

IMPLEMENTASI *QUESTION ANSWERING SYSTEM* TAFSIR AL-AZHAR MENGGUNAKAN LANGCHAIN DAN LARGE LANGUAGE MODEL BERBASIS CHATBOT TELEGRAM

Aji Bayu Permadi¹⁾, Nazruddin Safaat H²⁾, Lestari Handayani³⁾, Yusra⁴⁾

^{1,2,3,4}Jurusan Teknik Informatika

^{1,2,3,4}Fakultas Sains dan Teknologi

^{1,2,3,4}Universitas Islam Negeri Sultan Syarif Kasim Riau

E-mail : 12050113503@students.uin-suska.ac.id ¹⁾, nazruddin.safaat@uin-suska.ac.id ²⁾,

lestari.handayani@uin-suska.ac.id ³⁾, yusra@uin-suska.ac.id ⁴⁾

Abstract

Tafsir is a main gateway for a Muslim to study and understand the content of the verses in the Quran. One example is Tafsir Al-Azhar. Tafsir Al-Azhar is a commentary authored by Professor Dr. Hamka, which demonstrates how Dr. Hamka connects modern Islamic history with Quranic studies. Tafsir Al-Azhar has a large number of pages, requiring extra effort when searching for information within it. This research aims to create a system capable of receiving questions about Tafsir Al-Azhar in natural language and answering them in user-friendly terms. The technology used in this research includes Langchain and Large Language Models, implemented using a Telegram chatbot. Telegram was chosen for its popularity and user-friendly interface. The Question Answering system was tested using User Acceptance Testing (UAT) and the DeepEval framework. The UAT resulted in an accuracy score of 83.71%, while testing using the DeepEval framework yielded a hallucination score of 41%, contextual precision of 90%, and contextual relevancy of 79%.

Keywords : *Question Answering System, Langchain, Large Language Model , Tafsir Al-Quran , Chatbot*

Intisari

Tafsir merupakan sebuah gerbang utama bagi seorang muslim untuk mempelajari dan memahami isi kandungan dari ayat-ayat yang terdapat di dalam Al-Quran, salah satu contohnya adalah Tafsir Al-Azhar. Tafsir Al-Azhar merupakan tafsir karya Profesor Dr.Hamka yang memperlihatkan bagaimana cara Dr. Hamka menghubungkan antara sejarah islam modern dengan studi Al-Quran. Tafsir Al-Azhar mempunyai halaman yang sangat banyak, sehingga membutuhkan kerja yang lebih ketika kita ingin mencari suatu informasi dari tafsir tersebut. Penelitian ini bertujuan untuk membuat sebuah sistem yang mampu menerima pertanyaan seputar Tafsir Al-Azhar dalam bentuk bahasa alami, serta menjawab pertanyaan tersebut dengan kata-kata yang mudah dipahami oleh pengguna. Adapun teknologi yang akan digunakan pada penelitian ini seperti Langchain dan Large Language Model, untuk pengimplementasiannya menggunakan chatbot telegram, karena telegram merupakan aplikasi yang sudah terkenal dan sudah banyak digunakan oleh banyak orang, dan juga telegram mempunyai interface yang ramah, sehingga tidak membuat pengguna kebingungan ketika menggunakannya. Sistem Question Answering ini diuji menggunakan UAT (Unit Acceptance Test) dan framework DeepEval, hasil dari pengujian UAT berupa skor akurasi sebesar 83.71%, sedangkan hasil dari pengujian menggunakan framework DeepEval, berupa skor hallucination sebesar 41 %, contextual precision sebesar 90 % dan contextual relevancy 79 %.

Kata Kunci : *Question Answering, Langchain, Large Language Model, Tafsir Al-Qur'an, Chatbot*

1. PENDAHULUAN

Perkembangan *Artificial Intelligence* pada saat ini telah berpengaruh kepada banyak hal, sudah banyak teknologi-teknologi yang dibuat menggunakan *Artificial Intelligence* dengan tujuan untuk membantu manusia menjalani kehidupannya [1]. Penelitian yang dilakukan oleh Wahyudi [2] membandingkan antara

manusia dan *Artificial Intelligence* dalam memecahkan masalah. Hasilnya menunjukkan bahwa *Artificial Intelligence* memiliki keunggulan dalam hal analisis, memiliki kompleksitas rendah , dan tingkat ketidakpastian yang minimal. Dalam beberapa kasus, kecerdasan buatan bahkan mendapatkan skor yang lebih tinggi daripada manusia dalam kemampuan analitis. Oleh karena itu, kecerdasan buatan dapat menggantikan

beberapa aktivitas manusia, termasuk interaksi tanya-jawab. Situasi di mana kecerdasan buatan dapat menggantikan manusia terjadi ketika manusia tidak mampu memberikan jawaban karena informasi yang terlalu banyak, seperti dalam konteks tafsir Al-Qur'an.

