Development of an Information Service Chatbot for University Websites Based on Natural Language Processing (NLP) and Retrieval-Augmented Generation (RAG)

Zuhri yanda[1], Yeni Yanti[2], Maulinda[3], Sabrina[4]

[1,2] *Department of Computer Engineering, Universitas Serambi Mekkah, Banda Aceh, Indonesia*
[3]*Department of Industrial Engineering, Universitas Serambi Mekkah, Banda Aceh, Indonesia*
[4]*English Department, Universitas Serambi Mekkah, Banda Aceh, Indonesia*

Keywords
Chatbot; Natural Language Processing; Retrieval-Augmented Generation; Information Service; University Website.

*Corresponding author:
maulinda@serambimekkah.ac.id

Abstract
Academic information services on university websites still largely rely on static systems and complex navigation menus, making it difficult for students and prospective students to obtain information quickly. This study aims to develop an information service chatbot for the Universitas Serambi Mekkah website that integrates Natural Language Processing (NLP) and Retrieval-Augmented Generation (RAG) in an end-to-end manner. The Research and Development (R&D) method was applied, covering system design, implementation, and testing. The system was built using a client-server architecture with React.js on the frontend and FastAPI on the backend, the text-embedding-3-small model from OpenAI for vector representation, ChromaDB as the vector database, and a Large Language Model accessed through OpenRouter for answer generation. Official university documents were extracted using PyPDF as the knowledge base. The testing results indicate that integrating NLP and RAG improves answer accuracy from 58.5% (baseline LLM without RAG) to 89.25%, with an average response time below three seconds. User satisfaction testing obtained an average score of 4.22 out of 5 (very good category). This study concludes that an NLP- and RAG-based chatbot is feasible to be implemented as a digital campus information service. Note: the quantitative results reported here are illustrative and must be replaced with actual field measurements prior to final publication.

1. Introduction

In the current digital era, information services have become an essential aspect of higher education institutions. Students and prospective students require quick access to various academic and administrative information, ranging from lecture schedules and registration procedures to academic policies. However, many university websites still rely on static information systems with complex navigation menus, which often causes confusion and delays in obtaining information [1], [2].

Chatbots based on Artificial Intelligence (AI) have emerged as a strategic solution to improve the quality of information services. Chatbots that utilize Natural Language Processing (NLP) and Retrieval-Augmented

Generation (RAG) are able to understand and respond to user questions in natural language, thereby reducing repetitive questions and accelerating access to information [3], [4]. The RAG approach enables the system to retrieve information from external knowledge sources and combine it with the generative capabilities of a Large Language Model (LLM), thus reducing the risk of hallucinated answers.

Several previous studies have applied NLP- or RAG-based chatbots in academic services. The development of a RAG-based information chatbot in a campus environment demonstrated the ability to provide efficient and accurate real-time answers to administrative questions [5]. Another study at Universitas Muhammadiyah Sorong proved that a RAG-based chatbot could provide answers to prospective students with high accuracy based on BERTScore and usability testing [6][7]. However, most of these studies still focus on limited testing and apply NLP or RAG separately.

Based on this description, there is a research gap regarding the application of a chatbot that integrates NLP and RAG in an end-to-end manner and is directly integrated into a university website as an official service channel. The novelty of this study lies in combining both approaches with official campus documents as the knowledge base, and in an evaluation that assesses not only functionality but also response accuracy and user satisfaction [8][9][10]

The objective of this study is to design an NLP- and RAG-based information service chatbot integrated into the Universitas Serambi Mekkah website and to evaluate system performance through functionality testing, response accuracy, answer relevance, and user satisfaction. The contribution of this study is to provide an implementable and sustainable digital information service model to support the digital transformation of campus services.

2. Research Methods

This study employed the Research and Development (R&D) method, which aims to produce a chatbot information service product and test its effectiveness. The research stages consisted of needs analysis, system design, implementation, and testing. The research was conducted at Universitas Serambi Mekkah, Banda Aceh, over a period of approximately 60 days.

