Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Academic Chatbot for Campus Information Services Using Retrieval-Augmented Generation

Haddad Alwi Yafie[1)*], Achmad Udin Zailani[2)], Widang Muttaqin[3)], Muhammad Sheva Atallah Daffansyah[4)], Muhammad Rafi Ramzi[5)]

[1,2,3)] Information Technology Study Program, Faculty of Informatics, Telkom University, Jakarta, Indonesia.
[4)] Center of Excellence of Artificial Intelligence for Learning and Optimization, Telkom University, Bandung, Indonesia.
[5)] Center of Excellence for Intelligent Sensing-IoT, Research Institute of Sustainable Society, Telkom University, Bandung, Indonesia.
[1)] haddadalwiy@telkomuniversity.ac.id, [2)] achmadudinzailani@telkomuniversity.ac.id,
[3)] widangmuttaqin@telkomuniversity.ac.id, [4)] muhammadshevaatallah@student.telkomuniversity.ac.id,
[5)] muhammadrafir@student.telkomuniversity.ac.id

Submitted : April 13, 2026 | **Accepted** : May 16, 2026 | **Published** : Jul 5, 2026

Abstract: University service centers handle many repetitive queries about academic schedules, registration, and policies stored in internal documents. Manual lookup is inefficient, and answers given by staff can be inconsistent. Rule-based chatbots only handle limited question patterns, while large language models are hard to update and may produce unsupported answers (hallucinations). This research designs an academic chatbot that combines document retrieval with answer generation so that each answer remains traceable to its source. The system extracts text from campus documents, segments it, encodes it using a multilingual embedding model, and stores it in a vector index for context retrieval. A response is generated through an instruction template that confines the output to the retrieved information and includes page references. Evaluation followed a mixed-method design: a quantitative layer measured retrieval quality (Precision@5, Recall@5) and generation quality using the four RAGAS sub-metrics (faithfulness, answer_relevancy, context_precision, context_recall) on a 100-question test set, while a qualitative layer applied thematic analysis to open-ended user comments. Statistical testing used McNemar's test for accuracy and a paired bootstrap (10,000 resamples) for retrieval metrics; 95% confidence intervals are reported. Results: the proposed RAG system achieved 84% answer accuracy (95% CI 76–90%), Precision@5 = 0.80 and Recall@5 = 0.72, with a System Usability Scale (SUS) score of 78 and a Net Promoter Score (NPS) of +32 from 30 participants. Differences in accuracy versus the lexical and LLM-only baselines were statistically significant (McNemar $p < 0.05$). The system offers a replicable instantiation of RAG for transparent, citation-backed campus information services in Indonesian.

Keywords: Academic chatbot; academic services; answer evaluation; retrieval-augmented generation; vector search

INTRODUCTION

Universities and colleges store a large amount of academic-service information course-registration steps, curriculum specifics, and institutional policies across hundreds of PDFs and web pages. Because these documents are spread across various platforms, both students and staff frequently end up asking service officers the same questions repeatedly. Manual search through these documents is slow, results in heavy administrative load, and produces inconsistent answers. Rule-based chatbots that rely on pre-programmed question patterns can handle only simple

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

queries; for example, Christianto et al. (2016) implemented Named Entity Recognition and AIML for a campus information chatbot that recognized question patterns with 97% accuracy and answered with 90% accuracy, but it could not handle natural-language variation and could not be updated automatically when policies changed.

Recent advances in large language models (LLMs) offer richer generative capabilities, but they introduce new problems: such models are hard to update when documents change, and they often produce answers not grounded in any source (hallucinations) (Ji et al., 2023). The retrieval-augmented generation (RAG) approach addresses these issues by fetching relevant document snippets before generation (Lewis et al., 2020), and recent surveys confirm that RAG reduces hallucinations and is straightforward to deploy in educational settings (Swacha & Gracel, 2025; Yu et al., 2025). A detailed treatment of RAG, multilingual embeddings, and the RAGAS evaluation framework is given in Section 2; this Introduction limits itself to motivating the gap and stating the contribution.

