Comparative Analysis of RAG-Based Open-Source LLMs for Indonesian Banking Customer Service Optimization Using Simulated Data

Hendra Lijaya[1]*, Patricia Ho[2], Handri Santoso[3]
Department of Informatics, Science and Technology Faculty[1], [2], [3]
Universitas Pradita
Tangerang, Indonesia
hendra.lijaya@student.pradita.ac.id[1], patricia.ho@student.pradita.ac.id[2], handri.santoso@pradita.ac.id[3]

Abstract*— In the digital era, banks face challenges in delivering fast, accurate, and efficient customer service, especially for frequently asked simple questions. This study evaluates the effectiveness of three open-source Large Language Models (LLMs), namely Gemma2-9B-Sahabat-AI, Qwen2.5-14B-Instruct, and Mistral-Nemo-Instruct in supporting a Retrieval-Augmented Generation (RAG) question-answering system for the banking sector. Using 12,000 synthetic billing documents indexed with intfloat/multilingual-e5-large-instruct embeddings (1024 dimensions), model performance was assessed via semantic similarity metrics, LLM-as-a-Judge scores (GPT-4o-mini and Gemini 2.0 Flash), and human validation Gemma2-9B-Sahabat-AI achieved the highest semantic similarity score (0.9627), followed by Mistral (0.9614) and Qwen2.5 (0.9284). In LLM-as-a-Judge evaluations, Qwen2.5 ranked highest on GPT-4o-mini (92.2), while Gemma2 led under Gemini 2.0 Flash (88.4). Human evaluators gave perfect scores for factual questions (1–10), but all models struggled with arithmetic in question 13. Gemma2's average response time was 41 seconds, faster than Qwen2.5's 72 seconds and Mistral's 48 seconds, confirming Gemma2's balanced performance in accuracy, speed, and computational efficiency. These findings underscore the potential of locally operated open-source LLMs for banking applications, ensuring privacy and regulatory compliance. However, limitations include reliance on synthetic data, a narrow question set, and lack of user diversity. Future research should involve broader queries, real user testing, and numeric reasoning modules to ensure robust and scalable deployment in real-world banking customer service environments.

Keywords*— Bank Customer Service, Large Language Model (LLM), LLM-as-a-Judge, Semantic Similarity, Retrieval-Augmented Generation (RAG)

I. INTRODUCTION

In the digital era, banks face the challenge of meeting customers' increasing demands for fast, responsive, and accurate services, particularly for simple queries like product and billing information. However, customer service (CS) in banks still faces significant issues.

Traditional customer service operations in the banking industry often come with hefty price tags. These expenses cover not only ongoing training and salaries, but also the upkeep of infrastructure. Consider PT Bank Muamalat Indonesia, for example. To cut costs, they downsized their workforce and adopted branchless banking, which effectively reduced spending on both staff training and infrastructure maintenance. [1].

Additionally, one of the most common customer complaints is the lengthy wait time before interacting with a customer service representative. This problem not only reduces customer satisfaction, but it also harms the bank's reputation [2]. According to research, long wait times increase customers' psychological tension, which influences service quality and satisfaction rates [3]. Poor queue management mechanisms, such as the lack of wait time estimates and poor waiting area facilities, exacerbate the customer experience. While some banks try to make waiting time more productive by providing entertainment or relevant information, waiting time remains a major contributor to reduced customer satisfaction [4].

Service quality also varies according to the individual representative's knowledge and workload. High stress and one-size-fits-all training can reduce consistency [5]. Manual methods for searching data in multiple systems slow down answers and impair efficiency. According to research, employing technology such as data-driven automation and self-service platforms can cut response times and increase productivity while maintaining quality [6].

Even with significant investments in training, banks' efforts frequently fall short because they do not properly assess the results. Without comprehensive evaluation, training does not always lead to increased employee performance or organisational outcomes [7]. Research underscores the importance of performance-based evaluation, like Kirkpatrick Model, and collaborative, outcome-focused knowledge management [8].

In addition to that, training evaluation approaches need pressing improvement to meet organizational strategic objectives. The study in the banking industry of South Africa indicated that training programs developed without performance-based evaluation tend to get out of touch with their

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

applicability in practice [9]. Poor adoption of skills in the workplace is also common due to poor knowledge management, particularly when training systems have insufficiently strong evaluation elements including outcome-based learning as well as collaborative competency building [10].

In tackling these customer service challenges, question-answering systems that deliver precise, contextually relevant, and timely information have become indispensable. Retrieval-Augmented Generation (RAG) stands out as a promising approach that integrates Large Language Models (LLMs) with external information retrieval, as shown in Figure 1 [11]. By retrieving relevant document chunks and incorporating them into the prompt, RAG enhances the accuracy and relevance of responses.

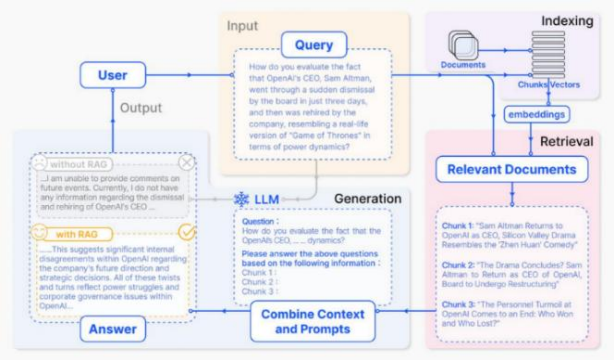

Fig. 1. Illustration of the Retrieval-Augmented Generation (RAG) Process [11]

In response to these challenges, various studies have sought to improve the performance gains of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) towards making question-answering systems more efficient and accurate in both customer service and financial contexts. Existing approaches were indicated to suffer from common shortcomings in RAG pipelines, including naive document chunking, overdependency on similarity-based retrieval, and underutilization of document structure [12]. Integrating Knowledge Graphs has been shown to improve retrieval precision [13], while retrieval-aware pretraining approaches like REALM offer greater modularity and scalability [14]. Dense Passage Retrieval (DPR) also performs better than earlier approaches such as BM25 [15], and hybrid architectures within RAG integrate retrievers and generators to support different NLP workflows [16]. In financial contexts, RAG pipelines with pre-retrieval filtering, hybrid retrieval, and reranking have achieved considerable improvements in quality [17], while benchmarks similar to FINDER provide authentic datasets to measure model performance in handling ambiguous queries [18].

But despite numerous attempts at the building of LLM and RAG models, their application to bank customer care remains a major technical challenge. Hallucination is one of the major problems, where models generate responses which appear correct but are entirely inaccurate, particularly an unsafe vulnerability as far as sensitive financial details are concerned [19]. The other major concern is the uninterpretability of these models, which makes it difficult to understand how exactly they make decisions and therefore reduce user faith in computerised systems [20]. Response bias, data protection, and the lack of support for local language or profound understanding of banking [19], [20] are also involved. Indonesia's banking sector has already made use of big data and artificial intelligence (AI) to identify fraud and target customers.

While RAG and LLMs are being used in customer service, they are not yet becoming mainstream. While advanced AI frameworks have the ability to transform initiatives like intelligent reporting and conversational banking, there are issues. Legacy IT systems, data silos, and burdensome data privacy rules, such as the Personal Data Protection Law's demands for express consent and strict adherence to Know Your Customer (KYC) and Anti-Money Laundering (AML) guidelines, all are impediments to their uptake [21]. Moreover, AI algorithm weaknesses and the lack of technical legal mechanisms to control the abuse of AI, for example, adversarial attacks, data poisoning, and tampering with fraud detection systems, inhibit the adoption of RAG and LLMs by Indonesia's banking sector [22].