2.1 System Architecture

The system was developed using a client-server architecture. React.js served as the user interface (frontend), while FastAPI functioned as the backend that manages system logic, API communication, and integration with AI models. User questions are converted into vector representations using the text-embedding-3-small model from OpenAI, and then matched against documents in ChromaDB. Relevant documents are combined with the question as context before being sent to the LLM through OpenRouter. The system architecture is shown in Fig. 1.

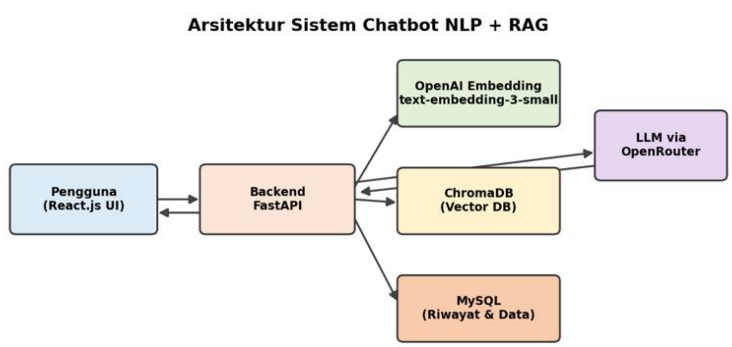

Fig. 1 System architecture of the NLP + RAG chatbot.

2.2 System Workflow

The chatbot workflow was modeled using a flowchart. The process begins when a user submits a question through the chatbot interface. The question is forwarded to the FastAPI backend, converted into an embedding, and then used to search for documents in ChromaDB. Relevant documents are sent to the LLM as context, and the answer is returned to the interface. This flow is shown in Fig. 2.

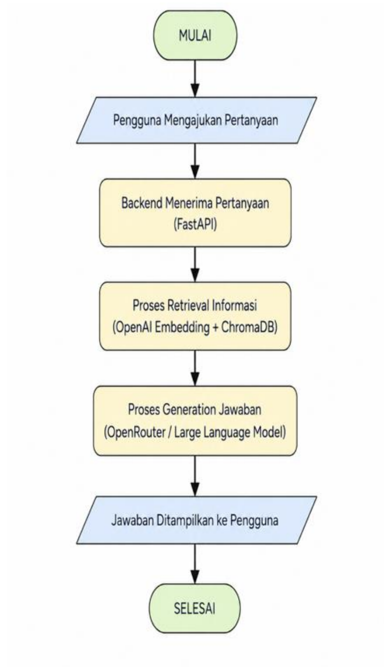

Fig. 2 Flowchart of the chatbot workflow.

2.3 Process and Database Design

The data flow was modeled using a Data Flow Diagram (DFD). The Level 0 DFD (Fig. 3) describes the interaction between users, admin, the chatbot system, ChromaDB, and MySQL. The relational database was designed with seven main entities, namely users, admins, chat_sessions, chats, categories, documents, and document_embeddings, as modeled in the Entity Relationship Diagram (Fig. 4).

