International Journal of Cyber and IT Service Management (IJCITSM)
Vol. 5, No. 2, October 2025, pp. 144−155
E-ISSN: 2808-554X | P-ISSN: 2797-1325, DOI:10.34306

Creation of RAG Chatbot in Answering Queries Related to Banking Terms Using Microsoft Azure

Cindy Aurora[1*], Tuga Mauritsius[2]
[1,2]Master of Information System Management, Bina Nusantara University, Indonesia
[1]cindy.aurora@binus.ac.id, [2]tmauritsus@binus.edu
***Corresponding Author**

Article Info

Article history:

Submission June 11, 2025
Revised July 22, 2025
Accepted August 26, 2025
Published September 9, 2025

Keywords:

Natural Language Processing
Large Language Model
Retrieval Augmented Generation
Chatbot
Microsoft Azure

ABSTRACT

This study investigates the use of Natural Language Processing (NLP), specifically Large Language Models (LLM), to implement technology intelligence as a solution for supporting developers in understanding information-related terms, abbreviations, and business processes in the banking sector. **The aim is to explore** the effectiveness of integrating AI-powered chatbots, particularly through Retrieval Augmented Generation (RAG), to enhance software development in banking by providing fast, accurate responses. **The methodology** involves profiling data analysis and the development of a RAG based AI chatbot using the Microsoft Azure platform, integrating advanced NLP and LLM techniques to assist developers in navigating complex banking terms and processes efficiently. **The results** demonstrate that the RAG chatbot significantly improves operational efficiency by offering real-time, context-aware responses, enabling faster decision-making and reducing time spent on manual searches for information, which leads to faster software development cycles. **This study contributes** to the fields of NLP and LLM, particularly in the banking sector, by showcasing the benefits of RAG chatbots in improving operational efficiency and software development quality. The use of AI technologies provides substantial improvements in the development process, leading to enhanced productivity in the banking industry.

DOI: https://doi.org/10.34306/ijcitsm.v5i2.209

1. INTRODUCTION

Developing banking software requires a deep understanding of both technical and business processes. Profiling data is crucial for ensuring the project stays on schedule. However, challenges arise in comprehending specific business terms and processes, making it essential to find accurate information. While tools like Google can help, they may not be effective for unique company processes. Alternatives like ChatGPT have limitations, as its knowledge only covers information up until 2022 and lacks access to specific documents from Bank Indonesia, leaving developers struggling with internal company documents [1–4].

This study focuses on developing AI-based solutions using NLP chatbots to speed up the search and understanding of banking-related terms, abbreviations, and business processes. By implementing LLM and RAG chatbots, the system quickly identifies context queries and delivers accurate responses, increasing operational efficiency and reducing the time spent on manual searches, allowing developers to focus on more complex tasks [5].

This study is expected to benefit organizations, developers, and academics. For organizations, the chatbot boosts operational efficiency by quickly answering queries, reducing manual work, and optimizing resources [6]. For developers, it provides insights and techniques to enhance chatbot performance, especially in responsiveness and handling diverse questions. Academically, this research contributes to understanding NLP and RAG chatbot theory in banking projects and can serve as a valuable reference for future studies. Scope study This covers exploration use of AI for increase efficiency and accuracy in search information related to business processes banking. Chatbot built is a RAG Chatbot that utilizes SEKI Metadata from Bank Indonesia as a knowledge base, with the Microsoft Bot Framework platform, and uses Language Python programming for implementation on the web [7].

1.1. Banking IT Project Development

Banking is crucial for collecting funds and supporting economic growth. Bank Indonesia (BI) ensures financial stability through monetary policy, supervision, and managing the payment system. Its main tasks include maintaining rupiah stability and controlling inflation through interest rate policies and open market operations [8, 9]. BI also supports sustainable economic growth and financial inclusion. BI provides key economic indicators like SEKI, ITEMs, and The Red Book for informed decision-making [10]. Internally, BI uses Agile and Scrum frameworks, emphasizing collaboration. Business metadata is essential for accurate decision-making, but developers face challenges in understanding terms and processes, making good documentation critical for project success.

