Journal of Applied Business and Technology

Agricultural Drone Technical Consultation Chatbot Based on Generative AI with RAG Approach

Enrique Aurelius [a], Dewi Nasien [a*]

[a] Faculty of Computer Science, Institut Bisnis dan Teknologi Pelita Indonesia, Indonesia

Article History
Received
25 April 2026
Received in revised form
6 May 2026
Accepted
18 May 2026
Published Online
31 May 2026

*Corresponding author
dewinasien@lecturer.pelitaindonesia.ac.id

Abstract

This research aims to develop an agricultural drone technical consultation chatbot system capable of providing accurate and responsive solutions to address farmer operational downtime. The research was conducted at PT. Smart Tech Solution International during the period of August to November 2025. The research method applied was Rapid Application Development (RAD), comprising the requirement planning, user design, construction, and cutover phases. The system was built using GPT-3.5 Turbo Large Language Model technology and the LangChain framework with ChromaDB vector database integration. Research results through comparative testing (A/B Testing) show that the RAG architecture significantly improved answer quality compared to the baseline model. Semantic Similarity increased by 59%, from 51% to 81%. The system recorded a Recall@5 value of 96% with an average response time of 2.41 seconds. Evaluation through User Acceptance Test (UAT) produced an average satisfaction score of 4.05 on a scale of 5.0. The conclusion of this research is that the implementation of the RAG method is effective in minimizing AI hallucinations and providing a reliable technical assistant to support after-sales service efficiency in the precision agriculture sector.

Keywords: Chatbot; Agricultural Drone; Generative AI; Retrieval-Augmented Generation; RAG
DOI: https://doi.org/10.35145/dkv8x122
SDGs: Industry, Innovation and Infrastructure (9); Zero Hunger (2); Decent Work and Economic Growth (8)

1.0 INTRODUCTION

The use of digital technology in the modern agricultural sector has developed rapidly over the last decade, along with the increasing need for efficiency and productivity to support food security. One prominent innovation is the use of agricultural drones for various functions such as pesticide spraying, land mapping, crop growth monitoring, and real-time data analysis.(Nurani et al., 2025) Based on the 2023 Agricultural Census by BPS, more than 16.7 million Indonesian farmers have used digital technology in their agricultural activities (BPS, 2023).

However, increased use of drones in the field is accompanied by increased risk of technical obstacles because agricultural drone systems are complex and dependent on a combination of hardware, sensors, firmware, and software. Data from PT. Smart Tech Solution International shows an average of 40 drone repair cases per month as of August 2025. Downtime due to technical disruptions can last between 3 to 7 days, depending on problem complexity and logistics conditions, significantly impacting agricultural operational activities (Briliyanti et al., 2025).

Current solutions for handling technical obstacles still show limitations. Almost all cases of damage—including minor issues—are often handled by operators bringing drone units directly to the service center. This pattern indicates that user manuals have not always been effective as a primary field solution, while creating high dependence on limited expert technicians. As a result, service queues tend to be long, response is slow, and operational downtime becomes increasingly difficult to reduce.

The development of Generative AI supported by Large Language Models (LLM) offers opportunities to create more intuitive technical assistance through natural language conversations. However, pure LLMs have intrinsic limitations as they can produce inaccurate, non-specific, or even hallucinating answers (Alansari & Luqman, n.d.)—especially when questions require narrow and specific technical domain information (Rachmat & Kesuma, 2024).

To address these limitations, the Retrieval-Augmented Generation (RAG) approach can be used as a grounding mechanism to make LLM answers more document-based. RAG combines the ability to retrieve information from trusted external sources with the LLM's answer generation capabilities. Research shows that RAG implementation can improve answer accuracy compared to pure generative models (Lewis et al., 2020; Siriwardhana et al., 2023).

Although RAG has been widely applied in various domains, research specifically testing the effectiveness of RAG for agricultural drone troubleshooting in Indonesian language is still limited. Furthermore, many previous studies tend to evaluate chatbot success only from final answer quality, without quantitatively and separately measuring the retrieval component. Therefore, this research contributes by building an agricultural drone technical consultation chatbot based on RAG and conducting evaluations that distinguish between retrieval performance and answer generation quality, making the results more relevant for operational needs and more scientifically traceable.

