SME Business Intelligence Support Using Retrieval-Augmented Generation and RFM Segmentation

Rosalina[1,*], Noor Lees Ismail[2], Genta Sahuri[3], Joseph Tedja Nugraha Wibawa[4]

[1,3,4]*Informatics Study Program, Faculty of Computer Science, President University, Bekasi, Indonesia*

[2]*School of Information Technology, UNITAR International University, Selangor, Malaysia*

(Received: September 25, 2025; Revised: November 22, 2025; Accepted: February 20, 2026; Available online: March 17, 2026)

Abstract

This study proposes and empirically evaluates an integrated SME business intelligence support system that combines Retrieval-Augmented Generation (RAG) with Recency–Frequency–Monetary (RFM) customer segmentation and embeds both capabilities directly into a mobile keyboard interface for everyday business communication. Unlike conventional chatbots or standalone analytics tools, the system delivers knowledge-grounded automated responses and actionable customer insights within the seller's existing messaging workflow, eliminating the need for separate applications, local infrastructure, or AI expertise. The framework constructs a structured SME knowledge base in Markdown, applies semantic chunking and Voyage-3 embeddings, and performs vector retrieval via PgVector to ensure high-fidelity grounding before generation using a cloud-based LLM. In parallel, historical invoice data are processed through an RFM engine to classify customers into Loyal, Moderate, and At-Risk segments for targeted promotions. Using real SME data collected over several weeks, the system was evaluated through retrieval faithfulness testing, correctness analysis with confidence intervals, silhouette validation of clusters, end-to-end latency measurement, and User Acceptance Testing with 18 sellers. Results show very high retrieval faithfulness (0.997), strong generative correctness (0.88), acceptable real-time latency (~5 seconds), and stable segmentation performance (Silhouette 0.61; ROC–AUC: At-Risk 0.93, Loyal 0.85). The key novelty lies in unifying RAG-based conversational support and lightweight customer analytics inside a keyboard-level interface, creating a practical, low-barrier pathway for AI adoption in small businesses while preserving natural communication practices.

Keywords: Mobile Keyboard Interface, Retrieval-Augmented Generation, RFM Segmentation, SME Business Intelligence

1. Introduction

Customer communication has become a key component of SME business intelligence, particularly as many Small and Medium Enterprises (SMEs) in Indonesia now conduct sales through digital platforms such as WhatsApp, Instagram, and Facebook [1]. [2], [3]. These channels enable sellers to provide product information, negotiate with buyers, and influence purchasing decisions in real time. Timely and accurate responses serve as indicators of professionalism and strongly affect customer trust. Prior studies highlight that responsive and high-quality communication substantially improves SME competitiveness and operational performance in digital commerce environments [4], [5], [6]. Effective communication is therefore essential not only for sustaining customer relationships but also for informing strategic decision-making. Despite this importance, many SMEs still rely on manual replies or conventional keyword-based tools. These systems depend on exact word matching and are unable to interpret varied or context-rich customer queries, often generating generic responses that fail to represent actual business information. Traditional retrieval techniques such as TF–IDF and cosine similarity lack semantic understanding, limiting their usefulness in natural conversation settings [7],[8], [9], [10], [11], [12].

More advanced solutions, including transformer-based models and fine-tuned large language models (LLMs), offer higher semantic reasoning but require substantial training data, computational resources, and engineering expertise requirements that exceed the capabilities of most SMEs [13], [14], [15], [16]. Even Parameter-Efficient Fine-Tuning (PEFT) techniques, designed to reduce computational cost, still demand technical proficiency and are more suitable

***Corresponding author: Rosalina (rosalina@president.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i2.1163

for larger enterprises [17], [18], [19], [20]. Retrieval-Augmented Generation (RAG) provides a practical alternative by generating responses grounded in verified business knowledge, reducing hallucination and eliminating the need for model retraining [21], [22]. When embedded directly in a mobile keyboard interface, RAG enables sellers to access automated, context-aware communication support inside their existing messaging platforms. Nevertheless, communication alone does not fully address SME needs. Many SMEs also require simple tools for understanding purchasing patterns and designing targeted promotions. RFM (Recency, Frequency, Monetary) segmentation commonly combined with clustering remains one of the most effective methods for deriving customer insights without requiring analytical expertise.

This study proposes a practical alternative in the form of a cloud-based service accessed directly by individual small enterprises through a mobile keyboard interface. The system does not require local infrastructure or model training. Instead, sellers interact with the service while replying to customers in their existing messaging applications, enabling lightweight adoption without disrupting established workflows. The framework integrates two complementary functions. First, a retrieval-grounded response engine generates replies based on verified business knowledge, ensuring that answers remain accurate and contextually relevant. Second, a recency–frequency–monetary segmentation module analyzes historical invoice data to classify customers into behavioral groups, enabling sellers to design targeted promotional strategies. By embedding both communication support and customer analytics into a single keyboard-based interface, the proposed system aims to improve response quality, operational efficiency, and data-driven decision making in everyday small business activities.

