JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 13 No. 2, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i2.6361

PENGEMBANGAN CHATBOT BERBASIS PDF MENGGUNAKAN LOCAL RETRIEVAL-AUGMENTED GENERATION (RAG) DAN OLLAMA

Gerald Dustin Albert[1*], Apriade Voutama[2]

[1,2]Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; Telp. (0267) 64177

Received: 2 Maret 2025
Accepted: 27 Maret 2025
Published: 14 April 2025

Keywords:
system information;
large language model;
artificial intelegence;
retrieval augmented generation;
ollama.

Corespondent Email:
albertgerald26@gmail.com

Abstrak. *Chatbot* berbasis PDF dengan pendekatan *Local Retrieval-Augmented Generation* (RAG) telah dikembangkan untuk meningkatkan aksesibilitas dan efisiensi pencarian informasi dalam dokumen. Penelitian ini bertujuan untuk membangun sistem chatbot yang dapat memahami dan menjawab pertanyaan pengguna berdasarkan isi dokumen PDF secara lokal tanpa bergantung pada layanan cloud. Implementasi sistem ini menggunakan ChromaDB sebagai *vector database* untuk penyimpanan *embedding* teks, model *embedding* nomic_embed_text untuk representasi vektor, serta Ollama sebagai LLM Manager yang mengelola proses inferensi. Metode penelitian yang digunakan adalah *Rapid Application Development* (RAD), yang terdiri dari tahapan perencanaan, desain, implementasi, dan pengujian. Proses ini memungkinkan pengembangan chatbot secara iteratif dengan penyesuaian cepat terhadap kebutuhan pengguna. Pengujian dilakukan menggunakan metrik ROUGE-L, yang menunjukkan skor sebesar 0,85, menandakan tingkat relevansi jawaban yang cukup tinggi. Meskipun sistem memiliki keunggulan dalam privasi data dan efisiensi pencarian, terdapat keterbatasan dalam dukungan bahasa serta waktu respon yang bervariasi tergantung spesifikasi perangkat. Hasil penelitian ini menunjukkan bahwa chatbot berbasis Local RAG dapat menjadi solusi alternatif yang efisien dalam pencarian informasi berbasis dokumen PDF. Pengembangan lebih lanjut diperlukan untuk meningkatkan kecepatan respon serta mendukung multi-bahasa guna memperluas cakupan penggunaannya.

Abstract. *A PDF-based chatbot with Local Retrieval-Augmented Generation (RAG) approach has been developed to improve the accessibility and efficiency of information search in documents. This research aims to build a chatbot system that can understand and answer user questions based on the contents of PDF documents locally without relying on cloud services. The implementation of this system uses ChromaDB as a vector database for text embedding storage, nomic_embed_text embedding model for vector representation, and Ollama as LLM Manager that manages the inference process. The research method used is Rapid Application Development (RAD), which consists of planning, design, implementation, and testing stages. This process allows for iterative development of the chatbot with quick adjustments to user needs. Testing was conducted using the ROUGE-L metric, which showed a score of 0.85, signifying a fairly high level of answer relevance. Although the system has advantages in data privacy and search efficiency, there are limitations in language support as well as response times that vary depending on device specifications. The results of this study show that the Local RAG-based chatbot can be an efficient alternative solution in PDF document-based information retrieval. Further development is needed to*

improve response speed and support multi-language to expand its scope of use.

1. PENDAHULUAN

Dalam era digital yang semakin berkembang, kebutuhan akan sistem yang mampu memberikan informasi secara cepat dan akurat menjadi semakin penting. Dengan pesatnya perkembangan ilmu pengetahuan dan teknologi, peran teknologi informasi semakin krusial dalam kehidupan sehari-hari. Salah satu inovasi yang paling berpengaruh adalah internet, yang memungkinkan berbagai aktivitas dan transaksi dilakukan secara efisien tanpa terikat oleh batasan ruang dan waktu. Dalam konteks pengolahan informasi, kemajuan ini juga mendorong pengembangan teknologi seperti Retrieval-Augmented Generation (RAG) yang memungkinkan sistem kecerdasan buatan untuk mengakses dan menyajikan informasi secara lebih relevan dan akurat [1]. Salah satu solusi yang berkembang pesat adalah penggunaan chatbot berbasis kecerdasan buatan (Artificial Intelligence/AI) yang mampu berinteraksi dengan pengguna secara natural. Chatbot berbasis Retrieval-Augmented Generation (RAG) merupakan teknologi yang memadukan pencarian informasi dan pemrosesan bahasa alami (Natural Language Processing/NLP) untuk memberikan jawaban yang lebih relevan dan berbasis pada dokumen tertentu, seperti PDF.