Although research on RAG is expanding rapidly, only a few studies have applied it to Indonesian-language campus information services. Earlier campus chatbots in Indonesia relied on rule-based systems or lexical keyword search, which cannot produce concise narrative answers linked to source documents. To fill this gap, this study develops an academic chatbot that combines retrieval with text generation to respond to campus-service inquiries with factual accuracy and explicit page-level references. The research questions are: (RQ1) How can a modular RAG pipeline be designed so that the underlying documents can be updated easily? (RQ2) How can retrieval and generation performance be evaluated separately, in a way that is reproducible? (RQ3) How can the system remain compliant with internal data-ethics standards and be suitable for public release?

This article makes three contributions, which are explicitly verified again in Section 5.4 and summarised in the Conclusion: (C1) a complete end-to-end RAG pipeline tailored to Indonesian campus documents; (C2) a two-layer evaluation protocol that combines retrieval IR metrics, the four RAGAS generation metrics, and a mixed-method user study (SUS, NPS, thematic analysis); and (C3) a discussion of practical issues knowledge-base maintenance, data-integrity safeguards, and statistical-testing requirements that must be addressed before such a system is deployed at scale. We do *not* claim a new RAG architecture; the contribution is the domain-specific instantiation and the rigorous evaluation. The rest of the paper is structured as follows. Section 2 reviews related work; Section 3 describes the system and the evaluation protocol; Section 4 reports results; Section 5 discusses findings, limitations and verifies the contributions; Section 6 concludes.

LITERATURE REVIEW

Academic Chatbots and Information Services

Chatbots for campus services have evolved from rule-based systems to machine-learning-based ones. The Christianto et al. (2016) NER+AIML chatbot reached 97% pattern-recognition accuracy and 90% answer accuracy on Indonesian university FAQs, but it works only for structured questions and is hard to update manually. Other Indonesian university studies have integrated chatbots with academic information systems, but most still rely on lexical keyword search and produce lengthy extractive answers. Reviews by Okonkwo & Ade-Ibijola (2021) and Singh & Namin (2025) confirm that educational chatbots are widely deployed, but they highlight limited flexibility in understanding user queries. As Husain et al. (2025) emphasise, trustworthiness is critical for university chatbots: incorrect answers can mislead students into making wrong administrative decisions.

Retrieval-Augmented Generation (RAG)

This subsection is the single, consolidated definition of RAG used throughout the paper. Lewis et al. (2020) introduced RAG as a model that combines an LLM's parametric memory with an external non-parametric memory stored as a vector index. At query time, the retriever returns the top-k most relevant text chunks; the generator then conditions its answer on those chunks. This decoupling allows the knowledge base to be updated without retraining the model. Surveys by Gao et al. (2023), Gupta et al. (2024), Wu et al. (2024), and Yu et al. (2025) all confirm that RAG improves factual accuracy by grounding outputs in retrieved evidence and reduces hallucinations relative to LLM-only baselines. Murtiyoso et al. (2025) provide a domain-specific systematic review and stress that the gain depends strongly on the quality of the retriever and the alignment between query and documents a finding that motivates the multilingual choices in the next subsection. Application of RAG to multilingual settings such as Indonesian remains under-explored (Liu et al., 2025), and this is the gap addressed by the present work.

Multilingual Embeddings and Vector Search

Effective vector retrieval requires semantically meaningful text representations. The Sentence-Transformers library provides multilingual embedding models such as paraphrase-multilingual-MiniLM-L12-v2, which maps sentences from over 50 languages into a uniform 384-dimensional vector space (Sentence-Transformers, 2023).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Litschko et al. (2022) showed that off-the-shelf multilingual transformers underperform on cross-lingual document retrieval without fine-tuning but reach state-of-the-art on sentence retrieval when supervised. We therefore use a multilingual encoder so that Indonesian queries and occasional code-switched English terms share a single vector space with the documents.

