



Pengembangan Chatbot Perpustakaan Berbasis Retrieval-Augmented Generation (RAG) pada Telegram

Elrica M. I. Tutu¹, Benny Pinontoan², Edwin Tenda³, Christian A. J. Soewoeh⁴, Mahardika I. Takaendengan⁵,
 Stephano C. W. Ngangi⁶

^{1*,2,3,4,5,6} Program Studi Sistem Informasi, Universitas Sam Ratulangi, Manado, Indonesia

*elricatutu106@student.unsrat.ac.id

ABSTRACT

The library of Sam Ratulangi University (UNSRAT) has a large collection of books and academic resources; however, users still experience difficulties in obtaining book availability information quickly and efficiently. This study aims to develop and evaluate a Telegram-based chatbot using a Retrieval-Augmented Generation (RAG) approach integrated with the GPT-4.1-mini Large Language Model (LLM). The system was developed using the Waterfall method and implemented through the n8n workflow automation platform by integrating Telegram Bot, MySQL, and Pinecone as a vector database. The chatbot applies a Text-to-SQL RAG mechanism, where user questions are converted into embeddings, matched with database context, and transformed into SQL queries limited to SELECT operations. System evaluation was conducted using Black Box Testing, User Acceptance Testing (UAT), and RAG evaluation metrics consisting of Answer Relevancy and Faithfulness. The results show that the chatbot successfully performs its main functions and achieved a UAT score of 82.3%, while Answer Relevancy and Faithfulness obtained scores of 100%. The developed system is capable of providing relevant and interactive information regarding library collections.

Keywords: Chatbot, Retrieval-Augmented Generation (RAG), Large Language Model (LLM), Telegram Bot, Library

ABSTRAK

Perpustakaan Universitas Sam Ratulangi (UNSRAT) memiliki koleksi buku dan sumber informasi akademik yang cukup besar, namun pengguna masih mengalami kesulitan dalam memperoleh informasi ketersediaan buku secara cepat dan efisien. Penelitian ini bertujuan mengembangkan dan mengevaluasi chatbot berbasis Telegram menggunakan pendekatan Retrieval-Augmented Generation (RAG) yang terintegrasi dengan Large Language Model (LLM) GPT-4.1-mini. Sistem dikembangkan menggunakan metode Waterfall dan diimplementasikan melalui platform *workflow automation* n8n dengan integrasi Telegram Bot, MySQL, dan Pinecone sebagai *vector database*. Chatbot menerapkan mekanisme *Text-to-SQL* RAG, di mana pertanyaan pengguna diubah menjadi *embedding*, dicocokkan dengan konteks database, kemudian diubah menjadi *query* SQL yang dibatasi hanya pada operasi SELECT. Evaluasi sistem dilakukan menggunakan *Black Box Testing*, *User Acceptance Testing* (UAT), serta evaluasi *Answer Relevancy* dan *Faithfulness*. Hasil penelitian menunjukkan bahwa chatbot berhasil menjalankan seluruh fungsi utama sistem dan memperoleh nilai UAT sebesar 82,3%, sementara evaluasi *Answer Relevancy* dan *Faithfulness* memperoleh nilai 100%. Sistem yang dikembangkan mampu memberikan informasi koleksi perpustakaan secara relevan dan interaktif.

Kata Kunci : Chatbot, Retrieval-Augmented Generation (RAG), Large Language Model (LLM), Telegram Bot, Perpustakaan

INFORMASI ARTIKEL

Submit
9, April, 2026

Diterima
10, Mei, 2026

Publish Online
30, Mei, 2026

PENDAHULUAN

Perpustakaan perguruan tinggi memiliki peran penting sebagai pusat sumber informasi dalam mendukung kegiatan Pendidikan dan penelitian. Seiring dengan perkembangan teknologi informasi, perpustakaan dituntut untuk menyediakan layanan yang lebih cepat dan efisien untuk diakses oleh pengguna. Namun, dalam praktiknya, proses pencarian informasi koleksi buku masih sering menghadapi berbagai kendala, terutama dalam hal aksesibilitas dan kecepatan layanan [1].

Perpustakaan UNSRAT memiliki banyak koleksi, mulai dari buku teks, karya ilmiah, hingga e-book internasional. Meskipun demikian, pengguna sering mengalami kesulitan dalam mengetahui ketersediaan buku. Proses pencarian masih dilakukan secara manual dengan menelusuri rak atau bertanya langsung kepada petugas, sehingga kurang efisien dan memerlukan waktu yang relatif lama. Selain itu, petugas perpustakaan juga sering menerima pertanyaan yang berulang dari pengguna, yang berdampak pada meningkatnya beban kerja dan menurunnya efektivitas layanan.

