A Hybrid Generative and Vectorization Approach for Retrieval Augmented Generation in Wholesale Stores

Wildan Suharso1,a), Bahrul Ulum1,b), Briansyah Setio Wiyono1,c), Ahmad Fauzi2,d), Hariyady Hariyady3,e), Hylmun Izhar4,f), Didih Rizki Chandranegara5,g),

1Faculty of Engineering, Universitas Muhammadiyah Malang, Malang, Indonesia.

2National Research and Innovation Agency, Jakarta Pusat, Indonesia.

3Faculty of Computing and Informatics, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Malaysia.

4ISDB Institute, Jeddah, Kingdom of Saudi Arabia.

5Faculty of Computer Science, Electronics and Telecommunication, AGH University of Science and Technology, Krakow, Poland.

a) Corresponding author: wsuharso@umm.ac.id

b) bahrululum383@webmail.umm.ac.id

c) brian@umm.ac.id

d) ahma068@brin.go.id

e) hariyady\_di21@iluv.ums.edu.my

f) hizhar@isdb.org

g) didihrizki@agh.edu.pl

**Abstract.** The rapid advancement of digital technology necessitates the retail sector, particularly wholesale stores, to adopt more sophisticated customer service solutions. Conventional FAQ systems often fall short in answering specific and contextual customer queries, creating a gap in customer interaction. To address this issue, our research introduces an innovative chatbot leveraging the Retrieval Augmented Generation (RAG) approach. Our method integrates two key components: vectorization using the BAAI/bge-small-en-v1.5 model for accurate information retrieval, and the Mistral-7B-Instruct-v0.3 generative model for constructing relevant responses. The system was tested on a wholesale store product catalog containing 219 items and a series of test questions. Evaluation results demonstrate the effectiveness of this approach, with information retrieval accuracy reaching 80%, and precision, recall, and F1-scores of 87.5% and 0.875, respectively. These findings confirm that RAG significantly enhances the relevance and accuracy of chatbot responses compared to static FAQ systems. Despite these promising results, the system has a limitation: it is not yet integrated with the inventory and point-of-sale systems, which prevents real-time stock data updates. This study paves the way for further development, including integration with a store's operational systems and the exploration of more advanced embedding and generative models to create more responsive and informative customer service in the future.

# Introduction

The rapid advancement of digital technology has created an imperative for the retail sector, particularly wholesale stores, to adopt more sophisticated customer service solutions. In this fast-paced environment, customers demand interactive, responsive, and accurate service. Service quality is a crucial factor that directly impacts customer satisfaction and business sustainability [1]. To meet these demands, many businesses are moving from manual interactions to automated solutions like chatbots, which can serve customers 24/7 and efficiently handle common queries [2, 3].

However, implementing chatbots for customer service, especially in the wholesale sector, presents significant challenges. Traditional systems relying on static FAQ databases often prove ineffective [4]. While they can answer standard questions, they fail to handle specific, contextual queries or those requiring a deep understanding of complex product data, such as inventory details or technical specifications [5]. This limitation can create a gap in customer service and lead to dissatisfaction. With recent advancements in Natural Language Processing (NLP), the need for smarter, more adaptive chatbots has become increasingly pressing. NLP enables chatbots to better understand conversation context, identify sentiment, and provide more relevant, human-like responses [6, 7].

The emergence of advanced generative models has ushered in a new era for chatbot development. These models can produce coherent, creative, and human-like text, far surpassing the capabilities of rule-based systems [8]. However, generative models are prone to "hallucinations"—the tendency to produce inaccurate or fabricated information. This is a serious issue in environments where factual accuracy is paramount, such as wholesale stores where product and stock details must be precise.

To overcome these weaknesses, the Retrieval-Augmented Generation (RAG) approach has emerged as a promising solution [9, 10]. RAG combines the power of a generative model with a process of retrieving information from a reliable external source. It works by first fetching relevant data from a knowledge base and then using that data as context to guide the generative model in forming a response. This ensures that the generated answers are not only fluent but also grounded in factual information. In this framework, vectorization plays a critical role. Vectorization is the process of converting data, such as text, into a dense numerical representation. Models are highly effective at producing high-quality vector representations that capture the semantic meaning of text [11, 12]. These representations enable the system to perform a far more accurate and relevant information search than traditional keyword-based methods.