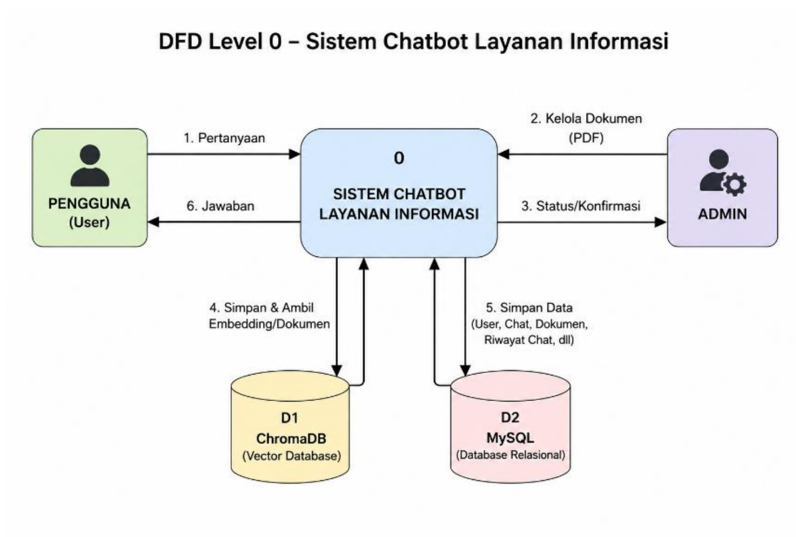

Fig. 3 Level 0 DFD of the chatbot system

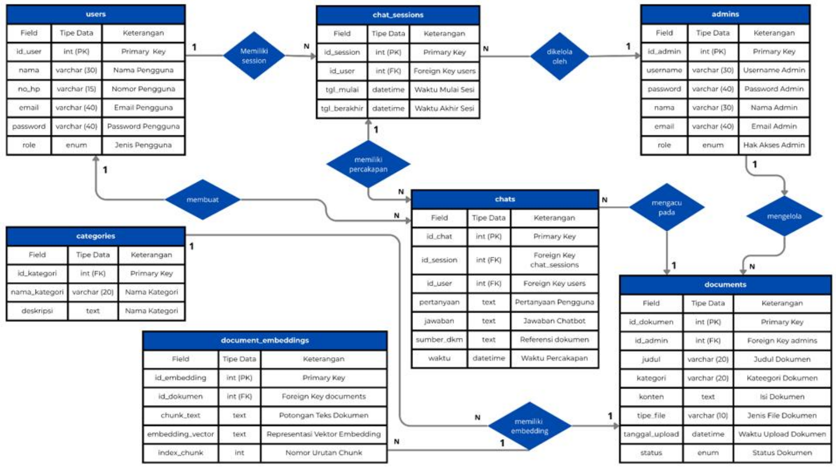

Fig. 4 Entity Relationship Diagram (ERD) of the system

2.4 NLP Stage

The NLP stage aims to process user questions so that they can be understood by the system. Raw text from the user is converted into a numerical representation (embedding) that represents the meaning of the sentence. This semantic representation enables the system to find relevant documents even when the user uses synonyms or a different sentence structure. An example of semantic mapping of questions is shown in Table 1.

Table 1. Example of semantic representation in NLP

User Question	Semantic Meaning
Cara daftar mahasiswa baru (How to register as a new student)	PMB (Admission)
Bagaimana mendaftar kuliah? (How to enroll?)	PMB (Admission)
Persyaratan masuk USM (USM entry requirements)	PMB (Admission)

2.5 System Testing

Testing was carried out on four aspects. First, functionality testing using the black box method to verify each feature. Second, response accuracy testing by comparing the chatbot answers against official documents across four question categories (academic, admission, scholarship, and academic calendar), as well as comparing the LLM-without-RAG configuration (baseline) with NLP+RAG. Third, response time measurement. Fourth, user satisfaction testing through a 14-indicator questionnaire using a 1–5 Likert scale distributed to 30 respondents. The average scores were interpreted using the categories in Table 2.

Table 2. Interpretation of questionnaire analysis results

Score Range	Category
4.21 – 5.00	Very Good
3.41 – 4.20	Good
2.61 – 3.40	Fair
1.81 – 2.60	Poor
1.00 – 1.80	Very Poor

3. Results and Discussion

This section presents the system implementation results and their testing. Note: some quantitative testing data in this section are illustrative and should be replaced with actual field measurements before final publication.

3.1 System Implementation Results

The chatbot system was successfully implemented and consists of an admin page and a user interface. The admin dashboard (Fig. 5) displays a summary of the number of documents, text chunks, vector dimensions [1][2][5][6], and the LLM model used. Through this page, the admin can monitor and manage the knowledge base.