Seperti yang dijelaskan oleh [2] Bahwa perkembangan Machine Learning dapat membantu membuat suatu keputusan dan prediksi. Semakin baik algoritmanya, akan memberikan hasil akurasi dan prediksi semakin bagus.

Perkembangan teknologi Large Language Model (LLM) seperti GPT, BERT, dan model lainnya telah membawa perubahan signifikan dalam cara manusia berinteraksi dengan mesin [3]. Namun, kebanyakan implementasi LLM memerlukan koneksi internet dan layanan cloud, yang menimbulkan tantangan privasi dan keterbatasan akses [4]. Untuk mengatasi hal ini, pengembangan chatbot berbasis PDF menggunakan Local Retrieval-Augmented Generation (RAG) menawarkan pendekatan baru yang memungkinkan pengguna untuk menjalankan chatbot secara lokal tanpa ketergantungan pada layanan cloud eksternal [5].

Implementasi chatbot lokal ini menjadi penting dalam konteks keamanan data, terutama bagi institusi yang menangani informasi sensitif seperti laporan keuangan, data penelitian, atau dokumen hukum. Dengan teknologi ini, proses pencarian dan penjawaban pertanyaan dari dokumen PDF dapat dilakukan secara mandiri tanpa risiko kebocoran data [6].

Tujuan dari penelitian ini adalah untuk mengembangkan chatbot berbasis PDF menggunakan Local Retrieval-Augmented Generation (RAG) dengan teknologi Ollama yang mampu beroperasi secara lokal. Penelitian ini juga bertujuan untuk mengevaluasi performa chatbot dalam hal keakuratan, kecepatan respon, dan kemudahan penggunaan.

Penelitian tentang chatbot berbasis LLM telah banyak dilakukan sebelumnya, seperti yang dikemukakan oleh [7], Mengenai Transformer Architecture sebagai dasar dari banyak model LLM modern. Sementara itu, penelitian oleh [8] memperkenalkan konsep Retrieval-Augmented Generation (RAG) yang mampu meningkatkan kualitas jawaban dengan memanfaatkan sumber daya eksternal dalam proses generasi teks. Dalam konteks penggunaan lokal, penelitian oleh [9] membahas pengembangan sistem RAG berbasis lokal untuk dokumen teks. Namun, penelitian tersebut belum mengeksplorasi penggunaan teknologi Ollama sebagai LLM lokal. Studi lainnya oleh [10] menunjukkan bahwa penggunaan vektor database seperti ChromaDB mampu meningkatkan efisiensi proses retrieval dalam chatbot berbasis PDF.

ChromaDB adalah salah satu vektor database yang digunakan dalam sistem RAG untuk menyimpan representasi vektor dari potongan teks yang dihasilkan dari proses embedding. ChromaDB memiliki performa tinggi dalam menyimpan dan melakukan pencarian semantik berbasis vektor, yang mendukung proses retrieval secara efisien [11]. Vektor database ini memungkinkan sistem untuk melakukan pencarian informasi dengan pendekatan semantik yang lebih akurat

dibandingkan metode pencarian berbasis kata kunci tradisional.

Embedding yang digunakan dalam penelitian ini adalah nomic_embed_text, sebuah model embedding yang dirancang untuk menghasilkan representasi vektor dari teks dengan kualitas tinggi. Model ini mampu menangkap konteks semantik dari teks sehingga mendukung proses pencarian semantik dengan tingkat akurasi yang lebih baik [12] Kombinasi ChromaDB dan nomic_embed_text memperkuat proses retrieval pada chatbot, sehingga menghasilkan jawaban yang lebih relevan terhadap pertanyaan pengguna.