2. Literature Review

Early retrieval systems relied on exact keyword matching, so they only returned results when the same words appeared in stored text [10], [11]. TF–IDF later improved ranking by weighting important terms, but it still depended on vocabulary overlap and failed when users phrased similar questions differently [7]. Cosine similarity refined this process but could not capture true meaning and remained sensitive to word frequency bias [23], [24]. Even modern retriever–reader models inherit these weaknesses because they use TF–IDF and cosine similarity as their retrieval backbone [9]. As a result, keyword-based approaches often produce irrelevant or generic answers in real customer interactions. Transformer-based models offer deeper semantic understanding, but fine-tuning them requires large datasets and high computational resources, which SMEs typically lack [13], [15]. Although Parameter-Efficient Fine-Tuning reduces this burden, it still demands technical infrastructure and expertise [17], [18], [19], [20]. Prompting LLMs directly with full business knowledge avoids training, yet it is constrained by limited context windows, prone to hallucinations, and costly at scale [25]. Given these constraints, we adopt Retrieval-Augmented Generation (RAG) as a more practical solution. RAG retrieves relevant knowledge from a curated database and uses it to guide generation, reducing hallucinations while avoiding expensive fine-tuning. In this study, we apply RAG in a smartphone keyboard setting so business knowledge can be injected directly into customer chats, improving both accuracy and everyday usability for SMEs.

Parallel to developments in retrieval–generation models, research on AI-assisted interfaces has explored embedding intelligent functions directly into user interaction tools, such as writing assistants, smart input methods, and conversational agents. These studies emphasize that integrating artificial intelligence into existing workflows lowers adoption barriers and improves usability, especially in environments where users are non-technical. However, most prior work focuses on desktop-based writing assistants or standalone chatbots, with limited attention given to mobile keyboard–level integration for business communication. This study extends the literature by situating retrieval-grounded generation within a mobile keyboard interface tailored to small enterprise operations. By combining hybrid retrieval–generation techniques with customer segmentation analytics in a single lightweight interface, the proposed framework bridges the gap between conversational support tools and business intelligence systems, an integration that has received limited empirical attention in previous research.

3. Methodology

This study adopts an implementation-focused methodology that integrates Retrieval-Augmented Generation (RAG) with RFM-based customer segmentation to enhance SME business intelligence. The system consists of six components:

ISSN 2723-6471

knowledge base construction, semantic chunking and embedding, vector retrieval, generative response modeling, mobile keyboard integration, and RFM segmentation. The system architecture is shown in figure 1.

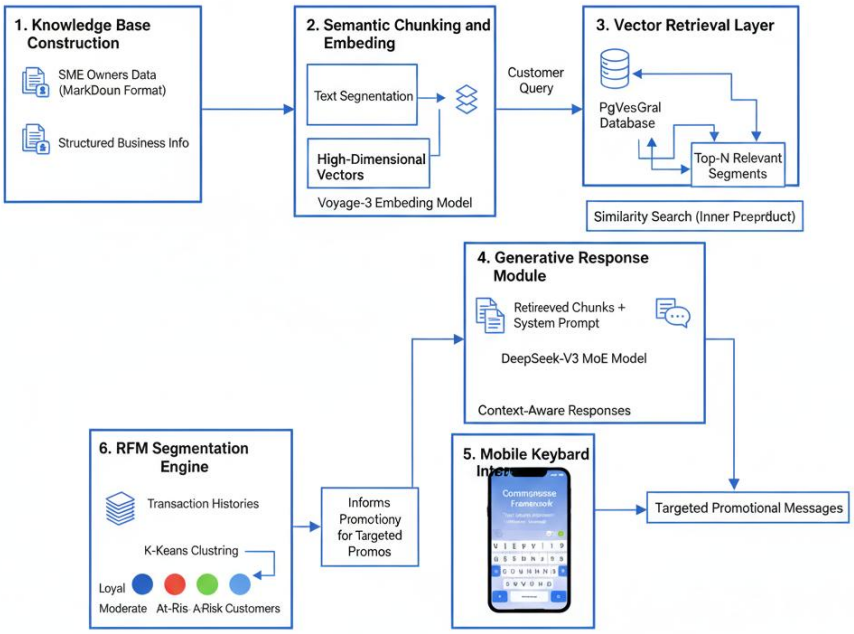


Figure 1. System architecture

3.1. Knowledge Base Construction

Structured business information, including product descriptions, pricing, store policies, and frequently asked questions, is collected directly from individual small enterprise owners. The content is standardized using Markdown formatting to ensure consistency and ease of maintenance. Markdown was selected due to its lightweight syntax, readability, and minimal technical overhead, enabling non-technical users to update business knowledge without specialized tools. Compared with more formal alternatives such as JSON-LD or XML, Markdown offers superior accessibility for small enterprises because it does not require strict schema definitions or specialized editors. Its hierarchical structure using headings and bullet lists also aligns naturally with the semantic chunking process employed in this study, allowing product categories, policies, and promotional information to be segmented into meaningful knowledge units with minimal preprocessing. This balance between structure and usability makes Markdown particularly suitable for small business environments where simplicity and maintainability are critical.

3.2. Semantic Chunking and Embedding

The Semantic Chunking and Embedding stage serves as the foundation of the proposed RAG framework. Its primary function is to transform structured business information into numerical vector representations suitable for efficient similarity-based retrieval. This stage ensures that the system captures the semantic meaning of SME knowledge, enabling accurate and contextually grounded responses during generation. As shown in figure 1, this stage consists of two sequential operations; (1) Semantic Chunking, which segments the text into meaningful units, and (2) Embedding, which converts each segment into a high-dimensional vector. Both processes are essential for maintaining semantic integrity and enabling precise retrieval. The input to this module is the structured business information formatted in Markdown. Instead of using simple fixed-length splitting which risks cutting sentences in unnatural locations and losing semantic coherence the system applies dynamic, content-aware chunking rules. Each chunk $Chunk_i$ is constructed from a sequence of sentences as in (1) and subject to the length constraint as in (2) where: s_j is a sentence, $|s_j|$ is a token length of a sentence, and L_{min}, L_{max} is a token thresholds ensuring coherent and manageable segments.