2.0 LITERATURE REVIEW

Chatbot and Natural Language Processing

Chatbot is a computer program (Alim, 2025; Yasmin & Fudholi, 2025) designed to conduct automated conversations with users through text or voice media (Albert & Voutama, 2025). Modern chatbot systems utilize Natural Language Processing (NLP) techniques to understand the intent of questions and generate contextually appropriate answers. In information service practice, chatbots are proven effective in handling repetitive questions (Hartono et al., 2025; Prastowo et al., 2025), thereby reducing helpdesk burden and accelerating information access.

Large Language Models and Generative AI

Large Language Models (LLM) are large-scale language models trained on massive text corpora to predict the next token in a sequence. LLMs are capable of summarizing documents (Cahyanti & Raya, 2025; Rizky, 2025), answering questions, and maintaining language style consistent with conversational context. However, pure LLMs may provide inaccurate or hallucinating answers if not equipped with a more controlled knowledge access mechanism (Alansari & Luqman, n.d.).

Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a method that combines information retrieval capabilities from external sources with answer generation capabilities from LLMs (Pokhrel et al., n.d.). The system retrieves relevant document snippets from the knowledge base, inserts them into the prompt, then the model generates an answer referencing the document content (Siriwardhana et al., 2023). The RAG flow includes document collection and cleaning, chunking, embedding creation, similarity search at query time, and answer generation (Mubarak & Sutopo, 2026).

Embedding and Vector Databases

Embedding is a technique representing text in high-dimensional numerical vector form so that semantic proximity can be computed mathematically. Embedding vectors are stored in vector databases and indexed using nearest-neighbor search algorithms such as FAISS or HNSW. ChromaDB is used as an open-source vector database that can be run locally (self-hosted) (Fanani, 2025; Swarawisesa et al., 2025).

Rapid Application Development

Rapid Application Development (RAD) is a software development method emphasizing rapid system building through prototype use and intensive user involvement. The RAD model consists of requirements planning, user design, construction, and cutover phases conducted iteratively. RAD is well-suited for developing RAG-based chatbot prototypes that require extensive feedback from field technicians and operators (Yasmin & Fudholi, 2025).

Prior Research

Several studies have applied RAG in various domains. Voutama developed a Local RAG-based chatbot with ChromaDB proven efficient for PDF document-based information retrieval, though response time varied for large documents (Albert & Voutama, 2025). Pujiono et al. showed RAG improves answer relevance in public services but requires adaptation testing in real environments with high user volumes (Pujiono et al., 2024). Tohir et al. proved RAG improves LLM answer accuracy in Indonesian, though response speed evaluation was incomplete (Tohir et al., 2024). Fanani achieved 80% answer correctness in academic proposal evaluation using RAG (Fanani, 2025). Chen et al. demonstrated RAG's capability in handling specific-domain terminology for industrial knowledge management (Chen et al., 2025). Xu et al. developed weighted RAG for enterprise system troubleshooting improving solution relevance (Xu et al., 2024). Alansari & Luqman concluded through a comprehensive survey that RAG is the most effective LLM hallucination mitigation solution currently available (Alansari & Luqman, n.d.).

However, research specifically testing RAG effectiveness for Indonesian-language agricultural drone troubleshooting remains limited, and this study fills that gap by also evaluating retrieval and generation components separately.

3.0 METHODOLOGY

This research was conducted at PT. Smart Tech Solution International during the period of August to November 2025. The research method applied was Rapid Application Development (RAD), comprising the requirement planning, user design, construction, and cutover phases. This approach was chosen because it enables rapid, interactive, and flexible system development through iterative cycles and functional prototypes.

RAG System Architecture

The proposed system is built with a client-server architecture consisting of two main phases. The first phase is ingestion (knowledge preparation), where manual documents and troubleshooting guides have their text extracted, cleaned, and broken into chunks. Text snippets are then converted into vector embedding representations using the text-embedding-3-small model and stored in the ChromaDB vector database. The second phase is inference (chatbot service), where the system receives user questions, performs semantic searches on the knowledge base, and uses GPT-3.5 Turbo to generate contextual answers based on found documents.