Tafsir merupakan gerbang utama yang bisa kita gunakan untuk mempelajari dan memahami isi kandungan dari Al-Quran [3]. Salah satu nya adalah Tafsir Al-Azhar, tafsir yang dibuat oleh Profesor Dr. Hamka yang memperlihatkan bagaimana cara Dr. Hamka menghubungkan antara sejarah islam modern dengan studi Al-Qur'an [4]. Tafsir Al-Azhar mempunyai 10 jilid dengan masing-masing jilid memiliki 800 lebih halaman, jumlah halaman yang banyak membuat susah nya melakukan pencarian sebuah tafsir pada tafsir Al-Azhar. Oleh karena itu, diperlukannya sebuah sistem yang mampu melakukan pencarian dengan konsep tanya-jawab , agar pengguna bisa lebih leluasa mencari isi tafsir yang diinginkan .

Question Answering System merupakan sebuah sistem yang digunakan untuk mencari jawaban dari setiap pertanyaan yang diajukan oleh pengguna dalam bentuk bahasa alami [5]. *Question Answering System* menjadi salah satu solusi yang tepat dari permasalahan pencarian yang meminta sistem memberikan informasi yang tepat dan relevan, sehingga pengguna tidak perlu memilah lagi informasi yang didapatkan [6].

Question Answering akan dibuat menggunakan *framework Langchain* dan *Large Language Model*. *Langchain* akan berperan sebagai penghubung antara data eksternal yakni data tafsir al-azhar dengan model bahasa besar [7], sebelum dihubungkan dengan model bahasa, data akan di proses oleh *langchain* dengan cara dipotong-potong, lalu di rubah menjadi *vector* dan disimpan ke dalam *vector store*. *Large Language Model* digunakan untuk memproses hasil jawaban sebelum ditampilkan kepada pengguna, dengan bantuan dari *Large Language Model* sistem mampu menghasilkan jawaban yang lebih relevan dan dapat dipahami oleh manusia [8].

Question Answering akan diimplementasikan ke dalam bentuk *chatbot*. *Chatbot* merupakan sebuah sistem yang mampu merespon pesan dari

pengguna dan memberikan informasi dengan cepat dan efisien [9]. Salah satu *platform* yang menyediakan *chatbot* ialah telegram. Telegram merupakan aplikasi message yang populer, menurut situs *Crunchbase* didapati bahwa telegram telah diunduh sebanyak 21 juta kali dalam 30 hari terakhir [10]. *Bot* yang disediakan oleh telegram mempunyai fungsi untuk menjawab perintah yang diberikan oleh pengguna [11].

Pada penelitian yang dilakukan oleh Made Agus Putra Subali dan teman-teman, yang berjudul *Sistem Question Answering* untuk Bahasa Bali menggunakan metode *Rule-Based* dan *String Similarity*. Metode yang digunakan diawali dengan memberi input pertanyaan, lalu mencari dokumen yang paling relevan berdasarkan pertanyaan yang diberikan dan sistem akan memberikan jawaban berdasarkan autan yang telah ditetapkan dari setiap pertanyaan. Sistem tersebut telah memperoleh hasil pengujian berupa akurasi terkait kebenaran dari jawaban yang diberikan sistem sebesar 40% [12].