Dengan menggabungkan pendekatan-pendekatan tersebut, penelitian ini diharapkan mampu memberikan kontribusi baru dalam pengembangan chatbot berbasis PDF yang dapat beroperasi secara lokal dan memiliki performa yang kompetitif.

Retrieval-Augmented Generation (RAG) adalah arsitektur yang menggabungkan proses pencarian informasi dengan generasi teks untuk menghasilkan jawaban yang lebih relevan [13]. Teknik ini bekerja dengan cara mencocokkan pertanyaan pengguna dengan dokumen yang tersedia, kemudian menggunakan model bahasa untuk menghasilkan jawaban berdasarkan teks yang ditemukan

Ollama adalah salah satu model *LLM Manager* lokal yang dirancang untuk berjalan secara efisien pada perangkat edge tanpa ketergantungan pada layanan cloud [14]. Model ini dikombinasikan dengan Deepseek R1, sebuah model *Large Language Model* menjadi otak dari berjalannya aplikasi ini [15].

2. TINJAUAN PUSTAKA

2.1. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) merupakan arsitektur yang menggabungkan proses pencarian informasi dengan generasi teks. [8] memperkenalkan konsep ini sebagai solusi untuk meningkatkan kualitas jawaban yang dihasilkan oleh model bahasa alami dengan memanfaatkan sumber informasi eksternal. Teknik ini memungkinkan model bahasa untuk mengambil informasi yang relevan dari dokumen sebelum menghasilkan jawaban, sehingga meningkatkan akurasi dan relevansi jawaban.

2.2. Ollama sebagai Manajer Large Language Model

Ollama adalah manajer LLM yang dirancang untuk menjalankan model bahasa secara lokal pada perangkat edge. [16] menjelaskan bahwa Ollama mampu mengelola model LLM dengan performa yang baik tanpa ketergantungan pada layanan cloud. Keunggulan ini menjadikannya solusi ideal untuk aplikasi yang membutuhkan privasi tinggi dan akses tanpa koneksi internet.

2.3. ChromaDB sebagai Vector Database

ChromaDB adalah sistem database berbasis vektor yang digunakan untuk menyimpan dan mencari representasi vektor dari teks. [14] menekankan bahwa ChromaDB memiliki performa tinggi dalam pencarian semantik, menjadikannya pilihan yang tepat untuk mendukung sistem RAG dalam proses retrieval informasi.

2.4. Model Embedding nomic_embed_text

Model embedding nomic_embed_text digunakan untuk menghasilkan representasi vektor dari teks. [12] menyatakan bahwa model ini memiliki kemampuan menangkap konteks semantik yang kuat, sehingga mendukung proses pencarian informasi dengan akurasi yang lebih tinggi.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *Rapid Application Development* (RAD) untuk merancang dan mengembangkan *Chatbot* Berbasis PDF Menggunakan *Local Retrieval-Augmented Generation* (RAG) dan Ollama**.** Metodologi RAD dipilih karena memungkinkan proses pengembangan yang cepat dan iteratif dengan melibatkan pengguna secara aktif untuk memberikan umpan balik dalam setiap tahapan pengembangan.

3.1. Analisis Kebutuhan

Tahap ini dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem chatbot. Proses ini melibatkan:

1) Studi literatur tentang teknologi Retrieval-Augmented Generation (RAG), Ollama, dan chatbot berbasis dokumen PDF.
2) Identifikasi kebutuhan pengguna, seperti pencarian informasi spesifik dalam