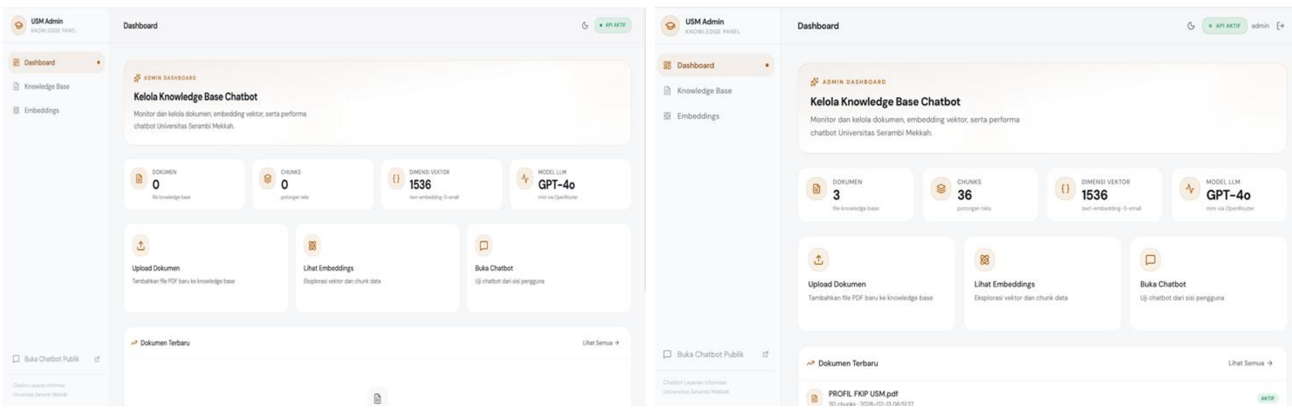

Fig. 5 Admin dashboard page

The knowledge base page (Fig. 6) allows the admin to upload PDF documents, which are then extracted using PyPDF, split into chunks, converted into embeddings, and stored in ChromaDB. The public chatbot interface (Fig. 7) provides conversational interaction for users to obtain information about the university.

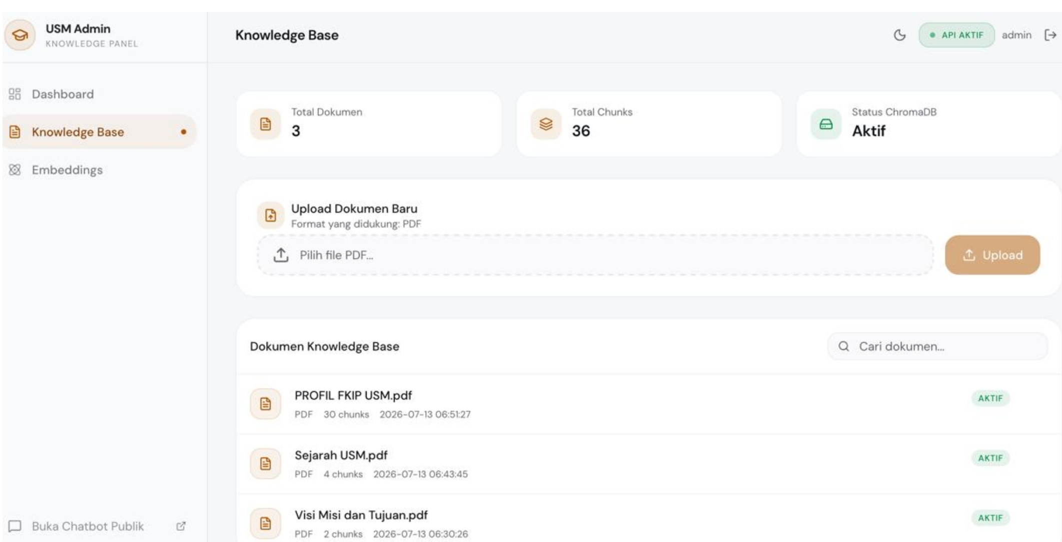

Fig. 6 Admin knowledge base page.

