ISSN 2964-0490

A RAG and Reciprocal Rank Fusion-Based Chatbot for New Student Admission (SPMB) Information Services at SMK Muhammadiyah 1 Wonosobo

Iqbal Ali Ar-Ridho[1], Nahar Mardiyantoro[2], Saifu Rohman[3]
[1,2,3]Informatics Engineering Study Program, Faculty of Engineering and Computer Science, Universitas Sains Al Qur'an Jawa Tengah di Wonosobo, Indonesia
Email: [1]iqbalali2606@gmail.com, [2]mardziyant@gmail.com, [3]saifurohman@unsiq.ac.id

The New Student Admission (SPMB) information service at SMK Muhammadiyah 1 Wonosobo faces a high volume of inquiries—987 registrants were recorded in 2025—handled through seven official channels limited to working hours, resulting in repetitive questions and inconsistent answers. This study aims to develop a Natural Language Processing (NLP)-based chatbot using the Retrieval-Augmented Generation (RAG) and Reciprocal Rank Fusion (RRF) methods to automate the SPMB information service on the school's official website. The system was built using the Prototype development method. The designed RAG pipeline adds three retrieval-enhancement mechanisms on top of naive RAG, namely query expansion, RRF with an asymmetric 4:1 weighting, and cross-encoder reranking, with Gemini 3 Flash Preview as the answer generator and a knowledge base of 44 official school text documents. Testing was conducted through Black Box Testing and RAGAS evaluation measuring faithfulness, answer relevancy, and context precision. Black Box Testing of 17 scenarios achieved a 100% functional success rate. RAGAS evaluation on the full (Enhanced) configuration obtained faithfulness 0.9812, answer relevancy 0.9146, and context precision 0.7991. Compared to the Baseline configuration, answer relevancy increased by 12.40% and context precision by 14.16%, while faithfulness remained in the "Very Good" category. This study concludes that applying these three techniques positively contributes to document relevance and the answer quality of the SPMB information chatbot.
Keywords: chatbot, Natural Language Processing, Retrieval-Augmented Generation, Reciprocal Rank Fusion, SPMB

Corresponding Author:
Iqbal Ali Ar-Ridho
Informatics Engineering Study Program, Faculty of Engineering and Computer Science, Universitas Sains Al Qur'an Jawa Tengah di Wonosobo, Indonesia
iqbalali2606@gmail.com

1. Introduction

The digital transformation of the education sector encourages the delivery of public services that are automated, transparent, and accessible at any time. In vocational secondary schools, this need is most acute during the New Student Admission (*Seleksi Penerimaan Murid Baru*, SPMB) period, when a school must serve a large volume of information requests from prospective students and their guardians. SMK Muhammadiyah 1 Wonosobo faces this condition directly; according to data on the school's official website, public interest in the 2025 admission reached 987 registrants.

This large number of registrants is accompanied by the complexity of the information channels the committee must manage. The school provides seven official channels—Email, Telephone, Website, Instagram, TikTok, WhatsApp, and Facebook—all of which operate only during working hours. Meanwhile, information requests may arrive outside working hours, creating a gap between service availability and public access needs. Because requests are answered directly by the committee, recurring questions—such as registration requirements, selection schedules, fees, and program details—must be answered repeatedly whenever posed by different registrants.

Given these conditions, a system is needed to handle SPMB information requests automatically on the school's official website. A chatbot based on Natural Language Processing (NLP)—the field concerned with the interaction between computers and human language [1]—is a relevant approach because it can serve questions without being limited to working hours, provide consistent answers, and reduce the committee's burden in answering recurring questions. This study is limited to SPMB information services and does not cover the formal student registration process.

Accordingly, the objectives of this study are to (1) design an NLP-based chatbot architecture that integrates the RAG and RRF methods; (2) determine the contribution of applying RRF together with Query Expansion and cross-encoder reranking to the relevance of the selected documents; (3) integrate the chatbot into the school's official website so that it is accessible without being limited to committee working hours; and (4) measure the chatbot's performance through Black Box Testing and RAGAS metric evaluation.

2. Literature Review and Problem Statement

Several prior studies have demonstrated various approaches to developing chatbots for public services. Rafi *et al.* [2] developed an SPMB chatbot at SMK Intensif Baitussalam using the Cosine Similarity method, which works by matching keyword patterns; this approach is limited to questions with high keyword similarity to the dataset, so question variations outside the dataset are not handled well. In the local Wonosobo context, Naufan [3] developed an information-service chatbot for the Regional Revenue, Finance, and Asset Management Agency (BPPKAD) of Wonosobo Regency using the Retrieval-Augmented Generation (RAG) method with the GPT-3.5 Turbo model, obtaining a faithfulness score of 0.83 and an answer relevancy of 0.89 in RAGAS testing. RAG itself combines document retrieval with text generation by a language model to produce source-grounded answers [6], while Reciprocal Rank Fusion (RRF) is an algorithm that merges several ranked document lists through the sum of reciprocal ranks and can improve the relevance of search results [4].