Perkembangan teknologi kecerdasan buatan telah mendorong munculnya berbagai solusi untuk meningkatkan layanan informasi, khususnya chatbot berbasis LLM, memberikan solusi untuk meningkatkan layanan informasi. Chatbot memungkinkan pengguna berinteraksi menggunakan Bahasa alami dan memperoleh jawaban secara langsung [2]. Namun, penggunaan LLM secara langsung masih memiliki keterbatasan, seperti potensi menghasilkan informasi yang tidak akurat (*hallucination*) serta keterbatasan dalam mengakses data spesifik. [3], [4].

Untuk mengatasi keterbatasan tersebut, pendekatan RAG dapat digunakan. Metode ini menggabungkan proses pencarian informasi dari sumber data eksternal dengan kemampuan generatif LLM dalam menyusun jawaban [5], sehingga dapat meningkatkan akurasi dan relevansi jawaban yang dihasilkan [6], [7]. Secara umum, arsitektur RAG terdiri dari tiga komponen utama, yaitu retriever, augmenter, dan generator, yang bekerja secara berurutan untuk menghasilkan respons yang sesuai dengan kebutuhan pengguna [3]

Beberapa penelitian sebelumnya telah menerapkan chatbot berbasis LLM dan RAG dalam berbagai bidang. Penelitian oleh Lewis et al. [7] menunjukkan bahwa integrasi chatbot dengan Telegram mampu meningkatkan aksesibilitas informasi secara *real-time*. Selain itu, Penelitian Samudra et al. [8] menyatakan bahwa penerapan RAG dapat meningkatkan kualitas jawaban dengan mengurangi kesalahan informasi yang dihasilkan oleh model. Hasil penelitian tersebut menunjukkan bahwa penggunaan chatbot berbasis RAG memiliki potensi besar dalam meningkatkan layanan informasi.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan chatbot berbasis Telegram menggunakan pendekatan RAG yang terintegrasi dengan database perpustakaan UNSRAT dan didukung oleh LLM. Sistem ini diharapkan dapat membantu pengguna dalam mencari informasi koleksi buku secara lebih cepat, mudah, dan interaktif.

TINJAUAN PUSTAKA

Retrieval Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) merupakan pendekatan yang menggabungkan proses pencarian informasi dengan kemampuan generatif *Large Language Model* (LLM) untuk menghasilkan jawaban yang lebih relevan dan akurat [7]. Pada

arsitektur RAG, sistem terlebih dahulu melakukan *retrieval* terhadap sumber pengetahuan eksternal sebelum menghasilkan respons. Pendekatan ini mampu mengurangi *hallucination* serta meningkatkan kualitas jawaban yang dihasilkan model [3].

Large Language Model (LLM)

Large Language Model (LLM) adalah model kecerdasan buatan yang dilatih menggunakan data teks dalam jumlah besar untuk memahami dan menghasilkan bahasa alami [5]. Pada penelitian ini digunakan GPT-4.1-mini dari OpenAI karena memiliki kemampuan pemrosesan bahasa yang baik serta respons yang cepat dalam percakapan.

Telegram Bot

Telegram Bot merupakan layanan otomatis pada platform Telegram yang memungkinkan pengguna berinteraksi melalui pesan instan. Telegram menyediakan *Bot API* yang bersifat terbuka sehingga dapat digunakan untuk pengembangan chatbot pada berbagai bidang layanan informasi [9]. Dalam penelitian ini, Telegram digunakan sebagai antarmuka utama chatbot perpustakaan.

Vector Database

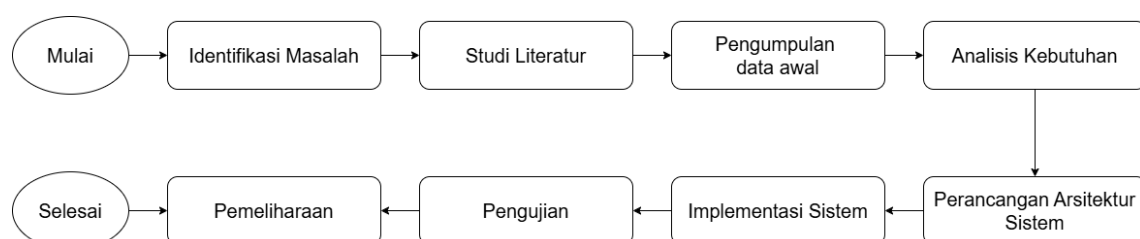
Vector database digunakan untuk menyimpan *embedding* dalam bentuk vektor numerik sehingga memungkinkan pencarian berbasis *semantic similarity* [10]. Penelitian ini menggunakan Pinecone sebagai *vector database* untuk mendukung proses *retrieval* konteks yang relevan pada sistem RAG.

n8n Workflow Automation

n8n merupakan platform *workflow automation* yang digunakan untuk menghubungkan berbagai layanan melalui API secara otomatis [11]. Pada penelitian ini n8n digunakan sebagai orchestration engine untuk menghubungkan Telegram Bot, OpenAI API, Pinecone, dan MySQL dalam satu *workflow* otomatis.