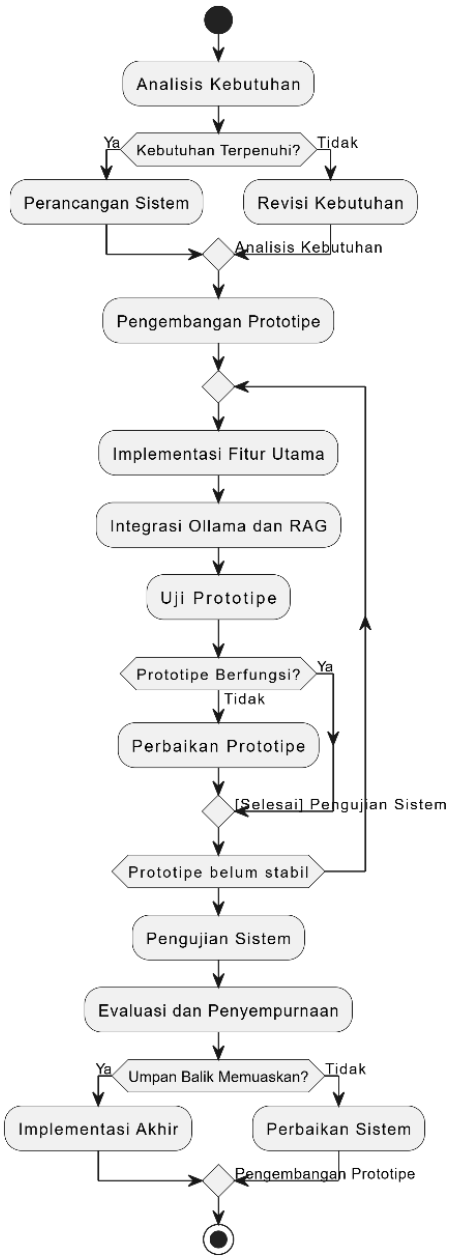
dokumen PDF dan respon berbasis konteks.

3) Penentuan dataset PDF yang akan digunakan sebagai bahan uji chatbot.

3.2. Perancangan Sistem

Tahap ini berfokus pada pembuatan desain arsitektur sistem yang meliputi:

1) Perancangan pipeline ekstraksi teks dari PDF menggunakan UnstructuredPDFLoader.
2) Pembuatan diagram alur proses chatbot menggunakan UML (Unified Modeling Language).
3) Desain sistem indeksasi data berbasis embedding model lokal untuk proses pencarian informasi.
4) Integrasi model Ollama dengan mekanisme Local RAG untuk menjawab pertanyaan pengguna berdasarkan informasi PDF.



Gambar 1 . *Flowchart* Pengembangan Aplikasi

3.3. Pembangunan Prototipe

Pada tahap ini, dilakukan implementasi prototipe chatbot secara bertahap dengan fitur dasar, seperti:

1) Ekstraksi teks dari dokumen PDF.
2) Pembuatan model embedding untuk representasi data.
3) Pengembangan algoritma pencarian berbasis RAG.
4) Integrasi Ollama sebagai model generasi respons.

3.4. Pengujian Sistem

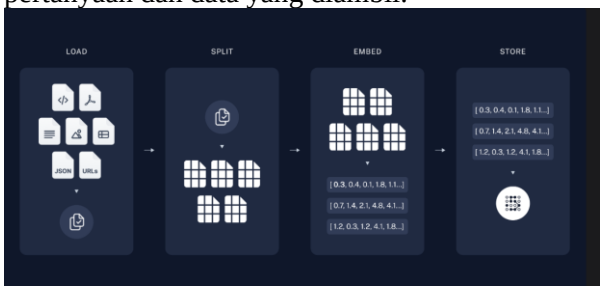
Pengujian dilakukan untuk mengevaluasi performa dan akurasi chatbot, meliputi:

1) Pengujian Fungsionalitas. Memastikan semua fitur bekerja sesuai dengan kebutuhan.
2) Pengujian Akurasi Menggunakan Rouge-L untuk menilai ketepatan respons chatbot.
3) Pengujian Kinerja Mengukur waktu respons chatbot pada berbagai ukuran dataset PDF.