Pada penelitian terkait lainnya yang dilakukan oleh Jafar A. Alzubi dan teman-temannya, yang berjudul *COBERT: COVID-19 Question Answering System Using BERT*. Sistem COBERT menggunakan dataset *COVID-19* yang terdiri dari 59 ribu makalah dan sekitar 41 ribu teks lengkap. Hasil dari pengujian sistem COBERT mendapatkan skor *Exact Match* sebesar 80,6/87,3, sistem tidak hanya mengeluarkan jawaban tetapi juga judul dokumen beserta paragraf tempat jawaban ditemukan [13]. Berdasarkan hal tersebut penelitian ini menggunakan *Question Answering System* untuk membantu mencari jawaban seputar pertanyaan tentang isi tafsir Al-Azhar.

2. METODOLOGI

Dalam proses pembuatan *chatbot* telegram seputar tafsir Al-Azhar, penulis melewati berbagai tahapan penelitian yang dimulai dari pengumpulan data, analisa dan perancangan, implementasi dan pengujian. Hasil akhir penelitian berupa *chatbot* telegram yang mampu menjawab pertanyaan seputar isi tafsir Al-Azhar.

2.1. Pengumpulan Data

Pada tahapan awal ini penulis mencari data yang dibutuhkan untuk penelitian. Adapun data yang diperlukan dalam penelitian ini ialah data tafsir Al-Azhar dalam bentuk digital seperti pdf, txt dan lain sebagainya.

2.2. Analisa dan Perancangan

Pada tahap analisa dan perancangan penulis melakukan perancangan diagram alir dari sistem, dengan tujuan untuk memastikan proses-proses yang akan di lakukan oleh sistem berjalan sesuai alur nya mulai dari proses memotong data, meng *embed* data, dan menyimpan hasil *embedding*. Lalu penulis melakukan analisa terhadap model bahasa yang akan digunakan oleh *question answering system*.

2.3. Implementasi

Setelah tahap analisa dan perancangan sistem proses selanjutnya adalah implementasi. Sistem dibuat menggunakan bahasa pemrograman *python* sebagai bahasa utama, dan didukung oleh *framework Langchain*, *Library FAISS*, *Flask* dan *Large Language Model*. Implementasi akan menggunakan penulisan kode program sesuai dengan diagram alir yang telah ditetapkan pada tahapan analisa dan perancangan. Hal ini mencakup penggunaan fitur-fitur yang disediakan oleh *Langchain* serta kemampuan yang dimiliki oleh *Large Language Model* untuk mendukung pembuatan sistem *Question Answering*.

2.4. Pengujian

Pengujian merupakan tahap terakhir didalam penelitian ini, pengujian dilakukan setelah proses implementasi telah selesai. Pengujian yang akan dilakukan pada penelitian ini adalah sebagai berikut :

- a. UAT (*Unit Acceptance Test*)
UAT adalah pengujian yang dilakukan oleh end user dengan tujuan untuk menguji kelayakan sebuah sistem sebelum digunakan secara umum [14].

Pengujian akan dilakukan dengan cara membandingkan jawaban yang

diberikan oleh *chatbot* dengan jawaban dari responden. Responden akan menilai sudah valid atau tidak jawaban yang diberikan oleh bot, adapun responden dari pengujian ini adalah 7 orang mahasiswa jurusan Ilmu Al-Qur'an dan Tafsir dari UIN Suska Riau.

- b. *DeepEval*
DeepEval merupakan sebuah *framework open source* yang digunakan untuk mengevaluasi *Large Language Model* dengan menggunakan bahasa pemrograman *python* [15]. *DeepEval* digunakan untuk mencari nilai *hallucination*, *contextual precision*, dan *contextual relevancy*.