METODE PENELITIAN

Penelitian ini dilakukan untuk mengembangkan sistem chatbot berbasis Telegram yang mampu membantu pengguna dalam mencari informasi koleksi buku secara cepat dan interaktif. Pendekatan yang digunakan adalah pengembangan sistem dengan memanfaatkan metode RAG yang mengintegrasikan proses pencarian informasi dan kemampuan generatif LLM. Proses penelitian dilakukan melalui beberapa tahapan yang ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Identifikasi Masalah

Tahap identifikasi masalah dilakukan melalui observasi dan wawancara di lingkungan Perpustakaan UNSRAT. Hasil identifikasi menunjukkan bahwa proses pencarian informasi koleksi buku masih dilakukan secara manual sehingga kurang efisien dan membutuhkan waktu yang relatif lama. Selain itu, pengguna juga mengalami kesulitan dalam memperoleh informasi terkait ketersediaan dan lokasi buku secara cepat.

Studi Literatur

Tahap studi literatur dilakukan dengan mengkaji berbagai penelitian terkait chatbot, LLM, serta metode RAG sebagai pendekatan dalam meningkatkan kualitas jawaban sistem. Studi ini juga mencakup penelitian terkait implementasi chatbot pada sistem informasi perpustakaan.

Penelitian yang dilakukan oleh Hidayat, Wijaya, dan Dwiyanaputra pada tahun 2025 [12] mengembangkan chatbot berbasis Telegram untuk layanan sistem informasi mahasiswa menggunakan pendekatan RAG. Sistem dibangun melalui tahapan survei kebutuhan, *preprocessing* data, Pembangunan indeks berbasis vector, konfigurasi model, hingga integrasi dengan Telegram. Hasil penelitian menunjukkan bahwa chatbot mampu memberikan jawaban dengan tingkat akurasi tinggi serta meningkatkan aksesibilitas informasi secara *real-time*.

Penelitian yang dilakukan oleh Lubis dkk pada tahun 2025 [2] mengembangkan sistem tanya jawab berbasis chatbot Telegram dengan memanfaatkan LLM dan framework Langchain. Sistem menggunakan Teknik chunking dan *embedding* untuk memproses data berbasis dokumen hukum. Evaluasi sistem dilakukan menggunakan metrik BERTScore dan ROUGE yang menunjukkan hasil cukup baik dalam hal relevansi dan kualitas jawaban. Penelitian ini membuktikan bahwa penggunaan LLM pada chatbot mampu memberikan respons yang informatif dan interaktif.

Pengumpulan Data

Tahap pengumpulan data dilakukan melalui observasi dan wawancara di lingkungan perpustakaan. Observasi di perpustakaan untuk memahami proses pencarian dan peminjaman koleksi buku secara langsung. Selain itu, wawancara dilakukan dengan petugas dan mahasiswa untuk mengidentifikasi kendala yang sering dihadapi dalam proses pencarian informasi koleksi buku. Data yang diperoleh pada tahap ini digunakan sebagai dasar dalam mengidentifikasi kebutuhan sistem yang akan dikembangkan.