$$Chunk_i = \{s_k, s_{k+1}, \dots \dots . s_{k+n}\} \quad (1)$$

$$L_{min} \leq \sum_{j=k}^{k+n} |s_j| \leq L_{max} \quad (2)$$

Each chunk is then transformed into a high-dimensional vector through the Voyage-3 text embedding model. This model encodes the semantic meaning of the text such that conceptually similar chunks occupy nearby positions within the vector space, each chunk is embedded as in (3) where $v_i \in \mathbb{R}^d$ and d is the embedding dimension. The resulting embedding vectors preserve semantic relationships as in (4). There vectors are then forwarded to the Vector Retrieval Layer and stored using PgVector, enabling high-accuracy inner-product similarity search during user queries.

$$v_i = \mathrm{f}_{Voyage-3}(Chunk_i) \tag{3}$$

$$similarity(v_i, v_j) \propto semantic_{relatedness}(Chunk_i, Chunk_j) \tag{4}$$

In practice, the system builds chunks by gradually grouping sentences until a preset token limit is reached. Once the limit is about to be exceeded, the current chunk is closed and a new one is started. Markdown section headers are used as soft boundaries so each chunk stays focused on a single topic, such as a specific product or policy. This keeps the content well organized without adding heavy preprocessing. For embeddings, the system relies on a commercial third-party API that converts short business texts into semantic vectors. The model is not trained internally, but selected for its ability to capture meaning accurately. Using an external service also avoids the need for local deployment, which makes the solution more practical for small businesses.

3.3. Vector Retrieval Layer

All embedding vectors generated during the Semantic Chunking and Embedding stage are stored in a PostgreSQL database equipped with the PgVector extension. PgVector enables efficient Approximate Nearest Neighbor (ANN) and exact vector similarity search, allowing the system to scale to hundreds or thousands of knowledge segments without compromising speed. When a customer submits a query via the mobile keyboard interface, the system converts the raw text into a query embedding vector using the same embedding model (Voyage-3) applied to the knowledge chunks as in (5) using the same model ensures that both the query and the stored knowledge vectors exists in the same semantic vector space, making similarity computation meaningful and consistent. In order to identify which part of the knowledge base are most relevant, the system computes the inner product similarity between the query vector q and each stored knowledge vector v_i as in (6) and to select Top-N knowledge chuck with the highest similarity scores as in (7), only these Top-N chunks are forwarded to the Generative Response Module.

$$\mathrm{q} = \mathrm{f}_{Voyage-3}(q) \tag{5}$$

$$score(q, v_i) = q.v_i \tag{6}$$

$$TopN(q) = arg \max_{i=1,\dots k}(q.v_i) \tag{7}$$

3.4. The Generative Response

The Generative Response Module represents the final stage of the RAG pipeline, where the retrieved knowledge is synthesized into a coherent, customer-ready response. Its primary function is to ensure that all generated answers remain grounded in the SME's verified information while maintaining clarity, fluency, and conversational relevance. Following the retrieval of relevant information, the module begins with an Input Assembly process, which prepares and structures all elements required for response generation. This stage integrates three components into a unified contextual sequence: (i) the system prompt P, which defines the model's operational role, tone, and grounding constraints; (ii) the ordered set of Top-N retrieved knowledge chunks $C = \{c_1, c_2, \dots, c_n\}$; and (iii) the original user query q. To ensure the transformer model receives these components in a consistent and instruction-optimized arrangement, the final input sequence is constructed as in (8) which compactly constructed as in (9).

$$\mathrm{Input} = \mathrm{P} \,\|c_1\|c_2\,\|\dots\|c_N||q \tag{8}$$

$$Input = concat(P, TopN(q), q \tag{9}$$

$$|Tokenizer(Input)| \leq \tau_{max} \tag{10}$$

Before inference, the assembled input must satisfy the model's maximum window constraint as in (10) where τ_{max} is the maximum number of tokens allowed by DeepSeek-V3. If the input exceeds this limit, lower-ranked chucks are truncated according to their similarity score as in (11) ensuring that only the most semantically relevant segments are

preserved. Then the entire concatenated input is then transformed into a token sequence as in (12) where m denotes the total number of token. During generation, the model attends over this sequence through its self-attention mechanism, defined at each layer as in (13), and grounding is enforced by constraining the model to generate an output $\hat{y}$ that is semantically derivable from the retrieved chucks as in (14) while simultaneously conditioned on user intent expressed in q.

$$s_i = q.v_i \tag{11}$$

$$T = [t_1, t_2, \dots t_m] = Tokenizer\ (Input) \tag{12}$$

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{13}$$

$$\hat{y} \subseteq \bigcup_{i=1}^{N} c_i \tag{14}$$

3.5. Mobile Keyboard Integration

The Mobile Keyboard Integration module functions as the operational front-end of the system, enabling real-time interaction with the RAG pipeline directly within the seller's existing communication workflow. Implemented as a custom Android Input Method Editor (IME), the module embeds AI-assisted response generation into any application that supports text entry, thereby eliminating the need to switch between interfaces. While responding to customers in messaging applications, the seller inputs or pastes the customer's message into the IME. A dedicated trigger mechanism (e.g., an "AI Reply" button) initiates the RAG pipeline by transmitting the query q to the backend as in (15) performed asynchronously to ensure the user interface remain responsive, the IME listens for the generated response $\hat{y}$ returned from the backed as in (16), upon receipt, the keyboard inserts the AI-generated draft either into the suggestion bar or directly into the text field.