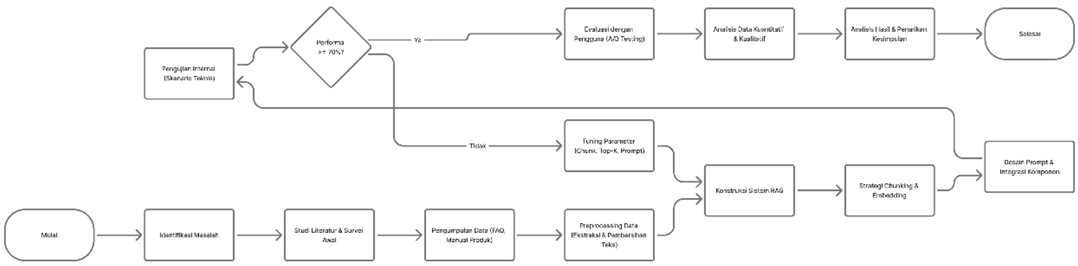

Figure 1. Research Framework of RAG-Based Agricultural Drone Chatbot System

Figure 1 illustrates the overall research framework of the proposed system, consisting of two main phases: (1) the Ingestion Phase, where DJI Agras T-series manual documents are extracted, chunked (500 characters, 10% overlap), embedded using the text-embedding-3-small model, and stored in ChromaDB; and (2) the Inference Phase, where user queries are vectorized, semantically matched against the knowledge base using top-k=5 retrieval, and GPT-3.5 Turbo generates contextual answers grounded on retrieved document chunks.

System parameter configurations were set based on best practices from RAG literature (Pokhrel et al., 2025). Parameters used consistently throughout the research were: (1) Chunk Size = 500 characters, (2) Chunk Overlap = 10% (50 characters), (3) Top-k = 5 documents, and (4) Temperature = 0.1–0.2. The complete development environment specifications are presented in Table 1.

Table 1. Development Environment Specifications

Component	Specification	Version
LLM Model	Gpt-3.5-turbo-0125	Temp-0.1
Embedding Model	Text-embedding-3-small	Dim=1536
Vector Store	ChromaDB	v.0.4.1
Framework	LangChain	V0.1.0
Hardware (Local Staging)	CPU Ryzen 5, 16GB RAM	-

Data Collection

Data collection was conducted through two methods. The observation method was used to identify patterns of technical problems that frequently occur in the field through direct observation of ongoing technical consultation service processes for one full week (5 working days). The documentation method was used to collect a knowledge corpus of at least 150 question-answer pairs from official troubleshooting websites and official user manuals of DJI Agras T-series drones.

Testing Techniques

System testing was conducted through three approaches. First, Black-Box Testing to ensure every system feature operates according to defined requirements. Second, A/B Testing to compare the performance of the RAG system (System B) with the LLM baseline without RAG (System A) using a dataset of 50 representative questions. Third, User Acceptance Test (UAT) involving 10 respondents consisting of drone technicians and operators to evaluate user acceptance from aspects of usability and answer clarity.

Quantitative evaluation used several metrics. For retrieval performance: Precision@k, Recall@k, F1-Score, and Mean Reciprocal Rank (MRR). For answer generation quality: BLEU (Bilingual Evaluation Understudy), ROUGE (Recall-Oriented Understudy for Gisting Evaluation), and Semantic Similarity. Performance targets were set based on industry standards for technical decision support systems: Precision@5 ≥ 85%, BLEU score ≥ 0.40, and average response time < 5 seconds.

The evaluation dataset consists of 50 technical questions selected representatively from a total of 213 FAQs in the knowledge base. Question distribution by category and difficulty level is presented in Table 2 and Table 3.

Table 2. Evaluation Dataset Distribution by Category

Category	Count	Percentage
Product Specifications	18	36%
Troubleshooting	10	20%
Operational	7	14%
Procedural	5	10%
Maintenance	4	8%
Features	3	6%
Error Codes	2	4%

Table 2 shows that product specification questions dominate the evaluation dataset at 36%, followed by troubleshooting (20%) and operational questions (14%). This distribution reflects real-world consultation patterns at PT. Smart Tech Solution International, where users most frequently inquire about technical specifications before performing field operations.

Table 3. Evaluation Dataset Distribution by Category

Difficulty	Count	Percentage
Easy	11	22%
Medium	34	68%
Hard	5	10%

Table 3 illustrates that the majority of evaluation questions (68%) are of medium difficulty, representing the typical complexity of field troubleshooting queries encountered by drone operators. Only 10% of questions are classified as hard, while 22% are easy, ensuring a balanced and realistic assessment of the chatbot's performance across varying complexity levels.

Table 4. Likert Scale Interpretation for UAT Results

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

Table 4 presents the five-point Likert scale interpretation categories used to evaluate UAT results. The system's overall average UAT score of 4.05 falls within the 'Good' category (3.41–4.20), indicating that drone technicians and operators consider the chatbot system effective, responsive, and suitable for field technical consultation use.