This research aims to effectively integrate vectorization and generative models within a RAG framework to build an accurate and reliable wholesale store chatbot. By utilizing advanced embedding models and generative models, we seek to create a system capable of providing more precise and contextual answers than conventional FAQ systems. This case study, using a wholesale store product catalog, will demonstrate how this hybrid approach can significantly improve customer service quality and ensure the reliability of information. Thus, this study contributes to the development of AI solutions that are not only intelligent in their interactions but also trustworthy in their factual delivery, a crucial prerequisite for modern business operations.

# research methods

Based on the study objectives, the methodology is divided into four main, interconnected stages. The flowchart below illustrates the entire workflow, from initial preparation to the final evaluation of the developed system.

Initial Preparation

Data Preprocessing, Indexing and Vectorization



System Development and Generative Modeling



System Testing and Evaluation

**Figure 1.** Stages of Research

The initial stage of this research focuses on the fundamental preparation for building an effective wholesale store chatbot. We began with an in-depth needs analysis, which involved discussions with wholesale store management and customer representatives. This process identified the most frequently asked questions (FAQs) and specific needs regarding product information, pricing, and stock availability. The goal was to ensure the developed chatbot could accurately address the gaps found in conventional customer service systems. Following the needs analysis, we proceeded to data collection. The primary data used was the wholesale store’s product catalog, including product names, descriptions, prices, and other relevant attributes. This data serves as the knowledge base for the system. We also gathered a set of test questions representing various customer interaction scenarios, from simple pricing queries to more complex product comparisons. This process was crucial to ensure the collected data was rich and representative, forming a solid foundation for the chatbot's accuracy and relevance in subsequent stages.

After data collection, the next step was data preprocessing. In this stage, the product catalog data was cleaned to remove irrelevant information, typos, and inconsistent formatting. This step ensured the data used in the subsequent processes was of high quality. The cleaned data was then broken down into smaller, more manageable units of information, or "chunks." These information units were subsequently converted into a numerical representation through vectorization using the BAAI/bge-small-en-v1.5 embedding model. This model was chosen for its ability to generate dense vectors that capture the semantic meaning of text, which is essential for accurate information retrieval. These vectors were then indexed and stored in a vector database, such as ChromaDB, to allow for fast and efficient searching. This indexing process is crucial as it creates the external "memory" for the generative model, where the relevance between a customer's question and product data can be found quickly and accurately.

At third phase, the Retrieval Augmented Generation (RAG) architecture was built by integrating the prepared components. First, the system was developed to receive input from a customer. This input is then vectorized using the BAAI/bge-small-en-v1.5 embedding model. The question's vector is then used to find the most relevant data from the previously indexed vector database. Once the data is retrieved, this relevant information, along with the customer’s original question, is compiled into a comprehensive prompt. This prompt is then passed to the generative model, Mistral-7B-Instruct-v0.3. This model's task is to craft a fluent, informative, and relevant response based on the context provided by the retrieved data. The selection of this model was based on its high performance in generating natural text from specific instructions. This process ensures that the chatbot's answers are not solely the product of the generative model but are also grounded in valid facts from the product catalog, thereby reducing the risk of "hallucinations" and increasing accuracy.

The final phase of this study is system testing and evaluation. We used a series of pre-prepared test questions to measure the chatbot's performance. The evaluation metrics included information retrieval accuracy, precision, recall, and F1-score to assess how effectively the system finds relevant data. Additionally, we conducted a qualitative evaluation to judge the relevance and quality of the generative model's responses. This testing aimed to compare the performance of our RAG-based chatbot against conventional FAQ systems. The results will provide a comprehensive overview of the developed system's strengths and limitations, serving as a basis for future development recommendations. This analysis will validate the hypothesis that integrating vectorization and a generative model can significantly enhance the accuracy and relevance of chatbot responses, making it a more reliable solution for customer service in wholesale stores.

# results and discussion

## Initial Preparation

The foundational phase of this study involved a comprehensive system requirements analysis for the chatbot, a critical step aimed at defining the essential functional and non-functional parameters the system must satisfy. This analysis was meticulously conducted through direct engagement, including structured interviews and in-depth discussions with key stakeholders, specifically the wholesale store's management, owners, and customer service staff. These sessions provided critical insights into the operational challenges and user expectations. The findings underscored a significant gap: customers require immediate and precise access to vital information, such as product details, real-time stock availability, and current pricing. Furthermore, they expressed a strong preference for swift, accurate responses to their inquiries, a necessity often unmet by staff who may be occupied with other patrons. From the business standpoint, the analysis highlighted a pressing need for a system that could alleviate the heavy workload on staff by automating responses to routine queries, thereby allowing human employees to dedicate their expertise to resolving more intricate, high-value customer issues. The synthesis of these insights culminated in a clear set of requirements. Functionally, the chatbot was mandated to deliver comprehensive product information and to process natural language queries with both speed and accuracy. From a non-functional perspective, the system was required to be easily accessible via a responsive web interface, ensuring seamless usability across a wide range of devices. This rigorous initial phase established a clear roadmap for the subsequent development and implementation of a solution designed to meet these specific business and customer needs. the results of the data collection phase are shown in Table 1/