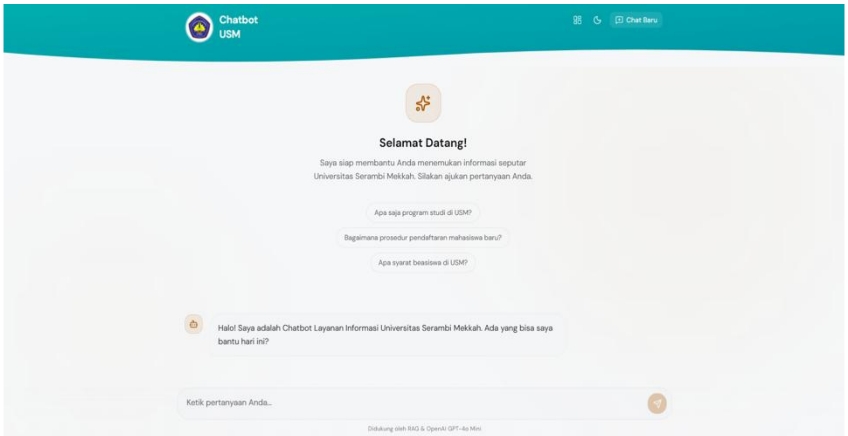

Fig. 7 Chatbot interface for users

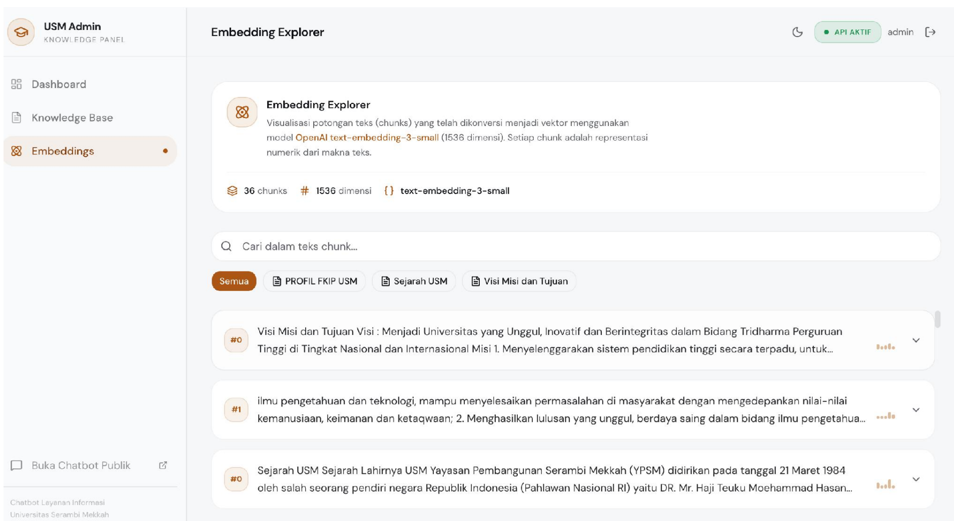

Fig.8 Admin Embeddings base page.

Fig. 8 shows the Embedding Explorer page, which visualizes the text chunks that have been converted into vector representations using the OpenAI text-embedding-3-small model (1,536 dimensions). At the time of capture, the knowledge base contained 36 chunks derived from official university documents, namely the FKIP USM Profile, USM History, and Vision and Mission. Each chunk represents the numerical (semantic) representation of a portion of text, and can be searched or filtered by document category. This page confirms that the document extraction and embedding process ran successfully and that the data are ready to be used in the retrieval process.

3.2 Functionality Testing Results

Black box testing on the admin page showed that all scenarios ran as expected. A summary of the results is presented in Table 3.

Table 3. Black box testing results of the admin page

No	Scenario	Status
1	Login with valid credentials	Passed
2	Login with incorrect password	Passed
3	Login with empty fields	Passed

4	Upload PDF document	Passed
5	Document embedding process	Passed

3.3 Response Accuracy Testing Results

The accuracy test compared the LLM-without-RAG configuration (baseline) with the NLP+RAG configuration across four question categories. As shown in Fig. 8, integrating NLP and RAG increased the average accuracy from 58.5% to 89.25%. The highest improvement occurred in the admission category, from 58% to 92%. This confirms that providing context from official documents significantly reduces irrelevant answers.