4.0 RESULTS AND DISCUSSION

Information System Design Analysis

The proposed system is designed to overcome the limitations of the manual system that requires long processing times for technical consultations. The RAG-based chatbot system provides 24/7 consultation access with the ability to automatically answer technical questions based on a verified knowledge base. The use case diagram describes the interaction of two main actors with the system: Users can perform technical consultations, view problem categories, and obtain troubleshooting solutions, while Admins can manage the knowledge base, synchronize data to the vector database, and monitor system activity.

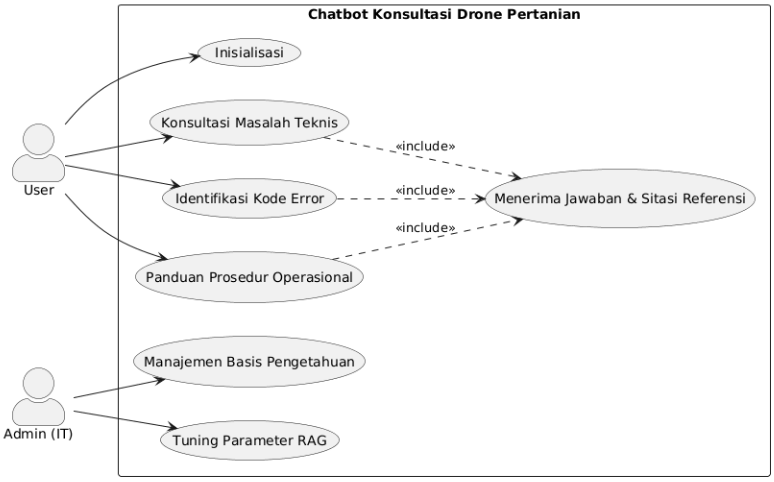

Figure 2. Use Case Diagram of the Proposed System

Figure 2 presents the use case diagram illustrating the interactions of two main actors with the RAG chatbot system. The User actor can submit technical consultation questions, browse problem categories, and receive troubleshooting solutions via the Telegram Bot interface. The Admin actor has elevated system privileges to manage the knowledge base, synchronize entries to the ChromaDB vector database, and monitor system activity through the admin dashboard.

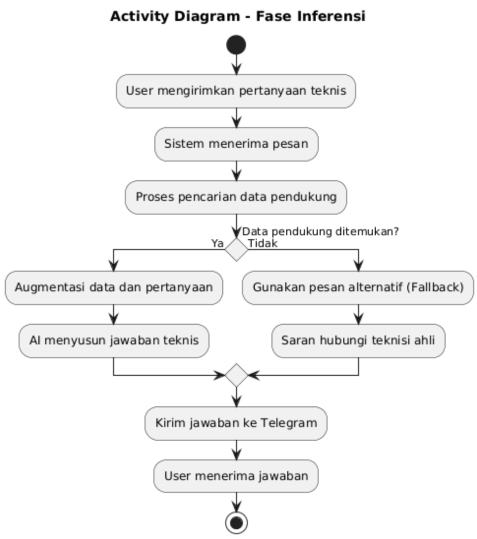

Figure 3. Use Case Diagram of the Proposed System

Figure 3 depicts the activity diagram for the chatbot's inference workflow. The flow begins when a user submits a question through the Telegram Bot. The system then converts the query into a vector embedding, performs semantic similarity search in ChromaDB to retrieve the five most relevant document chunks, constructs a context-augmented prompt, invokes GPT-3.5 Turbo, and returns the generated answer to the user. If no relevant document is found above the similarity threshold, the system returns a transparent fallback response.

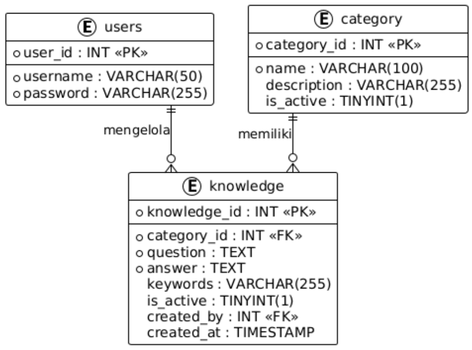

Figure 4. Use Case Diagram of the Proposed System

Figure 4 presents the Entity Relationship Diagram (ERD) for the chatbot's MySQL relational database. The schema consists of three main tables: Users (admin credentials for system access control), Category (master data for technical problem categories such as Error Codes, Calibration, Maintenance, and Operation), and Knowledge (the central repository of question-answer pairs, linked to Category via a one-to-many foreign key relationship). This structure supports full CRUD operations through the admin panel.

