarian berdasarkan kemiripan semantik.

### 3.1.5 Inferensi dan Pembangunan *Chatbot*

Setelah indeks terbentuk, *chatbot* diimplementasikan menggunakan antarmuka web berbasis *Chainlit*. Pertanyaan dari pengguna diubah menjadi *embedding*, kemudian dilakukan pencarian ke indeks semantik untuk menemukan chunk yang paling relevan. Potongan tersebut digunakan oleh model LLM via *Groq* untuk menghasilkan jawaban.

## 3.2 Pengujian

Pengujian dilakukan dengan menggunakan 10 sampel pertanyaan seperti pada tabel 1 dengan menghitung nilai *Precision*, *Recall*, dan *F1-Score* setiap pertanyaan.

**Tabel 1.** Sampel Pertanyaan

No	Pertanyaan
1.	Apa makna dari ayat "Alhamdulillah Rabbil 'Aalamiin" menurut Tafsir Jalalain?
2.	Siapakah yang dimaksud dengan "al-'aalamiin" dalam ayat kedua Surah Al-Faatihah?
3.	Mengapa Allah disebut "Maaliki Yawmiddiin"?
4.	Apa yang dimaksud dengan "ihdinaash-shiraathal mustaqiim"?
5.	Siapa yang dimaksud dengan "maghduubi 'alaihim" dan "dhaalliin"?
6.	Apa maksud dari ayat "Alif laam miim" dalam Al-Baqarah ayat 1?
7.	Apa fungsi Al-Qur'an menurut Surah Al-Baqarah ayat 2?
8.	Siapa yang dimaksud dengan "orang-orang yang bertakwa" dalam Al-Baqarah 2?
9.	Apa yang dimaksud dengan "yu'minuuna bimaa unzila ilayka wa maa unzila min qabluka"?
10.	Mengapa orang kafir tidak akan beriman meski diberi peringatan? (Al-Baqarah:6)

Berikut adalah contoh jawaban referensi dan jawaban prediksi:

- Jawaban Referensi soal no 1  
Artinya adalah "Segala puji bagi Allah, Tuhan semesta alam." Allah adalah Zat yang memiliki semua pujian dan seluruh makhluk yang berakal termasuk dalam kategori 'alam'.
- Jawaban Prediksi soal no 1  
Menurut Tafsir Jalalain, "Alhamdulillah Rabbil 'Aalamiin" berarti "Segala puji bagi Allah, Tuhan semesta alam". Allah adalah Zat yang memiliki semua pujian dan seluruh makhluk yang berakal termasuk dalam kategori 'alam. Lafal 'al-aalamiin' merupakan bentuk jamak dari lafal 'alam', yaitu dengan memakai huruf ya dan huruf nun untuk menekankan makhluk berakal/berilmu atas yang lainnya.
- Jawaban Referensi soal no 2  
Mereka adalah semua makhluk yang berakal, seperti manusia, jin, malaikat, dan lainnya. Kata ini berasal dari kata `alaamah (tanda), karena menjadi tanda keberadaan pencipta.
- Jawaban Prediksi soal no 2  
Menurut Tafsir Jalalain, "al-'aalamiin" dalam ayat kedua Surah Al-Faatihah berarti "semesta alam", yaitu semua makhluk yang berakal, termasuk jin dan manusia. Lafal 'al-aalamiin' merupakan bentuk jamak dari lafal 'alam', yaitu dengan memakai huruf ya dan huruf nun untuk menekankan makhluk berakal/berilmu atas yang lainnya.