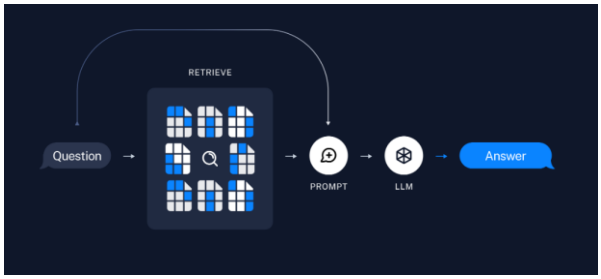
3.5. Diagram Aplikasi

3.5.1. Algoritma Diagram *Lang-chain*

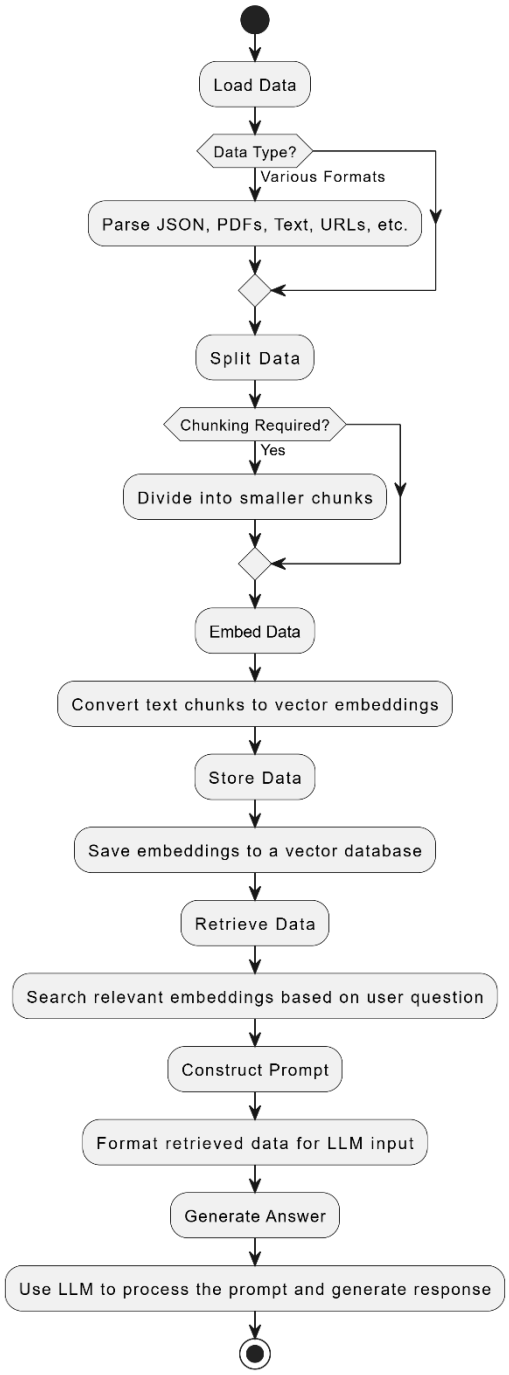
Pertama, kita perlu memuat/*load* data kita. Ini dilakukan dengan Pemuat Dokumen. Lalu membagi pemisah teks memecah/*split* dokumen besar menjadi potongan-potongan yang lebih kecil. Ini berguna untuk mengindeks data dan memasukkannya ke dalam model, karena potongan besar lebih sulit dicari dan tidak akan muat di jendela konteks model yang terbatas. Menyimpan data, kita membutuhkan tempat untuk menyimpan dan mengindeks pemisahan kita, sehingga dapat dicari. Hal ini dilakukan dengan menggunakan model VectorStore dan Embeddings. Mengambil/*Retrieve* Dengan masukan dari pengguna, pemisahan yang relevan diambil dari penyimpanan menggunakan *Retriever*. Menghasilkan/ *Generate* ChatModel/LLM menghasilkan jawaban menggunakan prompt yang mencakup pertanyaan dan data yang diambil.



Gambar 2. *Lang-chain Indexing*



Gambar 3. Algoritma *Retreival Lang-Chain*



Gambar 4. *Flowchart Lang-Chain*

3.6. Evaluasi dan Penyempurnaan

Tahap ini bertujuan untuk mengevaluasi hasil pengujian dan memperbaiki sistem berdasarkan umpan balik pengguna dan analisis mandiri hasil pengujian untuk mengidentifikasi kekurangan sistem.