The system uses a dual database architecture combining a relational database for content management and a vector database for semantic retrieval. This hybrid approach enables efficient structured data management while supporting semantic similarity-based search, which is central to the RAG architecture. The relational database is implemented using MySQL with three main tables: Users (for admin authentication), Category (master data for technical problem categories), and Knowledge (repository of question-answer pairs). The vector database is implemented using ChromaDB storing 1536-dimensional vector representations of each document chunk.

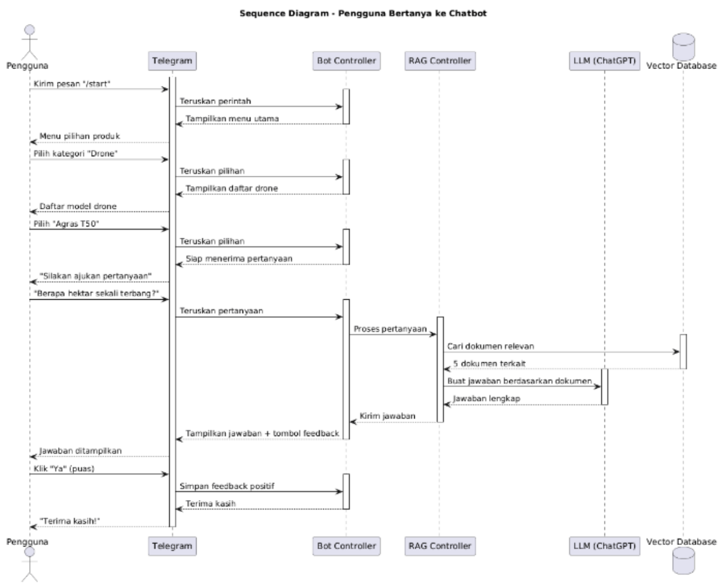

Figure 5. Sequence Diagram of the Technical Consultation Process

Figure 5 shows the sequence diagram detailing the communication flow between system components during a consultation session. The components include the Telegram Bot (user interface layer), Backend API (orchestration), Embedding Service (query vectorization using text-embedding-3-small), ChromaDB Vector Store (semantic retrieval), and LLM Service (GPT-3.5 Turbo). The diagram captures the complete lifecycle from user message receipt through document retrieval to final grounded response delivery.

Database Design

The system uses a dual database architecture combining a relational database for content management and a vector database for semantic retrieval. This hybrid approach enables efficient structured data management while supporting semantic similarity-based search, which is central to the RAG architecture. The relational database is implemented using MySQL with three main tables: Users (for admin authentication), Category (master data for technical problem categories), and Knowledge (repository of question-answer pairs). The vector database is implemented using ChromaDB storing 1536-dimensional vector representations of each document chunk.

Table 5. Vector Database Metdata Schema

Field	Data Type	Description
Id	String	Unique identify of each document chunk
Source	String	Name of source data file
Category	String	Technical category (ErrorCode, StepbyStep, etc)
Text_content	Text	Content of Q&A text snippet
Embedding	Float32[1536]	Vector from text-embedding-3-small
Created_at	Timestamp	Record creation time

Data Analysis

The system data source originates from official DJI Agras T-series manuals (3 PDF documents) and a FAQ database (150 question-answer pairs). Documents are processed into 500-character chunks with 50-character overlap using RecursiveCharacterTextSplitter. Each chunk is converted into a 1536-dimensional vector using the text-embedding-3-small model and stored in ChromaDB. This configuration was chosen based on analysis balancing adequate context and retrieval precision.

Implementation Results

The following figures present the implementation results of the RAG-based agricultural drone technical consultation chatbot system, including the Telegram Bot user interface and the web-based admin management dashboard.

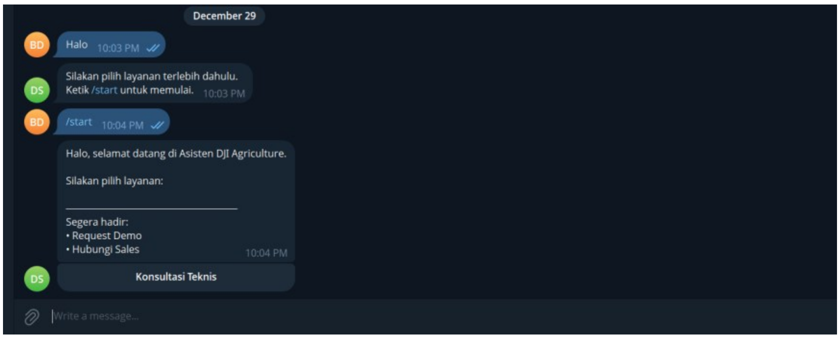

Figure 6. Telegram Bot Interface – Technical Consultation Session

Figure 6 illustrates the chatbot user interface accessed through the Telegram messaging platform. The screenshot shows a sample consultation session where a drone operator queries the system about a technical error. The chatbot responds with a precise, actionable troubleshooting solution sourced from the DJI Agras T-series knowledge base, demonstrating the RAG system's ability to deliver domain-specific guidance in natural conversational language without hallucination.