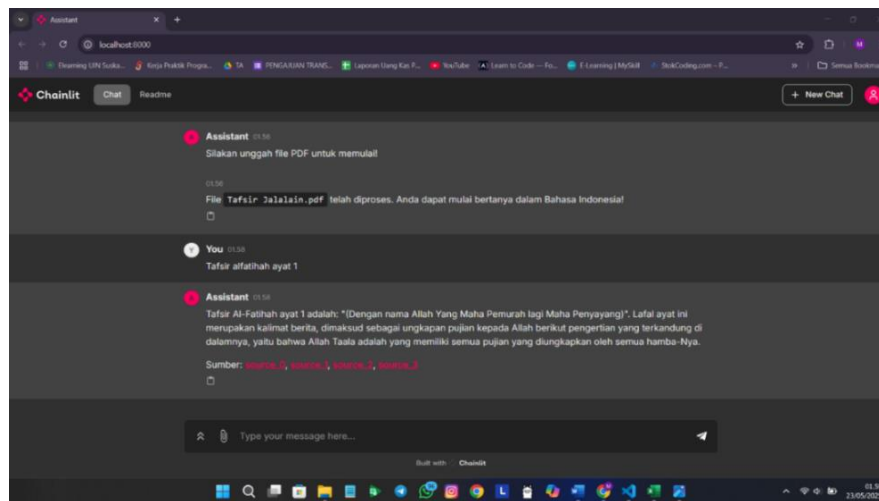
**Tabel 2.** Hasil Evaluasi *BERTScore*

Pertanyaan	<i>BERTScore</i> (%)		
	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	74,53%	89,82%	81,47%
2	67,57%	72,39%	69,89%
3	73,60%	79,18%	76,29%
4	69,57%	77,43%	73,29%
5	77,96%	78,67%	78,32%
6	77,05%	80,76%	78,86%
7	68,98%	73,22%	71,03%
8	68,97%	72,00%	70,45%
9	69,97%	80,23%	74,75%
10	70,16%	77,45%	73,63%
Rata-Rata	71,84%	78,11%	74,80%

Berdasarkan Tabel 2 yang menampilkan hasil evaluasi menggunakan metrik *BERTScore*, dapat disimpulkan bahwa performa sistem dalam menjawab sepuluh pertanyaan menunjukkan hasil yang cukup baik. Evaluasi dilakukan berdasarkan tiga metrik utama, yaitu *Precision*, *Recall*, dan *F1-Score*. Nilai rata-rata yang diperoleh adalah 71,84% untuk *Precision*, 78,11% untuk *Recall*, dan 74,80% untuk *F1-Score*. Hal ini menunjukkan bahwa sistem mampu menghasilkan jawaban yang cukup akurat dan relevan dengan konteks pertanyaan. Pertanyaan ke-6 menunjukkan performa terbaik dengan nilai *F1-Score* sebesar 78,86%, sedangkan performa terendah terdapat pada pertanyaan ke-2 dengan nilai *F1-Score* sebesar 69,89%. Secara keseluruhan, sistem memiliki kemampuan yang cukup seimbang antara ketepatan dalam memberikan jawaban dan kelengkapan informasi yang ditangkap dari pertanyaan.

### 3.3. Tampilan *Chatbot* Tafsir Ketika Digunakan

Berdasarkan implementasi dan pengujian yang telah dilakukan pada Gambar 5, sistem berhasil mengekstraksi teks dari file PDF, memecahnya menjadi bagian-bagian kecil (*chunks*), lalu melakukan *embedding* dan penerjemahan respons ke dalam Bahasa Indonesia. Namun, ditemukan bahwa agar sistem dapat memberikan jawaban yang akurat dan relevan, pertanyaan dari pengguna harus disampaikan secara jelas dan spesifik. Pertanyaan yang terlalu umum atau mengandung kesalahan pengetikan (*typo*) sering menyebabkan sistem kesulitan dalam memahami maksud dan konteks, sehingga jawabannya bisa menjadi kurang tepat.

**Gambar 5.** Tampilan *chatbot* ketika menjawab pertanyaan dari user

Untuk mengatasi hal ini, ke depannya dapat diterapkan *preprocessing* pada *input* pengguna, seperti *spell checking* otomatis, serta antarmuka bantu seperti *autocomplete* atau saran struktur kalimat. Selain itu, pelatihan ulang model dengan data pertanyaan berbahasa Indonesia yang telah dibersihkan dapat meningkatkan akurasi pemahaman. Menurut penelitian oleh Moslem *et al* [25] sistem NLP cenderung gagal ketika input berisi kesalahan linguistik, dan pendekatan berbasis koreksi otomatis terbukti meningkatkan performa pemahaman mesin terhadap bahasa alami.

Jika dibandingkan dengan penelitian Permadi *et al* [5], yang mengembangkan *chatbot* tafsir Al-Azhar berbasis Telegram dan mengevaluasi hasil menggunakan *User Acceptance Test (UAT)*, penelitian ini menunjukkan pendekatan evaluasi yang lebih objektif dengan menggunakan metrik *BERTScore*. Rata-rata *F1-Score* sebesar 74,80% mencerminkan kualitas jawaban yang relevan secara semantik, bukan hanya dinilai



secara manual. Selain itu, penelitian ini mengolah dokumen tafsir langsung dari format PDF, sedangkan penelitian Rizki *et al* [3] menggunakan data yang sudah terstruktur. Hal ini menunjukkan keunggulan sistem dalam menangani sumber data yang kompleks dan tidak selalu terorganisir.