One technical problem in chatbots based on Large Language Models (LLMs) is hallucination, a condition in which the model produces plausible-sounding answers that are not supported by source documents [5]. The RAG approach mitigates hallucination by requiring the model to construct answers from context retrieved from the knowledge base rather than from the model's internal memory [6]. To ensure answer quality in a measurable way, the RAGAS framework is used because it can measure faithfulness, answer relevancy, and context precision without requiring large-scale human annotation [7].

Based on this review, a research gap emerges: prior work in local information-service contexts still relies on keyword matching [2] or basic RAG [3] without a rank-fusion mechanism to strengthen the relevance of retrieved documents. Accordingly, the research problem of this study is how to design and measure a RAG-based chatbot that integrates Reciprocal Rank Fusion together with query expansion and cross-encoder reranking to improve document relevance while maintaining answer faithfulness in the SPMB information service.

3. Methods

Research Method and Data Collection

This study uses the Prototype system development method, which builds a system incrementally through prototypes that are iteratively tested and refined [8]. This method was chosen because the answer quality of a RAG-based chatbot must be refined iteratively through repeated testing. The research flow consists of

seven stages: problem identification, data collection, data analysis, system design, prototype development, testing and evaluation, and drawing conclusions.

The research object is the SPMB information service at SMK Muhammadiyah 1 Wonosobo, Wonosobo Regency, Central Java. Data were collected through literature study, observation, interviews, and documentation. The knowledge-base data were obtained primarily through *API requests* to the school website's official *endpoints*, because API data are in *JavaScript Object Notation* (JSON) with a standardized structure, making extraction more consistent than DOM-based *web scraping*, which is vulnerable to page-layout changes. Each JSON response was normalized into a *human-readable* text (.txt) file to fit the *TextLoader* module in the *LangChain* framework. A total of 44 text files were collected, comprising 42 files normalized from 42 API *endpoints* and 2 files from consultation with the school. The data-source mapping is presented in Table 1.

Table 1. Knowledge-Base Data-Source Mapping

Information Category	Files	Information Coverage
School Profile & Identity	5	School data, vision-mission, about, principal's welcome, homepage slider
Programs & Academics	4	Program list (6 programs), categories, special programs, class schedule
HR & Organization	3	Staff/teacher data, vice principals (WKS), organizational structure
Registration & SPMB	6	Admission tracks, registration paths, active waves, requirements, registrant list, total registrants
Activities & Homepage Content	7	Events, posts/news, photos, videos, achievements, announcements, category tags
Media & Documentation	3	Photos, videos, learning media
Finance & Administration	2	Payment details, Certificate of Graduation (SKL)
Student Affairs & Services	4	Student complaints, attendance/monitoring, DUDI (industry partners), info media
Feeder School Data	1	List of origin junior high schools (SMP/MTs)
Pagination Pages (mirror)	9	Paginated versions of events, posts, photos, achievements, announcements, videos, and PTK
Total	**44**	**All SPMB & school profile data**

RAG Pipeline Architecture

The core of the system is a Retrieval-Augmented Generation (RAG) pipeline that extends naive RAG by adding two retrieval-enhancement mechanisms—query expansion and Reciprocal Rank Fusion (RRF)—and one final filtering mechanism, a cross-encoder reranker. The pipeline comprises an indexing phase, executed once at start-up, and an inference phase, executed for every user question.

The indexing phase extracts text from the 44 files using *TextLoader*, splits the text into chunks using *RecursiveCharacterTextSplitter* (*chunk_size* 1,000 characters; *chunk_overlap* 200 characters), and transforms each chunk into a 1,024-dimensional vector using the *BAAI/bge-m3* embedding model [9], stored in a *FAISS* vector store [10]. The inference phase comprises six sequential stages:

1. *Query expansion* generating 2 question variants (*temperature* 0.2), for a total of 3 queries.
2. *Embedding* of the queries and parallel similarity search on *FAISS* (k = 25).

3. *Reciprocal Rank Fusion* merging the 3 ranked lists (k = 60; asymmetric weighting 4.0 : 1.0) into 12 candidate documents.
4. *Cross-encoder reranking* with *BAAI/bge-reranker-v2-m3* (score threshold 0.40; gap threshold 0.25) yielding at most 5 final documents.
5. *Prompt assembly* and answer generation: the final documents are assembled with the question into a prompt sent to Gemini 3 Flash Preview (*temperature* 0.0) to produce the final answer.