$$q \xrightarrow{IME\ Trigger} RAG\ pipeline \tag{15}$$

$$\hat{y} = f_{DeepSeek-V3}(Input), \quad IME \leftarrow \hat{y} \tag{16}$$

3.6. RFM Segmentation Engine

The RFM Segmentation Engine provides a complementary analytical capability within the framework by generating data-driven customer categories based on historical purchasing behavior. This module operates independently of the conversational pipeline and supports targeted promotional strategies generated by the LLM. Customer transaction histories are extracted from SME invoice records. For each customer i, three behavioral metrics are computed (a) recency (R) as in (17), (b) frequency (F) as in (18) where T_i is the number of transactions, (c) monetary (M as in (19). These metrics form the feature vector for each customer as in (20).

$$R_i = today - last\_purchase_i \tag{17}$$

$$F_i = \sum_{t=1}^{T_i} 1 = T_i \tag{18}$$

4. Results and Discussion

4.1. Retrieval Performance

We measured retrieval quality using the RAGAs faithfulness metric. As shown in figure 2, the system achieved a score of 0.997, meaning almost all retrieved chunks were highly aligned with the customer queries. This strong result comes from three key design choices: coherent semantic chunking, high-quality Voyage-3 embeddings, and precise inner-product search using PgVector. Because the generative model depends entirely on the retrieved content, this level of

retrieval accuracy is essential for keeping responses grounded in real business knowledge. To assess the robustness of the reported retrieval and generation performance metrics, 95% confidence intervals were computed across repeated experimental trials using non-parametric bootstrap resampling. For each metric, 1,000 bootstrap samples were drawn with replacement from the original evaluation set, and the percentile method was applied to estimate lower and upper confidence bounds. This procedure provides statistical assurance that the observed performance values are stable and not driven by random sampling effects.

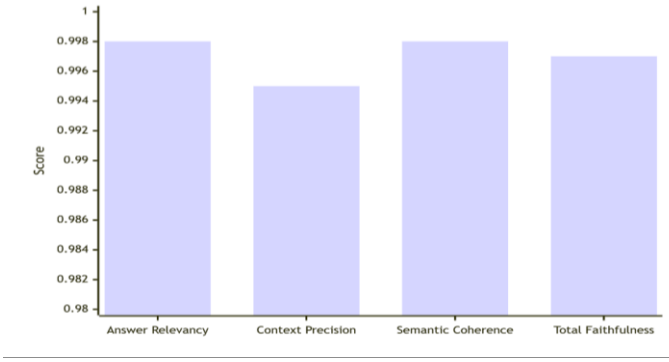


Figure 2. Performance metrics across trials

Table 1 shows the similarity scores produced by the vector retrieval layer for different knowledge chunks. These scores measure how closely each chunk matches the customer query, with higher values indicating stronger relevance. The results follow a clear descending pattern. Chunks like Market Expansion (0.92) and Recent Developments (0.85) are highly aligned with the query and are therefore prioritized. Medium scores such as Company Overview (0.67) reflect partial relevance, while lower scores like Financial Performance (0.43) and Products and Services (0.35) show weaker connections. This pattern confirms that the retrieval system can effectively separate highly relevant content from less useful material, helping ensure that only the most appropriate information is passed to the generative model.

Table 1. Pipeline Auto Text RAG Chunk Similarity Results

Chunk Heading	Similarity Rate
Market Expansion	0.92
Recent Developments	0.85
Company Overview	0.67
Financial Performance	0.43
Products and Services	0.35

The comparatively low similarity score observed for the “Products and Services” category does not indicate a failure of the retrieval mechanism. The evaluated customer query was focused on market expansion strategies rather than on specific product attributes, which led the semantic retrieval process to prioritize knowledge chunks related to growth planning and recent developments. As a result, product-oriented content exhibited weaker semantic alignment with the query and was therefore ranked lower, reflecting appropriate behavior of the vector retrieval process rather than retrieval error.

4.2. Generative Accuracy and Grounding Quality

We evaluated answer quality using the RAGAs correctness metric, which compares generated outputs with ground-truth responses from the SME knowledge base. As shown in figure 3, the model achieved a score of 0.88, indicating that most responses followed the retrieved facts closely. To see how much RAG contributes to this result, we compared it with an LLM-only model and a TF-IDF + cosine similarity baseline. We built the TF-IDF baseline using Scikit-learn with standard settings, and the LLM-only baseline used the same generative model but without any retrieval support. By testing all systems on the same set of queries, we ensured a fair comparison and confirmed that grounding generation with retrieval is the key driver of the performance gains we observed.