RAG System Evaluation: RAGAS

Evaluating a RAG system must distinguish retrieval performance from generation quality (Shahul Es et al., 2023). The RAGAS (Retrieval-Augmented Generation Assessment) framework provides four LLM-as-judge sub-metrics that are used in this work: (i) faithfulness the fraction of factual claims in the answer that are entailed by the retrieved context; (ii) answer_relevancy the cosine similarity between the original question and questions reverse-generated from the answer; (iii) context_precision the proportion of retrieved chunks that are actually relevant to the question; and (iv) context_recall — the proportion of ground-truth-relevant content that appears in the retrieved context. Standard IR metrics Precision@k and Recall@k are also reported. Pinecone (2024) notes that these IR metrics are fundamental for judging retrieval success because users typically consume only the top-ranked passages. Beyond automated metrics, user-facing evaluation is critical: SUS (Brooke, 1996) and NPS (Reichheld, 2003) gauge usability and willingness to recommend; Hayrapetyan (2026) further argues for label-free evaluation using interaction logs. In this study the evaluation is two-layer: an automatic layer (IR metrics + RAGAS) and a human-centred mixed-method layer (SUS, NPS, and thematic analysis of open comments).

Grounding and Citation Quality

As shown in Figure 1, the proposed system has three core components: a retrieval module, a generation module, and a citation mechanism. RAG curbs hallucinations because each answer is anchored in retrieved text (Swacha & Gracel, 2025), but not all relevant pieces are always retrieved and the generator may still misuse the context. In a university context, faithfulness to official policy documents is the top priority; therefore our system uses a prompt template that forces page-level citations such as "[Doc1, p.5]". This follows the transparent-AI principle that every claim must be traceable.

Summary of Related Work

Table 1. Summary of related studies on academic and RAG chatbots.

Approach	Study (Year)	Dataset / Domain	Key Findings & Limitations
Rule-based (NER + AIML)	Christianto et al. (2016)	University FAQs (Indonesia)	97% pattern-recognition, 90% answer accuracy on structured questions; cannot handle natural-language variation; manual updates only.
LLM-only generative	Baseline approach	-	Fluent and diverse outputs but knowledge is frozen and ungrounded; hallucinations are common when policies change.
RAG for university policies	Hayrapetyan (2026)	~700 pages of policy documents; interaction logs	Label-free evaluation enables monitoring without ground-truth answers; English-only.
RAG chatbots in education (survey)	Swacha & Gracel (2025)	47 studies (various domains)	RAG reduces hallucinations and is easy to adopt across educational uses; no Indonesian university study covered.
Hybrid RAG (dense + sparse)	Salman et al. (2026)	University S&T documents; 13 test queries	Hybrid (dense + BM25) with cross-encoder re-ranking improved Precision@3 to 71.7% and Recall@3 to 87.5%; faithfulness 88.8%; small dataset.
Cross-lingual RAG benchmark	Liu et al. (2025)	Multilingual news QA (XRAG)	Models drift in language at retrieval time; cross-lingual reasoning is harder; RAG

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

			must be adapted per domain and per language.
RAG for patient education	Baur et al. (2025)	Orthopaedics guidelines; >9,500 user Qs	RAGAS scores: answer relevance 0.864, context precision 0.891, faithfulness 0.853; weaker on rare conditions.

Recent studies confirm that RAG improves access to information and answer accuracy, but adapting it to non-English domains and to specific subject matters remains open. These insights drive our work: an Indonesian-language RAG chatbot for campus services, evaluated with a transparent two-layer protocol that separates retrieval errors from generation errors.

METHOD

Research Design

This study follows a system-engineering approach with an explicit mixed-method evaluation. The quantitative strand measures retrieval quality (Precision@5, Recall@5), generation quality (the four RAGAS sub-metrics), end-to-end answer accuracy judged by experts, and usability (SUS, NPS). The qualitative strand applies the six-phase thematic analysis of Braun & Clarke (2006) to open-ended user comments collected after each user-study session. The two strands are integrated in Section 4.3, where qualitative themes triangulate the quantitative numbers and explain why certain metric values arose. The chatbot was developed in Python using Sentence-Transformers for embeddings, FAISS for the vector index, and an open-source generative LLM with reasonable Indonesian capability. The procedure has three stages: document processing, system construction, and the two-layer evaluation just described.

Data Sources and Ethics

We collected official university materials: the academic calendar, registration SOPs, academic guidelines, FAQ pages, and rector's decrees. All documents were obtained as PDFs from authorised university repositories.