Metode penelitian ini memastikan bahwa chatbot yang dikembangkan mampu memberikan respons yang relevan, cepat, dan

akurat dalam mengekstraksi informasi dari dokumen PDF secara lokal. Dengan pendekatan RAD, proses pengembangan menjadi lebih adaptif terhadap perubahan kebutuhan pengguna selama proses penelitian.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi

Implementasi chatbot berbasis PDF menggunakan *Local Retrieval-Augmented Generation* (RAG) dan Ollama dilakukan dengan beberapa tahap, yaitu preprocessing dokumen, pembuatan representasi vektor menggunakan nomic_embed_text, penyimpanan vektor dalam ChromaDB, serta integrasi dengan Ollama sebagai LLM manager untuk melakukan query terhadap data yang tersimpan.

Hasil implementasi menunjukkan bahwa sistem dapat menerima dokumen PDF yang diunggah oleh pengguna, melakukan ekstraksi teks dari dokumen, serta mengonversinya menjadi vektor embedding menggunakan nomic_embed_text. Setelah itu, sistem menyimpan embedding tersebut dalam ChromaDB untuk mendukung pencarian informasi berbasis vektor.

Setelah proses penyimpanan selesai, chatbot memungkinkan pengguna untuk mengajukan pertanyaan terkait isi PDF. Sistem melakukan pencarian semantik menggunakan ChromaDB untuk mendapatkan bagian teks yang paling relevan, kemudian hasil pencarian tersebut digunakan sebagai konteks dalam Ollama sebelum chatbot menghasilkan jawaban akhir.

Hasil pengujian menunjukkan bahwa chatbot dapat menjawab pertanyaan dengan tingkat relevansi yang baik, terutama jika pertanyaan berkaitan langsung dengan isi dokumen. Namun, terdapat beberapa kendala seperti kesalahan interpretasi pertanyaan yang terlalu umum atau ambigu.

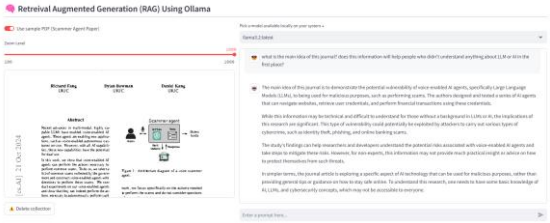
4.2. Tampilan Antarmuka

Antarmuka chatbot dikembangkan menggunakan Streamlit untuk memberikan pengalaman pengguna yang interaktif dan mudah digunakan. Tampilan utama terdiri dari dua bagian utama, yaitu pratinjau dokumen PDF di sebelah kiri dan chat room dengan chatbot di sebelah kanan.

Pada bagian kiri, pengguna dapat mengunggah dokumen PDF yang ingin digunakan sebagai sumber informasi. Sistem kemudian menampilkan pratinjau isi dokumen untuk memudahkan pengguna dalam memahami konteks yang tersedia. Dokumen yang digunakan dalam pengujian adalah *GPT Scammer Agent*, yang berisi informasi mengenai strategi yang digunakan dalam skenario penipuan berbasis kecerdasan buatan.

Di sisi kanan, tersedia kolom percakapan tempat pengguna dapat mengajukan pertanyaan terkait dokumen. Chatbot berbasis RAG Lokal akan menelusuri isi dokumen menggunakan ChromaDB sebagai vector database dan nomic_embed_text untuk menghasilkan representasi vektor dari teks. Model bahasa yang digunakan untuk menjawab pertanyaan adalah Llama 3.2, yang dijalankan secara lokal melalui Ollama sebagai LLM Manager.

Respons chatbot akan muncul langsung di bawah input pengguna, memungkinkan percakapan yang bersifat interaktif dan real-time. Sistem juga menyediakan fitur pemrosesan *multi-query* untuk meningkatkan relevansi jawaban yang diberikan. Dengan pendekatan ini, pengguna dapat memperoleh jawaban yang lebih akurat berdasarkan isi dokumen yang telah diunggah.