The Gemini 3 Flash Preview model is accessed through the OpenRouter service provider. According to the provider's documentation, the model supports multi-turn conversation, a large context window, and the Indonesian language, matching the interaction characteristics of the SPMB chatbot. It should be noted that the preview status and proprietary nature of this model have implications for reproducibility, since outputs may differ if the model version is updated. The complete RAG pipeline architecture is presented in Fig. 1.

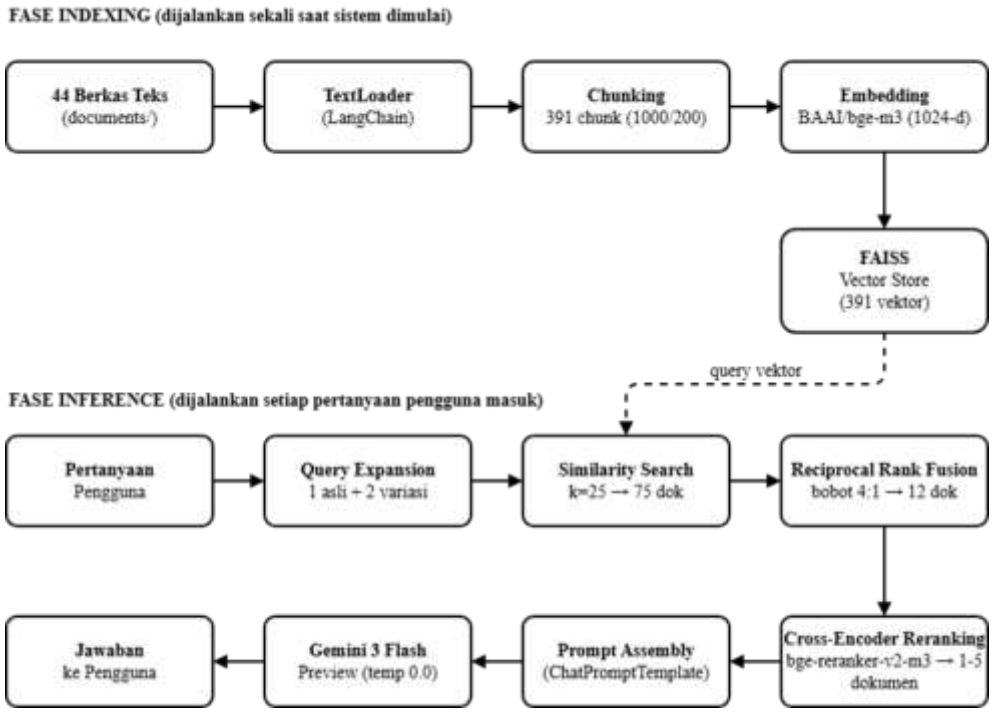


Figure 1. Retrieval-Augmented Generation (RAG) Pipeline Architecture

Reciprocal Rank Fusion Formula

In the Reciprocal Rank Fusion stage, the three ranked lists from the similarity search are merged into a single ranked list. The RRF score for a document d is computed using the weighted RRF formula:

$$\text{score}_{\text{RRF}}(d) = \sum_{i=1}^{n} w_i \cdot \frac{1}{k + r_i(d) + 1} \tag{1}$$

where n is the number of merged ranked lists (here n = 3), $r_i(d)$ is the rank of document d in list i, k is a smoothing constant, and w_i is the weight of list i. The constant k = 60 follows the recommendation of the original RRF paper [11], while adding one to the denominator accommodates ranks counted from zero (following programming-language indexing conventions), so that the relative behavior among documents remains consistent with the original formula.

Asymmetric weighting is applied by assigning $w_0 = 4.0$ to the ranked list from the original query and $w_1 = w_2 = 1.0$ to the two variants. The 4:1 ratio reflects the anchor query concept: the user's original question is placed as the primary anchor representing the true intent, while the LLM-generated variants act as coverage explorers. This principle of preserving the dominance of the original query is consistent with the Query2doc method, which explicitly repeats the original query so that its influence is not diluted by expansion results [12]. After RRF, the top 12 documents proceed to cross-encoder reranking using *BAAI/bge-reranker-v2-m3*, which scores query-document relevance directly and is therefore more accurate than the bi-encoder used in the similarity search [13].

Testing and Evaluation

Testing was carried out in two stages. First, Black Box Testing verified the functional correctness of the system based on a mapping to the system use cases [14]. Second, answer quality was evaluated using the RAGAS framework [7], which measures faithfulness (fidelity of the answer to the source documents), answer relevancy (relevance of the answer to the question), and context precision (relevance of the retrieved documents). The three metrics yield scores in the range 0–1, with higher values indicating better quality.