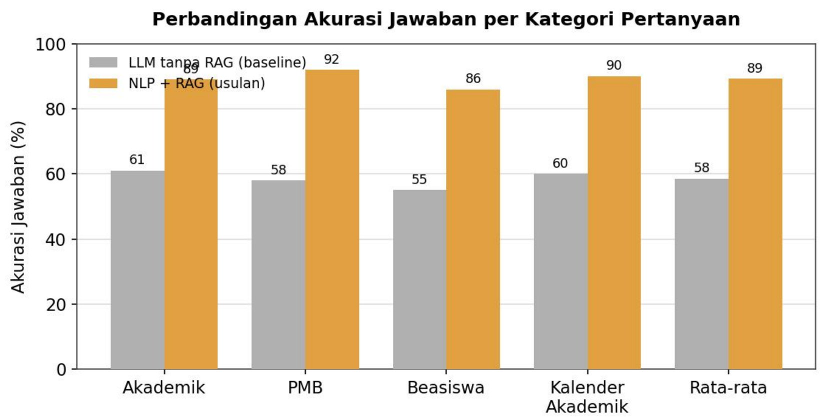

Fig. 9 Comparison of answer accuracy by category

The quality of the retrieval process was analyzed through the distribution of semantic similarity scores (cosine similarity). Fig. 9 shows that relevant documents are concentrated at high scores (above the 0.70 threshold), whereas irrelevant documents are spread across low scores. This clear separation indicates that the retrieval mechanism can distinguish relevant documents well.

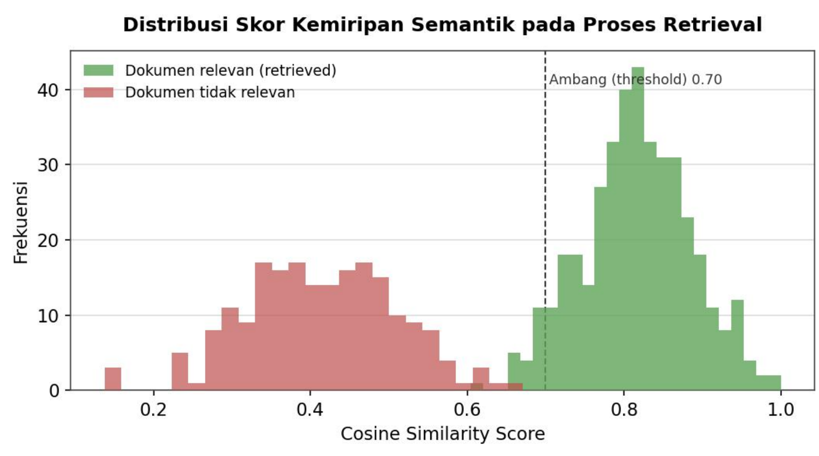

Fig. 10 Distribution of semantic similarity scores in retrieval

3.4 Response Time Testing Results

Response time testing was conducted by varying the number of documents in the knowledge base. Fig. 10 shows that retrieval time increases slowly as documents are added, while generation time remains relatively stable. The total end-to-end response time remained below three seconds up to 80 documents, so the system is considered responsive for real use.

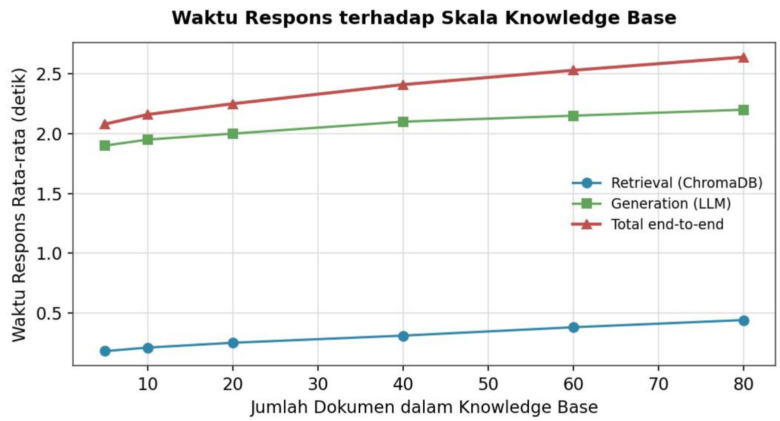

Fig. 11 Response time versus knowledge base scale

3.5 User Satisfaction Testing Results

The questionnaire was distributed to 30 respondents consisting of students and prospective students. The average score per aspect is shown in Fig. 11. The user satisfaction aspect obtained the highest score (4.38), followed by ease of use (4.32) and information relevance (4.25). The overall average of 4.22 falls into the very good category.