In view of such banking customer service concerns, LLM and RAG-related technology holds tremendous promise to help with service tasks, especially in responding to basic and routine questions. The technologies promise to improve cost and time efficiency and ensure a more uniform, responsive, and secure customer experience. This study thus seeks to test and compare the performance and efficiency of three top-rated LLMs, Gemma, Qwen, and Mistral, based on their semantic similarity, LLM-as-a-Judge evaluation methodologies, and human validation. The outcomes are likely to provide insights into the strengths and capabilities of these models in enhancing service quality and operational efficiency in the banking industry.

II. RESEARCH METHOD

Figure 2 shows the research workflow and is split into two broad phases, namely the preparation phase and implementation phase. Within the preparation phase, there exist three prominent stages. The first is the Data Generation process that entails developing data based on transaction documents and practice and test questions. The second stage is choosing the most appropriate embedding model for transforming text data into vector embeddings. This is succeeded by candidate LLM selection to decide on those Large Language Models that need to be tested and evaluated.

In the implementation phase, the RAG system and model integration are carried out. This involves building a Retrieval-Augmented Generation (RAG) system that combines document retrieval results from a vector database with the generative capabilities of the LLMs. Finally, model performance is evaluated to measure the effectiveness and quality of responses generated by each model in the context of banking customer service.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

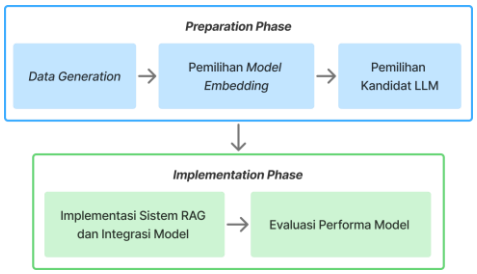

Fig. 2. Overview of the Research Framework

A. Data Generation

The data generation process begins by using validated bank billing documents as reference samples. Based on the phases in data generation, users send queries to a GPT model to generate simulated data in CSV format, containing a list of transactions based on the given examples. Each row in the CSV file represents a single customer transaction entry, which is then processed using the Google Sheet Sync plugin in Figma to generate a single PDF billing document. With a total of 12,000 data rows, this process produces 12,000 PDF documents as augmented data for credit card billing simulation.

Each generated PDF is then extracted into text format using LangChain, which enables automatic conversion from PDF to plain text. After extraction, the data is validated using regular expression methods to ensure that the document format adheres to the required standards. If the data is invalid, it is discarded and is not utilized. However, if it is valid, pertinent metadata including customer name and transaction date is extracted and added into the vector database during retrieval-based training data construction. As illustrated in Figure 3, this entire process is carefully engineered to provide real-world-like and organized data, which is essential for assessing the RAG system in banking customer service settings.

Synthetic data was chosen over real customer data to ensure compliance with data privacy regulations and to protect sensitive financial information. Real-world banking data often involves strict access restrictions and privacy concerns that could hinder the experimental process.

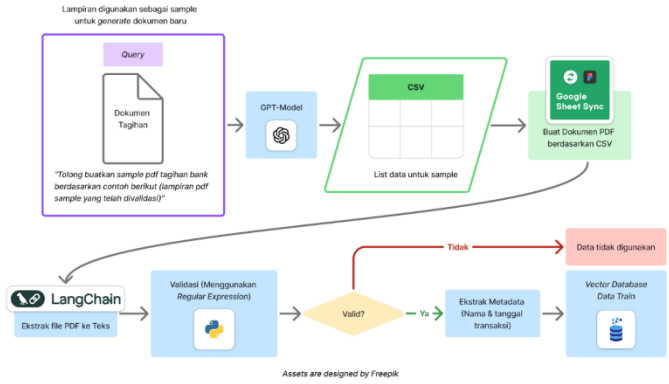

Fig. 3. Details of Each Phases in Data Generation

B. Embedding Model Selection

The second phase is choosing an embedding model to transform text data into vector embeddings. The selected model is intfloat/multilingual-e5-large-instruct due to its capability to effectively capture multilingual semantic representations strongly. This is a modern family of multilingual embeddings developed to handle over 100 languages and shows strong performance in high-resource and low-resource languages alike. The choice is based on evidence indicating that instruction-tuned methods with contrastive pre-training and supervised fine-tuning consistently yield high-quality embeddings that outperform earlier models on a wide array of multilingual retrieval tasks [23]. The model produces 1024-dimensional vector embeddings and is used as the embedding size in the vector database.

In addition, its design approach enables more accurate cross-lingual context understanding, making it highly suitable for applications such as semantic similarity-based retrieval and Retrieval-Augmented Generation (RAG). Recent advancements also show that embeddings of this kind can effectively unify cross-modal information, such as text and images, into a shared representation space without requiring costly multimodal training [24].

C. Selection of LLM Candidates

The third stage involves selecting the candidate LLMs to be evaluated. The models used in this study include Gemma2-9b-cpt-sahabatai-v1, Qwen2.5-14B-Instruct, and Mistral-Nemo-Instruct. The selection was based on several criteria, which are large parameter capacity, availability in the GGUF format for efficient inference, and the ability of each model to handle instruction-following tasks and support multilingual interaction, which aligns with the needs of the Indonesian banking sector. In addition, all three models represent high-performing open-weight approaches.

Gemma2-9B is utilized by Sahabat AI, a local LLM developing initiative, and thus is appropriate for testing models in local language and service applications. Specifically, Mistral-Nemo-Instruct and Qwen2.5 were selected over other models such as LLaMA because they are open source, highly performant, and offer superior multilingual support. Mistral has demonstrated exceptional performance across various NLP benchmarks and includes advanced features like grouped-query attention and sliding window attention for better efficiency. Qwen2.5, with up to 72B parameters in its family, has shown strong multilingual performance and versatility in handling complex tasks.

Also, the adoption of locally executed open-source models is further important from data security and regulatory points of view. For highly regulated banking organizations, execution of open models locally allows them to remain independent of third-party service providers and have all processes from extraction to response executed within a secure local system. This is important for privacy breach prevention and reducing chances of third-party sensitive data transfer. Local models allow flexibility in executing internal security policies and responding to national regulations as well [19].

In general, Gemma2 is an open-weight model family designed by Google DeepMind based on the Gemini model and

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

with parameter sizes up to 9B and 27B. Gemma2 is intended to strike a balance between performance efficiency and openness and exhibits superior performance in reasoning tasks, generation of code, and understanding language. It is even able to get close to commercial close models' capabilities [25].

Qwen2.5 is the newest line from Qwen with up to 72B parameters. It is capable of performing various tasks such as language comprehension, math, programming, and tool integration. The 14B version employed in this work is among the most developed open dense models and was trained using 18 trillion tokens and further improved using post-training methods such as supervised fine-tuning (SFT) and direct preference optimization (DPO). These methods enhance the capability of the model to process well-structured data and handle long-context data inputs and provide responses in accordance with human preference [26].

Mistral-Nemo-Instruct is founded upon the Mistral 7B design, a 7-billion parameter open source model with high efficiency. Features including grouped-query attention (GQA), sliding window attention (SWA), and rolling buffer cache enable this model to outperform and even surpass bigger models including LLaMA 2 (13B) in all evaluations on different benchmarks. Mistral's instruct version is particularly intended for accurate instruction following and allows for secure deployment by using adjustable prompt systems [27].

D. RAG System Implementation and Model Integration

Based on Figure 4, the process begins when the system receives a query from the user. This query is used to search the knowledge base, which consists of bank billing documents stored as vectors in a Vector Database (Knowledge Vector DB). In this implementation, the vector database is built using ChromaDB, which enables efficient semantic similarity-based retrieval. The application is developed using Flask as the backend framework to handle user requests and manage the flow between the retrieval component and the model processing. Meanwhile, the LLM is run locally using LM Studio, which supports loading open-weight models in GGUF format for inference.