The evaluation used 10 SPMB test questions, each paired with a manually constructed ground-truth answer based on the source documents and confirmed by the school. The evaluation was run in two modes: Enhanced (the full pipeline with query expansion, RRF, and cross-encoder reranking) and Baseline (naive single-query RAG without fusion or reranking). To reduce random variation in the LLM-based judgment, the evaluation was repeated three times and the final score for each metric was taken as the arithmetic mean. To aid interpretation, RAGAS scores were grouped indicatively into five categories: ≥ 0.85 "Very Good", 0.70–0.84 "Good", 0.55–0.69 "Fairly Good", 0.40–0.54 "Fair", and < 0.40 "Poor". These thresholds are a working reference calibrated against the score ranges of prior studies as a rough benchmark—Naufan [3] obtained 0.83–0.89 for a system deemed feasible, whereas Samudra *et al.* [15] obtained an answer relevancy of 0.57 deemed to still need improvement—and are not intended as an absolute standard. The RAGAS evaluation here uses an LLM as the judge without designating a judge model separate from the answer-generating model, and both the RRF weight selection and the final evaluation were performed on the same set of 10 questions; the implications of these two points are discussed in the Conclusion.

4. Results and Discussion

RAG Pipeline Architecture and Preprocessing Results

The developed RAG *pipeline* successfully prepared the knowledge base in the *indexing* phase. *Chunking* of the 44 text documents produced 391 UTF-8-encoded *chunks*, and each document was assigned a *source metadata* field containing its origin file name as a *provenance* trace for displaying sources in answers. The *chunk_size* of 1,000 and *chunk_overlap* of 200 characters were chosen so that context is preserved across *chunks*. The *bge-m3 embedding* model was selected because it supports more than 100 languages including Indonesian, can run without a GPU, and accepts inputs of up to 8,192 *tokens*—exceeding the *chunk_size*—so that no *chunk* is truncated during *embedding*.

In the *inference* phase, each question is expanded into three questions (one original and two variants). Each question yields 25 candidate documents via *similarity search*, producing 75 candidates before fusion. The three ranked lists are merged through RRF into 12 candidates, then filtered by *cross-encoder reranking* into 1–5 final documents that adapt to the characteristics of the question. This two-stage *retrieve-and-rerank* pattern lets RRF narrow the candidate pool first, so that the computationally more expensive *cross-encoder* only needs to score 12 query-document pairs per question.

Application of Reciprocal Rank Fusion

The choice of the 4:1 weighting ratio was supported by a *hyperparameter tuning* experiment over four weight ratios (1:1, 2:1, 3:1, and 4:1) run on the test questions using RAGAS. The results are presented in Table 2.

Table 2. RRF Weighting Parameter Tuning Results

Weight Ratio	Faithfulness	Answer Relevancy	Context Precision	Latency (s)
1 : 1	0.8571	0.8827	0.4286	9.70
2 : 1	0.7242	0.8837	0.8789	10.18
3 : 1	0.9750	0.9079	0.6667	10.66
4 : 1	0.9818	0.9147	0.8333	12.39

Note: the 4:1 row (highlighted) is the selected configuration.

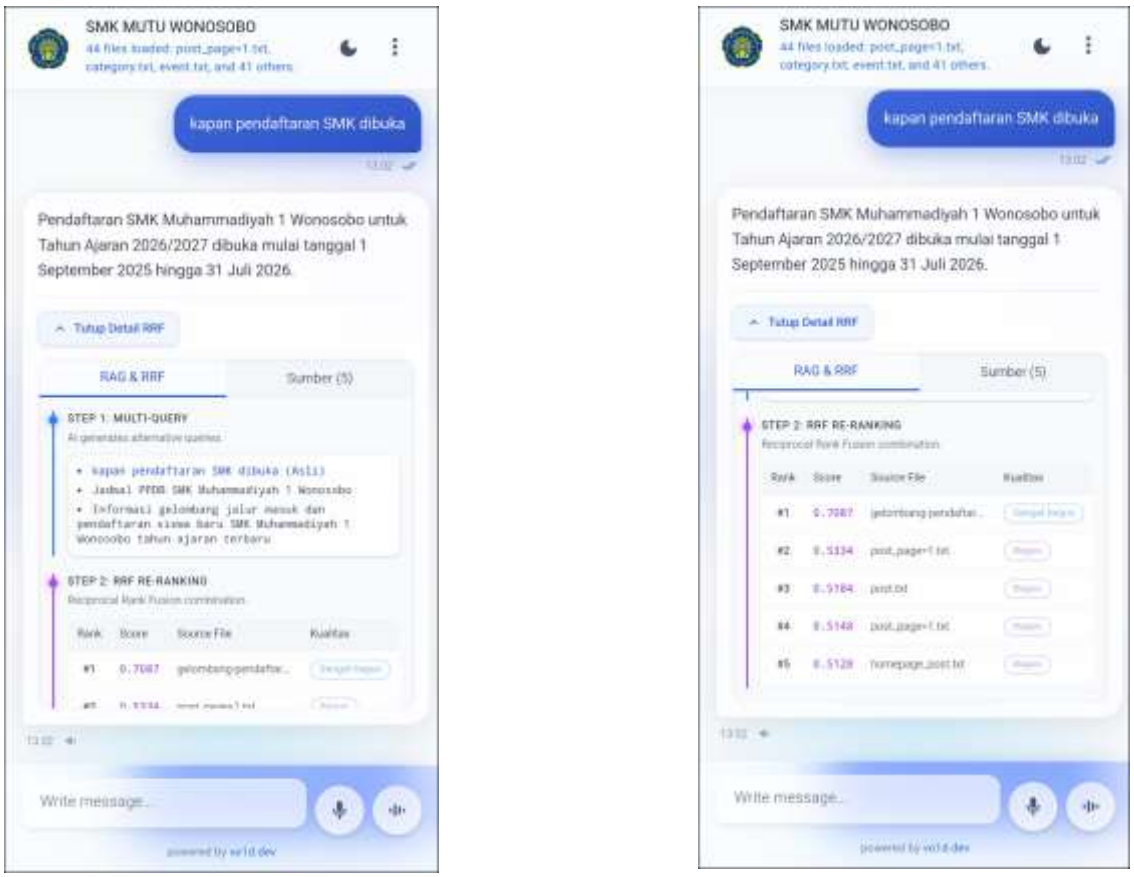
As Table 2 shows, the scores across ratios are non-monotonic and fluctuating—context precision at 2:1 (0.8789) is even higher than at 4:1 (0.8333), and faithfulness at 2:1 (0.7242) drops sharply relative to the other ratios. This fluctuation is unsurprising given that evaluation was performed on a limited question set, so the experiment is better read as a sensitivity analysis than as a precise search for an optimum.

