Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

Comparative Evaluation of Local Llama and Mistral Models in RAG for Helpdesk Documentation

Septian Pratama*, Sajarwo Anggai, Murni Handayani

Magister of Informatics Engineering, Universitas Pamulang, Tangerang Selatan, Indonesia
Email: [1,*]tianyama.id@gmail.com, [2]sajarwo@gmail.com, [3]dosen02710@unpam.ac.id
Correspondence Author Email: tianyama.id@gmail.com
Submitted: **09/12/2025**; Accepted: **30/06/2026**; Published**: 30/06/2026**

Abstract−The increasing volume of technical documentation and repetitive support requests at PT XYZ has made it difficult for helpdesk personnel to retrieve accurate information efficiently and consistently. The 114 internal applications that PT XYZ oversees have different documentation, troubleshooting protocols, and frequently asked questions, which leads to dispersed knowledge sources and drawn-out problem solving. This study uses locally deployed Large Language Models (LLMs), specifically Llama and Mistral, to design and assess a Retrieval-Augmented Generation (RAG) based chatbot for internal helpdesk knowledge retrieval in order to solve this issue. The suggested solution combines generative language models with semantic document retrieval in PostgreSQL using PGVector. The study compares the effectiveness of local LLMs in RAG and non-RAG configurations, employing 1,068 internal question-answer pairs for knowledge retrieval and 214 evaluation questions for performance assessment. ROUGE, BLEU, and Cosine Similarity measures are used for evaluation. According to experimental findings, RAG greatly enhances both models' performance. While Mistral improved from 0.0984 to 0.8983, Llama ROUGE-1 score rose from 0.1710 to 0.7345. With a ROUGE-1 score of 0.8983, a BLEU-1 score of 0.8161, and a Cosine Similarity score of 0.8916, Mistral with RAG performed the best of all setups. These results show that integrating RAG with locally installed LLMs greatly improves contextual correctness and response relevance, enabling safe and scalable implementation for enterprise helpdesk knowledge management.

Keywords: Retrieval-Augmented Generation; Local Large Language Model; Llama; Mistral; Helpdesk Knowledge Retrieval

1. INTRODUCTION

Financial services provider PT XYZ operates all around Indonesia. The Helpdesk section is in charge of managing technical problems and service requests for both internal and external users within its Information Management Division. Helpdesk services are a crucial part of IT service management because they guarantee service continuity and give users prompt technical support [1], [2]. To perform these responsibilities, helpdesk personnel rely on technical guidelines, troubleshooting documents, and frequently asked questions (FAQs) distributed across multiple information sources.

PT XYZ presently maintains 114 internal applications, each with its own documentation, operational procedures, and technical knowledge. Helpdesk staff must browse across several repositories to get pertinent information when answering customer enquiries since the amount of material keeps increasing. Longer issue resolution timeframes, inconsistent responses, and a greater reliance on personal experience are frequently the outcomes of this fragmented knowledge environment. Therefore, to increase the effectiveness and consistency of helpdesk services, an intelligent knowledge retrieval method is needed.

Artificial intelligence (AI) has emerged as a crucial technology for improving organizational decision-making and information management by allowing systems to process vast amounts of data and create contextual insights [3]. Retrieval-Augmented Generation (RAG) is a promising method that generates responses based on external knowledge sources by combining information retrieval techniques with Large Language Models (LLMs) [4]. RAG can enhance answer relevancy and lessen hallucination by combining document retrieval with natural language generation, which makes it appropriate for business helpdesk settings where data is dispersed across multiple documents. In order to increase operational effectiveness and service quality, AI technologies have also been extensively implemented in a variety of industries [5].

Previous research have shown that RAG-based chatbots are effective in a variety of fields using an accuracy score of 96.66% and a User Acceptance Test (UAT) score of 82.79%, a chatbot created using GPT-3.5 Turbo and LangChain for academic guidance services demonstrated great response accuracy and user satisfaction [6]. The efficacy of RAG in educational information retrieval was demonstrated by another study that used Gemini AI to implement a RAG-based chatbot, which reported context precision scores of 93%, faithfulness of 100%, answer relevance of 96%, context recall of 100%, answer correctness of 71%, and answer similarity of 93% [7]. Research testing retrieval approaches in domain-specific RAG chatbots indicated that the TF-IDF strategy had the greatest performance, with Recall@1, Recall@3, Recall@5, and Mean Reciprocal Rank (MRR) scores of 62.5%, 85%, 86.25%, and 72.33%, respectively, [8]. Additionally, a recent study that integrated GPT-4 Turbo, Mistral Devstral, and Xiaomi Mimo-v2 within a RAG framework demonstrated highly effective document retrieval and ranking performance with a Recall@3 score of 1.000, an MRR score of 0.756, and an NDCG@3 score of 0.864 [9].

Despite these promising results, several research gaps remain. Most existing studies focus on cloud-based models or evaluate only a single LLM architecture. In addition, limited research has systematically compared locally deployed open-source LLMs under both RAG and non-RAG configurations using the same dataset and evaluation framework. Previous studies tend to emphasize retrieval performance or chatbot usability, while fewer investigations

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

examine the impact of retrieval augmentation on response quality in real organizational environments. Furthermore, enterprise helpdesk services present unique challenges because they require accurate retrieval from large volumes of internal documentation while maintaining organizational control over knowledge resources.