ISSN 2723-6471

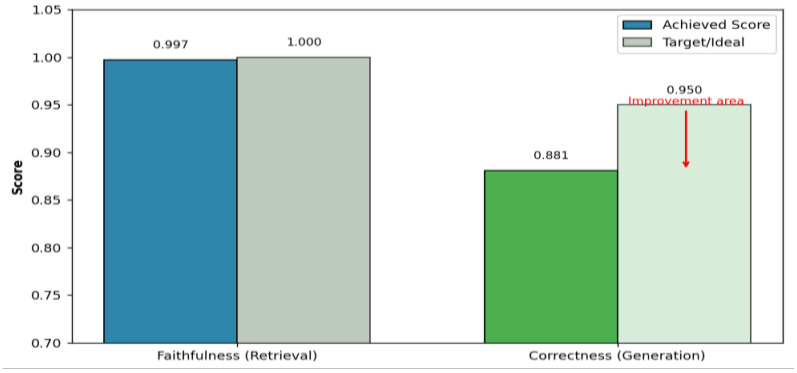
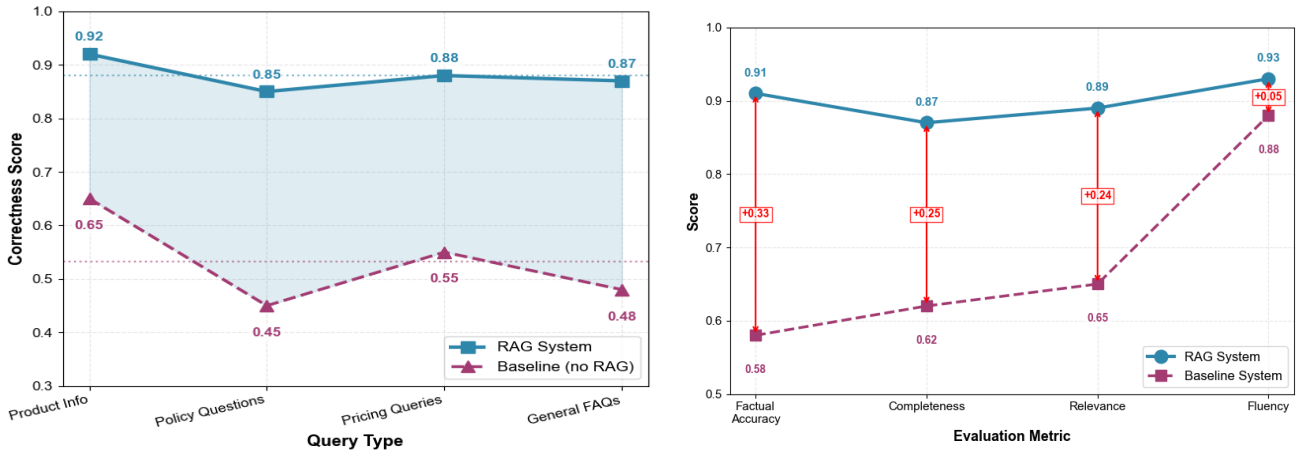


Figure 3. RAG pipeline performance comparison

Beyond overall correctness, a more detailed performance breakdown was evaluated across four qualitative metrics: factual accuracy, completeness, relevance, and fluency. Results presented in figure 4(a) show that the RAG system consistently surpasses the baseline across all dimensions, with the largest improvements observed in factual accuracy (+0.33) and completeness (+0.25). These findings highlight the role of embedded retrieval in grounding the model's outputs, especially for queries requiring specific policy details or product parameters.



(a) (b)

Figure 4. (a) Accuracy analysis by query category, (b) Detailed performance metrics comparison

To assess how the system performs across different information needs, accuracy was evaluated by query category. As shown in figure 4(b), the RAG model maintains high correctness across all four types of customer questions: product information, policy, pricing, and general FAQs. These categories reflect the most common queries faced by SMEs in daily operations. Product-related questions achieved the highest score (0.92), supported by well-structured knowledge entries. Policy queries, which are usually more complex, still performed strongly at 0.85 far above the baseline model's 0.45. Pricing and FAQ queries also showed reliable results, with scores of 0.88 and 0.87, confirming the system's consistent performance across varying query types. Meanwhile, table 2 presents the average RFM profiles for the Loyal, Moderate, and At-Risk segments. Loyal customers show frequent, recent, and high-value purchases, making them ideal for rewards and exclusive offers. Moderate customers display stable but average engagement and can be improved through cross-selling and regular communication. At-Risk customers have low activity and spending, indicating a high churn risk and the need for re-engagement strategies such as discounts and personalized reminders.

Table 2. Actionable Characteristics of Customer Clusters

Cluster	Avg. Recency	Avg. Frequency	Avg. Monetary	Suggested Action
Loyal	Low (Recent)	High	High	Reward programs, exclusive member benefits
Moderate	Medium	Medium	Medium	Cross-selling, product education, informational newsletters
At-Risk	High (Long ago)	Low	Low	Re-engagement campaigns, personalized discounts

4.3. System Latency and Operational Responsiveness

Latency evaluations were conducted on mid-range Android devices operating under typical mobile network conditions. The complete query–retrieve–generate cycle achieved an average end-to-end response time of 5.02 seconds, indicating that the system operates efficiently within real-world SME communication environments. Table 3 listed the breakdown of processing time across the three main components in the pipeline. User Acceptance Testing was conducted with 18 small enterprise sellers operating in retail and service sectors, each with at least six months of experience using mobile-based messaging platforms for customer communication. Participants were asked to perform routine customer response tasks using the keyboard interface over a one-week period and subsequently completed a structured questionnaire evaluating perceived response speed, workflow disruption, and overall usability.