Examined mechanistically, each ratio behaves in an interpretable way. At 1:1, equal weighting lets documents retrieved by the LLM-generated variants—which can drift semantically from the user's intent—occupy top ranks; without the original query anchoring the fusion, relevance is diluted and *context precision* collapses to 0.4286. At 2:1, moderate anchoring still admits enough variant-driven diversity that the generator receives a broad, heterogeneous context set, which raises the chance of statements not fully entailed by a single passage and pushes *faithfulness* down to 0.7242, even though the reranker keeps *context precision* high (0.8789). This decoupling—high precision but low faithfulness—illustrates that retrieval precision and generation faithfulness are distinct axes that need not move together. At 3:1, stronger anchoring restores *faithfulness* (0.9750) but under-weights useful variant evidence, leaving some relevant documents below the cutoff and yielding a moderate *context precision* (0.6667). At 4:1, a dominant anchor preserves the query intent (high *faithfulness*) while still retaining enough variant contribution to recall relevant documents (high *answer relevancy* and *context precision*), making it the only configuration that keeps all three metrics high simultaneously without any collapsing.

This pattern carries a transferable implication for RAG method development. Rackauckas [4] introduced RAG-Fusion with uniform reciprocal-rank fusion of expanded queries; the present results indicate that, on a small and domain-specific corpus, uniformly fusing LLM-expanded queries can actively degrade precision, and that anchor-weighting the original query—consistent with the query-preservation principle of *Query2doc* [12]—is an important refinement rather than a cosmetic one. This positions asymmetric weighting as an important design decision when RAG-Fusion is applied to a limited-size knowledge base. The 4:1 ratio was therefore selected on the combined basis of the *anchor query* rationale and this tuning. Because the ratio was selected using the same question set as the final evaluation, the result is treated as exploratory and potentially optimistic (see the Conclusion). Consequently, the 4:1 configuration also records the highest latency (12.39 s); this *trade-off* is acceptable because the SPMB information service is informational and asynchronous and thus tolerant of a few seconds' response, although latency optimization remains a target for improvement.

Beyond the numerical tuning, the retrieval stages are surfaced to the user through an RRF analysis panel (Figure 2). For the query "*kapan pendaftaran SMK dibuka*" (when does SMK registration open), Step 1 (Figure 2a) lists the original query together with the two LLM-generated variants, which restate it as an

admission-schedule query and as a broader query on admission waves and registration paths. Step 2 (Figure 2b) lists the five documents that reach the final context, each with its fusion score, source file, and a qualitative label: *gelombang-pendaftaran* (0.7087, labelled "*Sangat bagus*" / very good), *post_page=1.txt* (0.5334), *post.txt* (0.5184), *post_page=1.txt* again (0.5148), and *homepage_post.txt* (0.5128). The margin between the top-ranked document and the remainder is substantial (0.7087 versus 0.5334), and the leading document is precisely the file that stores the admission waves and their dates; the answer displayed above the panel—registration for the 2026/2027 academic year runs from 1 September 2025 to 31 July 2026—is grounded in it.