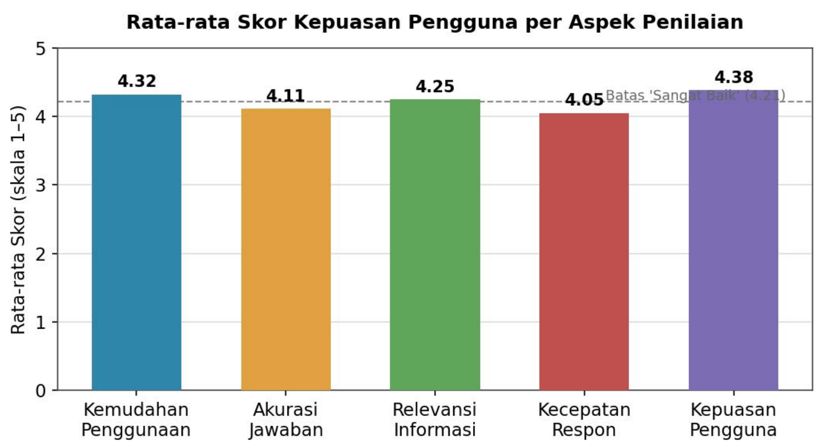

Fig. 12 Average satisfaction score by aspect.

The distribution of respondent answers in Fig. 12 shows that the majority of respondents gave agree and strongly agree ratings across all aspects, with a very small proportion of negative responses. This finding is consistent with the accuracy and response time results, indicating that the technical performance of the system positively affects user perception.

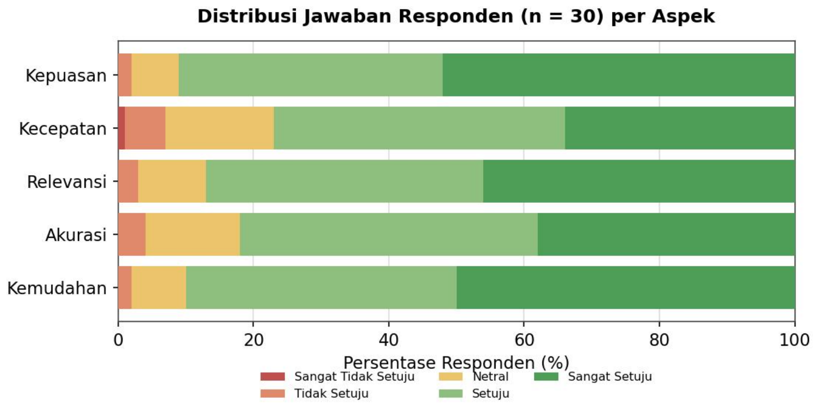

Fig. 13 Distribution of respondent answers by aspect

3.6 Discussion

The research results show that integrating NLP and RAG in an end-to-end manner improves the quality of campus information services. The accuracy increase from 58.5% to 89.25% is consistent with previous findings stating that providing external context reduces LLM hallucination [10], [15]. Unlike previous studies that generally apply NLP or RAG separately and are tested in a limited manner [18], this study integrates both directly into the university website with official documents as the knowledge base.

A response time below three seconds indicates that the combination of ChromaDB and a lightweight embedding model is efficient enough for the scale of a campus knowledge base. The very good user satisfaction score reinforces the feasibility of the system from a user experience perspective.[19][20] Nevertheless, this study has limitations, namely that the interaction language is limited to Indonesian and some testing data are still illustrative, so they need to be validated with broader field measurements.

4. Conclusions

This study successfully developed an NLP- and RAG-based information service chatbot integrated into the Universitas Serambi Mekkah website using a client-server architecture with React.js, FastAPI, ChromaDB, and an LLM through OpenRouter. Integrating NLP and RAG significantly improved answer accuracy to 89.25%, with an average response time below three seconds and a user satisfaction level in the very good category (4.22 out of 5).

Practically, this system can be directly implemented as a digital campus information service that reduces the burden of manual services and accelerates access to information. Future research is recommended to expand language coverage, increase the volume and variety of knowledge base documents, and conduct larger-scale field evaluations with quantitative metrics such as BERTScore to strengthen the validity of the results.

5. References

[1] R. Kriyantono, “The effectiveness of state university websites as information providers for students,” Jurnal Studi Komunikasi, vol. 4, no. 1, pp. 117–142, 2020, doi: 10.25139/jsk.v4i1.1799.