Secara teknis, sistem ini mengintegrasikan tiga komponen utama, yaitu *Chainlit* sebagai antarmuka *web* interaktif, *Groq* sebagai akselerator inferensi untuk mempercepat proses respon, dan *Ollama* sebagai solusi inferensi lokal tanpa bergantung pada layanan *cloud*. Kombinasi ketiganya belum ditemukan pada studi sebelumnya, menjadikan integrasi ini sebagai unsur kebaruan utama dalam penelitian. Hasilnya, sistem mampu memberikan jawaban yang cepat, kontekstual, serta menjaga privasi pengguna. Dengan pendekatan ini, sistem tidak hanya mendukung pembelajaran tafsir berbasis AI, tetapi juga menjadi langkah konkret dalam pengembangan media dakwah digital yang efisien, adaptif, dan mudah diakses.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian, sistem *chatbot* tafsir berbasis *Chainlit*, *Groq*, dan *Ollama* berhasil dikembangkan dan diimplementasikan sesuai dengan tujuan utama penelitian, yaitu menyediakan media pembelajaran tafsir Al-Qur'an yang interaktif, cepat, dan kontekstual terkhusus pada Tafsir *Jalalain*. Sistem mampu memproses dokumen tafsir dalam format PDF melalui tahapan ekstraksi teks, pembagian *chunk*, *embedding*, dan pencocokan semantik menggunakan *vector database Chroma*. Hasil evaluasi menggunakan *BERTScore* terhadap sepuluh pertanyaan menunjukkan performa yang baik dengan rata-rata *Precision* sebesar 71,84%, *Recall* 78,11%, dan *F1-Score* 74,80%, menunjukkan bahwa sistem dapat memahami dan menjawab pertanyaan secara relevan sesuai konteks tafsir. Sistem ini juga menunjukkan efisiensi dari sisi waktu respons dan kemudahan penggunaan berbasis *web*, meskipun terdapat tantangan seperti keterbatasan dalam menangkap maksud inti pertanyaan, terutama yang bersifat umum atau ambigu. Oleh karena itu, untuk pengembangan selanjutnya, disarankan untuk mengintegrasikan teknik seperti *query refinement*, *named entity recognition (NER)*, dan *semantic parsing* untuk meningkatkan pemahaman semantik sistem terhadap pertanyaan pengguna. Selain itu, eksplorasi terhadap model bahasa lokal yang telah di-fine-tune dengan data tafsir, serta dukungan multibahasa (Bahasa Arab dan Indonesia), menjadi langkah strategis untuk memperluas cakupan dan akurasi sistem dalam konteks pembelajaran Islam digital.

#### REFERENSI

- [1] A. S. Prihatinoto, N. Safaat Harahap, M. Irsyad, and I. Iskandar, "Question Answering System Tafseer Ibnu Katsir Using Large Language Models," *Jurnal Informatika & Rekayasa Elektronika*, vol. 7, no. 1, pp. 68–79, Apr. 2024, doi: 10.36595/jire.v7i1.1155.
- [2] I. Fadilatul Kasmar, F. Anwar, and U. N. Padang, "Metode Guru dalam Mengatasi Kesulitan Belajar Alquran Peserta Didik," *An-Nuha: Jurnal Pendidikan Islam*, vol. 1, no. 4, pp. 617–629, Nov. 2021, doi: <https://doi.org/10.24036/annuha.v1i4.140>.
- [3] F. Rizki, A. Sutiyo, N. S. Harahap, S. Agustian, and R. M. Candra, "KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi Question Answering Berbasis Chatbot Telegram Pada Tafsir Al-Jalalain Menggunakan Langchain dan LLM," *Media Online*, vol. 4, no. 5, pp. 2464–2472, Apr. 2024, doi: 10.30865/klik.v4i5.1784.
- [4] O. Topsakal and T. C. Akinci, "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast," in *International Conference on Applied Engineering and Natural Sciences (ICAENS), Vol. 1, No. 1, Halaman 1050–1056*, Jul. 2023. doi: 10.59287/icaens.1127.
- [5] Aji Bayu Permadi, Nazruddin Safaat H, Lestari Handayani, and Yusra, "Implementasi Question Answering System Tafsir Al-Azhar Menggunakan LangChain dan Large Language Model Berbasis Chatbot Telegram," *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, vol. 12, no. 1, pp. 62–69, Apr. 2024, doi: 10.21063/jtif.2024.v12.1.62-69.
- [6] A. Mansurova, A. Nugumanova, and Z. Makhambetova, "Development of a Question Answering Chatbot for Blockchain Domain," *Scientific Journal of Astana IT University*, vol. 15, pp. 27–40, Sep. 2023, doi: 10.37943/15xndz6667.
- [7] A. Pesaru, S. Gill, and A. 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/irjmet.42630.
- [8] F. Soygazi and D. Oguz, "An Analysis of Large Language Models and LangChain in Mathematics Education," in *Proceedings of the 15th International Conference on Educational Technology (ICET '23)*, ACM, Oct. 2023, pp. 92–97. doi: 10.1145/3633598.3633614.
- [9] M. Mulky Mar'arif, N. Safaat Harahap, and M. Affandes, "Sistem Tanya-Jawab Berbasis Chatbot Telegram Tentang Fiqih Kontemporer Menggunakan Langchain Dan LLM," *TEKNIKA*, vol. 19, no. 2, pp. 1–5, Apr. 2025.
- [10] T. Medeiros, M. Medeiros, M. Azevedo, M. Silva, I. Silva, and D. G. Costa, "Analysis of Language-Model-Powered Chatbots for Query Resolution in PDF-Based Automotive Manuals," *Vehicles*, vol. 5, no. 4, pp. 1384–1399, Dec. 2023, doi: 10.3390/vehicles5040076.