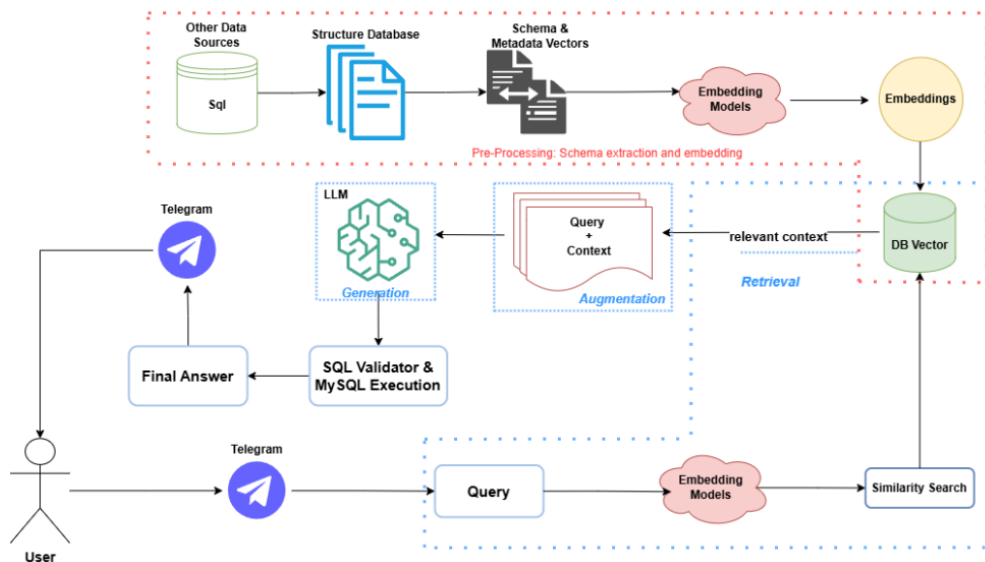
Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan berdasarkan data yang diperoleh dari hasil observasi dan wawancara. Pada tahap ini dilakukan identifikasi terhadap kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional meliputi kemampuan sistem dalam menerima pertanyaan pengguna, melakukan pencarian informasi koleksi buku, serta memberikan jawaban dalam bahasa alami. Sedangkan kebutuhan non-fungsional mencakup aspek kemudahan penggunaan, kecepatan respons, dan keakuratan informasi yang diberikan. Hasil dari tahap ini digunakan sebagai dasar dalam perancangan dan pengembangan sistem chatbot berbasis RAG.

Perancangan Arsitektur Sistem

Tahap perancangan arsitektur sistem dilakukan dengan menggunakan pendekatan RAG, yaitu metode yang menggabungkan proses pencarian informasi dari sumber eksternal dengan kemampuan generatif LLM untuk menghasilkan jawaban yang lebih akurat dan relevan [3], [5], [7]. Arsitektur sistem terdiri dari beberapa komponen utama, yaitu Telegram Bot sebagai antarmuka pengguna yang memanfaatkan Telegram *Bot API* yang bersifat gratis dan terbuka [9], platform n8n sebagai *workflow automation*, database MySQL sebagai media penyimpanan data koleksi, serta Pinecone sebagai *vector database* yang digunakan untuk proses *retrieval* berbasis *embedding*.

Arsitektur sistem *chatbot* yang akan dikembangkan, yang menunjukkan alur data dari pengguna hingga respons akhir, digambarkan pada Gambar 2 di bawah ini.



Gambar 2. Diagram Arsitektur Sistem Chatbot

Arsitektur sistem dibagi menjadi dua fase utama, yaitu fase pembangunan basis pengetahuan (*ingestion*) dan fase interaksi pengguna (*runtime*). Pertama, fase pembangunan basis pengetahuan merupakan tahap awal untuk membangun sumber informasi sistem. Pada tahap ini, struktur database MySQL yang terdiri dari tabel koleksi, peminjaman, dan pengembalian diekstraksi dan direpresentasikan dalam bentuk teks terstruktur yang memuat nama tabel, atribut, serta relasi antar tabel. Representasi tersebut kemudian diubah menjadi *embedding* untuk menangkap makna semantik dan disimpan dalam *vector database* menggunakan Pinecone guna mendukung proses pencarian berbasis kemiripan [10].

Selanjutnya, fase interaksi pengguna berlangsung secara *real-time* ketika pengguna berinteraksi melalui Telegram. Pertanyaan pengguna dikonversi menjadi *embedding* dan digunakan untuk melakukan *similarity search* pada *vector database* untuk memperoleh konteks yang relevan. Konteks tersebut kemudian digabungkan dengan pertanyaan pengguna untuk membentuk *prompt* yang dikirimkan ke LLM guna menghasilkan *query SQL* yang sesuai dengan struktur database [3]. *Prompt* tersebut dikirimkan ke LLM untuk menghasilkan *query SQL* yang sesuai dengan struktur database.

Query SQL yang dihasilkan dieksekusi pada database MySQL untuk memperoleh data yang relevan. Hasil tersebut kemudian diproses kembali oleh LLM untuk disusun

menjadi jawaban dalam bahasa alami yang faktual dan relevan, kemudian dikirimkan kepada pengguna melalui Telegram.