To address these gaps, this study develops and evaluates a RAG-based chatbot for internal helpdesk knowledge retrieval at PT XYZ using locally deployed Llama and Mistral models. These models were selected because they are widely adopted open-source LLMs that support local deployment and exhibit different architectural characteristics. The proposed system integrates PostgreSQL and PGVector for managing documents and vector embeddings and is implemented using the Ollama framework. The study contributes by: (1) developing a RAG-based chatbot utilizing knowledge resources from 114 internal applications; (2) systematically comparing Llama and Mistral under both RAG and non-RAG configurations; (3) evaluating the impact of retrieval augmentation using ROUGE, BLEU, and Cosine Similarity metrics; and (4) providing empirical evidence regarding the effectiveness of locally deployed LLMs for enterprise helpdesk knowledge retrieval.

2. RESEARCH METHODOLOGY

2.1 Research Stages

This study develops and evaluates a RAG-based chatbot for internal helpdesk knowledge retrieval at PT XYZ. The proposed system utilizes locally deployed Large Language Models (LLMs), namely Llama 3.1 8B and Mistral Instruct 7B, integrated with PostgreSQL and PGVector for semantic document retrieval. The research process consists of six main stages: data collection, document chunking and preprocessing, embedding generation, retrieval and augmentation, response generation, and evaluation. The knowledge base was constructed from internal technical documentation and Frequently Asked Questions (FAQs) used by the PT XYZ Helpdesk. The collected documents were transformed into vector embeddings and stored in PostgreSQL using PGVector. During inference, the system retrieves the most relevant document chunks and supplies them as contextual information to the language model before generating a response. Finally, the generated responses are evaluated using ROUGE, BLEU, and Cosine Similarity metrics. The overall research workflow is illustrated in Figure 1.

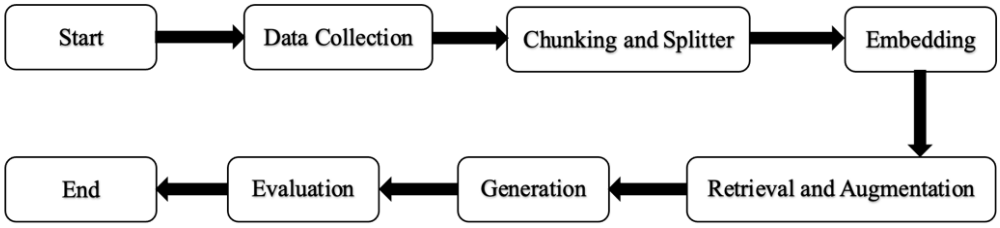

Figure 1. Research Stages

2.2 Data Collection

The dataset was gathered from the PT XYZ Helpdesk's internal knowledge resources, which include operational manuals, technical guidelines, troubleshooting techniques, and frequently asked questions (FAQs). These resources support 114 internal applications used across multiple business units within the organization. The collected documents were combined into a single PDF knowledge repository before being preprocessed and embedded. The chatbot's knowledge base was created by compiling 1,068 validated question-answer combinations from these publications. During evaluation, the same dataset was also used as the reference dataset, with the reference responses serving as the basis for automatic metric computations. To evaluate chatbot performance, a separate testing dataset of 214 questions was created. The generated response was compared to the matching reference answer from the verified knowledge source for every testing question.

Four domain experts consisting of one incident analyst, one application helpdesk leader, and two senior helpdesk specialists reviewed the entire set of question answer pairings to guarantee dataset quality and applicability. Using a five point likert scale, the validation evaluated the dataset's representativeness, relevance, completeness, compliance with operational documentation, and suitability for chatbot creation. The dataset was quite appropriate for use as the knowledge base of the suggested chatbot system, as evidenced by the total validation score of 92% [10].

2.3 Chunking and Splitter

Chunking is a text-processing method that preserves contextual information while breaking up lengthy publications into smaller sections [11]. By facilitating more accurate matching between user queries and document content, this procedure increases retrieval efficacy. In this study, the LangChain framework's RecursiveCharacterTextSplitter was used to chunk documents [12]. To preserve contextual continuity between neighbouring chunks, the chunk overlap was set at 200 characters and the chunk size was set at 500 characters. Documents were divided recursively while maintaining semantic coherence using RecursiveCharacterTextSplitter's default separator hierarchy. Important contextual information is preserved throughout document segments thanks to this setup.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

2.4 Embedding

Embeddings are numerical vector representations that facilitate effective similarity-based retrieval by capturing the semantic meaning of textual data [13]. Following chunking, the jeffh/intfloat-multilingual-e5-large:f32 embedding model was used to transform each document segment into a vector embedding. The PT XYZ documentation uses a mix of Indonesian and English technical terms that are frequently used in business application environments, which is why this approach was chosen. E5-Large is a multilingual embedding model that maintains robust retrieval performance while producing semantically relevant vector representations in several languages. In order to facilitate effective semantic search and vector similarity computations during the retrieval stage, the generated embeddings were saved in PostgreSQL using the PGVector extension.