**Table 1.** Data Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Content** | **Price** | **Stock** | **Category** |
| 1 | Sabun Cuci Piring 800ml – Cairan pembersih lemak efektif untuk mencuci peralatan makan. | 15.000 | 120 | Perlengkapan Dapur |
| 2 | Deterjen Bubuk 1kg – Formula anti-noda, menjaga warna pakaian tetap cerah | 28.000 | 85 | Perawatan Pakaian |
| 3 | Pembersih Lantai 1L – Wangi segar, efektif menghilangkan kotoran dan bau. | 20.000 | 60 | Kebersihan Rumah |
| 4 | Spons Cuci Piring (Isi 3) – Spons lembut dan tahan lama untuk membersihkan peralatan dapur. | 10.000 | 200 | Perlengkapan Dapur |
| 5 | Tisu Gulung 10 Roll – Tisu lembut, tebal, dan menyerap baik untuk keperluan sehari-hari. | 35.000 | 150 | Kebutuhan Harian |
| 6 | Cairan Pembersih Kaca 500ml – Membersihkan kaca tanpa meninggalkan bekas. | 18.000 | 90 | Kebersihan Rumah |
| 7 | Sapu Lantai – Bulu sapu kuat dan tahan lama, cocok untuk rumah maupun kantor. | 25.000 | 70 | Kebersihan Rumah |
| 8 | Ember Plastik 20L – Ember serbaguna dengan pegangan kuat. | 22.000 | 50 | Perlengkapan Rumah |
| 9 | Lap Pel Microfiber – Menyerap air dengan cepat, mudah dibersihkan. | 30.000 | 40 | Kebersihan Rumah |
| … |  | … | … | … |
| 219 | Kantong Sampah 60x80cm (Isi 20) – Plastik tebal untuk menampung sampah rumah tangga. | 12.000 | 130 | Kebersihan Rumah |

## Data Preprocessing, Indexing and Vectorization

The second phaseof this study is data preprocessing, indexing, and vectorization that crucial foundation that determines the effectiveness of the chatbot system. The quality of the processed data will directly influence the model's ability to provide accurate and relevant responses to users. In this research, the primary data is sourced from a wholesale store's product catalog, which includes comprehensive information such as product descriptions, prices, stock availability, and store policies. An example of the data structure used can be seen in Table 1, which displays five out of 219 product entries to provide a representative overview. In this stage, the raw data from the product catalog undergoes a series of processes. First, data cleaning is performed to eliminate inconsistencies and formatting errors. Next, the descriptive text is broken down into smaller units of information, or *chunks*. This process is essential for the model to process information more efficiently. After chunking, each data unit is converted into a numerical representation a vector through the vectorization process using the BAAI/bge-small-en-v1.5 embedding model. This model was chosen for its ability to generate dense vectors rich in semantic meaning, which allows for accurate information retrieval. These vectors are then stored and indexed in a vector database, such as Pinecone. The indexing process enables the system to quickly find and retrieve the data most relevant to a user's query, which is a key pillar of the *Retrieval Augmented Generation* (RAG) architecture.

The core dataset for this research was a comprehensive product catalog obtained directly from the wholesale store's proprietor. To ensure seamless integration and efficient processing within the system's architecture, this raw data was meticulously converted into a structured JSON format. The selection of JSON was a deliberate choice, driven by its inherent flexibility and universal compatibility with modern web and data processing technologies. Its ability to represent complex, hierarchical data structures proved ideal for organizing detailed product information, which includes descriptions, prices, and stock levels. This standardization was a critical preparatory step that significantly simplified subsequent data handling tasks, such as preprocessing, chunking, and the crucial vectorization stage. By leveraging a widely supported format like JSON, we established a robust and scalable foundation that streamlined the entire development pipeline and ensured data integrity for the Retrieval Augmented Generation (RAG) system, which is fundamental to the accuracy and reliability of the final chatbot.