Table 3. Latency Breakdown of the Query–Retrieve–Generate Pipeline

Component	Average Time (seconds)	Description
Embedding + Retrieval	1.90 s	Query embedding and vector similarity search using PgVector
LLM Inference	2.70 s	Response generation using DeepSeek-V3 MoE
Network & API Handling	0.42 s	Transmission latency, API routing, and response delivery
Total Latency	5.02 s	Full end-to-end processing time

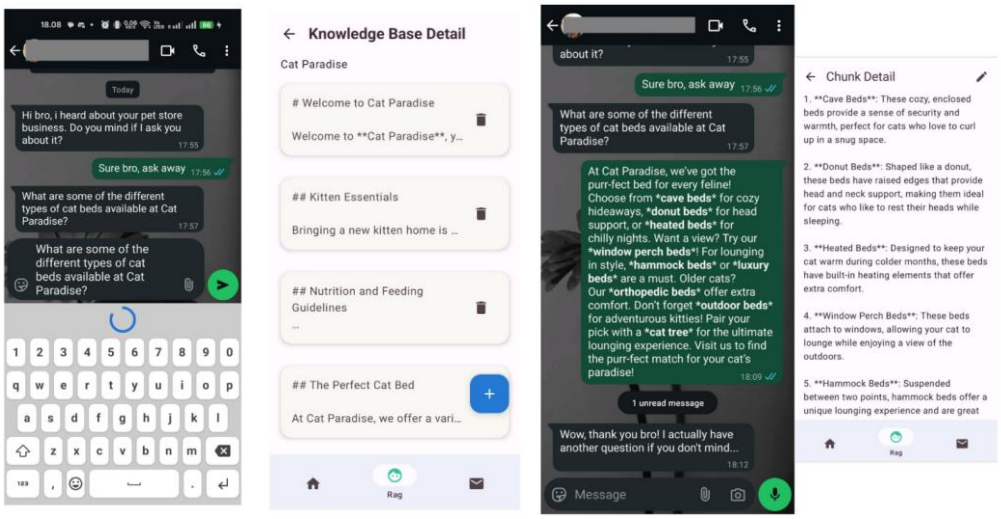
Despite the multi-stage nature of the pipeline, the measured latency remains well within acceptable bounds for real-time customer communication. SME sellers participating in the User Acceptance Testing (UAT) phase consistently reported that the AI-generated replies felt “near real-time” and did not interfere with ongoing conversations. Table 4 listed the UAT results related to latency perception and user experience. These findings confirm that the mobile keyboard integration is operationally viable and responsive in practical daily use.

Table 4. UAT Feedback on Latency and System Responsiveness

Evaluation Item	UAT Score / Response	Description
Perceived Response Speed	92% “near real-time”	Majority reported no noticeable delay in conversations
Impact on Conversation Flow	89% “no disruption”	AI replies did not interrupt messaging rhythm
Acceptability of 5-second Latency	94% “acceptable”	Latency judged reasonable for mobile business operations
Overall Responsiveness Satisfaction	91% “satisfied/very satisfied”	Users noted the system felt smooth and responsive

To complement the quantitative latency evaluation, figures 6 shows how these performance outcomes translate into real usage scenarios. Figure 7 presents the operational interfaces that enable fast interaction specifically, the keyboard-embedded RAG query input and the knowledge base management module that supports efficient retrieval. Meanwhile, figure 6(c) shows the resulting answer relevancy and correctness, demonstrating that the system not only responds quickly but also maintains high-quality outputs. To ensure data security and user trust, all query transmissions between the mobile keyboard and the backend service are encrypted using standard secure communication protocols, and no raw customer messages are permanently stored beyond the processing session. Retrieved knowledge is restricted to the seller’s own business repository, preventing cross-tenant data leakage. In terms of usability, the keyboard interface is designed to mirror standard mobile input behavior, minimizing the learning curve for SME users. During onboarding, sellers are provided with brief in-app guidance to familiarize them with the AI reply trigger mechanism, ensuring that the system can be adopted without disrupting existing communication habits.

ISSN 2723-6471



(a) (b) (c)

Figure 6. (a) Keyboard interface for RAG query (b) Knowledge base management and (c) relevancy and correctness interface

4.4. RFM Segmentation Outcomes

To check how well the RFM segments can be separated, we used ROC–AUC analysis. As shown in figure 7, the At-Risk group is the easiest to identify, with a very high AUC of 0.93. The Loyal group is also well distinguished, with an AUC of 0.85. However, the Moderate group performs poorly (AUC 0.52), which is expected because mid-level customers often show mixed behavior and overlap with other segments. The number of clusters was set to three to reflect the widely adopted and operationally interpretable customer categories of Loyal, Moderate, and At-Risk. This choice was further validated using Silhouette Score analysis, where values peaked at K=3 indicating the best balance between intra-cluster cohesion and inter-cluster separation. The Silhouette coefficient for the selected configuration was 0.61, which is considered acceptable for behavioral segmentation tasks and supports the suitability of the chosen cluster structure for practical SME applications.

The comparatively low ROC–AUC value observed for the Moderate customer cluster reflects the intrinsic behavioral overlap between medium-engagement customers and both Loyal and At-Risk groups. Customers in this segment frequently exhibit transitional purchasing patterns, such as declining frequency or sporadic high-value transactions, which reduces separability in the feature space. This phenomenon is consistent with established RFM theory and does not undermine the operational usefulness of the segmentation, as the framework primarily targets high-value Loyal customers and churn-prone At-Risk customers for decision support.

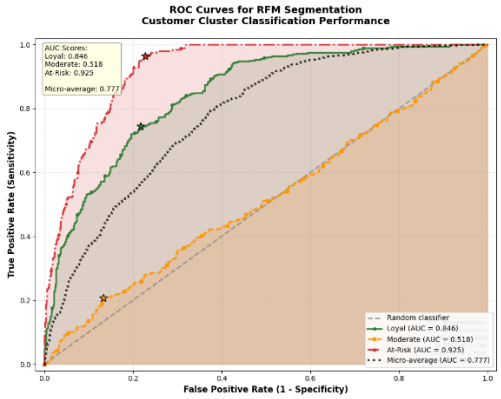


Figure 7. ROC Curves for RFM Segmentation of Customer Cluster Classification Performance

ISSN 2723-6471

To support the ROC–AUC results, figure 8 shows how customers are distributed across Recency, Frequency, and Monetary values. Loyal customers form a clear cluster with recent activity, high purchase frequency, and high spending, while At-Risk customers appear in the opposite region with long inactivity, low frequency, and low monetary value. Moderate customers lie between these two groups and overlap with both, which explains why this segment is harder to classify.