After receiving the query, the system determines whether relevant knowledge has been successfully retrieved. If relevant information is found, the corresponding documents are retrieved and passed along with the query to the LLM to generate a response. If no relevant documents are found, the query is processed directly by the LLM without additional context. The final output of this process is the response generated by the LLM, either document-based (if relevant knowledge is found) or purely based on the model's understanding of the query. This RAG-based approach has been proven to enhance the accuracy and relevance of responses, especially for complex and domain-specific tasks, by providing access to contextual information from an external knowledge base [28].

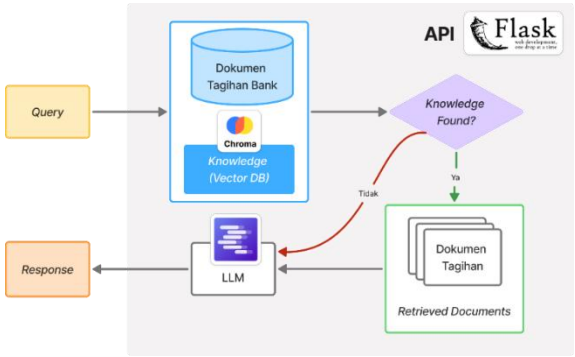

Fig. 4. Implementation Flow of RAG System in Answering Queries

E. Model Performance Evaluation

Model performance evaluation was carried out using two approaches, which are semantic similarity, LLM-as-a-judge, and human evaluation. Semantic similarity refers to measuring the closeness in meaning between the LLM-generated response and the reference answer (ground truth). This method leverages deep learning-based embeddings that represent sentences or documents as high-dimensional vectors, allowing comparisons between responses through mathematical measures such as cosine similarity [29]. Unlike traditional metrics based on n-grams or word overlap (such as BLEU or ROUGE), the semantic similarity approach captures semantic nuances and contextual alignment, making it capable of identifying whether two sentences have the same meaning even when phrased differently or paraphrased.

Previous studies have shown that this approach is highly effective for open-domain, knowledge-intensive, and retrieval-augmented generation (RAG) tasks, where there are often many possible correct answers expressed in different styles or sequences [29]. Additionally, the choice of encoder used to generate the embeddings plays a critical role. Larger and more complex encoders (such as MiniLM, Mpnet, DistilBERT, and XLM-RoBERTa) tend to produce richer semantic representations with higher precision, although they require more computational resources [30].

$$similiarity\ (A,\ B) = \cos(\theta) = \frac{A \times B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

In addition to the semantic similarity approach, this study also employs the LLM-as-a-Judge method to evaluate model performance. LLM-as-a-Judge is an approach that leverages the capabilities of Large Language Models (LLMs), such as GPT-4, to serve as automated evaluators that assess the quality, accuracy, and relevance of responses generated by other models. This method bridges the gap between human expert evaluations, which are often costly and inconsistent, and large-scale traditional automated metrics, enabling more efficient assessments that closely reflect human judgment [31]. In practice, LLM-as-a-Judge uses prompt templates and in-context learning examples to guide the evaluation process. This may include Likert scale scoring, yes/no judgments, pairwise answer comparisons, or best-answer selection. The choice of the evaluator model is also crucial. Although several fine-tuned

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

open-source judge models are available (such as PandaLM, JudgeLM, and Prometheus), studies have shown that GPT-4 remains the gold standard due to its high alignment with human evaluations, consistency across various evaluation schemes, and its ability to leverage advanced prompting techniques like chain-of-thought (CoT) reasoning [32].

Finally, to supplement these automated methods, human evaluation was implemented to ensure the quality of the model's responses. In this step, human reviewers carefully evaluated the LLM-generated answers for accuracy, clarity, and relevance to the context, ensuring that human judgment balanced out any potential biases from automated metrics. This approach is consistent with recent research [33], which emphasizes the importance of human validation in evaluating RAG system outputs, especially because automated metrics frequently overlook the nuanced aspects of answer quality across domains.

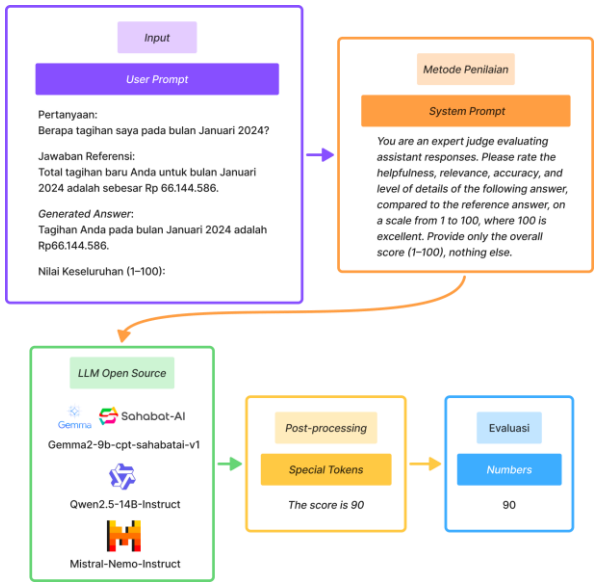

Fig. 5. Evaluation Pipeline LLM-as-a-Judge

During the experimental phase, user questions are received and key metadata is extracted, followed by embedding and information retrieval from the vector database. If relevant information is found, the LLM generates a response by combining context from the database. If no relevant context is retrieved, the LLM relies on its internal knowledge to answer the query. All responses from the three candidate models are then evaluated using the two predefined methods to gain a comprehensive understanding of each model's strengths, weaknesses, and implementation potential in supporting customer service in banking. Throughout the evaluation process, the models were executed with specific technical settings, which are temperature value of 0.4 was used to maintain response consistency and determinism, a maximum context length of 4096 tokens, and a batch size of 512 for evaluation. These settings were chosen to ensure both evaluation efficiency and the quality of the model outputs.

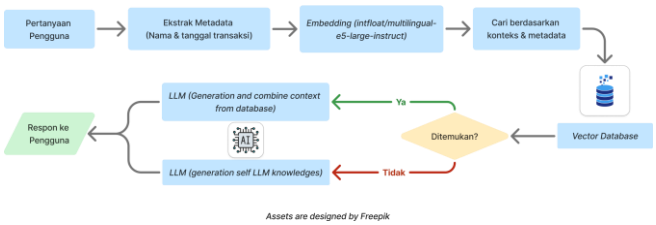

Fig. 6. Details of the Stages in the Experiment of Various Models Tested

III. Result And Discussion

A. Structure and Characteristics of Simulation Data

The data set simulated for this study is in CSV form and consists of 12,000 entries with every entry constituting one customer and one credit card billing statement. The data structure contains customer identifying data including full name, address, and customer number as well as bill data including billing date, due date, total amount due, minimum due amount, card category, and credit eligibility indicator. Each entry also carries a transactional record up to a maximum number of ten entries including transaction amount, date, merchant name, location, and transaction category. The data is further complemented by both original and masked account number, stamp duty amount, and various timestamps including final record date and subtotal period.