1.2. NLP and Chatbot

Chatbots can aid in understanding business metadata by offering an interface for managing business-related information [11]. They are classified into two types, Rule-Based and AI-Based, with architecture generally involving three stages, pre-processing (NLP), processing (NLU), and generation (NLG) to create responses [3]. Chatbots are also categorized by domain, Open Domain and Closed Domain, and they can process text or voice input [12]. In regulated sectors like banking, NLP-based chatbots face challenges, including the need to comprehend specialized financial terms and adhere to industry regulations, requiring focused development efforts for effective operation [1]. The Microsoft Bot Framework is a widely used open-source framework for building domain-specific chatbots [2].

1.3. LLM and RAG

A Large Language Model (LLM) is an AI that understands and generates human-like text, trained with deep learning techniques and millions to billions of parameters to recognize language patterns [13]. LLMs, like OpenAI's GPT, perform various tasks such as answering questions and translation, and are used in applications like chatbots and automatic writing. GPT models, including GPT-2, GPT-3, and GPT-4, have revolutionized AI with their human-like text generation [13]. RAG combines LLM text generation with information retrieval for more accurate responses, especially for recent data or specific queries [14]. RAG uses LLM to process input, retrieves data from external sources, and then generates a response, with Microsoft Azure's setup including application servers, Azure AI Search, and OpenAI GPT for final processing [15].

1.4. Azure AI Search

Azure AI Search enhances the chatbot experience in the RAG context by using the text-embedding-ada-002 model for various NLP tasks like semantic search, text classification, content recommendations, and clustering [14]. The main processes include indexing (vectorization) and retrieval, where documents are processed into tokens and vectors for search, with AI skills used to enrich the data. Search vectors use numerical representation to find similarities between vectors, enabling semantic or conceptual matching [16]. In RAG, Azure AI Search collects and indexes relevant data, searches based on user queries, and returns relevant results, which are then combined with a generative model like OpenAI GPT to produce more accurate and informative responses [14].

1.5. Azure OpenAI (GPT 3.5 Turbo)

GPT-3.5 Turbo plays a role as a generator model text in the implementation of RAG on Azure, which combines retrieved data via Azure AI Search for increased accuracy and relevance response. As part of RAG, GPT-3.5 Turbo does not only produce answers based on knowledge training general, but also integrates information from source relevant external for give more response informative and contextual [17]. GPT-3.5 Turbo is built with transformer architecture that uses an "attention" mechanism to understand input context and output appropriate text. This model is also trained using Reinforcement Learning from Human Feedback (RLHF) to increase responsiveness and ethical answers [10]. Designed For efficiency, GPT-3.5 Turbo can give a response with latency low, fit for real-time applications.

1.6. Evaluation & CRISP-DM

Stratified Split is a method of data sharing based on a theme to ensure every group reflects balanced proportions from every sector, allowing more analysis depth and understanding of trends in every sector of business [18]. Chatbot evaluation is divided into three sections: evaluation content, satisfaction users, and functionality, which can tested through functional testing, error testing, and the accuracy of chatbot answers [3]. CRISP-DM (Cross-Industry Standard Process for Data Mining) is a methodology standard for project data mining consisting of six phases main, Understanding Business, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Methodology This is flexible and iterative, allowing adjustments in each stage based on findings and bait-back [19].

1.7. Previous Research

Chatbots, supported by machine learning, are used in various fields to improve customer service and user performance [12]. BPMN-based chatbots, integrated with platforms like Microsoft Bot Azure, transform images into interactive tools for business processes [20]. In healthcare, chatbots analyze symptoms, offer telehealth services, and manage conditions like hypertension [21, 22]. They also educate on HIV, contraception, and pregnancy [23] and help isolated individuals seek mental health support [24, 25]. Chatbots are also used in software testing [4] and language education [26]. The Retrieval-Augmented Generation (RAG) chatbot combines retrieval and generation for accurate responses [27]. RAG chatbots assist in education and healthcare with text extraction, summarization, and medical responses [28]. Overall, chatbots enhance efficiency across industries.