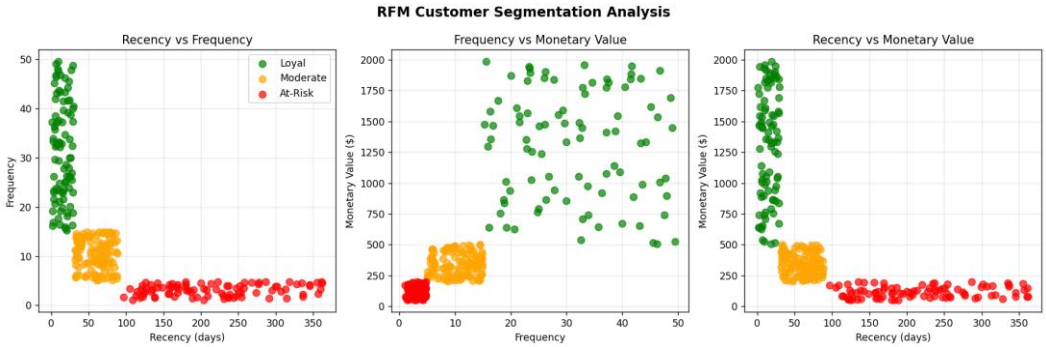


Figure 8. RFM customer segmentation analysis

5. Discussion

This study shows that combining RAG with RFM customer analytics improves how SMEs handle digital customer communication. High retrieval faithfulness reduces irrelevant or incorrect responses, which is especially important in a mobile keyboard setting where replies must be fast and accurate. The RAG system was also more stable than baseline models across different types of customer questions, from simple product checks to complex pricing issues. Although the system uses multiple processing stages, its response time still felt natural in real conversations, and positive UAT feedback confirms that the solution is not only accurate but also practical for everyday business use. Beyond improving communication quality, the RFM segmentation component adds a behavioral perspective that complements the conversational system. Loyal and At-Risk customers are clearly distinguished, while the Moderate group remains less defined, reflecting the typical nature of mid-range behavioral patterns. By linking these segments with generative responses, the system can tailor promotional messages and create a practical bridge between conversational AI and customer intelligence two areas that are often treated separately in SME research. The empirical findings of this study provide a foundation for extending the framework toward more expressive customer behavior modeling. While the current implementation relies on static recency, frequency, and monetary attributes, the strong retrieval faithfulness and stable segmentation outcomes suggest that the architecture can accommodate richer temporal features such as purchase sequences, seasonal effects, and lifecycle transitions. Integrating such temporal dynamics into the existing pipeline would enable the system to anticipate customer needs more proactively and further enhance the personalization of promotional strategies.

6. Conclusion

Within this research, a unified seller-support framework integrating Retrieval-Augmented Generation and RFM segmentation was developed to address the communication and analytical challenges faced by SMEs in digital marketplaces. The system combined semantic chunking, embedding-based retrieval, cloud-based generation, and mobile keyboard integration to deliver context-grounded, timely responses directly within existing messaging workflows. Experimental results demonstrated high retrieval faithfulness, strong generative correctness, practical response latency, and stable segmentation performance, confirming that the proposed architecture performs reliably under real usage conditions. The findings suggest that embedding AI-assisted retrieval and lightweight analytics into mobile interactions can enhance seller responsiveness and support more informed customer engagement strategies. Future work may expand the framework by implementing automated knowledge-base updating, integrating multilingual capabilities, and exploring more expressive customer models that incorporate temporal behavior or purchase sequences. Additional investigations involving larger and more diverse SME datasets could further validate

the system's generalizability. These extensions can be incorporated without redesigning the core pipeline, as the existing architecture already accommodates modular expansion through its cloud services and mobile interface.

7. Declarations

7.1. Author Contributions

Conceptualization: R., N.L.I., G.S., and J.T.N.W.; Methodology: R. and G.S.; Software: J.T.N.W.; Validation: R., N.L.I., and G.S.; Formal Analysis: R. and G.S.; Investigation: R.; Resources: N.L.I.; Data Curation: G.S.; Writing Original Draft Preparation: R. and J.T.N.W.; Writing Review and Editing: R., N.L.I., and J.T.N.W.; Visualization: G.S.; All authors have read and agreed to the published version of the manuscript.

7.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7.3. Funding

This research was funded by the International Funding Research Collaboration Program under the joint partnership between President University, Indonesia, and UNITAR International University, Malaysia, for the period of January 2026 to January 2027. The authors would like to express sincere appreciation to the Faculty of Computer Science at President University and to UNITAR International University for their academic support and collaborative research environment. Special thanks are also extended to the participating industry stakeholder who contributed valuable feedback during the User Acceptance Testing phase.

7.4. Institutional Review Board Statement

Not applicable.

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Adomako and M. Ahsan, "Entrepreneurial passion and SMEs' performance: Moderating effects of financial resource availability and resource flexibility," *Journal of Business Researc*h, vol. 144, no. 1, pp. 122–135, May 2022, doi: 10.1016/j.jbusres.2022.02.002.