Implementasi Sistem

Tahap implementasi dilakukan dengan mengintegrasikan keseluruhan komponen sistem dalam satu *workflow* menggunakan platform n8n. Sistem memproses pertanyaan pengguna secara otomatis mulai dari proses *retrieval*, pembentukan *query* SQL, eksekusi *query* pada database MySQL, hingga penyusunan jawaban dalam bahasa natural. Untuk menjaga keamanan sistem, *query* yang dihasilkan dibatasi hanya pada perintah SELECT sehingga tidak dapat melakukan manipulasi data pada database. Implementasi sistem dilakukan menggunakan beberapa *node* utama pada n8n, yaitu Telegram *Trigger Node* untuk menerima pesan pengguna, OpenAI *Node* untuk *embedding* dan *generation*, Pinecone *Node* untuk *similarity search*, MySQL *Node* untuk eksekusi *query*, Code *Node* untuk validasi *query* SQL, IF *Node* untuk pengkondisian alur *workflow*.

Pengujian Sistem

Pengujian dilakukan melalui tiga pendekatan, yaitu Black Box Testing, UAT, serta evaluasi teknis terhadap sistem RAG. Black Box Testing untuk memastikan bahwa setiap fitur pada chatbot dapat berjalan sesuai dengan yang diharapkan berdasarkan input dan output yang dihasilkan tanpa memperhatikan proses internal sistem [13]. UAT untuk mengukur tingkat penerimaan pengguna terhadap sistem yang dikembangkan [14], UAT dilakukan menggunakan kuesioner Skala Likert terhadap 20 responden. Selain pengujian dari sisi pengguna, dilakukan evaluasi terhadap kualitas jawaban sistem berbasis RAG. Evaluasi difokuskan pada tahap *generation*, yaitu proses ketika sistem menghasilkan jawaban berdasarkan data yang diperoleh. Pengukuran dilakukan menggunakan dua metrik utama, yaitu Answer Relevancy dan Faithfulness [8].

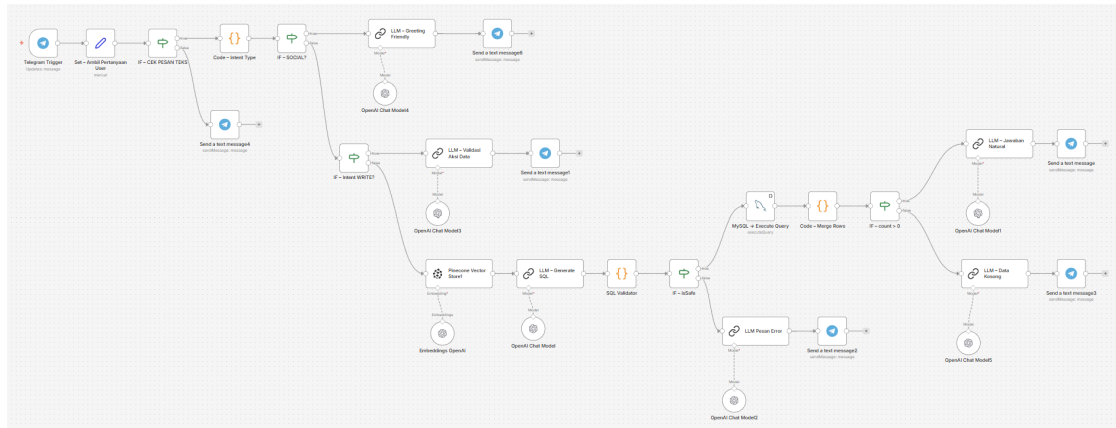
Pemeliharaan Sistem

Setelah sistem dijalankan, dilakukan tahap pemeliharaan untuk memastikan sistem tetap berfungsi dengan baik. Pemeliharaan meliputi perbaikan kesalahan atau bug yang mungkin belum terdeteksi pada tahap sebelumnya.

HASIL DAN PEMBAHASAN

Implementasi Alur Proses RAG

Alur kerja sistem diimplementasikan dalam bentuk *workflow* pada platform n8n yang mengatur proses mulai dari penerimaan pertanyaan pengguna, validasi input, proses *retrieval*, hingga *generation* jawaban. Alur *workflow* sistem pada n8n dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Workflow RAG

Alur proses RAG diimplementasikan menggunakan platform dalam bentuk *workflow* yang terdiri dari beberapa *node* yang saling terhubung. Setiap *node* memiliki fungsi spesifik dalam memproses pertanyaan pengguna hingga menghasilkan jawaban akhir.

Proses dimulai dari penerimaan input pengguna melalui Telegram, kemudian dilakukan tahap *preprocessing* dan klasifikasi intent untuk menentukan apakah pertanyaan termasuk dalam kategori pencarian informasi (*read*) atau bukan. Jika pertanyaan sesuai, sistem akan melanjutkan ke tahap *embedding query* dan proses *retrieval* menggunakan *vector database* Pinecone untuk memperoleh konteks yang relevan.