2. METHOD

The integration of Azure AI Search with OpenAI GPT and the custom Bank Indonesia metadata database is key to the chatbot's ability to generate accurate and context-aware responses. This combination of tools allows the chatbot to access up-to-date financial data, providing domain-specific responses tailored to the unique needs of the banking sector. The methodology used in this study is CRISP-DM which consists of six main stages, each of which has a series of specific tasks and activities as follows:

2.1. Business Understanding

Business Understanding in the CRISP-DM methodology is crucial for aligning objectives with software development, especially in banking. This study explores the use of AI, particularly NLP and LLM with the RAG approach, to help developers quickly access relevant information and improve operational efficiency [29]. The proposed RAG chatbot integrates real-time data retrieval with LLM-generated responses, offering more precise, domain-specific answers, making it more effective than traditional rule-based systems for the banking sector.

2.2. Data Understanding

This stage focuses on exploring and understanding the research data. The first step is downloading SEKI metadata from Bank Indonesia's website, which contains unstructured data in PDF format for analysis. The methodology lacks details on preprocessing and cleaning the data for chatbot integration. Key challenges include handling missing data, formatting issues, and ensuring compliance with industry standards, particularly for sensitive banking information. The next step is to stratify the data by sectors, Monetary, Government Financial, Real, and External, to guide the appropriate analysis for each sector.

2.3. Data Preparation

In the Data Preparation stage, selected data will be uploaded to Microsoft Azure, using Azure AI Document Intelligence for intelligent document processing. This cloud-based service automatically extracts data from PDFs and supports OCR to extract text from images or complex documents. It also handles various file types, including .docx, .pptx, .xlsx, .png, .jpg, and others, enabling efficient data processing for further analysis.

2.4. Modeling

In the Modeling stage, the main focus is to build a chatbot model that is by business objectives, specifically using OpenAI's Large Language Model (GPT-3.5 Turbo) for text interpretation. The steps taken include filling in Microsoft Azure credentials, defining the LLM role, running the Command Line Interface for deployment, and checking the service after deployment. GitHub is used for coding, versioning, and documentation management. Evaluation and optimization are carried out continuously to ensure optimal chatbot performance according to business needs.

2.5. Evaluation

To assess the quality and effectiveness of the model, several functional tests were carried out to evaluate its performance. These tests included Functional Testing, which ensures that the chatbot features and functionality operate as expected, Error Testing, which identifies chatbot responses to typos or ambiguous queries, and evaluates the accuracy of its answers to user questions.

2.6. Implementation

In the Deployment phase in CRISP-DM, the focus is on implementing the solution into a production environment. In this study, the chatbot was integrated into a website using the Azure App Service hosting service, allowing the chatbot to be accessed by users across multiple platforms and applications.

3. RESULT AND DISCUSSION

3.1. Business Understanding

To provide a deeper understanding of the economic and financial sectors of Indonesia, the following table presents the SEKI (Indonesian Economic and Financial Statistics) metadata. This table classifies the data into four main sectors: the monetary sector, government financial sector, real sector, and external sector, covering various key economic indicators essential for analyzing Indonesia's macroeconomic conditions. Below is the detailed breakdown of the data presented in this Table 1.

Table 1. SEKI Metadata (Indonesian Economic and Financial Statistics)

A. Monetary Sector:
1. Primary Money
2. Balance Sheet General Bank and BPR Analytics
3. Money Circulation and Factors Affecting It
4. Community Savings
5. Loans Given
6. Finance Company
7. Operation Monetary
8. Interest rate
9. Micro, Small and Medium Enterprises (MSMEs) Credit
B. Government Financial Sector
10. Operation Finance Central government
11. Government Securities
C. Real Sector:
12. Consumer Price Index (CPI)
13. Wholesale Price Index (IHPB)
14. Gross Domestic Product (GDP)
D. External Sector:
15. Indonesia's Balance of Payments
16. Foreign Trade
17. Indonesia's International Investment Position
18. International Economic Indicators
19. Nominal Exchange Rate Index Against Major Partner Currencies
20. Foreign Debt
21. State Foreign Exchange Reserves

This can be observed in Table 1. SEKI (Indonesian Economic and Financial Statistics) is a monthly publication by Bank Indonesia that provides comprehensive information on Indonesia's economy and finance, using internationally recognized methodology for comparative analysis with other countries' data. SEKI includes primary data from Bank Indonesia and secondary data from institutions like the Ministry of Finance, BPS, and LPS, covering four main sectors:

- Monetary Sector: Providing information about liquidity, money supply, balance sheet authority monetary, and dynamics of money markets and capital markets.
- Financial Sector Government: Focus on transparency and accountability implementation of the APBN, including data on revenue, expenditure, financing, and SBN status.
- Real Sector: Provides related data Income Gross Domestic Product and Consumer Price Index For evaluation of economics and policy.
- External Sector: Presents data about balance sheet payment, loan external, and indicators economy as well as monetary international, covering transaction export-import and obligations overseas.