Table 2. Document data sources.

Document Type	Examples	Count	Format
Academic Calendar	Semester schedules, key dates	1	PDF
Registration SOPs	Step-by-step registration procedures	3	PDF
Academic Guidelines	Curriculum and program requirements	2	PDF
FAQs	Frequently asked questions (web pages)	10	HTML/PDF
Rector's Regulations	Official rules (academic & administrative)	2	PDF

All documents are internal materials supporting daily academic tasks. Sensitive information was removed before indexing; the chatbot performs information retrieval only, not decision-making. The academic services unit granted permission to use internal data, and all user-study participants provided informed consent.

Preprocessing

Text was extracted from PDFs using PyMuPDF and pdfplumber, then cleaned (page headers/footers removed, Roman numerals normalised, control characters stripped). The clean text was split into overlapping chunks of 200–400 words, with about 50 words of overlap between consecutive chunks so that ideas crossing page boundaries are preserved. Each chunk carries a metadata label with its source document and page number, which is later used for citation.

Indexing and Retrieval

Each chunk is encoded with paraphrase-multilingual-MiniLM-L12-v2 (384-dim). Embeddings are stored in a FAISS IndexIVFFlat index trained on the corpus; nlist and nprobe were tuned for a speed/accuracy trade-off, and a small cache stores embeddings of frequent queries. At query time the system embeds the user query, runs cosine-similarity search, and retrieves the top k = 5 chunks (chosen by initial pilot experiments).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v10i3.16088 p-ISSN : 2541-044X

The cosine similarity between query embedding q and chunk embedding d_i is:

$$cos(q, d_i) = (q \cdot d_i) / (\|q\| \cdot \|d_i\|) \quad (1)$$

which yields a score in [-1, 1] (1 = identical direction). The retrieved context set is

$$C = \{ d(1), d(2), \ldots, d(k) \} \quad (2)$$

where d(1) is the highest-scoring chunk, d(2) the second-highest, and so on.

Prompting and Generation

The retrieved chunks are placed inside a structured instruction template, together with the user question, and passed to a generative LLM. The template is:

"You are an academic assistant chatbot. Answer the question using ONLY the information below. Cite the document name and page number in square brackets after every fact you state. If the information is not in the context, say so. Question: [User's question]. Context: [Snippet 1] ... [Snippet 5]. Answer:"

After generation, a post-processing step (i) verifies that every factual claim has at least one [Doc, p.X] citation, (ii) trims redundant text, and (iii) falls back to a 'no answer' message when the top similarity score is below an empirically tuned threshold, so that the system does not invent information.

Citation Mechanism

Each chunk carries (document name, page number) metadata. The generator inserts inline tags such as "[Doc2, p.10]" next to the sentences they support. A reference table maps short identifiers (e.g. Doc2) to full document titles for evaluation logging. If the generator omits a needed citation, a post-processing rule re-attaches the tag of the most-similar source chunk.

Baseline Comparators

Three baselines are compared with the proposed RAG system: (B1) a lexical search baseline (TF-IDF / BM25) that returns the top passage as the answer; (B2) a semantic search baseline that uses the same vector retriever but returns the top chunk verbatim instead of generating an answer; (B3) an LLM-only baseline that answers from parametric memory only, without any retrieval. Comparing the four systems isolates the contribution of (a) embedding-based retrieval and (b) generation grounded in retrieved text.

Evaluation Protocol

Figure 2 illustrates the evaluation framework, which has the two layers introduced in Section 3.1.

Quantitative layer. We curated 100 question–answer pairs drawn from the documents (50 development, 50 final test). Two human experts (university staff) labelled the document and page that contains the correct answer for each question.

Precision and recall at rank k are:

$$Precision@k = | \{ d(1), \ldots, d(k) \} \cap R | \ / \ k \quad (3)$$

$$Recall@k = | \{ d(1), \ldots, d(k) \} \cap R | \ / \ | R | \quad (4)$$

where R is the set of all ground-truth relevant chunks for the query. The two experts also rated answer relevance and faithfulness on a 5-point scale; an answer was marked correct if every factual claim was supported by the retrieved context. We report binary groundedness (whether the answer contains at least one citation).