2.5 Retrieval and Augmentation

When a user enters a query, the system compares it with the document embeddings kept in PGVector after first converting the query into an embedding vector. The Top-5 document chunks with the highest similarity scores are then chosen by the retrieval procedure [14]. The language model receives the retrieved chunks as contextual data once they are integrated into a prompt template. The chatbot can access outside information sources during response creation thanks to this RAG method, which lowers hallucinations and increases the relevance of responses [15].

Prior to the response production process, the augmentation stage seeks to enhance the user's query with pertinent contextual information obtained from the knowledge base. This technique improves factual accuracy and decreases hallucination by allowing the language model to provide replies based on organisational documents instead of only its pretrained knowledge [16]. The collected document chunks are integrated into a pre-made teaching template as part of the augmentation process, which employs a context-grounded prompting method. The instruction instructs the language model to respond to the user's query based solely on the context that was retrieved and to signal when the necessary data is not found in the knowledge base.

Table 1. Prompt Components Used in the Augmentation Stage

Component	Description
Assistant Role	PT XYZ Helpdesk Assistant
Response Style	Short, concise, and focused
Knowledge Source	Retrieved Top-5 document chunks
Context Usage	Answer only using the retrieved context
Unavailable Information	Politely state that the requested information is not available in the retrieved documents

The augmentation stage creates the input given to the language model by combining predefined prompt configurations, the user inquiry, and contextual data retrieved from the knowledge base, as shown in Table 1. The document chunks that were retrieved include domain-specific information that was taken from 114 business apps in PT XYZ's internal documentation. The suggested RAG framework allows the language model to produce responses based on organisational knowledge instead of only its pretrained parameters by explicitly adding the retrieved context into the prompt. This enhancement method lowers the possibility of hallucinated responses, increases answer relevance, and improves factual consistency.

2.6 Generation

Two locally deployed Large Language Models (LLMs), Llama 3.1 8B and Mistral Instruct 7B, are used in the generating step. Both models are connected with the suggested Retrieval-Augmented Generation (RAG) pipeline and run using the Ollama framework. These models were chosen because they are popular open-source LLMs that provide a balance between computing efficiency and language comprehension capacity, making them appropriate for usage in business settings without depending on outside cloud services. The configuration of the language models assessed in this study is compiled in Table 2.

Table 2. LLM Configuration

Model	Parameters	Deployment Framework	Experimental Configuration
Llama 3.1 8B	8.03 Billion	Ollama	RAG and Non-RAG
Mistral Instruct 7B	7.25 Billion	Ollama	RAG and Non-RAG

Both the user query and the contextual data acquired from the semantic retrieval stage are sent to the language model during response creation. After then, the model combines the external knowledge it has gathered from the document collection with its generative power to produce a natural language response. Because the response is created based on pertinent supporting context rather than just the model's parametric knowledge, Retrieval-Augmented Generation reduces hallucination and enhances answer grounding [17], [18]. The studies employed Ollama's default inference settings to guarantee a fair comparison across all experimental conditions. Temperature, top-p, and top-k generation parameters were not altered in any way. Llama 3.1 8B without RAG, Mistral Instruct 7B without RAG,

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

Llama 3.1 8B with RAG, and Mistral Instruct 7B with RAG were the four experimental configurations that were assessed in this study. The response generation workflow implemented in this study is illustrated in Figure 2.