Figure 7. Admin Dashboard – System Overview Page

Figure 7 shows the admin dashboard home page, providing a centralized overview of the chatbot system status. The dashboard displays key statistics including the total number of knowledge entries, number of active problem categories, and the current synchronization status between the MySQL database and ChromaDB vector store. This page serves as the primary control hub for system administrators to monitor and manage the chatbot's knowledge base.

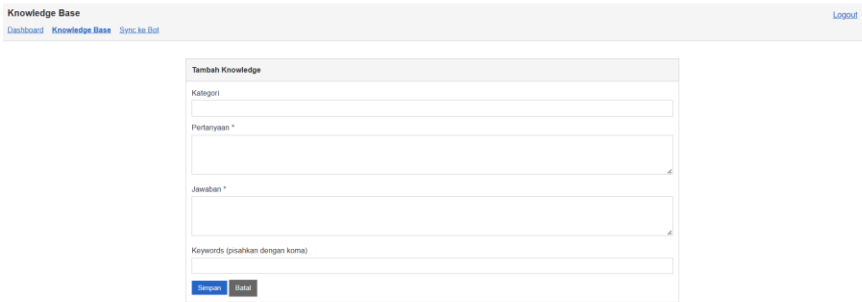

Figure 8. Admin Page – Add New Knowledge Entry Form

Figure 8 displays the knowledge entry form in the admin dashboard, where administrators can add new question-answer pairs to the system's knowledge base. Input fields include the technical question, its corresponding answer, category assignment, and source reference. Upon submission, entries are stored in the MySQL database and flagged for vector synchronization to ensure the ChromaDB retrieval layer remains up to date.

Figure 9. Admin Page – Knowledge Base Synchronization

Figure 9 illustrates the synchronization page of the admin dashboard. This feature allows administrators to re-index the ChromaDB vector database after any knowledge base modifications. During synchronization, all knowledge entries are re-embedded using text-embedding-3-small and the vector store is updated accordingly, maintaining consistency between the relational database and the semantic retrieval layer to ensure optimal chatbot response accuracy.

System Testing

Quantitative Evaluation

Evaluation was conducted on 50 representative questions covering various technical troubleshooting categories. The system recorded a 100% success rate with an average response time of 2.78 seconds. Retrieval performance showed Recall@5 of 96% and Precision@5 of 74%. The high recall value proves the system rarely misses important information and almost always successfully finds relevant documents from the knowledge base. However, the lower precision value indicates that some non-relevant documents are still included in the top document list, indicating potential for further optimization through re-ranking or hybrid retrieval techniques.

Table 6. Generation Metric Results

Metric	Avereage Value	Min Value	Max Value
BLEU Score	0.40	0.02	1.00
ROUGE-1	0.65	0.09	1.00
ROUGE-2	0.53	0.03	1.00
ROUGE-L	0.63	0.06	1.00
Semantic Similarity	0.81	0.50	1.00

Answer quality reached a Semantic Similarity metric of 0.81. Although the BLEU score was recorded at 0.40, this value is actually appropriate and highly adequate for generative Question Answering (QA) tasks. The BLEU metric has conservative computational properties as it is highly sensitive to lexical variations. Therefore, in architectures based on Large Language Models, the high Semantic Similarity (0.81) is a far more reliable and representative indicator proving the system's answers have precise contextual accuracy.

Comparative Testing

Comparative testing shows the RAG architecture provides significant improvement over the LLM baseline across all answer quality metrics.

Table 7. RAG vs Baseline Performance Comparison

Metric	RAG	Baseline	Improvement
Success Rate	100%	100%	-
Response Time	2.41 s	1.16 s	-52%
BLEU Score	0.40	0.03	+1233%
ROUGE-1	0.65	0.17	+282%
ROUGE-2	0.53	0.06	+783%
ROUGE-L	0.63	0.14	+350%
Semantic Similarity	0.81	0.51	+59%

Semantic Similarity increased by 59% from 0.51 to 0.81. The trade-off of a 1.25-second increase in response time is very acceptable given the drastic improvement in answer quality. The RAG system effectively prevents AI hallucinations by grounding on a verified knowledge base.