3.2. Data Understanding

There are 21 SEKI Metadata grouped into four sectors, with details of Monetary Sector 9 Metadata, Government Financial Sector 2 Metadata, Real Sector 3 Metadata, and External Sector 7 Metadata. The author chose to use 9 Metadata from the Monetary Sector, which are unstructured data in PDF format. Each PDF metadata has several pieces of information, including:

- Basic Information: Contains contact data, name, statistics organizer, address, and related information.
- Data Definition: Briefly explain the type of data contained.
- Data Coverage: State the data limitations, including influencing factors and institutional coverage.
- Publication Periodization: Provides information about the frequency of data publication (daily, monthly, yearly, etc.).
- Timeliness of Publication: State the exact time the publication was made.
- Forward Publication Schedule (ARC): Includes a schedule and examples of published data.
- Data Sources: Mention the institutions, applications, and portals used as data sources.
- Methodology: Describes concepts, definitions, collection methods, and factors influencing data.
- Data Integrity: Conveys data status (interim or final) and possible revisions.
- Data Access: Provides information about where or how to access published data.

3.3. Data Preparation

Before performing data preparation using Azure AI Document Intelligence Service, it is necessary to create a GitHub account and clone the Azure Samples Template first with the following steps:

- Create a GitHub account at https://github.com/codespaces and verify the account to be used.
- Once you have signed in, use the template from Microsoft Azure with the following link Azure-Samples/azure-search-openai-demo which is a sample app for the Retrieval-Augmented Generation pattern running in Azure, using Azure AI Search for retrieval and Azure OpenAI large language models to power ChatGPT-style and Q&A experiences (github.com).
- The next step is to create a new codespace using the Azure template at point b by clicking on the "<> Code" button, clicking the Codespaces menu then clicking the "Create codespace on main" button. After the codespace creation process is complete, then open the codespace page by clicking on the title of the codespace created.
- In default mode, Azure has prepared some sample files that can be used. After the GitHub project page opens, the first step is to delete the existing sample files and re-upload the files that will be used as a knowledge base.

3.4. Data Modeling

At this stage, the focus will be on creating a predictive model with the main goal of producing effective chatbot interactions that meet user needs. After the data preparation process is carried out, the model will be developed using an algorithm that matches the chatbot context, namely the GPT-35-Turbo large language model created by Open AI for text understanding. This modeling stage will be implemented with the help of GitHub as a collaborative repository for code, version tracking, and documentation of changes during the development process. The steps to be taken are as follows:

- Continuing the previous stage in the Data Preparation sub-chapter, still on the same GitHub, the next thing to do is to fill in the credentials used in the ".env" script file. In this environment setting, the author uses vectorization and data search with the text-embedding-ada-002 model which is an Azure AI Search service, and the GPT-35-Turbo model from Azure Open AI for the LLM model used.

- After the credentials are filled in, the next step is to define the role of LLM as a smart brain by setting the prompting in the app > backend > approaches > chatreadretrieveread. Py file folder.

```
@property
def system_message_chat_conversation(self):
    return """Kamu adalah asisten untuk para pengembang dalam memahami data SEKI (statistik ekonomi dan keuangan Indonesia).
    Jawab dengan singkat dan padat. Jawablah hanya sesuai dengan fakta dan sumber di bawah.
    Kalau tidak ada informasi yang cukup sesuai dengan pertanyaan, jawablah tidak tahu.
    JANGAN MENJAWAB DI LUAR SUMBER DI BAWAH. Apabila kamu membutuhkan pertanyaan lanjutan untuk memastikan sesuatu, tanya lah.
    Jawablah hanya dalam Bahasa Indonesia yang baik.
    Dalam melakukan sitasi, berikan nama file, diikuti dengan potongan informasi di dalam tanda kurung.
    Selalu sertakan sitasi dalam setiap responmu (apabila memungkinkan).
    Gunakan kurung kotak untuk sitasi, misal: [info1.txt]. Jangan digabung, jabarkan secara terpisah, seperti: [info1.txt] [info2.pdf]
    {follow_up_questions_prompt}
    {injected_prompt}
    """
```