The document indexing stage is an essential foundation for building an effective RAG-based chatbot system. In this phase, the product catalog data, which has been structured in JSON format, is transformed into vector representations or *embeddings* to enable efficient semantic search. Embedding is a key technique that converts text into fixed-dimensional numerical vectors, allowing a computer to mathematically interpret the semantic meaning of each product item. For this research, the BAAI/bge-small-en-v1.5 model was utilized to generate 384-dimensional vectors, ensuring every text entity can be compared within the same vector space. This model was selected for its optimal balance of performance and computational efficiency. The embedding process is implemented using the FastAPI framework, which is renowned for its high performance and asynchronous support. FastAPI manages the data flow from the user interface, through the vectorization process, and finally to the generative model. The output of this embedding process is a vector matrix with dimensions n × 384, where *n* is the number of product items. This matrix enables the system to perform a semantic similarity-based search, which is far superior to conventional keyword-matching methods. To measure the similarity between a user's query and the data, the *cosine similarity* metric is employed.

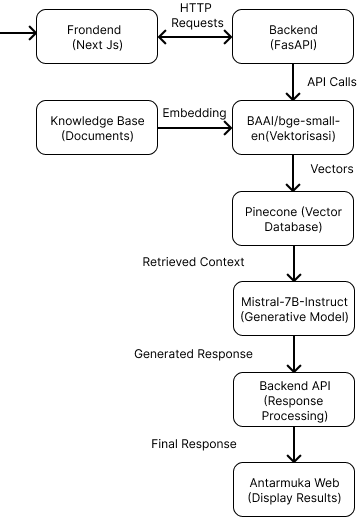
**Table 2.** Embedding Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Query (Q)** | **Item A (D1)** | **Item B (D2)** |
| 1 | 0.021 | 0.019 | 0.022 |
| 2 | 0.187 | 0.182 | 0.165 |
| 3 | 0.011 | 0.012 | 0.013 |
| 4 | 0.098 | 0.102 | 0.089 |
| 5 | 0.045 | 0.049 | 0.042 |
| 6 | 0.067 | 0.071 | 0.063 |
| 7 | 0.130 | 0.132 | 0.128 |
| 8 | 0.055 | 0.054 | 0.056 |
| 9 | 0.099 | 0.101 | 0.097 |
| 10 | 0.062 | 0.060 | 0.061 |
| ... | ... | ... | ... |
| 384 | 0.021 | 0.020 | 0.022 |

Each product item from the catalog and every user query is converted into a 384-dimensional vector using the specified embedding model. These vectors are a numerical representation of the text's semantic meaning. To measure the relevance between a user's query and a product item, the cosine similarity metric is employed. This method calculates similarity based on the angle between two vectors, where a value approaching 1 indicates a high degree of similarity. As an illustration, calculating cosine similarity involves three main steps: computing the dot product between the query vector and the item vector, calculating the norm (length) of each vector, and dividing them to obtain the similarity value. For example, if the cosine similarity between a query and Item A is 0.992, and with Item B is 0.894, this indicates that Item A is significantly more relevant to the query.

After these vectors are generated, they are stored in a vector database to facilitate fast and efficient searching. This research chose Pinecone as the vector storage and search platform due to its capability to handle large-scale, low-latency vector similarity searches. By storing these vectors in Pinecone, the system can perform semantic searches efficiently, retrieving the product items most relevant to a user's query based on vector similarity. This approach is far superior to traditional keyword-based searches because it captures context and meaning rather than just lexical matches. This ensures that the chatbot's responses are more accurate and relevant, making it a key foundation of a reliable Retrieval Augmented Generation (RAG) system.