Selanjutnya, sistem membentuk *prompt* yang menggabungkan pertanyaan pengguna dengan konteks hasil *retrieval*. *Prompt* tersebut dikirimkan ke LLM) untuk menghasilkan *query* SQL. *Query* yang dihasilkan kemudian divalidasi untuk memastikan hanya perintah *SELECT* yang dapat dijalankan, sehingga mencegah manipulasi data.

Setelah itu, *query* dieksekusi pada database MySQL untuk memperoleh data yang relevan. Hasil eksekusi kemudian diproses kembali oleh LLM untuk disusun menjadi jawaban dalam bahasa alami sebelum dikirimkan kembali kepada pengguna melalui Telegram. Jenis *Node* yang digunakan dapat dilihat pada Tabel 1 berikut.

Tabel 1. Node pada Workflow RAG dan Fungsinya

No	Nama Node	Fungsi
1	Telegram Trigger	Menerima input pertanyaan dari pengguna melalui Telegram
2	Set/Preprocessing	Membersihkan dan mempersiapkan teks pertanyaan pengguna
3	Intent Classification	Mengklasifikasikan jenis pertanyaan (<i>read/non-read</i>)
4	If Node (Percabangan)	Menentukan apakah pertanyaan akan diproses atau ditolak
5	Embedding Query	Mengubah pertanyaan menjadi <i>vector embedding</i>
6	Pinecone (Vector DB)	Melakukan <i>similarity search</i> untuk mengambil konteks yang relevan
7	Prompt Builder	Menggabungkan pertanyaan dan konteks menjadi <i>prompt</i>
8	LLM (Text-to-SQL)	Menghasilkan <i>query</i> SQL berdasarkan <i>prompt</i>
9	Validation Node	Memastikan <i>query</i> hanya berupa <i>SELECT (read-only)</i>
10	MySQL Node	Menjalankan <i>query</i> SQL ke database
11	LLM (Natural Language)	Mengubah hasil <i>query</i> menjadi jawaban bahasa alami
12	Telegram Response	Mengirimkan jawaban ke pengguna

Pada *workflow* yang dikembangkan, terdapat beberapa percabangan (*branching*)

yang diimplementasikan menggunakan *node IF* untuk mengatur alur proses sistem secara dinamis berdasarkan kondisi tertentu. Percabangan pertama digunakan untuk melakukan klasifikasi intent terhadap pertanyaan pengguna. Pada tahap ini, sistem membedakan apakah pertanyaan termasuk dalam kategori pencarian informasi (*read*) atau permintaan lain di luar cakupan sistem. Jika pertanyaan termasuk kategori *read*, maka sistem akan melanjutkan proses ke tahap *embedding* dan *retrieval*. Sebaliknya, jika tidak sesuai, sistem akan langsung memberikan respons penolakan kepada pengguna.

Percabangan kedua digunakan pada tahap validasi *query* SQL yang dihasilkan oleh model. Sistem akan memeriksa apakah *query* yang dihasilkan hanya berupa perintah *SELECT*. Jika *query* sesuai dengan aturan tersebut, maka *query* akan dieksekusi pada database. Namun, jika terdeteksi adanya perintah manipulasi data seperti *INSERT*, *UPDATE*, atau *DELETE*, maka sistem akan menolak eksekusi *query* tersebut untuk menjaga keamanan data.

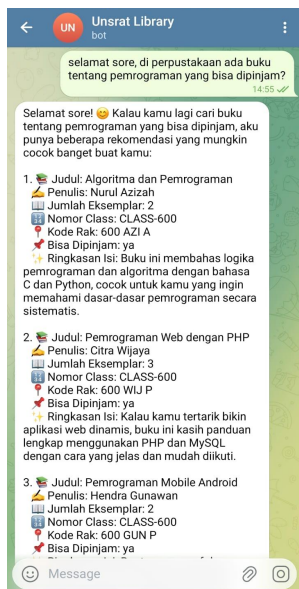
Percabangan ketiga digunakan untuk menangani hasil *retrieval* dari *vector database*. Jika sistem berhasil menemukan konteks yang relevan, maka proses akan dilanjutkan ke tahap pembentukan *prompt* dan *generation*. Namun, jika tidak ditemukan konteks yang sesuai, sistem akan memberikan respons bahwa data tidak tersedia atau tidak ditemukan dalam database.