Figure 1. LLM Instruction Script

- The script in Figure 1 LLM Instruction Script defines a property called system_message_chat_conversation, which generates a system message or instruction for the assistant (AI model) in performing a specific task. @property is a Python decorator that allows methods to be called like attributes. So, when system_message_chat_conversation is called it will produce a string without requiring parentheses like a function call. This message is designed to instruct the AI to act as a special assistant for developers in understanding SEKI data. The AI is asked to provide short, concise, and factual answers, only answer questions according to the information contained in the data source provided, avoid speculation or information outside the source, and answer "don't know" if there is not enough information and ask follow-up questions if there is a need for clarification.

- AI machines are also required to answer in formal and clear Indonesian. Include citations from data sources in every possible response, with the format [info1.txt] [info2.pdf]. Both variables (follow_up_questions_prompt) and (injected_prompt) indicate that it is possible to add additional instructions (injected_prompt) or follow-up questions (follow_up_questions_prompt) that will be added in a certain context to become an answer.

- To perform Deployment, run the CLI (Command Line Interface) command "and up". This command will automatically run the command that has been defined in the azure.yaml script, one of which will call the command at the start.sh script. If the "and up" command has been executed, then define the name of the environment to be used. Then log in with the command "and auth login". Next, log in using the code created by GitHub.

- After logging in, then run the "and up" command again. If the command is successful, you will be asked to fill in the Azure account subscription and the Azure server location to be used. Then the system will work and run the existing script until all the knowledge files used are uploaded, if the deployment is successful, a log will appear as shown in Figure 2 GitHub Success Log.

Figure 2. GitHub Success Log

Figure 2 illustrates the GitHub success log generated after the deployment process, confirming that the application was successfully deployed to Azure. The log provides details of the deployment duration, service endpoint, and access to the created resources, indicating that the system is ready for further use and testing.

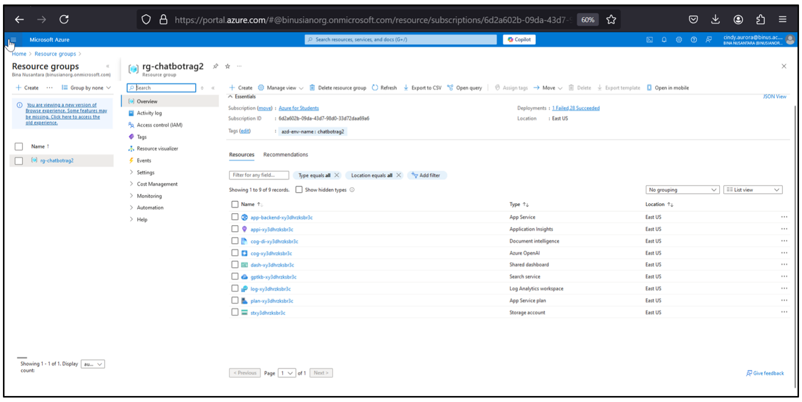

Figure 3. Resource Group Services Azure

- After the deployment is successfully executed, then you can check the service that has been formed on the Azure Portal. The service can be seen in the Resource Group menu > select the project that was formed > Overview > Resources as seen in Figure 3 Azure Resource Group Services. Make sure 9 services have been formed, namely:
  - App-backend: App Service for frontend and backend.
  - Appi: Application Inside for activity reporting.
  - Cog-di: Document Intelligence that acts as OCR.
  - Cog: Open AI LLM used.
  - Dash: Functions for the dashboard.
  - Gptkb: Functions for creating vectors for Azure AI Search.
  - Log: Application activity log.
  - Plan: Used for server hosting.
  - St: Storage Account where Knowledge Base is stored.

After a successful deployment, the next step is to ensure that all required services are properly established in the Azure portal. At this stage, critical components, such as backend services, activity reporting applications, and features supporting AI and analytics functionality, will be checked to ensure overall system integrity. This testing is crucial for ensuring smooth operations and ensuring that all necessary resources are properly configured.