(a) Step 1: multi-query expansion (b) Step 2: the five final documents

Figure 2. RRF Analysis Panel: (a) Query Expansion and (b) the Fused Ranking of the Five Final Context Documents

Two observations follow. First, ranks #2 and #4 refer to the same source file, indicating that fusion and *reranking* operate on *chunks* rather than whole files, so a single document may contribute more than one passage to the final context. Second, the panel header reports the knowledge base actually loaded (44 files), while the *Sumber* (Sources) tab reports the five documents used to compose the answer, matching the upper bound imposed by the *cross-encoder reranker*. Surfacing the scores, the source files, and the intermediate retrieval steps in this way makes the *pipeline* auditable to the end user rather than opaque—a property that matters for a public information service, in which an answer about admission dates must remain traceable to the official document that carries it.

Black Box Testing Results

Functional testing used *Black Box Testing* over 17 scenarios mapped from ten system *use cases* plus test variants under alternative conditions. Testing was performed on the Google Chrome browser on a *desktop*

device (1366×768-*pixel* resolution) against the production system (*frontend* on Vercel and *backend* on a *Virtual Private Server*). Scenarios covered submitting questions from the public and admin pages, automatic conversation-history loading, dark-mode activation, opening the RRF analysis panel, reviewing document sources, session management (creating a new session and clearing history), session *auto-expire* after 24 hours, and validation of empty and duplicate *input*.

All 17 scenarios passed with no failures, giving a functional success rate of 100% (17 of 17). This indicates that all *use-case* functionality was implemented as specified. However, this testing only verifies functional correctness and does not measure the substantive quality of the RAG *pipeline*'s answers, which requires the further RAGAS evaluation below.

Figure 3 shows the *chatbot* in operation on the school's official website, embedded as a *floating bubble* that can be opened from any page. In the example, the user asks which computer-related programs the school offers, phrased generically rather than by the official program names; the system nevertheless returns the three relevant programs, namely *Teknik Jaringan Komputer dan Telekomunikasi* (TJKT), *Rekayasa Perangkat Lunak* within the *Teaching Factory IoT* program, and *Pengembangan Perangkat Lunak dan Gim* (PPLG). Because the generic phrase used in the query is not one of the official program names stored in the knowledge base, the correct retrieval illustrates the semantic matching that keyword-based approaches such as *Cosine Similarity* [2] cannot provide.

Figure 3. Chatbot Widget Integrated into the School's Official Website

Answer Quality Evaluation with RAGAS

Answer quality was evaluated by comparing the *Enhanced* and *Baseline* modes on the same 10 test questions. The results, averaged over three runs, are presented in Table 3.

Table 3. RAGAS Score Comparison of Enhanced and Baseline Modes

Metric	Baseline	Enhanced	Difference	Interpretation (Enhanced)
Faithfulness	0.9924	0.9812	−0.0112	Very Good
Answer Relevancy	0.8137	0.9146	+0.1009	Very Good
Context Precision	0.7000	0.7991	+0.0991	Good
Average Latency (s)	2.83	12.07	+9.24	-

As Table 3 shows, the Enhanced mode raises answer relevancy from 0.8137 to 0.9146 (a relative increase of 12.40%) and context precision from 0.7000 to 0.7991 (a relative increase of 14.16%), while faithfulness decreases slightly from 0.9924 to 0.9812 (−0.0112).

The rise in *answer relevancy* stems from the recall-broadening effect of *query expansion*. Generating paraphrastic variants surfaces documents that a single similarity search would miss—particularly when the user's phrasing diverges lexically from the source text—after which RRF consolidates documents that are consistently relevant across variants toward the top, and the cross-encoder makes a final, direct query-document relevance judgment. The generator therefore receives context more tightly aligned with the question's intent and produces answers that address the query more directly. In the *Baseline*, documents are ranked solely by single-query embedding similarity, so any lexical or semantic gap between the user's phrasing and the source wording caps the achievable relevance.

The rise in context precision follows directly from the two-stage retrieve-and-rerank design. RRF rewards documents that rank highly across multiple lists (cross-variant consensus), promoting robustly relevant documents, and the cross-encoder then re-scores query-document pairs with full attention, demoting superficially similar but off-topic chunks that a bi-encoder similarity search tends to over-rank. The net effect is that the top-ranked contexts are more precisely relevant to the ground truth—exactly what context precision measures.

The slight decline in *faithfulness* is a predictable side effect of the very mechanisms that raise the other two metrics. By admitting a broader and more diverse context set (three fused query variants and up to five reranked documents), the *Enhanced* pipeline gives the generator more, and more varied, material to synthesize; a marginally wider context increases the surface for bridging statements that are not strictly entailed by a single source passage, nudging *faithfulness* down. The *Baseline*'s narrower single-query context is more homogeneous, making near-verbatim grounding slightly easier. Crucially, the drop is only 0.0112 and both values remain in the "Very Good" band (> 0.98), so the exchange is heavily favorable: a negligible faithfulness cost buys double-digit gains in relevance and precision. This mirrors a general tension in RAG, in which recall-oriented retrieval enhancement broadens the evidence base at a small cost to strict grounding.