[2] Indah Ramadhani, D. Dailami, Ulva Widiya, I. Yunita, Ridha Nurhuda, and M. Mutia, "The Influence of Response Speed and Information Quality on the Effectiveness of SME Sales Through Facebook," *Golden Ratio of Mapping Idea and Literature Format*, vol. 6, no. 1, pp. 117–124, Jul. 2025, doi: /10.52970/grmilf.v6i1.1377.

[3] S. W. Arista, Sigit Hermawan, S. W. Arista, and Sigit Hermawan, "Improving MSME Performance Based on Digital Marketing, Intellectual Capital, Product Innovation and Competitive Advantage," *Jurnal Manajemen Bisnis*, vol. 16, no. 2, pp. 526–557, Aug. 2025, doi: 10.18196/mb.v16i2.25192.

[4] M. S. Mahrinasari, S. Bangsawan, and M. F. Sabri, "Local wisdom and Government's role in strengthening the sustainable competitive advantage of creative industries," *Heliyon*, vol. 10, no. 10, pp. 1-13, May 2024, doi: 10.1016/j.heliyon.2024.e31133.

[5] F. S. Singagerda and S. Riadi, "Competitive Advantage of Family Business and the Barriers: Evidence from Indonesia SME's," *Proceedings of the Proceedings of the 1st Workshop on Multidisciplinary and Its Applications Part 1, WMA-01 2018*, 19-20 January 2018, Aceh, Indonesia, vol. 2019, no. Jan., pp. 1–10, 2019, doi: 10.4108/eai.20-1-2018.2281916.

[6] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[7] K. Zhou, K. Ethayarajh, D. Card, and D. Jurafsky, "Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words," *arXiv,* vol. 2022, no. May, pp. 1–12, 2022. https://arxiv.org/abs/2205.05092 (accessed Apr. 18, 2024).

[8] M. Ahmed, H. U. Khan, S. Iqbal, and Qutaibah Althebyan, "Automated Question Answering based on Improved TF-IDF and Cosine Similarity," *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, vol. 2022, no. Nov., pp. 1–6, 2022, doi: 10.1109/snams58071.2022.10062839.

[9] Z. FU, X. SUN, Q. LIU, L. ZHOU, and J. SHU, "Achieving Efficient Cloud Search Services: Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing," *IEICE Transactions on Communications*, vol. E98-B, no. Jan., pp. 190–200, 2015, doi: 10.1587/transcom.e98.b.190.

[10] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, Feb. 2016, doi: 10.1109/tpds.2015.2401003.

[11] Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), vol. 2017, no. Jul., pp. 1–12, 2017, doi: 10.18653/v1/p17-1171.

[12] C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins, "Synthetic QA Corpora Generation with Roundtrip Consistency," *arXiv,* vol. 2019, no. Jun., pp. 1–12, 2019, doi: 10.48550/arxiv.1906.05416.

[13] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," *arXiv,* vol. 2018, no. Jan., pp. 1–12, 2018, doi: 10.48550/arxiv.1801.06146.

[14] M. A. Bakker , "Fine-tuning language models to find agreement among humans with diverse preferences," *arXiv,* vol. 2022, no. Nov., pp. 1–12, 2022, doi: 10.48550/arxiv.2211.15006.

[15] C. Pornprasit and C. Tantithamthavorn, "Fine-tuning and prompt engineering for large language models-based code review automation," *Information and Software Technology*, vol. 2024, no. Jul., pp. 1–12, 2024, doi: 10.1016/j.infsof.2024.107523.

[16] N. Ding , "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, Mar. 2023, doi: 10.1038/s42256-023-00626-4.

[17] D. M. Anisuzzaman, J. G. Malins, P. A. Friedman, and Z. I. Attia, "Fine-Tuning LLMs for Specialized Use Cases," *Mayo Clinic Proceedings: Digital Health*, vol. 3, no. 1, pp. 1-12, Nov. 2024, doi: 10.1016/j.mcpdig.2024.11.005.

[18] B. Mohammadi, E. Abbasnejad, Y. Qi, Q. Wu, A. Van Den Hengel, and J. Q. Shi, "Parameter-efficient action planning with large language models for vision-and-language navigation," *Pattern Recognition*, vol. 172, no. 1, pp. 1-12, Apr. 2026, doi: 10.1016/j.patcog.2025.112462.

[19] L. Wang , "Parameter-efficient fine-tuning in large language models: a survey of methodologies," *Artificial Intelligence Review*, vol. 58, no. 8, pp. 1-12, May 2025, doi: 10.1007/s10462-025-11236-4.

[20] W. Fan , "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," in *KDD '24, Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 2024, no. Aug., pp. 6491–6501, 2024, doi: 10.1145/3637528.3671470

[21] Patrick , "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Neural Information Processing Systems*, vol. 33, no. May, pp. 9459–9474, May 2020, doi: 10.48550/arXiv.2005.11401.

[22] B. Kim, J. Oh, and C. Min, "Investigation on Applicability and Limitation of Cosine Similarity-Based Structural Condition Monitoring for Gageocho Offshore Structure," Sensors, vol. 22, no. 2, pp. 663-678, Jan. 2022, doi: 10.3390/s22020663.

[23] H. Steck, C. Ekanadham, and N. Kallus, "Is Cosine-Similarity of Embeddings Really About Similarity?," *arXiv*, vol. 2024, no. Mar., pp. 1–12, 2024, doi: 10.1145/3589335.3651526.

[24] L. Giray, "Prompt Engineering with ChatGPT: a Guide for Academic Writers," *Annals of Biomedical Engineering*, vol. 51, no. 1, pp. 2629–2633, Jun. 2023, doi: 10.1007/s10439-023-03272-4.