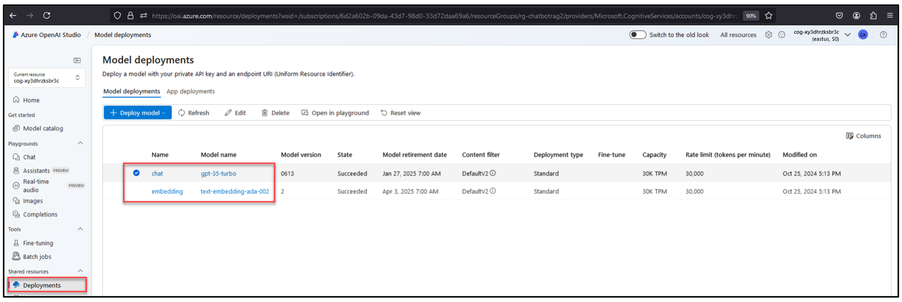

Figure 4. Chatbot Deployment Model

- In addition to checking the services formed, checks were also carried out for the Retrieval and Generation models formed. In Figure 4 the Chatbot Deployment Model can be seen after deployment on Azure Open AI, the models formed are GPT-35-turbo and text-embedding-ada-002.

3.5. Evaluation

- Feature Testing Scenario

  Before presenting the Feature Testing Scenarios, the following Table 2 outlines the key test scenarios conducted to evaluate the chatbot's functionality. The tests involved accessing the chatbot across different browsers and assessing its ability to display question suggestions based on the knowledge base when used for the first time. The results were analyzed to ensure proper functionality across various platforms.

Table 2. Feature Testing Scenarios

No	Test Scenario	Expected Output	Results
1	Accessing Chatbot using Chrome, Firefox, Edge browser	Chatbot successfully accessed	Success
2	Displays question suggestions based on the knowledge base when using the chatbot for the first time	Displaying question suggestions	Success

Based on Table 2 Feature Testing Scenario, the chatbot was tested on Google Chrome, Mozilla Firefox, and Microsoft Edge. While functional testing and error handling provided useful insights, adding statistical analysis (e.g., precision, recall, F1-score) and A/B testing would better measure accuracy and efficiency. The results showed the chatbot performed consistently across all browsers, and the second scenario confirmed that knowledge base question suggestions appeared correctly. This indicates that the chatbot, built on Microsoft Azure, is not only compatible with widely used browsers but also reliable in guiding users with relevant suggestions in diverse usage contexts.

- Scenario with Ambiguous Questions

  To further evaluate the robustness of the chatbot, error testing scenarios were conducted to examine its ability to handle incomplete or ambiguous queries. These tests aimed to verify whether the chatbot could still provide relevant and accurate responses even when the input contained missing letters, symbols, or other irregularities. The summary of these scenarios is presented in Table 3.

Table 3. Feature Error Testing Scenarios

No	Test Scenario	Expected Output	Results
1	If the letter from the inputted word No true/less	Chatbot successfully displays the correct result	Success
2	If you enter the wrong symbols or numbers	Chatbot successfully displays the correct result	Success

It can be seen in Table 3 (Feature Error Testing Scenario), using the first scenario, testing was carried out on the words from the reduced question query. In this scenario, the Author first entered a question query with a complete question without a question mark ”Explain what is primary money”, then the chatbot managed to provide an appropriate explanation in answering the question. Next, the Author entered a query with reduced letters ”Explain what is primary money”, the chatbot still provided an answer that was relevant to the question. In scenario 2, the Author re-inputted the question query by inserting a symbol in the question sentence ”Explain what {} | {} { {}{ is primary money @#!$%%”. In this question, the chatbot can also produce relevant answers and provide knowledge documents from the citations used.

This matter shows that the chatbot can understand context questions well even though the words used are incomplete and the sentences are not properly arranged, containing unnecessary symbols.

3.6. Accuracy Test Scenario Answer

To assess the accuracy of the chatbot's responses, a series of test scenarios were designed to compare the generated answers with the expected reference documents. These tests aimed to ensure that the chatbot could provide valid answers along with the correct source citations. The details of the accuracy evaluation are summarized in Table 4.