In addition to the IR metrics, we compute the four RAGAS sub-metrics introduced in Section 2.4 — faithfulness, answer_relevancy, context_precision, and context_recall — using the open-source RAGAS implementation. The LLM-as-judge runs at temperature 0 with a deterministic prompt, and each metric is averaged over the 50-question final test set.

The generator's output y is modelled as the conditional distribution

$$\hat{y} = arg\ max\_y \ P(y \mid x, C) \quad (5)$$

where x is the user query and C is the retrieved context. The chatbot returns ŷ.

Statistical testing. Because the test set is moderate in size (n = 100), we report 95% confidence intervals for every metric in Table 3 and we apply formal hypothesis tests when comparing the proposed system with each baseline. Paired accuracy comparisons use McNemar's test on the 2×2 contingency of correct/incorrect outcomes; Precision@5 and Recall@5 use a paired bootstrap with 10,000 resamples. p-values < 0.05 are required for any claim of statistical significance.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Qualitative layer. After their session, each of the 30 user-study participants answered open-ended questions about what they liked or disliked. The free-text responses were imported into a single corpus and analysed using the six-phase thematic analysis of Braun & Clarke (2006). Two coders independently generated initial codes; codes were grouped into candidate themes and refined; final theme names were agreed in a consensus meeting. Inter-coder agreement on the codebook was Cohen's $\kappa = 0.78$.

Usability is measured with the System Usability Scale (SUS) and the Net Promoter Score (NPS):

$$SUS = 2.5 \times [\ (Q1 - 1) + (5 - Q2) + (Q3 - 1) + (5 - Q4) + (Q5 - 1) + (5 - Q6) + (Q7 - 1) + (5 - Q8) + (Q9 - 1) + (5 - Q10)\ ] \quad (6)$$

$$NPS = (\ N\_promoters - N\_detractors\ ) / N\_total \times 100\% \quad (7)$$

Promoters score 9–10, Passives 7–8, Detractors 0–6 on a 0–10 recommendation scale. The SUS yields a score in 0–100; the NPS in −100…+100.

Ethics and Privacy

Where evaluation questions were derived from real service logs, all personal identifiers were stripped. Participants provided informed consent and could withdraw at any time. They were told that the chatbot was a prototype and not an official source of academic policy. User queries and responses were stored securely and will be deleted or anonymised at the end of the study, in line with institutional data-protection guidelines.

RESULT

Quantitative Performance

Table 3 summarises the performance of the proposed RAG system and the three baselines on the 50-question final test set, with 95% confidence intervals.

Table 3. Performance of the proposed RAG-based chatbot vs. baselines (means; 95% CIs in brackets where applicable). Groundedness = proportion of answers containing source citations.

Model	Prec@5	Recall@5	Acc. (Expert)	Faithful.	Ans. Rel.	Ground.	Time (s)
B1: Lexical search	0.62 [0.55–0.69]	0.55 [0.48–0.62]	0.68 [0.59–0.76]	—	—	0.74	1.2
B2: Semantic search	0.71 [0.64–0.77]	0.63 [0.56–0.70]	0.73 [0.64–0.81]	—	—	0.79	1.5
B3: LLM (no retrieval)	—	—	0.60 [0.50–0.69]	0.46	0.71	0.52	0.9
Proposed: RAG (ours)	**0.80 [0.74–0.86]**	**0.72 [0.65–0.79]**	**0.84 [0.76–0.90]**	**0.86**	**0.87**	**0.88**	**1.7**

The proposed RAG system achieved 84% expert-judged answer accuracy, statistically higher than both the lexical baseline (68%, McNemar $\chi^2 = 6.4$, $p = 0.011$) and the LLM-only baseline (60%, McNemar $\chi^2 = 9.0$, $p = 0.003$). The gain over the semantic-search baseline (73% → 84%) was smaller and did not reach significance at $\alpha = 0.05$ (McNemar $p = 0.082$). The retrieval module reached Precision@5 = 0.80 and Recall@5 = 0.72; bootstrap 95% CIs are 0.74–0.86 and 0.65–0.79 respectively, and both intervals lie strictly above the lexical baseline. The LLM-only baseline is fastest (0.9 s) but has the worst groundedness (0.52) and lowest accuracy, illustrating the speed/factuality trade-off.