Percabangan keempat digunakan setelah eksekusi *query* database. Jika hasil *query* mengandung data, maka hasil tersebut akan diproses oleh model untuk disusun menjadi jawaban dalam bahasa alami. Sebaliknya, jika hasil *query* kosong, sistem akan memberikan respons bahwa informasi yang dicari tidak tersedia.

Percabangan terakhir digunakan untuk menangani kondisi error atau input yang tidak valid. Jika terjadi kesalahan dalam proses, seperti format pertanyaan yang tidak sesuai atau kegagalan sistem, maka chatbot akan memberikan pesan yang informatif kepada pengguna agar dapat melakukan perbaikan pada input yang diberikan.

Implementasi Antarmuka Telegram Bot

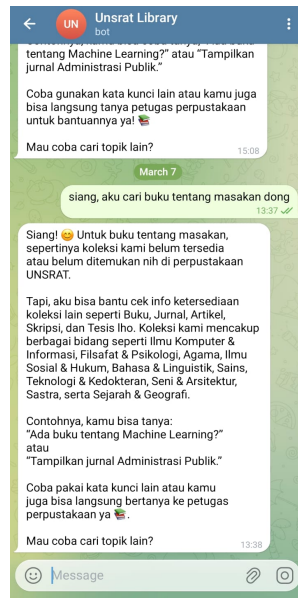
Gambar 4 dibawah ini menampilkan contoh interaksi ketika pengguna mengajukan pertanyaan mengenai ketersediaan buku dengan topik tertentu.



Gambar 4. Tampilan hasil pencarian buku pada telegram

Pada Gambar 4, chatbot mampu menampilkan informasi buku secara lengkap, meliputi judul, penulis, lokasi rak, jumlah eksemplar, status ketersediaan, serta ringkasan isi buku. Hal ini menunjukkan bahwa sistem mampu menghasilkan jawaban yang informatif dan berbasis data.

Selanjutnya, sistem juga dirancang untuk memberikan respons yang sesuai ketika pengguna mencari buku yang tidak tersedia pada database perpustakaan. Gambar 5 dibawah ini menampilkan contoh dengan pertanyaan yang tidak ada dalam database.



Gambar 5. Tampilan respons chatbot ketika buku tidak ditemukan

Berdasarkan Gambar 5, sistem memberikan respons yang sesuai ketika data tidak ditemukan, disertai saran pertanyaan lain. Hal ini menunjukkan bahwa chatbot tetap mampu memberikan pengalaman pengguna yang baik meskipun informasi tidak tersedia.

Pengujian Fungsional Sistem

Pengujian fungsional menggunakan metode black-box testing untuk memastikan seluruh fitur chatbot berjalan sesuai kebutuhan sistem. Berdasarkan hasil pengujian, seluruh skenario seperti pencarian buku, validasi keamanan *query*, dan penolakan manipulasi data berhasil dijalankan dengan baik.

Tabel 2. Pengujian fungsional sistem

No	Skenario Pengujian	Input Pengguna	Output yang diharapkan	Hasil Sistem	Status
1	Perintah inisialisasi	/start	Menampilkan pesan sambutan	sesuai perancangan	Berhasil
2	pencarian buku tersedia	“Buku pemrograman?”	Menampilkan data buku lengkap	Informasi relevan	Berhasil
3	Buku tidak tersedia	“buku tentang topik X?”	Menampilkan pesan tidak tersedia	Sesuai harapan	Berhasil
4	Manipulasi data	“Tambahkan buku baru”	Permintaan ditolak	Penolakan Jelas	Berhasil

Pengujian UAT

Pengujian User Acceptance Testing (UAT) dilakukan terhadap 20 responden menggunakan kuesioner skala Likert. Hasil pengujian memperoleh nilai sebesar 82,3% yang termasuk dalam kategori "Sangat Sesuai". Mayoritas responden menilai chatbot mudah digunakan, mampu memberikan informasi yang relevan, serta membantu proses pencarian buku di perpustakaan. Namun, beberapa responden masih menilai interaksi chatbot belum sepenuhnya natural dibandingkan sistem pencarian berbasis website.

Evaluasi Metrik Sistem RAG

Selain itu, dilakukan evaluasi kualitas jawaban sistem RAG menggunakan metrik Answer Relevancy dan Faithfulness terhadap 15 pertanyaan uji. Hasil evaluasi menunjukkan nilai Answer Relevancy sebesar 100% dan Faithfulness sebesar 100%, yang menunjukkan bahwa jawaban chatbot relevan dengan pertanyaan pengguna serta sesuai dengan data pada database perpustakaan.

KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan chatbot perpustakaan berbasis Telegram menggunakan pendekatan RAG yang terintegrasi dengan GPT-4.1-mini. Sistem mampu membantu pengguna memperoleh informasi koleksi buku secara cepat, relevan, dan interaktif melalui bahasa natural. Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan dengan baik, nilai UAT memperoleh 82,3% dengan kategori "Sangat Sesuai", serta evaluasi Answer Relevancy dan Faithfulness memperoleh nilai 100%.

Pengembangan selanjutnya dapat dilakukan dengan mengintegrasikan data peminjaman secara *real-time*, menambahkan fitur rekomendasi buku, serta memperluas implementasi chatbot pada platform website maupun aplikasi mobile. Selain itu, evaluasi sistem dapat dikembangkan menggunakan metrik khusus RAG seperti Answer Relevancy dan Faithfulness untuk mengukur kualitas jawaban secara lebih mendalam.

DAFTAR PUSTAKA

- [1] Fardi C. F. P., Anwar R. K., Amar S. C. D., and Rukmana E. N., "Perkembangan Pelayanan Perpustakaan Perguruan Tinggi," *Jurnal Perpustakaan dan Kearsipan*, vol. 4, no. 2, pp. 60–72, 2024, doi: <https://doi.org/10.24821/jap.v4i2.9610>.
- [2] Lubis A. T. U., Harahap N. S., Agustian S., Irsyad M., and Afrianty I., "Question Answering System on Telegram Chatbot Using Large Language Models (LLM) and Langchain (Case Study: Health Law)," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 955–964, Jul. 2024, doi: <https://doi.org/10.57152/malcom.v4i3.1378>.
- [3] Gao Y. *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," pp. 1–21, Mar. 2024.
- [4] Bang Y. *et al.*, "AMultitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," *Centre for Artificial Intelligence Research (CAiRE)*, Nov. 2023.
- [5] Brown T. B. *et al.*, "Language Models are Few-Shot Learners," pp. 1877–1901, Dec. 2020, Accessed: Sep. 07, 2025. [Online]. Available: <https://dl.acm.org/doi/abs/10.5555/3495724.3495883>

- [6] Tohir H., Merlina N., and Haris M., "Utilizing Retrieval-Augmented Generation In Large Language Models To Enhance Indonesian Language NPL," *JTIK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, vol. 10, no. 2, pp. 352–360, 2024, doi: <https://doi.org/10.33480/jitk.v10i2.5916>.
- [7] Lewis P. *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," <https://arxiv.org/abs/2005.11401>, 2021.
- [8] Samudra G., Zy A. T., and Ermanto, "Implementasi Retrieval Augmented Generation dalam Perancangan Chatbot Kesehatan Pencernaan," *JSAI: Journal Scientific and Applied Informatics*, vol. 8, no. 1, pp. 181–188, 2025.
- [9] Ghozi W., Rahmawan E., and Setiono O., "Design of a Telegram Chatbot to Control Internet Connection on Computer Laboratory," *Jurnal Mantik*, vol. 6, no. 3, pp. 3591–3599, 2022.
- [10] Pujiono I., Agtyaputra I.M., and Ruldeviyani Y., "Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context," *Jurnal Ilmu Pengetahuan dan Teknologi Komputer*, vol. 10, no. 1, pp. 216–223, 2024, doi: <https://doi.org/10.33480/jitk.v10i1.5572>.
- [11] Ramahdani A., Yantoro M. D., Akmal M. F., Mahfud M., and Fauzi, "Chatbot Otomatis dengan N8N dan AI untuk Analisis Data dan Pelaporan Hasil," *Jurnal Riset Teknik Komputer*, vol. 2, no. 2, pp. 18–23, 2025, doi: <https://doi.org/10.69714/x1p94182>.
- [12] Hidayat L. R., Wijaya G. P. S., and DwiYansaputra R., "Optimalisasi Layanan Sistem Informasi Mahasiswa dengan Integrasi Telegram : Chatbot Retrieval-Augmented-Generation berbasis Large Language Model," *Jurnal Teknologi Informasi, Komputer dan Aplikasinya (JTika)*, vol. 7, Mar. 2025, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [13] Jailani A. and Yaqin M.A., "Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis," *Journal Automation Computer Information System*, vol. 4, no. 2, pp. 60–66, 2024, doi: <https://doi.org/10.47134/jacis.v4i2.78>.
- [14] Aliyah, Hartono N., and Muin A. A., "Penggunaan User Acceptance Testing(UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang," *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 1, pp. 84–100, 2025, doi: <https://doi.org/10.62951/switch.v3i1.330>.