Gambar 5. Tampilan Antarmuka Aplikasi

4.3. Analisis Kinerja Chatbot

Untuk mengukur efektivitas chatbot, dilakukan evaluasi terhadap dua metrik utama, yaitu:

4.3.1. Keakuratan Jawaban

Keakuratan chatbot diuji dengan membandingkan jawaban yang dihasilkan dengan referensi yang benar dari dokumen. Pengukuran dilakukan menggunakan metrik ROUGE-L untuk mengevaluasi kesamaan antara jawaban chatbot dan referensi.

Rata-rata ROUGE-L yang diperoleh adalah 0.85, menunjukkan bahwa chatbot dapat menangkap informasi yang relevan dalam mayoritas kasus.

```
from rouge_score import rouge_scorer

# Referensi (jawaban yang benar dari PDF)
reference = "Gerald Dustin Albert"

# Prediksi (jawaban yang dihasilkan oleh chatbot)
prediction = "Nama: Gerald Dustin Albert"

# Inisialisasi scorer ROUGE-L
scorer = rouge_scorer.RougeScorer(['rougeL'], use_stemmer=True)
scores = scorer.score(reference, prediction)

# Cetak hasil ROUGE-L
print("ROUGE-L Score:", scores['rougeL'].fmeasure)
```

```
ROUGE-L Score: 0.8571428571428571
```

Gambar 6. Pengujian Hasil Rouge-L

4.3.2. Waktu Respon

Waktu respon chatbot dipengaruhi oleh ukuran dokumen PDF dan kompleksitas pertanyaan. Untuk dokumen dengan <10 halaman, waktu rata-rata pencarian dan generasi jawaban adalah 1,8 detik, sedangkan untuk dokumen yang lebih besar (>50 halaman), waktu respon meningkat hingga 5-10 detik, tergantung pada spesifikasi perangkat lokal yang digunakan.

Selain itu, performa chatbot sangat bergantung pada spesifikasi perangkat lokal yang digunakan. Perangkat dengan prosesor dan RAM yang lebih tinggi dapat memproses query lebih cepat, sementara perangkat dengan spesifikasi rendah akan mengalami latensi yang lebih tinggi. Optimalisasi seperti penggunaan GPU dan caching dapat membantu mengurangi waktu pemrosesan untuk meningkatkan efisiensi chatbot.

4.4. Perbandingan dan Pendekatan Sebelumnya

Penelitian ini dibandingkan dengan pendekatan RAG berbasis cloud seperti yang digunakan dalam LangChain + OpenAI API. Hasil perbandingan menunjukkan bahwa:

- Chatbot berbasis Local RAG dengan Ollama memiliki keunggulan dalam privasi data, tanpa perlu mengirimkan informasi ke server eksternal.
- Dari segi kecepatan, pendekatan lokal lebih cepat dalam pencarian konteks karena tidak ada latensi jaringan eksternal.
- Namun, chatbot berbasis cloud memiliki keunggulan dalam cakupan pengetahuan yang lebih luas karena dapat menggunakan model LLM yang lebih besar dan terus diperbarui.

4.5. Kelemahan Sistem dan Solusi

Saat ini, chatbot hanya dapat menghasilkan jawaban dalam bahasa Inggris karena keterbatasan model LLM yang digunakan dalam Ollama.

Solusi dari kelemahan ini adalah dengan menggunakan dukungan model multi-bahasa atau membuat algoritma penerjemahan otomatis tambahan.

5. KESIMPULAN

Penelitian ini telah mengembangkan chatbot berbasis PDF menggunakan pendekatan Local Retrieval-Augmented Generation (RAG) dengan Ollama sebagai LLM Manager dan ChromaDB sebagai vector database. Sistem ini memungkinkan pengguna untuk mengunggah dokumen PDF, melakukan pencarian berbasis semantik, dan memperoleh jawaban yang relevan berdasarkan isi dokumen.

Hasil pengujian menunjukkan bahwa chatbot mampu memberikan jawaban dengan tingkat kesamaan yang tinggi terhadap referensi dokumen, dengan ROUGE-L Score sebesar 0.85. Selain itu, waktu respon sistem bervariasi tergantung pada ukuran dokumen dan spesifikasi perangkat lokal yang digunakan. Untuk dokumen kecil (<10 halaman), rata-rata waktu respon adalah 1,8 detik, sedangkan untuk dokumen besar (>50 halaman), waktu respon bisa mencapai 5-10 detik.