3. HASIL DAN PEMBAHASAN

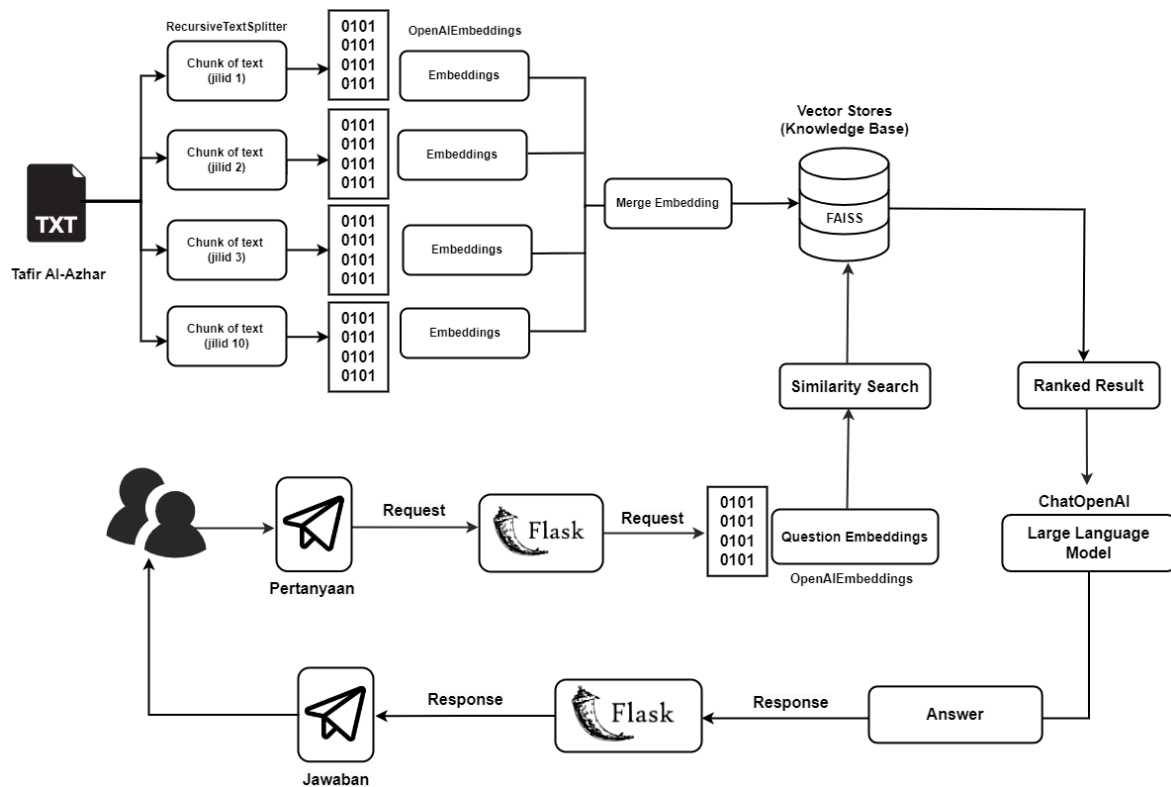
3.1. Pengumpulan Data

Data tafsir Al-Azhar yang digunakan pada penelitian ini didapatkan dari sebuah penerbit yakni Pustaka Nasional PTE LTD Singapura, dalam bentuk pdf dan berjumlah 10 jilid, dengan masing-masing jilid berjumlah 800 halaman lebih.

Data yang awalnya pdf dirubah menjadi txt dengan tujuan memaksimalkan hasil pada proses pemotongan dan *embedding*, lalu dilakukan *text processing* berupa pembersihan data dengan menghilangkan karakter-karakter tidak penting dengan menggunakan *tools* berupa website dengan nama *txtcleaner.net*.

3.2. Analisis dan Perancangan

Pada penelitian ini penulis menggunakan model bahasa dari salah satu platform yakni *OpenAI* dengan nama model "*gpt-3.5-turbo*", dan juga "*text-embedding-ada-002*". Adapun rancangan diagram alir dari *question answering system* berbasis *chatbot* telegram yang akan diterapkan dalam proses implementasi sistem ditunjukkan pada Gambar 2.

Gambar 1. Diagram Alir *Question Answering System*

Pada gambar 1 proses pertama yang dilakukan adalah pemotongan teks (*chunks of text*) pada data, pemotongan teks dilakukan untuk mengubah teks-teks yang panjang menjadi beberapa bagian [16]. Pemotongan teks dilakukan dengan menggunakan salah satu kelas dari *framework langchain* yakni *RecursiveTextSplitter* dengan ukuran pemotongan sebesar 1000 karakter dan pengecekan tumpang tindih karakter sebesar 200 karakter, pengecekan tumpang tindih ini dilakukan untuk memastikan tidak ada informasi yang hilang atau terpotong pada saat proses pemotongan teks.

Selanjutnya adalah proses embedding, hasil potongan-potongan teks dirubah menjadi bentuk *vector* [17]. Proses *embedding* ini dilakukan dengan menggunakan salah satu kelas dari *framework langchain* yakni *OpenAIEmbeddings*, kelas ini mengembed data dengan menggunakan bantuan dari model “*text-embedding-ada-002*”.