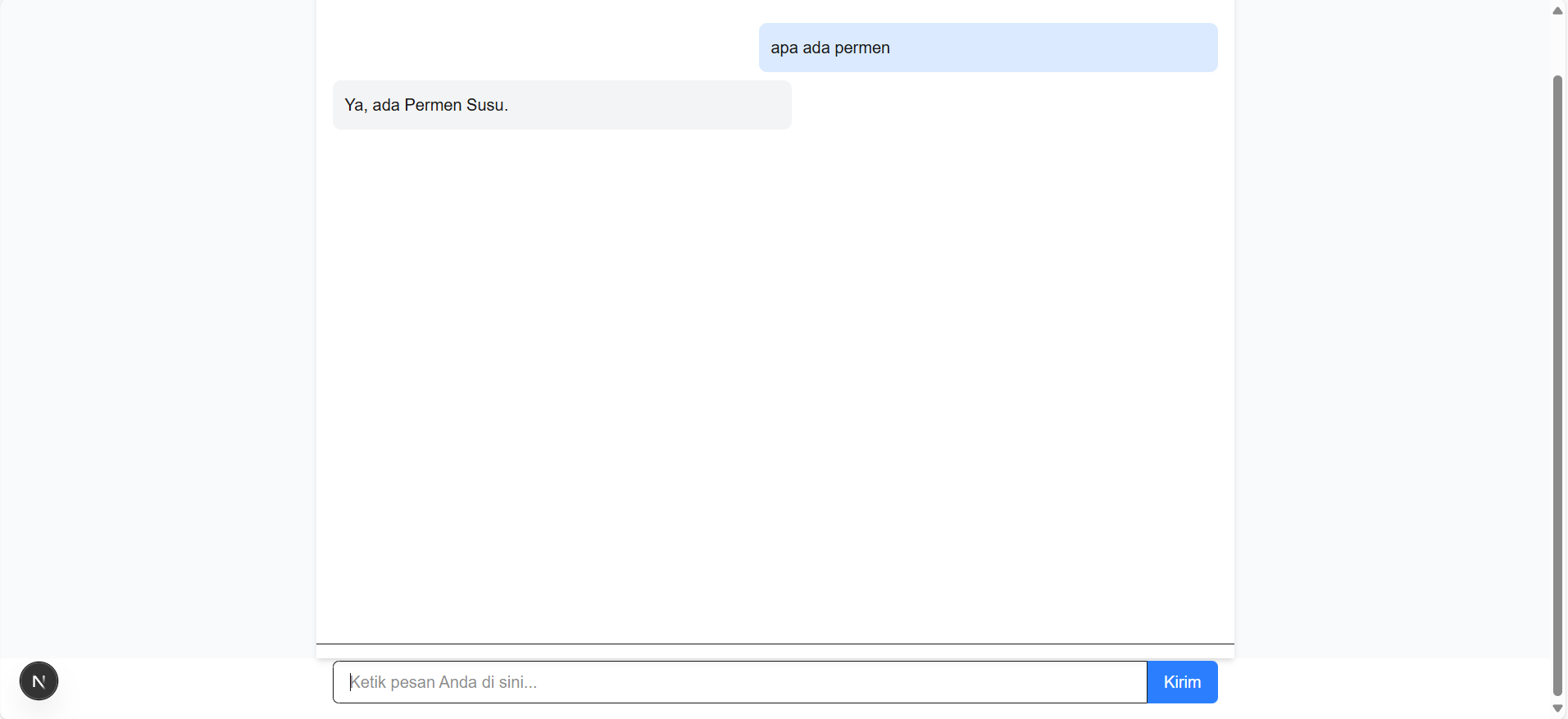
## System Development and Generative Modeling



**Figure 2.** Architecture of Retrieval Augmented Generation Based Chatbot System

Figure 2 presents the architecture of a chatbot system designed and implemented for customer service in wholesale stores, focusing on the Retrieval Augmented Generation (RAG) approach. This architecture integrates several key components to ensure an efficient workflow from user input to an accurate output. The system comprises two main components: a user interface (frontend) developed using Next.js and a server-side component (backend) built with FastAPI. The web interface serves as the initial point of interaction, where users input their queries. These user requests are then sent to the backend for comprehensive processing. The core of the backend processing is the implementation of RAG. The process begins when a user query is received. This query is converted into a numerical vector representation using the BAAI/bge-small-en-v1.5 embedding model. This query vector is then used to search for the most relevant data from a pre-indexed knowledge base. This knowledge base is stored in Pinecone, a vector database optimized for semantic similarity search. The most relevant documents from this search are then combined with the original user query and passed to Mistral-7B-Instruct, a generative model responsible for composing a coherent and contextual answer. The response generated by this model is then sent back to the backend for further processing and is subsequently transmitted to the web interface, where the results are displayed to the user. This architecture synergistically combines the efficient search capabilities of vector-based retrieval with the creativity of a generative model to provide relevant, accurate, and informative responses.

Next step are the core of the Retrieval Augmented Generation (RAG) approach, which synergistically combines relevant product items with a generative model's capabilities to produce comprehensive and contextual answers. This process begins with the formulation of a precise prompt, a crucial stage known as prompt engineering. An effective prompt is vital for determining the quality of the generative model's output, guiding it to produce responses that are accurate, relevant, and useful to the user. In this research, the prompt is constructed by combining the user's original question with contextual information retrieved from the product catalog. The developed prompt structure consists of three main components: (1) Task Instruction, which defines the model's role as a wholesale store assistant and provides general guidelines on the expected response format; (2) Context, which provides the relevant information from the retrieved product items; and (3) Question, which is the original query from the user that needs to be answered. Once the prompt is assembled, it is sent to the Mistral-7B-Instruct-v0.3 model to generate a response. This model, as the generative component, synthesizes the information provided in the prompt into a natural and easily understandable answer. The resulting answer from the model is then returned to the backend for further processing, such as formatting adjustments or validation. After the response is finalized, it is sent back to the frontend to be displayed via a user-friendly web interface. This workflow ensures that the chatbot's answers are not merely the output of a generative model but are also supported by factual data from an external knowledge base, thereby significantly enhancing their accuracy and reliability.