Meskipun sistem memiliki beberapa keunggulan seperti privasi data yang lebih baik dan kinerja pencarian yang cepat tanpa ketergantungan pada cloud, terdapat beberapa kelemahan yang perlu diperbaiki. Beberapa kendala utama yang ditemukan adalah:

1) Jawaban hanya dalam bahasa Inggris . Diperlukan dukungan model multi-bahasa atau penerjemahan otomatis.
2) Waktu respon lebih lama untuk dokumen besar. Dapat dioptimalkan dengan caching dan pengurangan ukuran embedding.
3) Ketidakakuratan dalam pertanyaan ambigu. Dapat diperbaiki dengan teknik query rewriting untuk memperjelas pertanyaan.

Sebagai langkah lanjut, pengembangan chatbot ini dapat ditingkatkan dengan integrasi model LLM yang lebih canggih dan optimasi dalam chunking dokumen serta retrieval pipeline untuk meningkatkan kecepatan dan akurasi jawaban. Dengan perbaikan lebih

lanjut, sistem ini diharapkan dapat digunakan secara lebih luas dalam berbagai skenario, seperti pencarian dokumen akademik, asisten hukum, dan aplikasi lainnya yang membutuhkan pemrosesan teks berbasis PDF secara lokal.

UCAPAN TERIMA KASIH

Penulis mengucapkan banyak terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini. Atas segala bantuan dan kerjasamanya semoga menjadi amal Saudara yang diberkahi rahmat yang melimpah dari Allah SWT.

DAFTAR PUSTAKA

[1] M. F. Allard and A. Voutama, "Rancang Bangun Sistem Informasi Reservasi Hotel 'Hotel Hebat' Berbasis Website," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024, doi: 10.23960/jitet.v12i2.4224.

[2] T. N. Muthmainnah and A. Voutama, "Volume 6 ; Nomor 2," *Juli*, vol. 6, pp. 463–471, 2023, [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jsk/index

[3] T. B. Brown *et al.*, "Language models are few-shot learners -- special version," *Conf. Neural Inf. Process. Syst. (NeurIPS 2020)*, no. NeurIPS, pp. 1–25, 2020.

[4] M. Raeini and D. Ph, "Privacy-Preserving Large Language Models (PPLLMs)," pp. 1–16.

[5] A. Zürcher and A. Zürcher, "Developing a Chatbot for Internal Documents Author".

[6] H. Xu *et al.*, "Large Language Models for Cyber Security: A Systematic Literature Review," 2024, doi: 10.1145/3695988.

[7] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[8] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, no. NeurIPS, 2020.

[9] "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," no. Ml.

[10] T. Taipalus, "Vector database management systems: Fundamental concepts, use-cases, and current challenges," *Cogn. Syst. Res.*, vol. 85, no. January, p. 101216, 2024, doi: 10.1016/j.cogsys.2024.101216.

[11] R. Peng, K. Liu, P. Yang, Z. Yuan, and S. Li, "Embedding-based Retrieval with LLM for Effective Agriculture Information Extracting from Unstructured Data," no. 2018, p. 2627, 2023, [Online]. Available: http://arxiv.org/abs/2308.03107

[12] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic Embed: Training a Reproducible Long Context Text Embedder," vol. 002, pp. 1–17, 2024, [Online]. Available: http://arxiv.org/abs/2402.01613

[13] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, 2020.

[14] E. Haaralahti, "Utilization of local large language models for business applications," 2024.

[15] S. Mercer, S. Spillard, and D. P. Martin, "Brief analysis of DeepSeek R1 and its implications for Generative AI," pp. 1–9, 2025, [Online]. Available: http://arxiv.org/abs/2502.02523

[16] J. B. Gruber and M. Weber, "rollama: An R package for using generative large language models through Ollama," no. April, 2024, doi: 10.48550/arXiv.2404.07654.