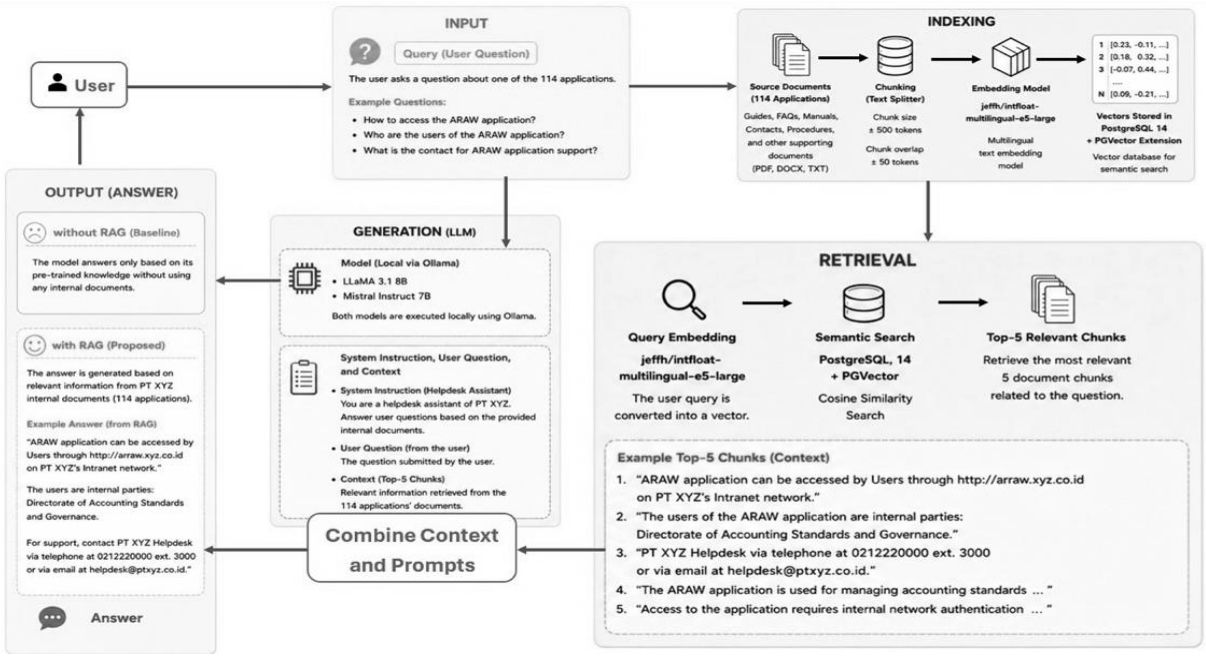

Figure 2. Proposed RAG Workflow for the PT XYZ Helpdesk Chatbot

A user's query about one of PT XYZ's 114 internal programs is first transformed into an embedding vector using the jeffh/intfloat-multilingual-e5-large embedding model, as shown in Figure 2. Semantic similarity search is then used to compare the embedding vector with document embeddings kept in PostgreSQL 14 using the PGVector extension. The Top-5 most pertinent document chunks that offer context for the supplied topic are returned by the retriever. Before being forwarded to the chosen language model, these retrieved chunks are integrated with the user inquiry and the specified system command to create an augmented prompt. Then, rather than depending just on its pretrained parameters, the language model produces a response based on the retrieved organisational knowledge. A non-RAG baseline was also assessed for comparison. The retrieval step is skipped in this arrangement, and the user's query is sent straight to the language model using the same instruction prompt without any contextual data retrieved from the knowledge base. This comparison allows ROUGE, BLEU, and Cosine Similarity metrics to be used to statistically assess semantic retrieval's contribution.

2.7 Evaluation

The evaluation phase seeks to determine the calibre and applicability of the chatbot's responses. 214 test questions and their matching reference answers were used to assess performance. Three metrics for automatic evaluation were used:

2.7.1 ROUGE

Lexical overlap between generated responses and reference answers is measured by ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [19];. In this work, linguistic similarity and content coverage were assessed using ROUGE-1, ROUGE-2, and ROUGE-L. Equations (1)–(3) provide the mathematical formulation [20], [21], [22].

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count\ (gram_n)} \quad (1)$$

$$ROUGE - N = \frac{p}{q} \quad (2)$$

$$ROUGE - N = \frac{LCS}{m} \quad (3)$$

2.7.2 BLEU

BLEU (Bilingual Evaluation Understudy) uses modified n-gram precision to assess how similar generated solutions are to reference answers. Greater lexical and structural similarity between generated and reference texts is indicated by higher BLEU ratings. Equations (4)–(6) provide the mathematical formulation [23].

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

$$BP_{BLEU} = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \leq r \end{cases} \quad (4)$$

$$P_n = \frac{\sum_{C \in corpusn} - \sum_{gram \in C} count_{clip}\ (n-gram)}{\sum_{C \in corpusn} - \sum_{gram \in C} count\ \ (n-gram)} \quad (5)$$

$$BLEU = BP_{BLEU}.\ e^{\sum_{n=1}^{N} w_n log\ Pn} \quad (6)$$

2.7.3 Cosine Similarity

By computing the cosine of the angle between their vector representations, Cosine Similarity assesses the semantic similarity between generated responses and reference answers [24]. The calculated value ranges from 0 to 1, where a value close to 1 indicates a high degree of similarity [25]. Equation (7) presents the mathematical formulation.

$$(dj, q) \quad = \frac{\sum_{i=l}^{t} (w_{ij}.w_{iq})}{\sqrt{\sum_{i=l}^{t} w_{ij}^2 . \sum_{i=l}^{t} w_{iq}^2}} \quad (7)$$

By assessing lexical overlap, structural similarity, and semantic significance, the combination of ROUGE, BLEU, and Cosine Similarity offers complementary viewpoints on chatbot performance. While qualitative error analysis and human review may offer more information about hallucinations and response quality, they were outside the purview of this study and are suggested for further investigation. ROUGE, BLEU, and Cosine Similarity together offer complementary viewpoints for assessing chatbot performance. BLEU assesses lexical and structural similarity, ROUGE quantifies lexical overlap between generated and reference responses, and Cosine Similarity captures semantic similarity. When combined, these measures offer a thorough quantitative evaluation of the suggested RAG-based chatbot's efficacy. While qualitative error analysis and human-centered evaluation of chatbot responses were outside the purview of this study, they offer potential avenues for future investigation to evaluate response quality and usefulness.

3. RESULT AND DISCUSSION

3.1 Dataset Description

The PT XYZ Helpdesk documentation, which includes Technical Guidelines (Juknis), troubleshooting manuals, operational procedures, and Frequently Asked Questions (FAQs), provided the dataset used in this study. The organization's 114 internal applications are supported by these documents. 1,068 validated question-answer pairs were created by combining the gathered materials into a single knowledge repository. In addition to being the reference dataset for evaluation, this dataset functioned as the RAG system's knowledge base. To assess chatbot performance, 214 test questions were also created. Four domain experts two Senior Helpdesk specialists, one Application Helpdesk Leader, and one Incident Analyst reviewed the dataset to guarantee its quality. Representativeness, relevance, completeness, compatibility with operational documentation, and eligibility for chatbot creation were all evaluated during the validation process. The dataset was quite appropriate for use as the chatbot knowledge base, as evidenced by the overall validation score of 92%. Examples of the question-answer combinations utilised in this study are shown in Table 3.