TABLE I. Structure of Credit Card Billing Simulation Dataset

Category	Column Name	Description
Customer Identity	*Indonesian Full Names, Customer Number*	Full name and unique identification number of the customer
	Company Name_x, Postal Code_x	Region/company name and postal code
	Address_1, Address_2	Full address of the customer
Billing Information	*Account Date_x, Due Date_x*	Print date and due date of the bill
	New Bill Amount_x, Minimum Payment_x	Total bill and minimum payment
	Card Type, Credit Quality_x	Card type and credit quality status
	Subtotal Amount, Materai	Summary of transaction values and stamp duty
	Nomor Rekening, Nomor Rekening Censor	Original card number and censored version
Transaction History	*Transaction Amount 1–10*	Amount of each transaction
	Transaction Date 1–10	Transaction date (DD-MM format)
	Merchant 1–10, Location 1–10	Merchant name and location
	CR 1–10	Transaction type (all "CR" for credit)
Supporting Date	*End Date 1–10*	Last date of recording for each transaction
	Subtotal Date 1–2, Subtotal End Date 1–2	Summary date and end of subtotal period

B. Embedding Representation Validation

Based on Figure 7, the distribution of cosine similarity scores between a query and all documents in the vector database

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

shows a range of values from 0.93 to close to 1.00, with a peak distribution around 0.95. This indicates that most documents have a high level of semantic similarity to the query, indicating that the embedding process has succeeded in representing the contents of the document into the vector space densely and consistently. The shape of the distribution curve, which resembles a normal distribution, further suggests that the retrieval system achieves a reasonable and stable similarity spread, enabling effective document selection based on semantic relevance.

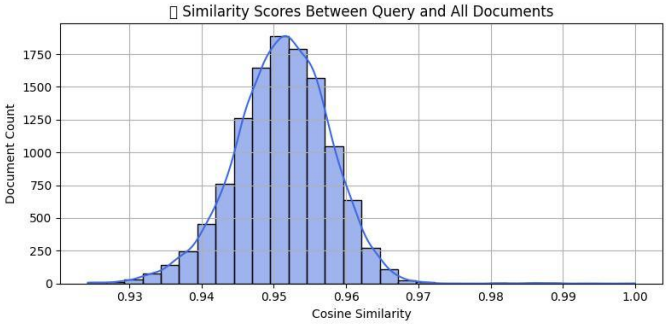

Fig. 7. Distribution of Cosine Similarity Scores between Query and All Documents

Furthermore, to complement this similarity analysis, the two-dimensional projection of the document embedding vectors using the UMAP (Uniform Manifold Approximation and Projection) algorithm is presented. Each point represents a single document in the compressed vector space. This pattern indicates that the embedding effectively captures the semantic diversity among documents. The presence of faint clusters in the central area suggests similarities in topic or structure between documents, while points located at the edges represent documents that are semantically more unique. Taken together, these insights validate the embedding representation's quality before it is leveraged in the retrieval and context injection processes within the RAG system.

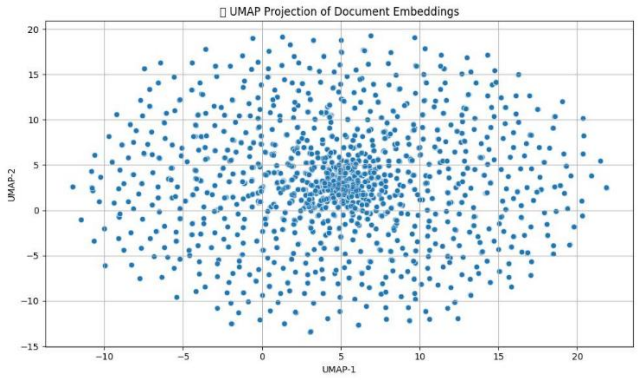

Fig. 8. UMAP Projection of Whole Document Embedding

C. *Experiment Prompt Preparation*

This study utilized a total of 15 experimental prompts, carefully chosen to reflect key categories of information that banking customers frequently inquire about. These include inquiries about monthly billing details, minimum payment amount, due date, account creation date, credit quality status, total transaction value, the presence of stamp duty, automatic payment status, and transaction details at specific locations (such as South Jakarta or Padang).

For the experiment, all questions were tested using a single user profile, namely Tgk. Drajat Handayani, S.Gz, who represents a typical banking customer seeking credit card bill information. For each prompt, a reference answer (ground truth) was prepared, serving as the correct and accurate response in accordance with standard information typically provided by banking systems. This ground truth served as the main benchmark for evaluating how well the candidate models Gemma2-9b-cpt-sahabatai-v1, Qwen2.5-14B-Instruct, and Mistral-Nemo-Instruct could generate relevant and contextually appropriate responses, as detailed in the following table listing the 15 questions along with their ground truth answers.

TABLE II. LIST OF QUESTION AND GROUND TRUTH FOR PROMPTING MODEL TESTING

Num	Question	Ground Truth
1	Berapa tagihan saya pada bulan Januari 2024?	Total tagihan baru Anda untuk bulan Januari 2024 adalah sebesar Rp 66.144.586.
2	Berapa pembayaran minimum yang harus dibayar pada bulan Januari 2024?	Pembayaran minimum yang harus dibayarkan pada bulan Januari adalah Rp 3.307.229.
3	Kapan tanggal jatuh tempo tagihan bulan Januari 2024?	Tanggal jatuh tempo tagihan untuk bulan Januari 2024 adalah 01 Februari 2024.
4	Kapan tanggal rekening dibuat untuk tagihan bulan Januari 2024?	Tanggal rekening dibuat adalah 16 Januari 2024.
5	Apa kualitas kredit dari pemilik kartu ini?	Kualitas kredit dari pemilik kartu ini adalah lancar.
6	Berapa total transaksi pada bulan Januari 2024?	Total nilai transaksi yang dilakukan selama bulan Januari 2024 adalah sebesar Rp 66.134.586.
7	Apakah ada bea materai dalam tagihan bulan Januari 2024?	Ya, terdapat bea materai sebesar Rp 10.000 dalam tagihan ini.
8	Apakah ada pembayaran otomatis pada tagihan bulan Januari 2024?	Ya, terdapat pembayaran otomatis sebesar Rp 66.134.586 pada tagihan ini.
9	Apakah ada transaksi di Jakarta Selatan pada bulan Januari 2024?	Ya, terdapat transaksi dengan PT HASSANAH di Jakarta Selatan sebesar Rp 892.000 pada bulan Januari 2024.
10	Apakah ada transaksi di Padang pada bulan Januari 2024?	Ya, terdapat transaksi dengan PT HANDAYANI PUTRA (PERSERO) TBK di Padang sebesar Rp 5.798.490 pada bulan Januari 2024.
11	Apa saja kategori pengeluaran utama Tgk. Drajat Handayani yang dibayarkan menggunakan kartu kredit selama Januari 2024, dan apa yang dapat disimpulkan dari kebiasaan tersebut?	Kategori pengeluaran utama selama bulan Januari 2024 mencakup berbagai kebutuhan rumah tangga dan jasa, tercermin dari beberapa transaksi berikut: CV Gunawan Adriansyah Mojokerto: Rp 27.470.620, PERUM NAMAGA