User Testing

The User Acceptance Test evaluation involved 10 drone technician and operator respondents, resulting in an average satisfaction score of 4.05 on a scale of 5.0 ("Good" category).

Table 8. User Acceptance Test Results

Aspect	Score	Percentage	Category
Ease of Use	4.00	80.0%	Good
Answer Quality	4.08	81.6%	Good
Response Speed	4.10	82.0%	Good
User Satisfaction	3.98	79.5%	Good
Comparison to Alternatives	4.07	81.3%	Good
Overall Average	**4.05**	**80.9%**	**Good**

Response Speed recorded the highest score (4.10), confirming that the 2.41-second response time meets field user expectations. Answer Quality (4.08) validates that the RAG approach successfully presents accurate information aligned with technical needs. It should be noted that the relatively small respondent sample size (n=10) limits generalization of user acceptance findings; further testing with larger operational user involvement is recommended to comprehensively validate these findings.

Error Analysis

Although the system has a 100% success rate, qualitative evaluation found several cases where retrieved documents were less relevant or answers were incomplete. Details of error types and their frequencies from 50 test questions are presented in Table 9.

Table 9. Error Distribution Analysis in RAG Architecture

ErrorType	Frequency	Primary Case
Interrupted Multi-Step Procedures	4 cases	Context fragmentation between chunks
Technical Term Ambiguity	2 cases	Embedding model limitations in differentiating abbreviations
Truncated Context	2 cases	Insufficient overlap

Analysis results show that context fragmentation in multi-step procedural questions is the main obstacle. This occurs when a troubleshooting guide is longer than 500 characters, causing it to split into separate chunks, resulting in the LLM losing the complete instruction sequence. These findings strongly indicate the need for more advanced chunking strategies, such as semantic chunking, to minimize truncation of crucial technical context.

Research Results

Research results prove that RAG implementation is highly effective in improving chatbot answer quality for technical consultations. RAG's advantages lie in three main mechanisms: (1) grounding on an external knowledge base ensures factual and traceable answers, (2) retrieval injects DJI Agras product-specific context so answers are not generic, and (3) a fallback mechanism prevents hallucinations with transparency when information is unavailable (Lewis et al., 2020). The system provides a real solution for PT. Smart Tech Solution International handling an average of 40 repair cases per month, with 24/7 consultation access via Telegram reducing technician burden and enabling operators to obtain independent solutions, potentially reducing operational downtime with significant economic impact during planting seasons.

5.0 CONCLUSION

This research successfully developed an agricultural drone technical consultation chatbot system based on Retrieval-Augmented Generation (RAG) that is effective in providing accurate and responsive troubleshooting solutions. The system achieved high retrieval performance with Recall@5 of 96% and Precision@5 of 74%, as well as good generation quality with Semantic Similarity of 0.81. Comparative testing proved the RAG architecture provides significant improvement over the LLM baseline, with Semantic Similarity increasing 59% and BLEU Score increasing 1233%. The system successfully prevented AI hallucinations through a grounding mechanism on a verified knowledge base. User Acceptance Test evaluation produced an average satisfaction score of 4.05 on a scale of 5.0, with an average response time of 2.41 seconds meeting operational targets. System implementation provides real impact by providing 24/7 consultation access that reduces technician burden and operational downtime in agricultural fields. As a broader theoretical and practical implication, this research confirms that for specific technical domain problems with a limited knowledge base, the RAG approach is proven capable of achieving an optimal balance between high recall and accurate semantic fidelity, without requiring complex and computationally expensive model fine-tuning processes.

Future research can improve Precision@5 through implementation of re-ranking or hybrid search techniques to increase document relevance. Development of more sophisticated chunking strategies such as semantic chunking can address answer quality variability on multi-step procedural questions. Implementation of feedback loop mechanisms for continuous improvement and longer trial periods with larger numbers of respondents are needed to validate long-term effectiveness and increase system adoption. Future research can explore RAG implementation in other agricultural domains and evaluate system scalability on larger knowledge bases.

References

Alansari, A., & Luqman, H. (n.d.). *Large Language Models Hallucination : A Comprehensive Survey*. 1–36.

Albert, G. D., & Voutama, A. (2025). Pengembangan Chatbot Berbasis Pdf Menggunakan Local Retrieval-

Augmented Generation (Rag) Dan Ollama. *Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(2). https://doi.org/10.23960/jitet.v13i2.6361

Alim, A. A. (2025). *RETRIEVAL-AUGMENTED GENERATION (RAG) PADA CHATBOT WHATSAPP UNTUK LAYANAN INFORMASI AKADEMIK*. 866–872.