Dikarenakan data yang terlalu banyak , ini menyebabkan limit ketika proses *embedding*, jadi penulis membagi proses chunk teks dan embedding menjadi berjilid dari tafsir Al-Azhar.

Setelah itu baru dilakukan penggabungan menggunakan fungsi *merge_from* dari *langchain*. Hasil akhir dari *embeddings* disimpan ke dalam *vector stores* menggunakan *library FAISS* (Facebook AI Similarity Search) dan disebut sebagai *knowledge Base* [18].

Setelah terciptanya *knowledge base*, proses selanjutnya adalah menggunakan *knowledge base* tersebut sebagai database dari sistem *Question Answering*, dan mengaksesnya menggunakan *Application Programming Interfaces* (API) yang dibuat menggunakan *microframework* yakni *Flask* pada *python* [19].

API akan menjadi penghubung antara *knowledge base* hasil *embedding* data tafsir Al-Azhar dengan *chatbot* telegram [20]. *Chatbot* akan mengirim pertanyaan berdasarkan inputan dari user, lalu *flask* akan melakukan *request*, agar *request* dari *flask* dapat diterima dan diproses oleh *knowledge base*, pertanyaan yang dibawa akan *embed* terlebih dahulu menggunakan kelas *OpenAIEmbeddings*.

Hasil *embed* pertanyaan akan diproses menggunakan metode pencarian dari *library Faiss*, yakni *similarity search*, dengan cara

menemukan item yang mirip antara hasil *embed* pertanyaan dengan hasil *embed* di *knowledge base* lalu dibandingkan dari segi karakteristik dan yang memiliki kemiripan tertinggi [21].

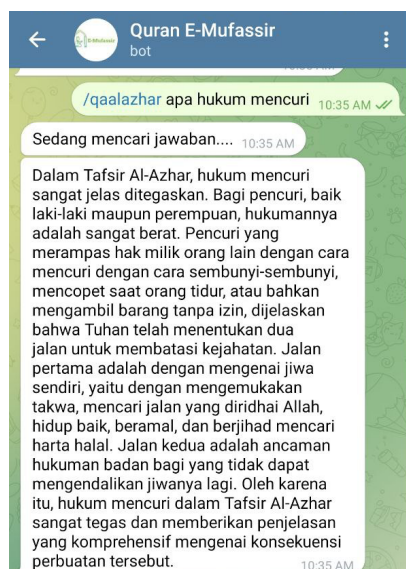
Hasil dari proses pencarian ini, akan dirangking menurut kemiripan dan secara default nya akan di ambil 4 item, dan kemudian dirubah kembali kebentuk teks menggunakan kelas *OpenAIEmbeddings*. Setelah dapat 4 item yang mirip *Large Language Model* akan memproses 4 item ini dengan menggunakan kelas *ChatOpenAI* yang menggunakan model “*gpt-3.5-turbo*”, dengan temperatur sebesar 0.1, temperatur diberikan agar jawaban yang di hasilkan lebih sesuai dengan 4 item yang di berikan, semakin tinggi temperatur yang diberikan maka akan semakin liar jawaban yang di berikan oleh *Large Language Model*.

Hasil jawaban akan diberikan kepada *flask* dalam bentuk *response* yang kemudian ditangkap oleh *chatbot* untuk kemudian ditampilkan kepada pengguna.

3.3. Implementasi

Hasil implementasi *Question Answering System* pada data tafsir al-azhar berbasis *chatbot* telegram adalah sebagai berikut :

- 1) Tampilan ketika bertanya dan *chatbot* menjawab



Gambar 2. Tampilan ketika bertanya

3.4. Pengujian

Pengujian UAT dilakukan dengan memberikan 10 pertanyaan yang berbeda kepada *chatbot* dan

juga responden, adapun yang akan menjadi responden dalam pengujian adalah 7 orang mahasiswa jurusan Ilmu Al-Qur'an dan Tafsir dari UIN Suska Riau. Berikut adalah 10 pertanyaan yang akan ditanyakan :

Tabel 1. Pertanyaan pengujian akurasi jawaban sistem

No	Pertanyaan
1	Mengapa Al-Fatihah dinamai sebagai ummul-kitab ?
2	Mengapa kita harus percaya hari akhirat ?
3	Apa arti dari kata kufur ?
4	Apa saja pekerjaan dan tugas dari malaikat ?
5	Apa itu sifat Taqiyah ?
6	Kenapa orang-orang arab jahiliyah membunuh anak-anak perempuan ?
7	Apa jaminan Allah untuk menghadapi kecemasan ?
8	Sumber hukum pertama di dalam agama islam ?
9	Apa hukum laki-laki yang lemah imannya menikah dengan perempuan ahlul kitab ?
10	Apakah boleh seorang muslim memelihara anjing ?