**Figure 3.** Fieldtext Chatbot

Figure 3 shows the fields resulting from the development of the chatbot system. The indexing process for the product catalog data resulted in a 384-dimensional vector representation for each item, using the BAAI/bge-small-en-v1.5 embedding model. The resulting vector database was capable of efficiently storing and indexing all 219 product items in the store's catalog. This vector representation enabled a semantic search that is far more advanced than traditional keyword-based searches, allowing the system to understand the context and meaning of customer queries. The implementation of the RAG pipeline with the Mistral-7B-Instruct-v0.3 generative model successfully combined semantic search capabilities with contextual response generation. The system was able to respond to various types of user questions, ranging from specific product information to store policies.

## System Testing and Evaluation

System testing was conducted using a method of comparing expected output and actual outcomes. To obtain a comprehensive overview of the performance, we designed 10 test scenarios that represented various types of customer queries, ranging from simple questions (e.g., "Berapa harga beras premium 5 Kg?") to more complex and contextual ones (e.g., "Produk apa yang cocok untuk membuat roti?"). Each test scenario was executed on the chatbot, and the generated responses were recorded. These actual outcomes were then compared against a set of predefined expected answers. To quantitatively measure the system's effectiveness, we utilized standard evaluation metrics such as Accuracy, Precision, Recall, and F1-Score. These metrics provided an objective assessment of how well the system retrieved correct information (precision), found all relevant information (recall), and delivered accurate responses overall (accuracy and F1-score). Through this systematic testing approach, we were able to thoroughly evaluate the performance of the RAG chatbot and validate the research hypotheses.

**Table 3.** Expected and Actual Outcomes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Scenario** | **Input** | **Expected Outcomes** | **Actual Outcomes** |
| 1 | Menampilkan harga produk | Berapa harga Indomie Goreng? | 3500 | 3500 |
| 2 | Menampilkan stok produk | Stok Yupi Gummy Pizza berapa? | 85 | 85 |
| 3 | Menampilkan kategori produk | Kategori Yupi Gummy Pizza | permen | permen |
| 4 | Menangani pertanyaan umum | Apa itu Indomie Goreng? | Menjelaskan isi konten produk | Menjelaskan isi konten produk |
| 5 | Menangani input dengan kesalahan ketik | Harga Indomee goleng? | 3.500 (tetap mengenali) | 3500 |
| 6 | Menangani pertanyaan tidak relevan | Ada iPhone? | "Tidak memiliki informasi produk" | "Tidak memiliki informasi produk" |
| 7 | Menangani input ambigu | Berapa harga? | “Harga yang Anda tanyakan tidak spesifik” | “Harga yang Anda tanyakan tidak spesifik” |
| 8 | Menjawab pertanyaan dengan sinonim | Berapa biaya Yupi Gummy Pizza | 9500 | 9500 |
| 9 | Menangani pertanyaan dengan format kalimat panjang | Saya ingin tahu berapa harga dan stok dari Indomie Goreng ya | Menampilkan harga dan stok | 3500, 200 |
| 10 | Respons terhadap permintaan produk tidak tersedia | Harga Tanggo? | "Tidak memiliki informasi produk" | Tidak ada produk yang sesuai dengan nama "Tanggo" dalam konteks yang diberikan. |

Based on the test results presented in the Table 3, it can be concluded that the chatbot system demonstrates very good performance in handling various scenarios. Out of the 10 test scenarios, the chatbot successfully provided accurate and expected responses in most cases, including direct questions about product price, stock, and category (No. 1, 2, 3). The system's ability to handle general questions and even typing errors (typos) also proved effective (No. 4, 5). This confirms that the implemented RAG approach is capable of understanding the semantic meaning behind user queries, not just matching keywords. The system also showed reliability in handling irrelevant or ambiguous questions (No. 6, 7), as well as questions using synonyms (No. 8). However, there was a slight difference in scenario No. 10, where the actual response provided a more detailed explanation about the absence of a product matching the name queried, demonstrating a more flexible generative capability of the model. Overall, these results validate that the RAG system is capable of providing responses that are far more intelligent and accurate than a static FAQ-based system.