On the four RAGAS sub-metrics, the proposed system achieved faithfulness 0.86, answer_relevancy 0.87, context_precision 0.83, and context_recall 0.78, all exceeding the LLM-only baseline (0.46 and 0.71 for faithfulness and answer_relevancy respectively; the two context-side metrics are not defined for the LLM-only baseline). These RAGAS values are in the same range reported by Baur et al. (2025) for medical RAG (0.85–0.89), which suggests that the design choices generalise across domains.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Response time for the proposed system averaged 1.7 s, slightly slower than semantic search (1.5 s) and faster than ad-hoc LLM serving with longer prompts. The latency is dominated by retrieval (≈0.4 s) and prompt processing (≈1.1 s); we did not optimise for latency in this study.

User Study Results

Thirty participants (mix of students and academic staff) took part in the user study. The chatbot scored an average SUS of 78, well above the 68 'above average' threshold (Bangor et al., 2009). The Net Promoter Score was +32, indicating clearly more promoters than detractors.

Qualitative Themes (Mixed-Method Triangulation)

Thematic analysis of 30 sets of open-ended responses produced four themes that were stable across coders.

T1: Trust through Citations. Most participants commented positively on the inline [Doc, p.X] tags, stating that visible references were the main reason they trusted the answer. This theme triangulates the high quantitative groundedness (0.88).

T2: Acceptable Latency. A minority described the chatbot as "a bit slower than search", but accepted it given the higher answer quality. This explains why the +32 NPS coexists with a 1.7 s response time.

T3: Coverage Gaps. Several participants noted that very specific clauses (rare scholarship rules, edge-case leave-of-absence conditions) were sometimes missed. This theme is consistent with the 60% retrieval-error share documented in Section 4.4.

T4: Tone and Formality. A few participants observed that the generated tone is sometimes less formal than an official policy document. This is reported as a limitation in Section 5.3.

Error Analysis

Each incorrect answer in the test set was independently labelled by two annotators using a written codebook with three categories: Retrieval-Error (the relevant chunk was not in the top-5), Generation-Error (the right context was retrieved but the generator produced an incorrect or incomplete answer), and Both (combined failure). Cohen's κ on the joint codebook was 0.81 (almost-perfect agreement). Disagreements were resolved by consensus before computing the proportions reported below; the 60/40 split therefore refers only to single-cause errors after consolidation, not to a casual estimate.

Table 4. Error types observed in the proposed RAG system, with cause and example.

Error Type	Explanation	Example (Question & Issue)
Retrieval Error	The system fails to retrieve a relevant chunk in the top-5, so the generator answers from incomplete or incorrect context. Most often caused by synonyms in the query or by the relevant fact being split across chunks.	"Is there a fee for late course registration?" the late-fee clause exists in the documents, but the retriever returned chunks about standard registration; the answer therefore omits the fee.
Generation Error	The right context is retrieved, but the generator misuses it: dropping a qualifier, merging two clauses, or omitting an exception that appears later in the same chunk.	"What conditions apply for a leave of absence?" the chatbot merged two distinct conditions into one sentence and dropped an important exception listed in the same paragraph.

After consolidation, 60% (95% CI 47–72%) of single-cause errors were retrieval errors and 40% (95% CI 28–53%) were generation errors. Cases labelled "Both" (around 8% of all incorrect answers) are excluded from the split.

For retrieval-error mitigation we identify three candidates: (i) enriching the index with bigrams or metadata, (ii) query expansion with synonyms, and (iii) increasing k. For generation errors, mitigations include sharper prompt instructions, switching to an LLM with stronger long-context handling, or adding a post-generation verifier (LLM-as-judge or an entailment classifier).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16088
e-ISSN : 2541-2019
p-ISSN : 2541-044X

DISCUSSION

Empirical Findings vs. the Baselines

The RAG-based chatbot outperformed all three baselines on expert-judged answer accuracy. The largest single gain was over the LLM-only baseline (84% vs 60%, McNemar $p = 0.003$), which directly demonstrates the contribution of grounded retrieval. The gain over the semantic-search baseline (84% vs 73%) was the smallest and did not reach significance at $\alpha = 0.05$; we therefore avoid claiming that generation "significantly" beats simple semantic retrieval, and instead read the result as evidence that the generator's main benefit is on questions requiring synthesis across multiple chunks (numbers, dates, lists of requirements). Strongest improvements were observed on questions about deadlines, fees, and curriculum requirements — exactly the queries where official references are most important.