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

Num	Question	Ground Truth
		WIJAYA (PERSERO) TBK Payakumbuh: Rp 9.824.460, PERUM NURDIYANTI Denpasar: Rp 7.740.670, UD PRAYOGA IRAWAN CIREBON: Rp 7.754.680, PERUM SALAHUDIN Surabaya: Rp 5.831.810, PT HANDAYANI PUTRA (PERSERO) TBK Padang: Rp 5.798.490, PD SIHOTANG TBK Langsa: Rp 821.856, PT HASSANAH Jakarta Selatan: Rp 892.000. Total nilai transaksi adalah Rp 66.134.586, yang menunjukkan pola pengeluaran yang konsisten dan strategis.
12	Bandingkan distribusi lokasi transaksi (misalnya, Langsa, Denpasar, Surabaya, Cirebon, Mojokerto, Padang, Payakumbuh, Jakarta Selatan) untuk melihat apakah terdapat pola pengeluaran yang signifikan secara geografis pada bulan Januari 2024	Terdapat distribusi pengeluaran di beberapa kota besar dan menengah di Indonesia. Transaksi terbesar di Mojokerto (Rp 27.470.620) dan terkecil di Jakarta Selatan (Rp 892.000), menunjukkan pola pengeluaran yang luas dan beragam
13	Berapa sisa saldo tagihan pada kartu kredit setelah pembayaran otomatis pada tanggal 2 Januari 2024?	Saldo tagihan menjadi Rp 0 karena pembayaran otomatis pada tanggal 2 Januari 2024 sebesar Rp 66.134.586 sudah melunasi saldo tagihan sebelumnya
14	Sebutkan dua transaksi terbesar yang dilakukan oleh Tgk. Drajat Handayani selama bulan Januari 2024	Dua transaksi terbesar adalah dengan CV Gunawan Adriansyah Mojokerto sebesar Rp 27.470.620 dan PERUM NAMAGA WIJAYA (PERSERO) TBK Payakumbuh sebesar Rp 9.824.460
15	Berapa saldo tagihan yang tersisa setelah melakukan pembayaran minimum sebesar Rp 3.307.229 pada bulan Januari 2024?	Saldo tagihan tersisa adalah Rp 66.144.586 - Rp 3.307.229 = Rp 62.837.357.

The detailed classification of these 15 questions allows for a structured evaluation of model performance across different levels of complexity and customer service scenarios.

TABLE III. CATEGORIZATION OF EXPERIMENTAL PROMPTS BY TYPE, COMPLEXITY, AND DOMAIN

Question Number	Type of Answer	Complexity	Domain
1	Descriptive	Simple	Billing
2	Descriptive	Simple	Payment
3	Descriptive	Simple	Billing
4	Descriptive	Simple	Account
5	Descriptive	Simple	Account
6	Descriptive	Simple	Transaction
7	Boolean (Yes/No)	Simple	Billing
8	Boolean (Yes/No)	Simple	Payment
9	Boolean (Yes/No)	Simple	Transcation
10	Boolean (Yes/No)	Simple	Transcation
11	Descriptive + Analytical Summary	Complex	Spending Pattern Analysis
12	Comparative + Descriptive Analysis	Complex	Spending Pattern Analysis
13	Descriptive	Moderate	Billing
14	Descriptive	Moderate	Transaction
15	Descriptive	Moderate	Billing

D. Model Computational Performance Evaluation

The average response time curve for all three models is shown in figure 9 and includes gemma2-9b-cpt-sahabataiv1-instruct with 9 billion parameters, mistral-nemo-instruct with 12 billion parameters, and qwen2.5-14b-instruct with 14 billion parameters. The figure also adds an overall average line of about 53.99 seconds as a general reference among all the models.

Interestingly, Gemma2, despite being the smallest in parameter count, achieved the lowest average response time, around 41 seconds, which is significantly below the overall average. Mistral Nemo also performed close to the average line, with an average response time slightly below 50 seconds. In contrast, Qwen2.5, the largest model, displayed the longest average response time, exceeding 70 seconds. This pattern shows that larger models like Qwen2.5 typically have longer processing times, likely due to their higher computational demands, while smaller models such as Gemma2 can achieve quicker response times despite having fewer parameters. However, the data also shows that parameter count is not the only factor, Gemma2's performance suggests that architecture and implementation play a significant role in responsiveness, with its average well below the overall average despite its smaller size.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

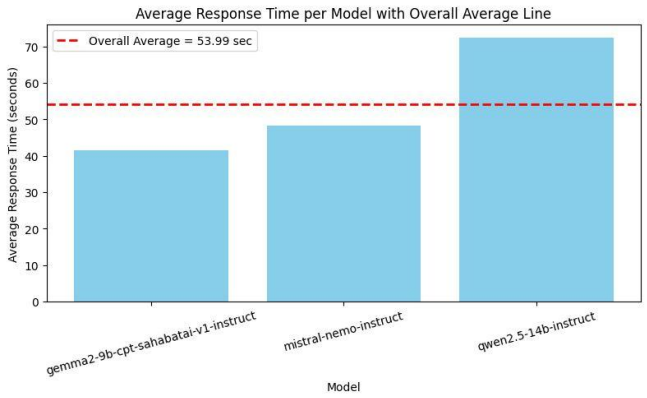

Fig. 9. Average Response Time per Model Used in the Study

Based on Table IV, it is clear that Qwen 2.5 consistently shows longer response time than the other two models on almost all queries. Meanwhile, Gemma2-Sahabat-AI is sometimes even faster than Mistral, even with a smaller number of parameters. This strengthens the finding that Gemma2-Sahabat-AI has excellent efficiency and can be an attractive alternative for applications that require fast response with lighter computational load.

TABLE IV. COMPARATIVE MODEL TESTING FOR EACH MODEL BASED ON RESPONSE TIME

Question Number	Response Time (seconds)		
	Gemma2 Sahabat AI	Qwen2.5	Mistral
1	95,07	28,70	54,93
2	16,64	23,14	15,26
3	21,15	25,19	16,60
4	25,78	28,25	17,74
5	79,30	33,67	27,97
6	33,72	48,16	35,58
7	44,00	43,10	34,12
8	43,27	52,75	68,23
9	82,48	58,82	116,22
10	31,90	55,80	48,77
11	34,91	142,34	61,31
12	40,50	125,17	45,30
13	25,48	152,29	65,75
14	23,63	117,93	59,28
15	24,32	149,02	55,79
Average Score	41,48	72,29	48,19

Figure 10 show how three language models, namely Gemma2 Sahabat AI, Qwen2.5, and Mistral Nemo, use system resources over time, focusing on RAM (blue line) and GPU (red line) usage. Gemma2-Sahabat-AI shows a sharp rise in GPU usage that quickly levels off. This means that once it stabilizes, there will be a steady demand for GPUs. On the other hand, its RAM usage is less predictable, with big changes showing that it is using dynamic memory management, which is probably because it is caching data or storing temporary data during inference.

On the other hand, Qwen2.5 shows a steady rise in RAM usage, which means that the session is taking up more memory as it goes on. However, its GPU usage is significantly more unpredictable, with frequent rises and drops. This pattern suggests that computational demands fluctuate frequently, most likely due to the complexity of the queries being processed. Mistral Nemo has some similarities in that its GPU usage varies greatly, with peaks and troughs indicating intensive computational bursts. Meanwhile, RAM usage rises initially and then falls slightly, indicating that Mistral Nemo stabilizes after the initial acceleration period.

Overall, these patterns highlight some noticeable differences. Gemma2's GPU usage smoothes out, but RAM remains jumpy; Qwen2.5's RAM demands continue to rise alongside volatile GPU activity; and Mistral Nemo's resource usage remains consistently turbulent for both GPU and RAM.