The latency overhead (12.07 s versus 2.83 s; +9.24 s) is the sum of the added stages executed largely in series: *query expansion* requires an extra LLM generation call; three similarity searches replace one (though these are parallelizable); RRF adds negligible CPU cost; and *cross-encoder reranking* is the dominant contributor, since it performs full pairwise query-document attention over the fused candidates and is far heavier than the *bi-encoder* similarity search. For an asynchronous SPMB information service this cost is acceptable, but it is the primary target for future optimization through GPU acceleration or caching of retrieval results.

Placed against prior research, these results are broadly consistent while adding nuance; Table 4 summarizes the comparison. Compared with Naufan [3], whose basic RAG obtained *faithfulness* 0.83 and *answer relevancy* 0.89, the *Enhanced* configuration here reaches 0.9812 and 0.9146; part of this gap is attributable to the added retrieval enhancements and a stronger multilingual embedding (*bge-m3*) plus cross-encoder reranking, although the different domain, knowledge base, and generator model (GPT-3.5 versus Gemini 3 Flash) make the comparison indicative rather than controlled. Compared with Samudra *et al.* [15], whose RAG health-information chatbot reported *faithfulness* 0.62 and *answer relevancy* 0.57, the values obtained here (0.9812 and 0.9146) are markedly higher, consistent with the view that retrieval enhancement together with a curated, well-structured knowledge base materially lifts answer relevance, notwithstanding differences in domain difficulty. Relative to the keyword-matching approach of Rafi *et al.* [2], the semantic RAG pipeline handles paraphrase and out-of-vocabulary phrasing that *cosine similarity* cannot, a qualitative gain in coverage. Finally, the findings corroborate Rackauckas [4] that RAG-Fusion improves answer quality over naive RAG, and extend that work by showing that asymmetric anchor weighting is needed to avoid precision degradation on small corpora; the near-ceiling *faithfulness* (> 0.98) also supports the premise in the broader RAG literature [5], [6] that grounding generation in retrieved context strongly suppresses hallucination.

Table 4. Quantitative Comparison with Prior RAG-Based Chatbot Studies

Study	Retrieval Method	Generator	Domain	Faith.	Ans. Rel.	Ctx. Prec.
Rafi *et al.* [2] (2025)	Cosine similarity (keyword matching)	–	Vocational school admission (SPMB)	n/r	n/r	n/r
Naufan [3] (2025)	Naive RAG (single query)	GPT-3.5 Turbo	Regional tax and finance	0.83	0.89	n/r
Samudra *et al.* [15] (2025)	Naive RAG	n/r	Health information	0.62	0.57	n/r
Rackauckas [4] (2024)	RAG-Fusion (query expansion + uniform RRF)	n/r	n/r	n/r	n/r	n/r
This study – Baseline	Naive RAG (single query)	Gemini 3 Flash Preview	Vocational school admission (SPMB)	0.9924	0.8137	0.7000
This study – Enhanced	Query expansion + weighted RRF (4:1) + cross-encoder reranking	Gemini 3 Flash Preview	Vocational school admission (SPMB)	0.9812	0.9146	0.7991

Note: n/r = not reported. Scores are reproduced as reported by each study. Because the studies differ in domain, corpus, test-set size, judge configuration, and generator model, the comparison is indicative rather than controlled. Samudra *et al.* [15] additionally report a *Mean Reciprocal Rank* of 0.93, a retrieval metric that is not directly comparable to *context precision*; Rackauckas [4] evaluates RAG-Fusion without reporting RAGAS scores.

Overall, these results carry several implications for RAG method development. On small, curated, domain-specific knowledge bases, *faithfulness* tends to saturate near its ceiling, so the binding constraint on answer quality is retrieval relevance; engineering effort is therefore best directed at the retrieval stack (expansion, fusion, and *reranking*) rather than at the generator. The observed *trade-off*—higher relevance and precision at the cost of a small *faithfulness* drop and substantial latency—implies that configuration should be service-aware: latency-tolerant, accuracy-critical services such as SPMB information favor the *Enhanced* pipeline, whereas latency-critical settings may prefer the *Baseline* or a lighter variant. However, because these gains are the joint effect of three techniques, isolating the specific contribution of RRF requires an ablation study, so the improvement should not yet be attributed to RRF alone. Moreover, since the evaluation used 10 questions without a statistical significance test, the magnitude of the improvement should be read as an indication of the direction of change rather than an exact figure.