Perhitungan presentasi hasil UAT :

$$SS = 21 \times 5 = 105$$

$$S = 41 \times 4 = 164$$

$$C = 8 \times 3 = 24$$

$$TS = 0 \times 2 = 0$$

$$STS = 0 \times 1 = 0$$

$$\text{Total} = 293$$

$$X = \text{skor tertinggi} \times (\text{jumlah pertanyaan} \times \text{jumlah responden}) \quad (1)$$

$$X = 5 \times (10 \times 7) = 350$$

Nilai X merupakan nilai pembagi dikarenakan pertanyaan yang di tuju merupakan pertanyaan positif

$$\text{Akurasi} = \frac{\text{total akurasi}}{x} \times 100 \% \quad (2)$$

$$\text{Akurasi} = \frac{293}{350} \times 100 \%$$

$$= 83,71 \%$$

Dari hasil pengujian yang dilakukan, didapati bahwa *chatbot* memiliki akurasi kevalidan sebesar 83.71%.

Pengujian selanjutnya ialah pengujian menggunakan *framework DeepEval*. Nilai yang akan dicari pada pengujian ini adalah :

1. *Hallucination*

Nilai yang digunakan untuk mengukur seberapa banyak *Large Language Model* menghasilkan teks yang tidak sesuai dengan konteks.

2. *Contextual Precision*

Nilai yang digunakan untuk mengukur seberapa akurat *Large Language Model* menghasilkan teks yang relevan dengan konteks yang diberikan.

3. *Contextual Relevancy*

Nilai yang digunakan untuk mengukur seberapa relevan teks yang dihasilkan *Large Language Model* dengan konteks yang diberikan.

Berikut adalah hasil dari pengujian menggunakan pengkodean *DeepEval* :

Tabel 2. Hasil Pengujian *Hallucination*

	<i>Hallucination</i>
Pertanyaan 1	0.66
Pertanyaan 2	0.0
Pertanyaan 3	0.25
Pertanyaan 4	0.25
Pertanyaan 5	0.6
Pertanyaan 6	0.0
Pertanyaan 7	0.75
Pertanyaan 8	0.4
Pertanyaan 9	0.75
Pertanyaan 10	0.44
Rata-rata	0.41

Pada tabel 2, nilai diperoleh dari pengkodean pada gambar 3.

```

80 test_case = LLMTTestCase(
81     input=question,
82     actual_output=output,
83     context=listContext
84 )
85 metric = HallucinationMetric(threshold=0.5)
86 metric.measure(test_case)

```

Gambar 3. *Code Hallucination*

Pada gambar 3 nilai *input* didapat dari pertanyaan, lalu *actual output* didapat dari hasil jawaban, dan *context* diambil dari hasil *similarity search*, ke-3 argumen tersebut di proses menggunakan kelas *HallucinationMetric* dari *DeepEval*.

Tabel 3. Hasil Pengujian *Contextual Precision*

	<i>Contextual Precision</i>
Pertanyaan 1	1.0
Pertanyaan 2	1.0
Pertanyaan 3	0.91
Pertanyaan 4	0.80
Pertanyaan 5	0.33
Pertanyaan 6	1.0
Pertanyaan 7	1.0
Pertanyaan 8	1.0
Pertanyaan 9	1.0
Pertanyaan 10	1.0
Rata-rata	0.90

Pada tabel 3, nilai diperoleh dari pengkodean pada gambar 4.

```

56 metric = ContextualPrecisionMetric(
57     threshold=0.7,
58     model="gpt-4",
59     include_reason=True
60 )
61 test_case = LLMTTestCase(
62     input=question,
63     actual_output=output,
64     expected_output=output,
65     retrieval_context=listContext
66 )
67
68 metric.measure(test_case)

```

Gambar 4. *Code Contextual Precision*

Pada gambar 4 nilai *input* didapat dari pertanyaan, *actual output* dari jawaban, *expected output* dari data yang digunakan, dan *retrieval output* digunakan dari hasil *similarity search*, ke-4 argumen diproses menggunakan kelas *ContextualPrecisionMetric* dari *DeepEval*.