Based on the quantitative evaluation of the 10 test scenarios at Table 4, the chatbot's performance was measured using standard evaluation metrics. Out of the total 10 scenarios, there were 7 True Positive (TP) cases, which indicates accurate and expected responses, as well as 1 False Positive (FP) case, 1 False Negative (FN) case, and 1 True Negative (TN) case. With this data, the performance metrics were calculated as follows: The system's Accuracy reached 80%, which represents the overall proportion of correct responses. A Precision value of 87.5% indicates that out of all responses claimed to be relevant by the system, the majority were indeed relevant. Recall also showed a figure of 87.5%, meaning the system successfully found most of the relevant information it was supposed to find. Finally, the F1-score, which is the harmonic mean of precision and recall, reached 0.875, confirming a good balance between the two metrics and indicating strong overall system performance.

**Table 4.** Evaluation Matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan Pengguna** | **Jawaban Chatbot** | **Jawaban Seharusnya** | **Relevan (Y/T)** | **TP** | **FP** | **FN** | **TN** |
| 1 | Berapa harga Indomie Goreng? | 3.500 | 3.500 | Y | 1 | 0 | 0 | 0 |
| 2 | Stok Tora Bika Cappuccino berapa? | 95 | 95 | Y | 1 | 0 | 0 | 0 |
| 3 | Produk apa saja dari kategori mie instan? | Mie Goreng Pedas Gila, Mie Sedaap Kari Spesial, Mie ABC, Mie Sedaap Goreng, dan Mie Gaga 100. | Termasuk juga Indomie Goreng, dll | Y | 0 | 0 | 1 | 0 |
| 4 | Harga SilverQueen 65g? | Tidak ada informasi tersedia mengenai harga Silver Queen 65g dalam konteks yang diberikan. | Produk tidak tersedia | Y | 1 | 0 | 0 | 0 |
| 5 | Siapa presiden Indonesia? | Tidak memiliki informasi tersebut dalam konteks yang diberikan. | Tidak relevan | T | 0 | 0 | 0 | 1 |
| 6 | Yang paling murah? | 13500 | Chocolatos 1500 | Y | 0 | 1 | 0 | 0 |
| 7 | Apakah ada Jetz? | Tidak, toko grosir ini tidak menjual produk Jetz. | Produk tidak tersedia | Y | 1 | 0 | 0 | 0 |
| 8 | Kategori dari Indomie Goreng? | Mie instan | Mie instan | Y | 1 | 0 | 0 | 0 |
| 9 | Harga Joshua? | Tidak ada produk dengan nama "Joshua" dalam informasi yang tersedia | Produk tidak tersedia | Y | 1 | 0 | 0 | 0 |
| 10 | Stok Boncabe? | Stok Boncabe tidak tersedia dalam konteks yang diberikan. | Produk tidak tersedia | Y | 1 | 0 | 0 | 0 |

The performance testing of the vector-based search yielded promising results in terms of precision and relevance. The cosine similarity metric served as the basis for measuring the semantic proximity between user queries and product items in the catalogue. Testing with various queries demonstrated that the system was able to identify relevant products even when the user's query did not use the exact same terminology as the product descriptions in the catalogue. Experiments with different top-N parameter values in the retrieval process showed that N=5 provided the optimal balance between precision and recall. This value was sufficient to provide adequate context for the generative model without introducing excessive noise from less relevant items.

The test results indicate that the integration of vectorization (BAAI/bge-small-en-v1.5) and a generative model (Mistral-7B-Instruct-v0.3) within the RAG pipeline successfully enhanced the relevance and accuracy of the chatbot's answers. The high precision and recall values demonstrate that the system is effective at identifying and displaying product information that corresponds to user questions.

However, the 80% accuracy also suggests that there are still cases where the system could not provide a precise answer, particularly for ambiguous questions or those requiring more complex reasoning. This could be attributed to limitations of the embedding data or the generative model's incomplete understanding of the wholesale store domain context. A comparison with conventional FAQ systems shows that the RAG approach offers a significant improvement in flexibility and natural language understanding. Nevertheless, to achieve an accuracy above 90%, further optimization is needed.