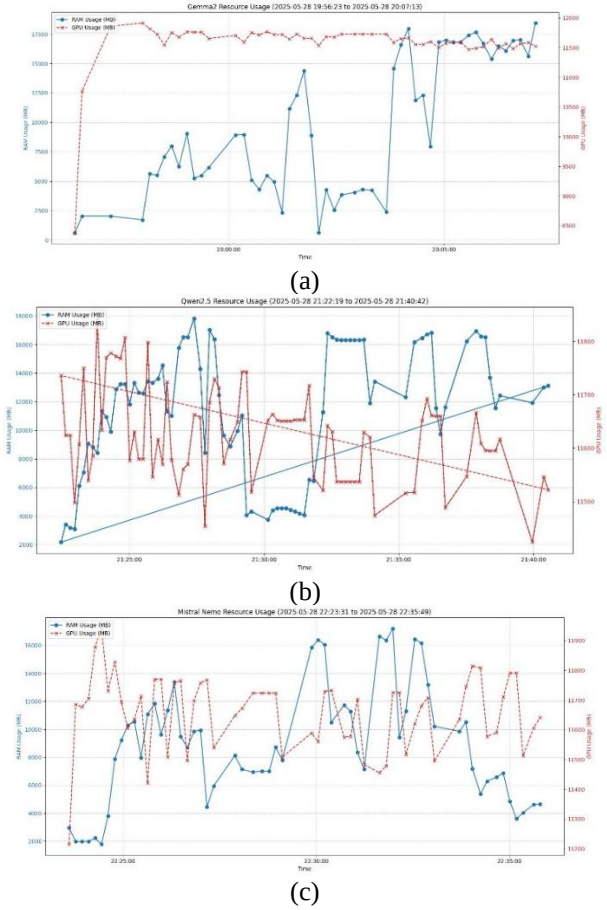

Fig. 10. Resource Utilization of (a) Gemma2 Sahabat AI, (b) Qwen2.5, and (c) Mistral Nemo Over Time

E. Semantic Similiarity Evaluation

The comparison results of semantic similarity scores highlight the performance of the three evaluated models: Gemma2 Sahabat AI, Qwen2.5, and Mistral. These results are based on fifteen experimental questions designed to reflect real-world banking customer service scenarios. Each score is measured on a scale from 0 to 1, where values closer to 1 indicate a high degree of semantic similarity between the model's response and the reference answer (ground truth),

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

regardless of differences in word arrangement or phrasing.

The analysis of Table V shows that Gemma2 Sahabat AI consistently achieved the highest average score of 0.9627, closely followed by Mistral at 0.9614, while Qwen2.5 recorded a slightly lower average of 0.9284. For most straightforward questions, such as those about due dates, transaction summaries, and payment amounts, all three models demonstrated high semantic similarity, with Gemma2 and Mistral consistently scoring above 0.98. However, in more challenging queries involving complex details, like automatic payment status or specific location-based transactions, Qwen2.5 showed slightly lower semantic alignment, although still remaining above 0.91.

TABLE V. SEMANTIC SIMILARITY SCORE COMPARATIVE MODEL TESTING FOR EACH MODEL

Question Number	Semantic Similiarity Score		
	Gemma2 Sahabat AI	Qwen2.5	Mistral
1	0,9877	0,9431	0,9877
2	0,9751	0,9215	0,9741
3	0,9981	0,9442	0,9936
4	0,9976	0,9355	0,9965
5	0,9825	0,8974	0,9834
6	0,9932	0,9189	0,9923
7	0,9296	0,9358	0,9517
8	0,9598	0,9138	0,9464
9	0,9887	0,9507	0,9561
10	0,9889	0,9604	0,9823
11	0,9291	0,9332	0,9434
12	0,8893	0,8944	0,9270
13	0,9454	0,9347	0,9233
14	0,9254	0,9072	0,9165
15	0,9506	0,9348	0,9463
Average Score	0,9627	0,9284	0,9614

F. LLM-as-a-Judge Evaluation

The LLM-as-a-Judge results demonstrate a comparative evaluation of model outputs scored by two primary evaluators, which are OpenAI GPT-4o-mini and Gemini 2.0 Flash, using a 0-to-100 scale. This approach assesses the quality, relevance, and factual accuracy of each model's answer compared to the ground truth. Based on the updated data, Qwen2.5 recorded the highest average score of 92.2 under the GPT-4o-mini evaluations, with Mistral and Gemma2 closely following at 89.33 and 90.33 respectively. Meanwhile, Gemini 2.0 Flash assessments placed Gemma2 slightly ahead at an average of 88.4, while Qwen2.5 and Mistral scored 87.53 and 87.33 respectively.

Despite some variation in individual question scores, for instance, in question 13, all models performed poorly (scoring between 10 and 30) under both evaluators, the general trend across the board is consistent high performance in simple factual questions (1-10) and a noticeable decline in more complex numerical and reasoning questions (11-13). Notably, Qwen2.5 generally received the highest average rating from OpenAI, suggesting a slight edge in overall output quality for typical tasks.

Model performance was uniformly strong on simple factual questions 1-10, with clear, concise answers across all systems. However, moderate-complexity numeric questions, especially in question 13 revealed inconsistencies, while Gemma2 provided a direct numeric answer, Qwen2.5 noted a residual fee, and Mistral detected a negative balance discrepancy, suggesting models diverge when precision arithmetic is required. LLMs still struggle with arithmetic because their vocabularies were built for words, not numbers, a multi-digit figure is often split into several rare sub-tokens, so the model loses place-value information before it even attempts the calculation. As a result, they show good fluency with words but noticeably poor, inconsistent performance on anything beyond very small or well-seen numbers [34].

Complex reasoning questions 11 and 12 also highlighted differences. Qwen2.5 and Mistral offered structured, bullet-point responses, while Gemma2 stuck to concise summaries. Despite these differences, all models consistently provided factual transaction details, with interpretation and conclusion-drawing being the most variable elements. To illustrate these differences, representative questions and summarized generated answers are provided in Appendix B (Table IX).

TABLE VI. LLM-AS-A-JUDGE SCORE COMPARATIVE MODEL TESTING FOR EACH MODEL

Question Number	LLM-as-a-Judge Score					
	OpenAI 4o-mini			Gemini 2.0 Flash		
	Gemma2 Sahabat AI	Qwen 2.5	Mistral	Gemma2 Sahabat AI	Qwen 2.5	Mistral
1	100	90	100	99	100	100
2	100	95	100	100	100	100
3	100	95	100	100	95	100
4	100	100	100	100	98	100
5	100	95	100	100	100	100
6	100	85	100	100	95	98
7	90	95	100	98	100	100
8	100	85	85	100	98	100
9	95	90	90	100	99	90
10	95	95	95	100	100	100
11	75	85	75	90	95	40
12	60	85	85	90	45	95
13	30	30	30	10	25	10
14	95	85	95	98	65	98
15	100	100	100	98	98	95
Average Score	89,33	90,33	87,33	92,20	87,53	88,4

G. Human Evaluation of LLM Response

The human evaluation conducted in this study focused on assessing the answers generated by the models themselves. For simple factual questions (questions 1 to 10), human evaluators gave consistently high scores (mostly 100), indicating that the models produced accurate and clear answers. However, for more complex questions (11 and 12), human evaluators gave lower scores, especially to Mistral (50 for question 11 and 70 for question 12). These questions required deeper reasoning,

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

such as drawing inferences about spending patterns and analyzing transaction locations. Human evaluators valued not just linguistic similarity, but also the factual correctness and logical coherence of the answers. This led to lower scores for models that failed to provide complete or well-reasoned explanations. For moderate complexity numeric questions (13 to 15), the human evaluations showed a mix of results: question 13 saw all models scoring zero due to clear numerical errors, while questions 14 and 15 had near-perfect scores for most models, reflecting their ability to handle more straightforward numeric queries.

When comparing these human scores to automated metrics, important patterns emerge. Semantic similarity scores tended to be consistently high across all questions, including the complex ones (11 and 12). This suggests that while the model-generated answers looked similar to the ground-truth answers in wording and phrasing, this did not always reflect true factual accuracy or strong reasoning. The LLM-as-a-Judge scores showed variable agreement with human evaluations. For question 11, for example, Gemini 2.0 Flash scored Mistral's answer as very low (40), while OpenAI 4o-mini rated it at 95, illustrating that LLM judges themselves can diverge in their perceptions of complex answers. In moderate complexity numeric questions, there was more agreement across human and automated metrics, especially for question 13, where all methods flagged the models' answers as poor.