Table 4. Accuracy Test Scenario Answer

No	Test Scenario	Expected output	The output issued	Results
1	Explain what is meant by money circulation	PDF file 3_Money_Circulation_and _Factors_That_Influence_It_Indo	PDF file 3_Money_In _Circulation_and _Factors_That _Influence_It _Indo and 02_Analytical _Balance_of _General_Banks _and_BPRs	Valid
2	When is the publication time for Community Savings data?	PDF file 04_Community_Savings	PDF File 04_Public _Savings, 3_Money_In _Circulation _and_Factors _That_Influence _It_Indo and 02_Analytical _Balance_of _General_Banks _and_BPRs	Valid
3	What are the tenor classifications of Monetary Operation instruments?	PDF file 7_Monetary _Operations_DPM-SEKI_Rev_2016	PDF file 7_Monetary _Operations _DPM-SEKI_Rev_2016	Valid

The final functionality test is to perform an accuracy test answer, in this test scenario expected chatbot can give the right answer along with the source knowledge that is quoted. As can be seen in Table 4 Accuracy Test Scenario Answer, there are 3 questions compiled by the Author. For do a trial along with the expected citation file. Answer results chatbot displays conclusion relevant answers and correct citation files by the expected scenario. This proves that Retrieval and Generation performance carried out by chatbots function with good and have accurate valid answers.

3.7. Implementation

Implementing Azure Bot on a website provides a solution that utilizes ability intelligence artificial For increase the experience of the user in a direct way. Azure Bot Service provides infrastructure that enables the development, implementation, and management of chatbots with diverse capabilities. This website-based chatbot integration utilizes Azure Cloud Service. Website-based chatbot interface compatible with the browser in general used such as Google Chrome, Edge, and Mozilla Firefox. Website as an implementation platform gives high accessibility, making it easy for users to interact with the bot without the need to download application additional. Azure Bot Service supports integration through a website-based chat channel that can embedded in a website page using JavaScript, making it flexible For various types of websites, good For service customers, information portals, and support technical. Once the deployment is complete, the chatbot can accessed with URL app-backend-xy3dhrzksbr3c.azurewebsites.net, The Chatbot Homepage view that has been made can seen in Figure 5 Chatbot Homepage.

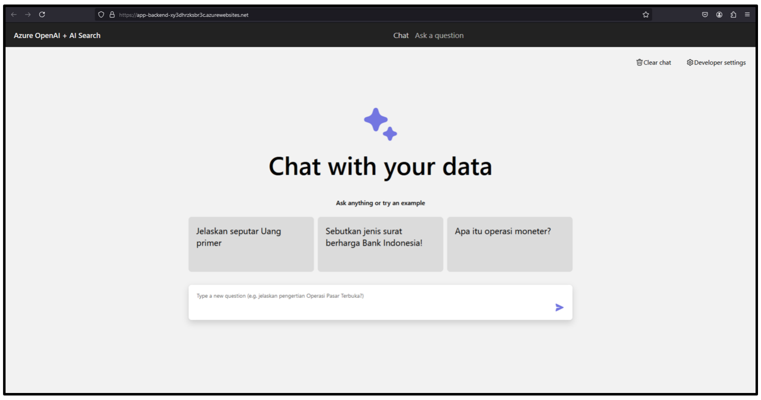

Figure 5. Chatbot Homepage

Figure 5 shows the chatbot homepage after deployment, where users can directly interact with SEKI data through a simple and accessible web-based interface. This interface demonstrates the successful integration of Azure Bot Service into a real-time application.

4. MANAGERIAL IMPLICATIONS

The integration of RAG-based chatbots using NLP and LLM in the banking sector offers significant managerial implications. By automating the retrieval and generation of business-specific information, it enhances operational efficiency, reducing the time developers spend searching for relevant data, leading to faster project delivery. This can also result in cost savings, as the need for manual searches and administrative support is minimized. The chatbot enables improved decision-making by providing managers and developers with fast and accurate responses, ensuring decisions are based on up-to-date data. Furthermore, the chatbot's scalability allows for easy adaptation to handle increasing data and more complex queries, facilitating future growth without a proportional rise in operational costs. Banks adopting AI-powered chatbots gain a competitive advantage by speeding up software development, staying ahead in an evolving market, and improving employee productivity by allowing staff to focus on higher-value tasks. Additionally, the seamless integration of the chatbot with existing systems ensures that banks can enhance their infrastructure without disrupting established workflows. Overall, the use of AI technologies positions banks to be more efficient, cost-effective, and competitive, benefiting both operational and strategic aspects of management.