- 
- [11] X. Zhong, J. Tang, and A. J. Yepes, "PubLayNet: largest dataset ever for document layout analysis," *Nat Commun*, vol. 8, no. 1, Aug. 2019, doi: 10.1038/s41467-017-00439-1.
- [12] M. Y. Firdaus, "Digitalisasi Khazanah Ilmu Al-Qur'an dan Tafsir di Era Digital: Studi Analisis pada Website Tanwir.id," *Reslaj: Religion Education Social Laa Roiba Journal*, vol. 5, pp. 2710–2716, 2023, doi: 10.47476/as.v5i6.2552.
- [13] W. Grosky and T. Ruas, *Data Science for Software Engineers*, 9th ed. McGraw-Hill Education, 2019.
- [14] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 EMNLP (Systems Demonstrations)*, Nov. 2020, pp. 38–45. doi: 10.18653/v1/2020.emnlp-demos.6.
- [15] N. S. Harahap, A. Saad, and H. Ubaidullah, "Comprehensive Bibliometric Literature Review of Chatbot Research: Trends, Frameworks, and Emerging Applications," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, p. 2025, 2025, [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [16] S. Xiang *et al.*, "HeteroFlow: An Accelerator Programming Model with Decoupled Data Placement for Software-Defined FPGAs," in *FPGA 2022 - Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Association for Computing Machinery, Inc, Feb. 2022, pp. 78–88. doi: 10.1145/3490422.3502369.
- [17] L. Yuan, D.-J. Han, S. Wang, and C. G. Brinton, "Local-Cloud Inference Offloading for LLMs in Multi-Modal, Multi-Task, Multi-Dialogue Settings," *ArXiv*, vol. 1, pp. 1–15, Feb. 2025, doi: 10.48550/arXiv.2502.11007.
- [18] C. Dasanayaka, K. Dandeniya, M. B. Dissanayake, C. Gunasena, and R. Jayasinghe, "Multimodal AI and Large Language Models for Orthopantomography Radiology Report Generation and Q&A," *Applied System Innovation*, vol. 8, no. 2, Apr. 2025, doi: 10.3390/asi8020039.
- [19] K. Hrushikesh Koundinya, A. K. Palakurthi, V. Putnala, and A. K. Kumar, "Smart College Chatbot using ML and Python," in *2020 International Conference on System, Computation, Automation and Networking, ICSCAN 2020*, Institute of Electrical and Electronics Engineers Inc., Jul. 2020. doi: 10.1109/ICSCAN49426.2020.9262426.
- [20] A. Pak, A. Ziyaden, T. Saparov, I. Akhmetov, and A. Gelbukh, "Word Embeddings: A Comprehensive Survey," *Computacion y Sistemas*, vol. 28, no. 4, pp. 2005–2029, 2024, doi: 10.13053/CyS-28-4-5225.
- [21] E. Meiliverani and C. Yefta, "Web-Based Chatbot with Natural Language Processing and Knuth-Morris-Pratt (Case Study: Universitas Internasional Batam)," *Jurnal Sains dan Teknologi*, vol. 11, pp. 132–141, 2022, doi: 10.23887/jst-undiksha.v11i1.
- [22] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, Apr. 2019. doi: 10.48550/arXiv.1904.09675.
- [23] M. Clinciu, A. Eshghi, and H. Hastie, "A Study of Automatic Metrics for the Evaluation of Natural Language Explanations," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, Mar. 2021. doi: 10.18653/v1/2021.eacl-main.202.
- [24] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," in *Proceedings of the Chinese Computational Linguistics (CCL 2019), Lecture Notes in Computer Science*, Springer, May 2019. doi: 10.1007/978-3-030-32381-3\_16.
- [25] Y. Moslem, R. Haque, and A. Way, "Arabisc: Context-Sensitive Neural Spelling Checker," in *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA)*, E. Yang, E. Xun, B. Zhang, and G. Rao, Eds., Suzhou: Association for Computational Linguistics, Feb. 2020, pp. 11–19. doi: <https://doi.org/10.18653/v1/2020.nlpatea-1.2>.