Tabel 4. Hasil Pengujian *Contextual Relevancy*

	<i>Contextual Relevancy</i>
Pertanyaan 1	0.80
Pertanyaan 2	0.80
Pertanyaan 3	0.56

Pertanyaan 4	1.96
Pertanyaan 5	0.65
Pertanyaan 6	0.95
Pertanyaan 7	0.86
Pertanyaan 8	0.57
Pertanyaan 9	0.96
Pertanyaan 10	0.80
Rata-rata	0.79

Pada tabel 4, nilai diperoleh dari pengkodean pada gambar 5.

```

58 metric = ContextualRelevancyMetric(
59     threshold=0.7,
60     model="gpt-4",
61     include_reason=True
62 )
63 test_case = LLMTestCase(
64     input=question,
65     actual_output=output,
66     retrieval_context=listContext
67 )
68 metric.measure(test_case)

```

Gambar 5. Code Contextual Relevancy

Pada gambar 5 nilai *input* didapatkan dari pertanyaan, *actual output* dari jawaban, dan *retrieval output* diambil dari hasil *similarity search*, ke-3 argumen di proses menggunakan kelas *ContextualRelevancyMetric* dari *DeepEval*.

4. KESIMPULAN DAN SARAN

Berdasarkan dari penelitian yang sudah dilakukan menggunakan *Langchain* dan *Large Language Model*, didapati bahwa penulis telah berhasil melakukan implementasi *question answering system* pada data tafsir Al-Azhar berbasis *chatbot* telegram dengan hasil pengujian UAT sebesar 83.71%, dan pengujian menggunakan *DeepEval* mendapatkan nilai *hallucination* sebesar 0.41 yang menunjukkan kategori bagus, karena berada di bawah ambang batas 0.5. Nilai lainnya seperti *contextual precision*, dan *contextual relevancy* yang masing-masing mencapai nilai di atas 0.5, juga menunjukkan kualitas baik. Namun terdapat kekurangan pada *Question Answering System* yang dibuat, yaitu ada beberapa pertanyaan yang tidak bisa dijawab oleh sistem, meskipun konteks yang ditanyakan terdapat pada tafsir Al-Azhar. Kekurangan pada penelitian ini mungkin bisa dianalisis kembali untuk pengembangan penelitian selanjutnya. Saran untuk penelitian selanjutnya yaitu melakukan pemrosesan pada pertanyaan, agar saat pencarian sistem mampu mencari pokok dari pertanyaan.

DAFTAR PUSTAKA

- [1] R. J. Sahputra and A. Muzakir, "Penerapan AI Melalui Pendekatan Heuristik Semilaritas Pada Game Edukasi Anak Usia Dini," *J. Pengemb. Sist. Inf. dan Inform.*, vol. 1, no. 4, pp. 209–219, 2021, doi: 10.47747/jpsii.v1i4.547.
- [2] T. Wahyudi, "Studi Kasus Pengembangan dan Penggunaan Artificial Intelligence (AI) Sebagai Penunjang Kegiatan Masyarakat Indonesia," *Indones. J. Softw. Eng.*, vol. 9, no. 1, pp. 28–32, 2023, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/ijse>
- [3] A. Latif, "Spektrum Historis Tafsir Al-Qur'an Di Indonesia," *TAJIDID J. Ilmu Ushuluddin*, vol. 18, no. 1, pp. 105–124, 2020, doi: 10.30631/tjd.v18i1.97.
- [4] O. Yulianda, "Istidraj Menurut Hamka dalam tafsir Al-Azhar," vol. 10, p. 6, 2021.
- [5] A. Dhandapani and V. Vadivel, "Question Answering System over Semantic Web," *IEEE Access*, vol. 9, pp. 46900–46910, 2021, doi: 10.1109/ACCESS.2021.3067942.
- [6] R. A. Yunmar and I. W. W. Wisesa, "Pengembangan Mobile-Based Question Answering System Mobile-Based Question Answering System Development With Ontology Based Knowledge," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 693–700, 2020, doi: 10.25126/jtiik.202072255.
- [7] O. Topsakal and T. C. Akinci, "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast," *Int. Conf. Appl. Eng. Nat. Sci.*, vol. 1, no. 1, pp. 1050–1056, 2023, doi: 10.59287/icaens.1127.
- [8] B. D. Laraswati, "Large Language Models, Cikal Bakal Lahirnya Chat GPT." Accessed: Nov. 08, 2023. [Online]. Available: <https://blog.algorit.ma/large-language-models/>
- [9] A. Hariansyah, E. Haerani, and M. Affandes, "Implementation of Telegram Chatbot as Information Service of