Table 3. Sample Question–Answer Dataset

No.	Question	Answer
1.	Please explain the general purpose of the service catalog?	The Information System (IS) and Application Service Catalog is a structured documentation containing information about the Services and Applications provided by the System Management Department as the service and application operations manager. This IS Service Catalog serves as a guide for IS Users and service managers in the provision of IS Services. Through the IS Service Catalog, IS Users can obtain information regarding the IS Services provided by the service manager and the available Applications.
2.	What are the benefits of the service catalog?	1. Facilitates IS Users in requesting IT Services. 2. Facilitates IS Users in reporting complaints or issues related to Information System Services they experience. 3. Provides information on the agreed service level availability guarantees between IS Users and service owners. 4. Provides information regarding operational Applications.
3.	What information is included in the service catalog?	1. Service Description. 2. IS Customers and Users. 3. Service dependency information. 4. Information for related Work Units. 5. Service Contact. 6. Information on operational Applications.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

No.	Question	Answer
…	…….....	…….
…	…….....	…….
1066.	How to access the ARAW application?	ARAW application can be accessed by Users through http://arraw.xyz.co.id on PT XYZ's Intranet network.
1067.	Who are the users of the ARAW application?	The users of the ARAW application are internal parties: Directorate of Accounting Standards and Governance.
1068.	What is the contact for ARAW application support?	PT XYZ Helpdesk via telephone at 0212220000 ext. 3000 or via email at helpdesk@ptxyz.co.id.

3.2 System Implementation

3.2.1 Chunking and Embedding Configuration

The LangChain framework's RecursiveCharacterTextSplitter was used to handle the source documents. To maintain contextual continuity across document portions, a chunk size of 500 characters and a chunk overlap of 200 characters were used. The multilingual embedding model jeffh/intfloat-multilingual-e5-large was used to transform each chunk into vector embeddings. Because PT XYZ documentation uses a combination of English and Indonesian technical terms that are frequently used in corporate application environments, this format was chosen. To facilitate effective semantic retrieval, the created embeddings were saved in PostgreSQL using the PGVector extension. The chunking and embedding settings employed in this investigation is summarised in Table 4.

Table 4. Chunking and Embedding Configuration

Parameter	Value
Chunk Size	500
Chunk Overlap	200
Embedding Model	jeffh/intfloat-multilingual-e5-large
Vector Database	PostgreSQL with PGVector

3.2.2 Semantic Retrieval Using PGVector

The same embedding model used for indexing is used to convert each user query into a vector representation during the retrieval stage. The query vector is then compared to stored document embeddings in PGVector to find semantically similar chunks. The most pertinent contextual data is retrieved from the knowledge base by the system via semantic similarity search. The retriever is set up to return the Top-5 most pertinent document chunks for each query. The language model uses these retrieved chunks as contextual information when generating responses once they have been concatenated. Instead of depending only on pretrained model parameters, the recovered data provides grounded context, enabling the system to produce responses based on internal organisational knowledge.

3.2.3 Example of Retrieval Augmented Generation

To demonstrate the RAG process, Table 5 has an example workflow that depicts the transformation from user query to generated response.

Table 5. Example of RAG Workflow

Component	Example
User Query	How to access the ARAW application?
Retrieved Context (Top-5 Chunks)	ARAW application can be accessed through http://arraw.xyz.co.id on PT XYZ intranet network.
Prompt Input to LLM	Use the following context to answer the question : [retrieved context]. Question: How to access the ARAW application?
Generated Response	Users can access the ARAW application through the PT XYZ intranet at http://arraw.xyz.co.id.

This example shows the entire RAG pipeline, in which the system initially uses PGVector-based similarity search to retrieve semantically relevant document chunks. The language model then uses the retrieved context to create a response based on internal organisational knowledge by injecting it into the prompt.

3.3 Experimental Result

In order to compare model performance under RAG and non-RAG conditions, four experimental setups were assessed. Llama 3.1 8B and Mistral Instruct 7B models are among the configurations that have been tested both with and without the RAG mechanism. To guarantee a fair comparison across experiments, all models were run using the Ollama framework with the same inference settings. Lexical overlap, n-gram precision, and semantic similarity between generated responses and reference answers were evaluated using 214 test questions using ROUGE, BLEU, and Cosine Similarity measures.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

Table 6. Evaluation Result

Model	R1	R2	RL	RLs	B1	B2	B4	CS
Llama Non RAG	0.1710	0.0712	0.1560	0.1580	0.1082	0.0673	0.0317	0.1842
Mistral Non RAG	0.0984	0.0307	0.0849	0.0874	0.0517	0.0278	0.0099	0.1595
Llama RAG	0.7345	0.6625	0.7232	0.7206	0.5918	0.5553	0.4903	0.6917
Mistral RAG	0.8983	0.8747	0.8943	0.8916	0.8161	0.7999	0.7626	0.8916

The quantitative evaluation findings for each experimental configuration are compiled in Table 6. The findings show that, on all evaluation metrics, both RAG-based models significantly outperform their respective Non-RAG models. With ROUGE-1 scores of 0.8983, BLEU-4 scores of 0.7626, and Cosine Similarity scores of 0.8916, Mistral RAG outperformed the other four configurations. Regardless of the underlying language model, Llama RAG also showed notable gains over its non-RAG counterpart, demonstrating that adding external knowledge retrieval greatly improves answer quality. To make it easier to compare the four experimental setups across all evaluation measures, Figure 3 provides further illustrations of the quantitative data shown in Table 6.