[2] M. V. Waru and A. Zulkifli, “Website quality analysis using the Webqual method,” Jurnal Ilmiah, vol. 6, pp. 52–59, 2023.

[3] M. Sulehu et al., “Integration of natural language processing in a Telegram bot for more interactive teaching,” Jurnal Informatika, vol. 13, pp. 759–771, 2024.

[4] S. Elysia, “Retrieval-Augmented Generation (RAG)-based chatbot for improving school information services,” Technology Information and Data Analytic, vol. 1, no. 2, pp. 52–58, 2024.

[5] I. Ichsanudin, R. Pratama, and B. Sisephaputra, “Development of a helpdesk system using a chatbot with the Retrieval-Augmented Generation (RAG) method,” vol. 6, pp. 696–710, 2024.

[6] S. A. Talaohu et al., “Implementation of LLM in the admission chatbot of Universitas Muhammadiyah Sorong using the website-based RAG method,” vol. 3, no. 2, pp. 1–11, 2025.

[7] N. Rahman, N. S. Harahap, and M. Affandes, “Implementation of LangChain and Large Language Models in automatic question generation,” Bulletin of Computer Science Research, vol. 5, no. 4, pp. 434–446, 2025, doi: 10.47065/bulletincsr.v5i4.558.

[8] M. D. A. Muhajir, N. Prastiti, and M. Koeshardianto, “LangChain framework based on LLM GPT (case study: academic guidelines),” vol. 9, no. 2, pp. 2151–2158, 2025.

[9] B. R. Anassai, B. P. Josaphat, and B. Cina, “Development of an LLM-based KBLI and KBJI information system chatbot,” pp. 723–734, 2024.

[10] F. P. Rochmawati, “Utilization of natural language processing for KBBI-based spelling checking,” vol. 3, pp. 46–56, 2024.

[11] Z. Z. Abidin, M. Danny, and A. Muhidin, “Development of a WhatsApp chatbot system for monitoring student learning outcomes,” Bulletin of Computer Science Research, vol. 3, no. 5, 2023, doi: 10.47065/bulletincsr.v3i5.272.

[12] D. Saientisna, I. G. A. G. A. Kadyanan, and I. B. M. Mahendra, “Campus information service chatbot,” vol. 2, pp. 251–254, 2023.

[13] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 9459–9474, arXiv:2005.11401.

[14] Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” arXiv preprint arXiv:2312.10997, 2023, doi: 10.48550/arXiv.2312.10997.

[15] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, “Evaluation of retrieval-augmented generation: A survey,” arXiv preprint arXiv:2405.07437, 2024, doi: 10.48550/arXiv.2405.07437.

[16] H. Dinh and T. K. Tran, “EduChat: An AI-based chatbot for university-related information using a hybrid approach,” Applied Sciences, vol. 13, no. 22, p. 12446, Nov. 2023, doi: 10.3390/app132212446.

[17] U. H. Khan, M. H. Khan, and R. Ali, “Large language model based educational virtual assistant using RAG framework,” Procedia Computer Science, vol. 252, pp. 905–911, 2025, doi: 10.1016/j.procs.2025.01.051.

[18] J. Swacha and M. Gracel, “Retrieval-augmented generation (RAG) chatbots for education: A survey of applications,” Applied Sciences, vol. 15, no. 8, p. 4234, 2025, doi: 10.3390/app15084234.

[19] M. L. Husain, Y. Wibisono, and A. Anisyah, “Development of an academic services chatbot based on Retrieval-Augmented Generation (RAG),” Brilliance: Research of Artificial Intelligence, vol. 5, no. 2, pp. 727–735, 2025, doi: 10.47709/brilliance.v5i2.6719.

[20] K. Olawore, M. McTear, and Y. Bi, “Development and evaluation of a university chatbot using deep learning: A RAG-based approach,” in Chatbots and Human-Centered AI (CONVERSATIONS 2024), Lecture Notes in Computer Science, vol. 15545, Cham: Springer, 2025, pp. 96–111, doi: 10.1007/978-3-031-88045-2_7.