- Madani Hospital Pekanbaru,” vol. 11, no. 3, pp. 188–198, 2023.
- [10] “Telegram Messenger - Tech Stack, Apps, Patents & Trademarks.” Accessed: Feb. 09, 2024. [Online]. Available: <https://www.crunchbase.com/organization/telegram-messenger/technology>
- [11] Heri Khariono, Rizky Parlika, Haidar Ananta Kusuma, and Dimas Arif Setyawan, “Pemanfaatan Bot Telegram Sebagai E-Learning Ujian Berbasis File,” *J. Inform. Polinema*, vol. 7, no. 4, pp. 65–72, 2021, doi: 10.33795/jip.v7i4.696.
- [12] M. A. P. Subali and P. Wijaya, “Sistem Question Answering untuk Bahasa Bali menggunakan Metode Rule-Based dan String Similarity,” *Techno.Com*, vol. 20, no. 2, pp. 300–308, 2021, doi: 10.33633/tc.v20i2.4390.
- [13] J. A. Alzubi, R. Jain, A. Singh, P. Parwekar, and M. Gupta, “COBERT: COVID-19 Question Answering System Using BERT,” *Arab. J. Sci. Eng.*, vol. 48, no. 8, pp. 11003–11013, 2023, doi: 10.1007/s13369-021-05810-5.
- [14] P. Pujianto, M. Mujito, D. Prabowo, and B. H. Prasetyo, “Pemilihan Warga Penerima Bantuan Program Keluarga Harapan (PKH) Menggunakan Metode Simple Additive Weighting (SAW) dan User Acceptance Testing (UAT),” *J. Inform. Univ. Pamulang*, vol. 5, no. 3, p. 379, 2020, doi: 10.32493/informatika.v5i3.6671.
- [15] “DeepEval - The Open-Source LLM Evaluation Framework.” Accessed: Mar. 12, 2024. [Online]. Available: <https://docs.confident-ai.com/>
- [16] “Text Splitters | Langchain.” Accessed: Feb. 26, 2024. [Online]. Available: https://python.langchain.com/docs/modules/data_connection/document_transformers/
- [17] K. Pandya and B. V. Mahavidyalaya, “Automating Customer Service using LangChain,” *Comput. Lang. (cs.CL); Comput. Soc. (cs.CY); Mach. Learn.*, vol. 1, pp. 28–31, 2023, [Online]. Available: <https://arxiv.org/abs/2310.05421>
- [18] M. Douze *et al.*, “The Faiss library,” 2024, [Online]. Available: <http://arxiv.org/abs/2401.08281>
- [19] Graciela Fausten Novindri and P. Ocsa Nugraha Saian, “Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada Pt Xyz Berbasis Website,” *J. Mnemon.*, vol. 5, no. 2, pp. 81–85, 2022, doi: 10.36040/mnemonic.v5i2.4670.
- [20] B. P. Putra and Y. A. Susetyo, “Implementasi Api Master Store Menggunakan Flask, Rest Dan Orm Di Pt Xyz,” *Sistemasi*, vol. 9, no. 3, p. 543, 2020, doi: 10.32520/stmsi.v9i3.899.
- [21] M. Schwarz, K. Chapman, and B. Häussler, “Multilingual Medical Entity Recognition and Cross-lingual Zero-Shot Linking with Facebook AI Similarity Search,” *CEUR Workshop Proc.*, vol. 3202, 2022.