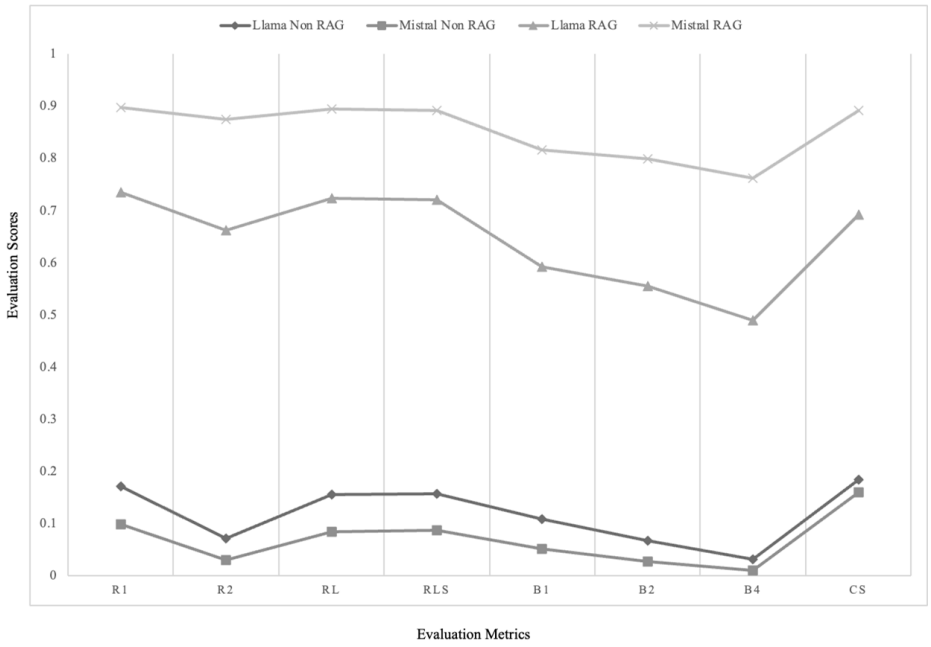

Figure 3. Performance Comparison of Llama and Mistral Models

RAG integration greatly enhances both Llama and Mistral models, as Figure 3 illustrates. In every evaluation metric, the RAG-based setups consistently perform better than the non-RAG models. Specifically, Mistral with RAG performs the best, demonstrating a greater capacity to use retrieved contextual knowledge to produce precise and contextually relevant replies. Since ROUGE-1 is the most general indicator of lexical overlap between generated and reference texts, Table 7 shows the relative improvement in ROUGE-1 score to further examine the effects of RAG.

Table 7. ROUGE-1 Improvement after RAG Integration

Model	Non-RAG	RAG	Improvement
Llama	0.1710	0.7345	329.5%
Mistral	0.0984	0.8983	812.9%

The findings show that RAG integration considerably enhances lexical and semantic performance in all models. Mistral with RAG, in particular, has the highest overall ratings, demonstrating its superior capacity to use retrieved contextual information for response generation. This enhancement implies that Mistral gains from external context injection more successfully than Llama, probably as a result of improved instruction-following behaviour and contextual integration in generation.

3.4 Discussion

RAG reliably improves chatbot performance across all assessment criteria, such as ROUGE, BLEU, and Cosine Similarity, according to the experimental results. Integrating external contextual information through the retrieval component significantly improves the performance of both the Llama and Mistral models.The Mistral model shows the most improvement, with ROUGE-1 rising from 0.0984 in the Non-RAG option to 0.8983 with RAG inclusion. In a similar vein, Cosine Similarity shows a significant improvement in semantic alignment between generated responses

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

and reference answers, rising from 0.1595 to 0.8916. These enhancements verify that retrieval augmentation is essential for firmly establishing responses in domain-specific knowledge.

Mistral's instruction-tuned optimisation, which enhances its capacity to efficiently use external context during generation, may be the reason for its better performance in the RAG situation as compared to Llama. Additionally, how each model incorporates retrieved data into cohesive answers may vary depending on model design and token-level attention efficiency. Mistral seems to provide more accurate and comprehensive responses by better preserving and utilising contextual cues from retrieved document fragments. On the other hand, both Non-RAG configurations perform noticeably worse on every metric. This suggests that without access to external material, standalone language models find it difficult to correctly respond to domain-specific Helpdesk requests, despite their great general knowledge capabilities. The findings emphasise how crucial it is to base generative models on retrieval mechanisms when working with enterprise knowledge systems.

These conclusions are further supported by a qualitative examination. Because their training data lacks relevant organisational knowledge, Non-RAG models often produce general or inadequate answers in situations like questions concerning the ARAW application access protocol. On the other hand, RAG-based models produce responses that closely match reference answers after successfully retrieving pertinent document chunks from the knowledge base. From an operational standpoint, integrating locally installed LLMs with PGVector-based semantic retrieval offers a practical method for enhancing internal knowledge accessibility in Helpdesk settings. This architecture guarantees that answers are based on validated organisational documents in addition to being linguistically coherent.