BPS. (2023). *Hasil Pencacahan Lengkap Sensus Pertanian 2023 - Tahap I*. 2023. https://www.bps.go.id/id/pressrelease/2023/12/04/2050/hasil-pencacahan-lengkap-sensus-pertanian-2023---tahap-i.html

Briliyanti, R. D., Maulana, I., & Jannah, E. M. (2025). *The Effectiveness of Drones in Improving Efficiency and Accuracy of Oil Palm Plantation Management in Indonesia : A Literature Review*. *5*(1), 142–149.

Cahyanti, F. L. D., & Raya, R. D. A. (2025). Perancangan Sistem Informasi Chatbot Retrieval Augmented Generation Berbasis Website Pada PT. Revolusi Cita Edukasi. *Indonesian Journal Computer Science*, *4*(1), 15–21. https://doi.org/10.31294/m75d4782

Chen, L. C., Pardeshi, M. S., Liao, Y. X., & Pai, K. C. (2025). Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model. *Computer Standards and Interfaces*, *94*(December 2024), 103995. https://doi.org/10.1016/j.csi.2025.103995

Fanani, I. (2025). Implementasi Retrieval Augmented Generation Untuk Evaluasi Proposal Tugas Akhir Mahasiswa. *Jurnal Teknologi Komputer Dan Informatika*, *3*(2). https://doi.org/10.59820/tekomin.v3i2.336

Hartono, L. S., Setiawan, E. I., & Singh, V. (2025). *Retrieval Augmented Generation-Based Chatbot for Prospective and Current University Students*. *5*(3), 268–277.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *2020-Decem*.

Mubarak, M. H., & Sutopo, J. (2026). *Metode Retrieval-Augmented Generation dalam Pengembangan Chatbot Large Language Model pada Service Informasi Publik Disdukcapil Anambas*. *09*(1), 10–18.

Nurani, A., Taqiya, H., Nabila, A., & Herlambang, I. B. (2025). *PERAN ARTIFICIAL INTELLIGENCE DALAM SISTEM IOT UNTUK PERTANIAN CERDAS : SYSTEMATIC LITERATURE REVIEW*. *9*(1), 1446–1455.

Pokhrel, S., Bina, K. C., & Shah, P. B. (n.d.). *A Practical Application of Retrieval- Augmented Generation for Website-Based Chatbots : Combining Web Scraping , Vectorization , and Semantic Search*. *6*(4), 424–442.

Prastowo, I., Putro, H., Antoni, J., Adhitya, M. K., & Herawati, N. A. (2025). *Retrieval-Augmented Generation (RAG) Chatbot for Handling Customer Complaints in the Energy Sector*. *10*(2), 105–111.

Pujiono, I., Agtyaputra, I. M., & Ruldeviyani, Y. (2024). Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *10*(1), 216–223. https://doi.org/10.33480/jitk.v10i1.5572

Rachmat, N., & Kesuma, D. P. (2024). Implementasi LLM Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android. *Jurnal Ilmu Komputer (JUIK)*, *4*(1), 40. https://doi.org/10.31314/juik.v4i1.2831

Rizky, M. A. (2025). *Analisis Efektivitas Dua Jenis Gaya Prompt dalam Model LLM Berbasis RAG*. *9*(1), 76–86.

Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., & Nanayakkara, S. (2023). *Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering*. *11*, 1–17.

Swarawisesa, G., Putra, E., Suhendra, A., N, A. B. M. Q., & Juarna, A. (2025). *Implementation of digital human avatar virtual assistant with augmented generation retrieval technology in interactive systems for nutrition education*. *Llm*, 231–244.

Tohir, H., Merlina, N., & Haris, M. (2024). Utilizing Retrieval-Augmented Generation in Large Language Models To Enhance Indonesian Language Nlp. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *10*(2), 352–360. https://doi.org/10.33480/jitk.v10i2.5916

Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024). Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. In *SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3626772.3661370

Yasmin, S. M., & Fudholi, D. H. (2025). *Pengembangan Chatbot Informasi Hukum Layanan Publik Berbasis Retrieval-Augmented Generation Menggunakan LangChain dan OpenAI di Ombudsman DIY Development of a Legal Information Chatbot for Public Services Using RAG , LangChain , and OpenAI at Ombudsman DIY*. *5*(9), 2548–2565.