5. Conclusion

This study successfully developed an NLP-based *chatbot* using the RAG and RRF methods to automate the SPMB information service on the website of SMK Muhammadiyah 1 Wonosobo. The architecture consists of an *indexing* phase (document loading, *chunking*, and *embedding* with *BAAI/bge-m3* into a *FAISS vector store*) and a six-stage *inference* phase (*query expansion*, *similarity search*, *Reciprocal Rank Fusion*, *cross-encoder reranking*, *prompt assembly*, and answer generation with Gemini 3 Flash Preview), drawing on a knowledge base of 44 official school text documents.

Applying the three retrieval-enhancement techniques contributed positively to document relevance, shown by an increase of 14.16% in *context precision* and 12.40% in *answer relevancy* relative to the *Baseline*. The *chatbot* was successfully integrated into the school's official website through an *embedded widget* (a *floating bubble*), making it accessible at any time. In testing, *Black Box Testing* over 17 scenarios achieved 100% functional success, and RAGAS evaluation obtained *faithfulness* 0.9812, *answer relevancy* 0.9146, and *context precision* 0.7991, with the first two in the "Very Good" category and *context precision* in the "Good" category. These gains were measured on a limited test set and are the joint effect of the three techniques, so they should be interpreted as an early indication.

This study has several limitations that constrain the generalization of the results: no ablation study was conducted to isolate the contribution of RRF on its own; the 4:1 RRF weighting was selected on the same 10 questions as the final evaluation, without a separate validation split; the RAGAS evaluation used an LLM judge without designating a judge model distinct from the answer generator; the number of test questions is relatively small (10) and reported without a statistical significance test; and the *Enhanced* mode's latency (±12 s) remains high. Accordingly, future work should conduct an ablation study to isolate each technique's contribution, use a judge model distinct from the answer generator, enlarge the test-question set with significance testing, and optimize response time through, for example, GPU acceleration or caching of retrieval results.

Acknowledgements

The authors thank SMK Muhammadiyah 1 Wonosobo for its data support and research opportunity, in particular Mr. Ari Setiawan, S.Kom., Staff of the Vice Principal, who assisted in validating the knowledge-base data and the ground-truth answers, and the Informatics Engineering Study Program, Universitas Sains Al Qur'an Jawa Tengah di Wonosobo.

6. References

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713–3744, 2023, doi: 10.1007/s11042-022-13428-4.

[2] R. R. Rafi, A. Setiawan, A. A. Rosadi, and A. Sanjaya, "Rancang bangun chatbot penerimaan murid baru berbasis cosine similarity pada SMK Intensif Baitussalam," *Prosiding Seminar Nasional Teknologi dan Sains*, vol. 4, 2025.

[3] F. M. Naufan, "Pengembangan chatbot AI berbasis NLP dengan metode Retrieval-Augmented Generation (RAG) untuk otomatisasi layanan informasi pajak dan keuangan daerah di website BPPKAD Kabupaten Wonosobo," Undergraduate thesis, Universitas Sains Al-Qur'an Jawa Tengah di Wonosobo, 2025.

[4] Z. Rackauckas, "RAG-Fusion: a new take on retrieval-augmented generation," *International Journal on Natural Language Computing*, vol. 13, no. 1, pp. 37–47, 2024, doi: 10.5121/ijnlc.2024.13103.

[5] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: a survey," *arXiv preprint*, 2023, doi: 10.48550/arXiv.2312.10997.

[6] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, doi: 10.48550/arXiv.2005.11401.

[7] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. European Chapter of the ACL (EACL): System Demonstrations*, 2024, doi: 10.48550/arXiv.2309.15217.

[8] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York: McGraw-Hill Education, 2020.

[9] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," in *Findings of the ACL 2024*, 2024, doi: 10.48550/arXiv.2402.03216.

[10] M. Douze *et al.*, "The Faiss library," *arXiv preprint*, 2024, doi: 10.48550/arXiv.2401.08281.

[11] G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. 32nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2009, pp. 758–759, doi: 10.1145/1571941.1572114.

[12] L. Wang, N. Yang, and F. Wei, "Query2doc: query expansion with large language models," in *Proc. 2023 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2023, doi: 10.48550/arXiv.2303.07678.

[13] C. Li, Z. Liu, S. Xiao, Y. Shao, and D. Lian, "Llama2Vec: unsupervised adaptation of large language models for dense retrieval," in *Proc. 62nd Annual Meeting of the ACL*, 2024, doi: 10.48550/arXiv.2312.15503.

[14] S. Nidhra and J. Dondeti, "Black box and white box testing techniques – a literature review," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, 2012, doi: 10.5121/ijesa.2012.2204.

[15] G. Samudra, A. Turmudi Zy, and Ermanto, "Implementasi Retrieval-Augmented Generation (RAG) pada chatbot layanan informasi kesehatan," *JSAI: Journal Scientific and Applied Informatics*, vol. 8, no. 1, 2025, doi: 10.36085/jsai.v8i1.7678.