These findings highlight that while human evaluations provide a deep understanding of factual correctness and reasoning, automated metrics, particularly semantic similarity, focus more on surface-level linguistic overlap and may miss underlying flaws. LLM-as-a-Judge metrics serve as a helpful bridge but can still vary based on the specific LLM model used as a judge.

TABLE VII. HUMAN VALIDATION SCORE COMPARATIVE MODEL TESTING FOR EACH MODEL

Question Number	Human Validation		
	Gemma2 Sahabat AI	Qwen2.5	Mistral
1	100	100	100
2	100	100	100
3	100	100	100
4	100	100	100
5	100	100	100
6	100	100	100
7	100	100	100
8	100	100	100
9	100	100	100
10	100	100	100
11	85	70	50
12	50	80	70
13	0	0	0
14	100	80	100
15	100	100	100
Average Score	89,00	88,67	88,00

IV. CONCLUSION

This study demonstrates that the end-to-end RAG pipeline developed for credit-card billing queries is technically robust, yet it also exposes some clear boundaries in current large-language-model capabilities. The 12 000-row synthetic data set, complete with identity, billing transaction and metadata fields, proved rich enough to mimic real statements. The multilingual-e5-large embeddings represented that data cleanly, which are cosine-similarity scores clustered tightly around 0.95, and a UMAP projection revealed well-separated yet coherent groups of documents. These findings confirm that the retriever is passing a well-behaved vector space to the generator.,

Across the 15 prompts, which ranged from simple factual look-ups to arithmetic and higher-level spending-pattern analysis, the three models displayed markedly different speed profiles. Gemma2 Sahabat AI, despite its nine-billion-parameter size, responded in roughly forty-one seconds on average, well below the overall mean. In contrast, the fourteen-billion-parameter Qwen 2.5 routinely required more than seventy seconds. Resource-utilisation traces reinforce this picture, Gemma2's GPU demand stabilised quickly, whereas Qwen 2.5 and Mistral Nemo showed volatile GPU spikes and steadily rising RAM footprints.

Semantic-similarity scores paint a uniformly positive picture, with Gemma2 edging out Mistral (0.963 vs 0.961) and Qwen 2.5 trailing slightly (0.928). Yet this surface-level metric masks deeper weaknesses that emerge under closer scrutiny. When answers were judged by LLMs themselves, OpenAI GPT-4o-mini tended to prefer Qwen 2.5, whereas Gemini 2.0 Flash gave a slight nod to Gemma2, illustrating that even automated judges diverge on what constitutes the best answer. Human evaluators, however, told a more nuanced story: they awarded perfect scores for every model on the ten straightforward factual questions, but penalised all systems heavily when numerical precision or multi-step reasoning was required. In particular, Question 13, which asked for a residual balance after an automatic payment, yielded a score of zero across the board because each model performed the arithmetic incorrectly. This mirrors broader findings that sub-tokenised numbers erode place-value information, leaving LLMs fluent in prose but unreliable in calculation.

Taken together, the study suggests that Gemma2 offers the most attractive balance of latency and semantic accuracy for routine customer-service dialogues, while Mistral remains a close second with only a modest speed penalty. Qwen 2.5 delivers strong language quality according to one judge model, yet its computational overhead may limit its practicality. Crucially, none of the three systems can be trusted to handle financial arithmetic or complex analytic reasoning without supplementary safeguards. Moving toward production, organisations should incorporate a deterministic calculator module for numeric tasks, consider fine-tuning on domain-specific financial tokens, and maintain human-in-the-loop checks for nuanced or high-risk queries. With these additions, a fast, accurate conversational assistant for credit-card billing support is within reach as long as the arithmetic gap is explicitly addressed.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

Despite these promising results, several limitations remain, such as the reliance on synthetic data that lacks real-world noise and the testing of only a single user profile, which limits insights into how the system might handle diverse customer needs. The small number of questions and absence of real conversational variability further restrict generalizability. LLM-as-a-Judge evaluations may be influenced by biases and do not fully account for Indonesian language or cultural nuances. Future research should include testing with real users, expanding the question pool, integrating more evaluation metrics like factual consistency and latency, and incorporating security, auditing, and transparent error handling. Developing a working prototype and embedding numeric modules to address arithmetic weaknesses will be key for transitioning this system from simulation to a reliable customer-facing tool.

APPENDIX

A. Experimental Setup

TABLE VIII. EXPERIMENTAL HARDWARE CONFIGURATION

Component	Spesification
Processor	AMD Ryzen 9 5900X, 12-core, 3.70 GHz
RAM	32 GB DDR4, 2133 MHz
Graphics Card	NVIDIA GeForce RTX 3080 Ti, 12 GB VRAM

B. Representative Answer Samples

TABLE IX. REPRESENTATIVE ANSWER SAMPLES FOR QUESTIONS 11 - 13

Question Number	Model	Summarized Generated Answer
11	Gemma2 Sahabat AI	Major expenses in January 2024 were property-related (e.g., Perum Nurdiyanti, Perum Salahudin, PT Handayani Putra), suggesting involvement in business/investment. This habit shows that Tgk. Drajat Handayani has an active lifestyle and is involved in various economic activities.
	Qwen2.5	Breakdown of main expenses by company type, with highest payments to PT/Persero and CV/UD businesses, indicating business-related transactions. The credit card spending was primarily focused on small and medium enterprises (SMEs), followed by state-affiliated operational entities, then large corporations or public companies (PT/Persero), and lastly on minor general purchases.
	Mistral	The credit card transactions in 16 January 2024 primarily reflect large operational or project-related expenses, with significant payments made to companies, other business entities, and small merchants. While most spending was concentrated in high-value business-related transactions, there were also a few minor routine purchases.
12	Gemma2 Sahabat AI	The credit card transactions in January 2024 were geographically distributed across various cities in Indonesia (Langsa, Denpasar, Surabaya, Cirebon, Mojokerto, Padang, Payakumbuh, dan Jakarta Selatan), with no dominant spending pattern in any specific region.
	Qwen2.5	The credit card transactions in January 2024 were spread across multiple regions in Indonesia, with notable spending concentrated in East Java, particularly Mojokerto. While some high-value transactions also occurred in Bali and West Sumatra, no strong geographic spending pattern was identified overall.
	Mistral	The January 2024 credit card transactions were primarily concentrated in Mojokerto, Denpasar, and Surabaya, indicating these as the main spending locations, with additional smaller transactions occurring in other cities across Indonesia.
13	Gemma2 Sahabat AI	Remaining balance after auto-payment on Jan 2, 2024 is Rp 66,134,586.
	Qwen2.5	After auto-payment of Rp 66,134,586, only Rp 10,000 remains as Bea Meterai.
	Mistral	The credit card statement shows a negative remaining balance of Rp 9,999 after an automatic payment on January 2, 2024, suggesting a possible calculation error or unrecorded additional payment, and it is recommended to verify the billing details with the bank.

REFERENCES

[1] N. Izzah and M. Z. Rachmawan, "Penerapan Strategi Cost Efficiency (Efisiensi Biaya) Pada PT Bank Muamalat Indonesia, Tbk. Tahun 2017," *Abiwara: Jurnal Vokasi Administrasi Bisnis*, vol. 1, no. 1, pp. 37–44, 2019, doi: https://doi.org/10.31334/abiwara.v1i1.500.

[2] N. Lekhawichit, C. Chavaha, K. Chienwattanasook, and K. Jermsittiparsert, "THE IMPACT OF SERVICE QUALITY ON THE CUSTOMER SATISFACTION: MEDIATING ROLE OF WAITING TIME," 2021. Accessed: May 03, 2025. [Online]. Available: http://www.psychologyandeducation.net/pae/index.php/pae/article/view/2552/2228

[3] A. Z. Desta and T. H. Belete, "The Influence of Waiting Lines Management on Customer Satisfaction in Commercial Bank of Ethiopia," *Institutions and Risks*, vol. 3, no. 3, pp. 5–12, 2019, doi: 10.21272/fmir.3(3).