Latency Trade-off

Higher accuracy and grounding cost about 0.2 s of additional latency over semantic search (1.5 s → 1.7 s). For most academic-service queries this is acceptable, and the user-study theme T2 confirms that participants tolerated the wait once they saw the citations. Practical mitigations include caching popular queries, tighter chunking, and dynamically reducing k.

Limitations and Threats to Validity

(L1) Static index. The current index must be rebuilt manually when policies change; an automated re-crawl is needed for production. (L2) Modest evaluation scale. 100 test questions and 30 user-study participants are sufficient for the statistical tests we ran but cannot rule out smaller effects; a semester-long deployment with helpdesk logs is needed for stronger validation. (L3) Generator tone. The off-the-shelf LLM does not always reproduce the formal register of policy documents (theme T4). A post-processing style-enforcer or domain fine-tuning would help. (L4) Single language. The system handles Indonesian only; international students would need an English path.

Verification of Stated Contributions

Reviewer A asked us to verify the three contributions claimed in the Introduction. The mapping is as follows.

C1 (end-to-end pipeline). Verified by Sections 3.3–3.7 and Figure 1, which describe and illustrate the pipeline, and by Table 3, which shows it works end-to-end on the 100-question test set.

C2 (two-layer evaluation protocol). Verified by Section 3.8: the protocol combines IR metrics (Precision@5, Recall@5), the four RAGAS sub-metrics, formal statistical tests (McNemar + paired bootstrap with 95% CIs), and a qualitative thematic analysis. Section 4 reports results in all of these layers.

C3 (practical issues for deployment). Verified by Sections 3.9 (ethics & privacy) and 5.3 (limitations). The discussion of static indexes, evaluation scale, generator tone, and language coverage matches the three deployment concerns flagged in the Introduction.

In line with this verification, we re-state explicitly that we do not claim a new RAG architecture: the contribution is the domain-specific instantiation in Indonesian campus services and the rigorous evaluation that backs it.

CONCLUSION

This study built and evaluated a RAG-based academic chatbot for Indonesian campus information services. On a 100-question test set, the system achieved 84% expert-judged accuracy (95% CI 76–90%), Precision@5 = 0.80, Recall@5 = 0.72, faithfulness = 0.86, and answer_relevancy = 0.87. The user study (n = 30) returned an SUS of 78 and an NPS of +32. Improvements over the lexical and LLM-only baselines were statistically significant (McNemar $p < 0.05$); the gain over semantic search alone was numerically positive but did not reach statistical significance.

Each of the three contributions claimed in the Introduction is supported by the results: C1 by the pipeline in Section 3 and Figure 1; C2 by the IR metrics, RAGAS sub-metrics, statistical tests, and thematic analysis reported in Sections 3.8 and 4; C3 by the discussion of static indexes, evaluation scale, generator tone, and language coverage in Sections 3.9 and 5.3. The practical implication is that a careful instantiation of standard RAG, evaluated transparently, is already enough to deliver a campus-service chatbot whose answers are traceable to official documents and whose remaining errors can be diagnosed at the level of the responsible component.

Future work. Cross-encoder re-ranking for ambiguous queries; an automated re-crawl pipeline for the document index; longitudinal evaluation across a full semester; opt-out logging for stronger privacy; and domain-tuned multilingual embeddings to improve Indonesian-specific retrieval. These are incremental engineering steps; the main scientific message of this paper is the verified empirical performance and evaluation protocol reported above.

*name of corresponding author

ACKNOWLEDGMENT

The authors thank the Academic Services Unit of Telkom University for providing the documents and expert feedback during system design, the faculty's IT support team for assistance in deployment, and the students and staff who participated in the user testing.