One of the main limitations of the system is that the chatbot is not yet integrated with a point-of-sale (POS) or cashier system. Consequently, the chatbot is unable to update product stock data in real-time. The information on product availability provided still relies on the static catalog data indexed at the beginning of development. This limitation has the potential to cause discrepancies between the stock information provided by the chatbot and the actual conditions in the store, especially when direct sales or restocking events occur. This could affect the accuracy of the information and customer satisfaction. To address this, future development is recommended to integrate the chatbot with the POS system via an API, allowing stock data to be updated automatically and in real-time. This integration will significantly enhance the reliability of the information and the overall user experience.

# conclusion

This research successfully developed an AI-based chatbot using the Retrieval Augmented Generation (RAG) approach, which combines vectorization techniques with the BAAI/bge-small-en-v1.5 model and the Mistral-7B-Instruct-v0.3 generative model for wholesale store customer service. System evaluation showed an information retrieval accuracy of 80%, with precision and recall both at 87.5%, and an F1-score of 0.875. The RAG approach proved effective in enhancing the relevance and accuracy of the chatbot's answers compared to conventional FAQ systems, by generating contextual and natural responses. However, the system still has limitations in handling complex questions and is not yet integrated with the point-of-sale system, which prevents real-time product stock updates. Further research is recommended to integrate the chatbot with the cashier system to ensure stock information is always accurate and up-to-date, and to improve the quality of the embedding and generative models to better understand natural language and a wider variety of user queries.

# References

1. Wijayanti, and Anton Agus Setyawan. “DAMPAK KUALITAS LAYANAN TERHADAP KEPUASAN PELANGGAN: STUDI PADA TOKO MATAHARI DEPARTMENT STORE SOLO SQUARE.” manajemen-dirgantara, vol. 16, no. 2, pp. 609–619, Jan. 2024, doi: 10.56521/manajemen-dirgantara.v16i02.1042.
2. Y. Y. Baek, S. Park, and Y. B. Park. “Chatbot Program for Proposed Requirements in Korean Problem Specification Document.” IJACSA, vol. 14, no. 8, 2023, doi: 10.14569/IJACSA.2023.0140875.
3. Dr.Arthy Rajakumar, Vignesh V K, and Sandeep Joe A. “AI-Powered Cardiovascular Health Chatbot: Design and Development.” Int Res J Adv Engg Hub, vol. 3, no. 03, pp. 532–539, Mar. 2025, doi: 10.47392/IRJAEH.2025.0075.
4. Wiradinata, Antonius Sakti, and Viny Christanti Mawardi. "Abstractive Text Summarization Berita Bahasa Indonesia Menggunakan Retrieval-Augmented Generation." Jurnal Ilmu Komputer dan Sistem Informasi 13.1 (2025).
5. H. S. Bank and D. R. Herber. “CatalogBank: A Structured and Interoperable Catalog Dataset with a Semi-Automatic Annotation Tool (DocumentLabeler) for Engineering System Design.” 2024, arXiv. doi: 10.48550/ARXIV.2408.08238.
6. S. Labat, T. Demeester, and V. Hoste. “EmoTwiCS: a corpus for modelling emotion trajectories in Dutch customer service dialogues on Twitter.” Lang Resources & Evaluation, vol. 58, no. 2, pp. 505–546, Jun. 2024, doi: 10.1007/s10579-023-09700-0.
7. A. A. Mutiari, L. Edi Nugroho, and M. A. Khadija. “Utilizing Natural Language Processing in Chatbots for Integrated Complaint Manage-ment Application: A Case Study of PT PLN (Persero) ULP Padalarang.” in 2024 7th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia: IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/ISRITI64779.2024.10963544.
8. K. Nishida, K. Nishida, H. Asano, and J. Tomita. “Natural Language Inference with Definition Embedding Considering Context On the Fly.” in Proceedings of The Third Workshop on Representation Learning for NLP, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 58–63. doi: 10.18653/v1/W18-3007.
9. P. Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” 2020, arXiv. doi: 10.48550/ARXIV.2005.11401.
10. Fanani, Ikhsan. "IMPLEMENTASI RETRIEVAL AUGMENTED GENERATION UNTUK EVALUASI PROPOSAL TUGAS AKHIR MAHASISWA." Jurnal Teknologi Komputer dan Informatika 3.2 (2025).
11. Suharso, Wildan, et al. "Music Features Pada Bidang Ilmu Komputer Menggunakan Modularity Clustering." Just IT: Jurnal Sistem Informasi, Teknologi Informasi, dan Komputer (2022).
12. Yuriansyah, Dwiki Adi, et al. "A cluster analysis of information data poverty people in Indonesia." AIP Conference Proceedings. Vol. 3179. No. 1. AIP Publishing LLC, 2025

.