3.5 Limitations

Even though there have been significant performance gains, there are a few things to be aware of. First, ROUGE, BLEU, and Cosine Similarity are the only automatic metrics used in this study's evaluation. These metrics offer quantitative insights into lexical and semantic similarity, but they fall short in terms of real-world task efficacy, answer utility, and user pleasure. A more thorough assessment could be obtained by human evaluation with Helpdesk staff.

Second, despite covering 114 internal applications, the dataset comes from a single firm (PT XYZ). Because of this, the results might not apply to other organisational settings with different terminology, workflows, or documentation structures. Third, operational performance parameters like latency, throughput, and computing cost were not assessed; instead, the study concentrated on response quality and retrieval efficacy. When evaluating the viability of production-level deployment, several elements are crucial.

Lastly, even if RAG greatly increases the relevance of responses, retrieval errors can still happen when pertinent material is absent, poorly worded, or not sufficiently represented in the embedding space. This drawback emphasises how RAG systems rely on the accuracy and comprehensiveness of the underlying knowledge base. To further improve system reliability, future research may investigate enhanced retrieval techniques like hybrid search, reranking processes, and human-in-the-loop evaluation.

4. CONCLUSION

This study used two locally installed Large Language Models, Llama 3.1 8B and Mistral Instruct 7B, to construct and assess an internal Helpdesk chatbot based on RAG. The system was created to facilitate knowledge retrieval from PT XYZ's internal documentation, which comprises 1,068 pairs of questions and answers gathered from FAQs and technical guides in 114 internal applications. The incorporation of RAG significantly enhances chatbot performance as compared to non-RAG configurations, according to the experimental results on 214 test questions. With the greatest ROUGE, BLEU, and Cosine Similarity scores across the four assessed scenarios, the Mistral RAG model performed the best overall, demonstrating greater lexical overlap and semantic alignment with the reference responses. While both non-RAG models fared significantly worse when responding to organization-specific questions, the Llama RAG model also shown notable improvements over its non-RAG counterpart. These results validate the need of retrieval augmentation in establishing enterprise-specific information as the foundation for local LLMs. This study provides actual proof that internal Helpdesk knowledge access in organisational settings can be efficiently supported by locally deployed LLMs in conjunction with PGVector-based semantic retrieval. Practically speaking, the suggested strategy can preserve organisational control over proprietary documentation while enhancing the coherence and applicability of answers to internal support enquiries. This study could be expanded in the future by adding human review, tracking operational performance metrics like response latency and resource utilisation, and evaluating the approach's generalisability in other organisational areas.

REFERENCES

[1] D. A. Susanto, “Perancangan Aplikasi Sistem Informasi Helpdesk pada Kantor ABC,” *Jurnal Tera*, vol. 2, no. 2, pp. 26–33, Sep. 2022, doi: 10.59832/jt.v2i2.119.

[2] W. M. H. Sasmita, S. Sumpeno, and R. F. Rachmadi, “Improving Government Helpdesk Service with an AI-Powered Chatbot Built on the Rasa Framework,” *Jurnal RESTI*, vol. 9, no. 2, pp. 393–403, Apr. 2025, doi: 10.29207/resti.v9i2.6293.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 450−458
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.8895