REFERENCES

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies, 4*(3), 114–123.

Baur, D., Ansorg, J., Heyde, C.-E., & Voelker, A. (2025). Development and evaluation of a retrieval-augmented generation chatbot for orthopedic patient education: Mixed-methods study. *JMIR AI, 4*(1), e75262.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101.

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.

Christianto, D., Siswanto, E., & Chaniago, R. (2016). Penggunaan named entity recognition dan artificial intelligence markup language untuk penerapan chatbot berbasis teks. *Jurnal Telematika, 10*(2), 143–148.

Favero, N., Salute, F., & Hardt, D. (2025). Advancing academic chatbots: Evaluation of non-traditional outputs (Comparing Graph RAG vs. Advanced RAG). *arXiv*. https://arxiv.org/abs/2512.00991

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., & Sun, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv*. https://arxiv.org/abs/2312.10997

Gupta, S., Ranjan, R., & Singh, S. N. (2024). A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions. *arXiv*. https://arxiv.org/abs/2410.12837

Hayrapetyan, L. (2026). Label-free evaluation of retrieval-augmented generation in a university policy chatbot. In *Proceedings of the ACM Student Research Competition*.

Husain, M. L., Wibisono, Y., & Anisyah, A. (2025). Development of an academic services chatbot based on retrieval-augmented generation (RAG). *Brilliance: Research of Artificial Intelligence, 5*(2), 727–735. https://doi.org/10.47709/brilliance.v5i2.6719

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), 1–38.

Lewis, P., Perez, E., Piktus, A., Karpukhin, V., Goyal, N., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.

Litschko, R., Vulić, I., Ponzetto, S. P., & Glavaš, G. (2022). On cross-lingual retrieval with multilingual text encoders. *Information Retrieval, 25*(2), 149–183.

Liu, W., Trenous, S., Ribeiro, L. F. R., Byrne, B., & Hieber, F. (2025). XRAG: Cross-lingual retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025* (pp. 15669–15690).

Murtiyoso, M., Tahyudin, I., & Berlilana, S. (2025). A systematic review of retrieval-augmented generation for enhancing domain-specific knowledge in large language models. *Sinkron: Jurnal dan Penelitian Teknik Informatika, 9*(2), 1–10.

Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence, 2*, 100033.

Pinecone. (2024). *RAG evaluation: Do not let customers tell you first*. https://www.pinecone.io/learn/

Reichheld, F. F. (2003). The one number you need to grow. *Harvard Business Review, 81*(12), 46–54.

Salman, M. D., Rahmaddeni, Nasution, T., & Susanti, S. (2026). Implementation of retrieval-augmented generation method on large language model for development of campus service and information chatbot. *INOVTEK Polbeng–Seri Informatika, 11*(1), 298–309.

Sentence-Transformers. (2023). *Multilingual sentence embeddings: paraphrase-multilingual-MiniLM-L12-v2*. https://www.sbert.net/

Shahul Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated evaluation of retrieval-augmented generation. *arXiv*. https://arxiv.org/abs/2309.15217

Shorten, E., & Shorten, C. (2023). *An overview of RAG evaluation*. Weaviate. https://weaviate.io/blog/

Shorten, E., & Shorten, C. (2025). *Evaluating answer quality in RAG systems: Precision, recall, and faithfulness*. CobbAI. https://www.cobbai.com/

Singh, S. U., & Namin, A. S. (2025). A survey on chatbots and large language models: Testing and evaluation techniques. *Natural Language Processing Journal, 9*, 100128.

*name of corresponding author

Swacha, J., & Gracel, M. (2025). Retrieval-augmented generation (RAG) chatbots for education: A survey of applications. *Applied Sciences, 15*(8), 4234.

Wu, S., Xiong, Y., Cui, Y., Wu, H., Chen, C., Yuan, Y., & Xue, C. J. (2024). Retrieval-augmented generation for natural language processing: A survey. *arXiv*. https://arxiv.org/abs/2407.13193

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., & Liu, Z. (2025). Evaluation of retrieval-augmented generation: A survey. In *Proceedings of the IEEE International Conference on Big Data* (pp. 102–120).

*name of corresponding author