[4] J.-H. Yang and A.-S. Park, "Analysis on Relationship between Waiting Time and Customer Satisfaction of General Hospitals," 2021. Accessed: May 03, 2025. [Online]. Available: https://www.nveo.org/index.php/journal/article/view/248/223

[5] A. Subyantoro, D. Tri Mardiana, M. S. Zulfikar, and M. Hasan, *PELATIHAN DAN PENGEMBANGAN SUMBER DAYA MANUSIA*. ZAHIR PUBLISHING, 2022. Accessed: May 03, 2025. [Online]. Available: http://eprints.upnyk.ac.id/34931/1/Buku%20Pengembangan%20Sumber%20Daya%20Manusia.pdf

[6] J. Wirtz and V. Zeithaml, "Cost-effective service excellence," *J Acad Mark Sci*, vol. 46, no. 1, pp. 59–80, Jan. 2018, doi: 10.1007/s11747-017-0560-7.

[7] F. O. Edeh, N. M. Zayed, V. Nitsenko, O. Brezhnieva-Yermolenko, J. Negovska, and M. Shtan, "Predicting Innovation Capability through Knowledge Management in the Banking Sector," *Journal of Risk and Financial Management*, vol. 15, no. 7, Jul. 2022, doi: 10.3390/jrfm15070312.

[8] K. Bahl, R. Kiran, and A. Sharma, "Evaluating the effectiveness of training of managerial and non-managerial bank employees using Kirkpatrick's model for evaluation of training," *Humanit Soc Sci Commun*, vol. 11, no. 1, Dec. 2024, doi: 10.1057/s41599-024-02973-y.

[9] M. Gumede, "THE IMPACT OF TRAINING AND DEVELOPMENT ON EMPLOYEE PERFORMANCE: A CASE STUDY OF CAPITEC BANK IN DURBAN," Durban University of Technology, 2021. Accessed: May 03, 2025. [Online]. Available:

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025

https://openscholar.dut.ac.za/server/api/core/bitstreams/72d4bd9f-4019-473d-b04b-aae4f34ec578/content
[10] A. Vilard *et al.*, "The Effects of Training and Development on Employees Performance: The Case of the National Financial Credit Bank (NFCB) of the Centre Region of Cameroon," *International Journal of Science and Business*, vol. 4, no. 6, pp. 88–106, 2020, doi: 10.5281/zenodo.3897174.
[11] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *ArXiv*, Dec. 2023, [Online]. Available: http://arxiv.org/abs/2312.10997
[12] S. Setty, H. Thakkar, A. Lee, E. Chung, and N. Vidra, "Improving Retrieval for RAG based Question Answering Models on Financial Documents," *ArXiv*, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2404.07221
[13] Z. Xu *et al.*, "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*, Association for Computing Machinery, Inc, Jul. 2024, pp. 2905–2909. doi: 10.1145/3626772.3661370.
[14] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," *ArXiv*, Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.08909
[15] V. Karpukhin *et al.*, "Dense Passage Retrieval for Open-Domain Question Answering," *ArXiv*, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.04906
[16] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *ArXiv*, May 2020, [Online]. Available: http://arxiv.org/abs/2005.11401
[17] S. Kim, H. Song, H. Seo, and H. Kim, "Optimizing Retrieval Strategies for Financial Question Answering Documents in Retrieval-Augmented Generation Systems," *ArXiv*, Mar. 2025, [Online]. Available: http://arxiv.org/abs/2503.15191
[18] C. Choi *et al.*, "FinDER: Financial Dataset for Question Answering and Evaluating Retrieval-Augmented Generation," *ArXiv*, Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.15800
[19] D. Staegemann, C. Haertel, C. Daase, M. Pohl, M. Abdallah, and K. Turowski, "A Review on Large Language Models and Generative AI in Banking," in *Proceedings of the 7th International Conference on Finance, Economics, Management and IT Business*, SCITEPRESS - Science and Technology Publications, 2025, pp. 267–278. doi: 10.5220/0013472600003956.
[20] G. Olaoye and H. Jonathan, "EasyChair Preprint The Evolving Role of Large Language Models (LLMs) in Banking," 2024.
[21] K. A. Maspul and N. K. Putri, "Will Big Data and AI Redefine Indonesia's Financial Future?," *Jurnal Bisnis dan Komunikasi Digital*, vol. 2, no. 2, p. 21, Feb. 2025, doi: 10.47134/jbkd.v2i2.3739.
[22] M. Nadzirin Anshari Nur and G. K. Kassymova, "The Potential Misuse of Artificial Intelligence Technology Systems in Banking Fraud," *Universitas Diponegoro*, vol. 21, no. 1, p. 17, Feb. 2025, Accessed: May 28, 2025. [Online]. Available: https://www.researchgate.net/publication/390122387_The_Potential_Misuse_of_Artificial_Intelligence_Technology_Systems_in_Banking_Fraud
[23] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 Text Embeddings: A Technical Report," *ArXiv*, Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.05672
[24] T. Jiang *et al.*, "E5-V: Universal Embeddings with Multimodal Large Language Models," *ArXiv*, Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.12580
[25] M. Riviere *et al.*, "Gemma 2: Improving Open Language Models at a Practical Size," *ArXiv*, Jul. 2024, [Online]. Available: http://arxiv.org/abs/2408.00118
[26] A. Yang *et al.*, "Qwen2.5 Technical Report," *ArXiv*, Dec. 2024, [Online]. Available: http://arxiv.org/abs/2412.15115
[27] A. Q. Jiang *et al.*, "Mistral 7B," *ArXiv*, Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.06825
[28] M. A. Oumano and S. M. Pickett, "Comparison of Large Language Models' Performance on 600 Nuclear Medicine Technology Board Examination–Style Questions," *J Nucl Med Technol*, vol. 00, p. jnmt.124.269335, May 2025, doi: 10.2967/JNMT.124.269335.
[29] A. Mahboub, M. E. Za'ter, B. Al-Rfooh, Y. Estaitia, A. Jaljuli, and A. Hakouz, "Evaluation of Semantic Search and its Role in Retrieved-Augmented-Generation (RAG) for Arabic Language," *ArXiv*, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.18350
[30] D. Chandrasekaran and V. Mago, "Evolution of Semantic Similarity -- A Survey," *ArXiv*, Apr. 2020, doi: 10.1145/3440755.
[31] J. Gu *et al.*, "A Survey on LLM-as-a-Judge," *ArXiv*, Nov. 2024, [Online]. Available: http://arxiv.org/abs/2411.15594
[32] H. Huang *et al.*, "An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4," *ArXiv*, Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.02839
[33] E. Oro, F. M. Granata, A. Lanza, A. Bachir, L. De Grandis, and M. Ruffolo, "Evaluating Retrieval-Augmented Generation for Question Answering with Large Language Models," in *4th National Conference on Artificial Intelligence*, CINI, May 2024, pp. 1–6. Accessed: May 30, 2025. [Online]. Available: https://ceur-ws.org/Vol-3762/495.pdf
[34] J. Ahn, R. Verma, R. Lou, D. Liu, R. Zhang, and W. Yin, "Large Language Models for Mathematical Reasoning: Progresses and Challenges," *ArXiv*, vol. 1, p. 114, Jan. 2024, [Online]. Available: http://arxiv.org/abs/2402.00157

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i3.2383, Copyright ©2025
Submitted : Mey 19, 2025, Revised : Mey 27, 2025, Accepted : June 5, 2025, Published : July 28, 2025