[3] R. H. Nugroho, I. R. Kusumasari, V. Febrianto, M. A. Farhan N. H, and M. R. Mahardika, "Strategi Teknologi Artificial Intelligence (AI) dalam Pengambilan Keputusan Bisnis di Era Digital," *Jurnal Bisnis dan Komunikasi Digital*, vol. 2, no. 2, p. 7, Dec. 2024, doi: 10.47134/jbkd.v2i2.3476.
[4] G. D. Albert and A. Voutama, "Pengembangan Chatbot Berbasis PDF Menggunakan Local Retrieval-Augmented Generation (RAG) dan Ollama," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 2, Apr. 2025, doi: 10.23960/jitet.v13i2.6361.
[5] M. S. Y. Lubis, "Implementasi Artificial Intelligence pada System Manufaktur Terpadu," in *Prosiding Seminar Nasional Teknik Universitas Islam Sumatera Utara (UISU)*, 2021, pp. 1–7.
[6] M. D. A. Muhajir, N. Prastiti, and M. Koeshardianto, "Implementasi Chatbot Menggunakan Framework Langchain Berbasis LLM GPT (Studi Kasus : Panduan Akademik Universitas Trunojoyo)," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 2, 2025, doi: 10.36040/jati.v9i2.13003.
[7] Y. Tribber, Kusnadi, and M. Asfi, "Implementasi Retrieval Augmented Generation untuk Layanan Informasi Kampus dengan Chatbot Berbasis AI," in *Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK) ke-8*, 2024, pp. 594–600.
[8] Asmaidin and C. B. Santoso, "Evaluasi Metode Retrieval pada Chatbot Domain Khusus Berbasis Retrieval-Augmented Generation," *Journal Scientific and Applied Informatics*, vol. 09, no. 1, pp. 105–111, Jan. 2026, doi: 10.36085/jsai.v9i1.9897.
[9] Daerobby, Tukiyat, and A. Musyafa, "Analisis dan Evaluasi Kinerja Chatbot Penerimaan Mahasiswa Baru Berbasis LLM dengan Pendekatan RAG," *Jurnal Innovation and Future Technology*, vol. 8, no. 1, pp. 53–61, Feb. 2026, doi: 10.47080/iftech.v8i1.4452.
[10] F. R. Fatonah, D. S. Maylawati, and E. Nurlatifah, "Chatbot Edukasi Pra-Nikah Berbasis Telegram Menggunakan Bidirectional Encoder Representations from Transformers (BERT)," *Jurnal Algoritma*, vol. 21, no. 2, pp. 29–40, Nov. 2024, doi: 10.33364/algoritma/v.21-2.1657.
[11] T. Helviansyah, N. S. Harahap, M. Irsyad, and B. S. Negara, "Sistem Tanya Jawab Berbasis Chatbot Website Menggunakan Gemini AI pada Data Fiqih Kontemporer," *Journal of Information System Management*, vol. 7, no. 1, pp. 38–47, Jun. 2025, doi: 10.24076/joism.2025v7i1.2082.
[12] N. N. Qoniah and D. Ramadhani, "Optimization of Feminacare Chatbot Application Using SeaLLM Model," *Indonesian Journal of Informatic Research and Software Engineering*, vol. 5, no. 1, pp. 68–78, Mar. 2025, doi: 10.57152/ijirse.v5i1.2043.
[13] A. G and V. K, "RAG Based Chatbot Using LLMs," *Interantional Journal of Scientific Research in Engineering and Management*, vol. 8, no. 6, pp. 1–4, Jun. 2024, doi: 10.55041/IJSREM35600.
[14] A. Bora, "Comparative Analysis of Retrieval-Augmented Generation (RAG) Based Large Language Models (LLM) for Medical Chatbot Applications," Master's thesis, University of Lincoln, Lincoln, 2024. doi: 10.13140/RG.2.2.33650.93124.
[15] T. Muhammad, R. Rahardiansyah, R. Setya Perdana, and T. N. Fatyanosa, "Analisis Teknik Embedding Model NV-Embed pada Large Language Models Berbasis Retrieval Augmented Generation," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 2, pp. 2548–964, Feb. 2025, [Online]. Available: http://j-ptiik.ub.ac.id
[16] M. Klesel and H. F. Wittmann, "Retrieval-Augmented Generation (RAG)," *Business and Information Systems Engineering*, vol. 67, pp. 551–561, Jun. 2025, doi: 10.1007/s12599-025-00945-3.
[17] K. Fathoni *et al.*, "EMasjid, Islamic Chatbot Application With RAG (Retrieval Augmented Generation)," *Informatics, Electrical and Electronics Engineering (Infotron)*, vol. 5, no. 1, pp. 49–58, May 2025, doi: 10.33474/infotron.v3i2.22894.
[18] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv preprint arXiv:2312.10997*, pp. 1–21, Dec. 2023, doi: 10.48550/arXiv.2312.10997.
[19] M. Barbella and G. Tortora, "ROUGE Metric Evaluation for Text Summarization Techniques," *SSRN Electronic Journal*, pp. 1–31, May 2022, doi: 10.2139/ssrn.4120317.
[20] F. F. Kartamanah, A. R. Atmadja, and I. Budiman, "Analyzing PEGASUS Model Performance with ROUGE on Indonesian News Summarization," *Jurnal dan Penelitian Teknik Informatika (Sinkron)*, vol. 9, no. 1, pp. 31–42, Jan. 2025, doi: 10.33395/sinkron.v9i1.14278.
[21] I. Haromain, S. Munir, and A. Rahmah, "Analisa Prompt Engineering pada Large Language Model dengan Retrieval-Augmented Generation untuk Informasi Obat dan Vitamin," *Indonesian Journal Computer Science*, vol. 4, no. 2, Oct. 2025, doi: 10.31294/ijcs.v4i2.10005.
[22] M. Yang, "Application of Multimodal Generation Model in Short Video Content Personalized Generation," *Informatica*, vol. 49, no. 21, Dec. 2025, doi: 10.31449/inf.v49i21.9838.
[23] J. Ilmiah, T. Harapan, A. Bunga Permata, T. Listyorini, and E. Supriyati, "Pengembangan Mesin Penerjemah Bahasa Indonesia ke Bahasa Daerah Kudus Menggunakan NMT Berbasis RNN-GRU," *Jurnal Ilmiah Teknologi Harapan*, vol. 14, no. 1, pp. 15–27, Mar. 2026, doi: 10.35447/jitekh.v14i1.1314.
[24] A. Sanjaya, A. Bagus Setiawan, U. Mahdiyah, I. Nur Farida, A. Risky Prasetyo, and U. Nusantara PGRI Kediri, "Measurement Of Meaning Similarity Using Cosine Similarity and Word Synonyms Database," vol. 10, no. 4, 2023, doi: 10.25126/jtiik.2023106864.
[25] M. Qamal, M. Iqbal, and Y. Afrillia, "Utilizing Text Mining For Encryption Algorithm Recommendation Using Content-Based Filtering," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 11, no. 4, pp. 1272–1280, May 2026, doi: 10.33480/jitk.v11i4.7663.