5. CONCLUSION

This study indicates that the implementation of AI based chatbots using NLP and LLM technologies can significantly enhance efficiency in supporting project development processes in the banking sector. The chatbot, developed using the RAG approach, has proven capable of delivering fast and accurate responses to queries submitted by software developers. This effectiveness is achieved through the utilization of a domain-specific knowledge base tailored to the banking industry, which helps overcome the limitations of manual searches related to internal terminology, abbreviations, and processes typically known only within the organization.

Trial results demonstrate that the chatbot, deployed via Microsoft Azure, performs reliably across various browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It is able to suggest relevant questions to assist users in composing inputs, understand incomplete or symbol-laden queries, and deliver responses that are both accurate and contextually appropriate. The chatbot's ability to cite relevant knowledge documents and respond meaningfully has been validated. With its contextual understanding, the chatbot holds the potential to improve developer productivity by reducing time spent on unproductive manual searches, allowing them to focus more on complex tasks in banking software development projects.

For further development, several recommendations are proposed. First, continuously updating the knowledge base with current documents and collaborating with business and compliance teams will ensure answer accuracy and regulatory alignment. Second, enhancing contextual learning through fine-tuned datasets reflective of banking language can help the chatbot handle more complex inquiries. Third, deeper integration with internal company systems will allow access to real-time and updated information. Periodic performance evaluations, interactive interface enhancements with visual aids or step-by-step guidance, and strengthening data privacy and security through encryption, audits, and access controls are essential to ensure the chatbot remains adaptive, user-friendly, and compliant with banking standards.

6. DECLARATIONS

6.1. About Authors

Cindy Aurora (CA) -

Tuga Mauritsius (TM) https://orcid.org/0000-0002-5525-5591

6.2. Author Contributions

Conceptualization: CA; Methodology: TM; Software: CA; Validation: TM and CA; Formal Analysis: TM and CA; Investigation: CA; Resources: TM; Data Curation: CA; Writing Original Draft Preparation: TM; Writing Review and Editing: CA; Visualization: TM; All authors, CA and TM, have read and agreed to the published version of the manuscript.

6.3. Data Availability Statement

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

6.4. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.5. Declaration of Conflicting Interest

The authors declare that they have no conflicts of interest, known competing financial interests, or personal relationships that could have influenced the work reported in this paper.

REFERENCES

[1] M. R. Haque and S. Rubya, "An overview of chatbot-based mobile mental health apps: insights from app description and user reviews," *JMIR mHealth and uHealth*, vol. 11, no. 1, p. e44838, 2023.

[2] M. A. Kuhail, I. Taj, S. Alimamy, and B. Abu Shawar, "A review on polyadic chatbots: trends, challenges, and future research directions," *Knowledge and Information Systems*, vol. 67, no. 1, pp. 109–165, 2025.

[3] B. Zarouali, T. Araujo, J. Ohme, and C. De Vreese, "Comparing chatbots and online surveys for (longitudinal) data collection: an investigation of response characteristics, data quality, and user evaluation," *Communication Methods and Measures*, vol. 18, no. 1, pp. 72–91, 2024.

[4] R. Regin, S. S. Rajest, T. Shynu *et al.*, "An automated conversation system using natural language processing (nlp) chatbot in python," *Central Asian Journal of Medical and Natural Science*, vol. 3, no. 4, pp. 314–336, 2022.

[5] A. Khamaj, "Ai-enhanced chatbot for improving healthcare usability and accessibility for older adults," *Alexandria Engineering Journal*, vol. 116, pp. 202–213, 2025.

[6] U. Rusilowati, F. P. Oganda, R. Rahardja, T. Nurtino, and E. Aimee, "Innovation in smart marketing: The role of technopreneurs in driving educational improvement," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, no. 3, pp. 305–318, 2023.

[7] S. M. Abdullah, A. M. Al-Bakry, and A. K. Farhan, "Conversational health bots for telemedicine services: Survey," *Iraqi Journal for Computers and Informatics*, vol. 50, no. 2, pp. 156–172, 2024.

[8] B. of Indonesia. (2024) Tujuan kebijakan moneter. Accessed: Jan. 01, 2024. [Online]. Available: https://www.bi.go.id/id/fungsi-utama/moneter/Default.aspx
[9] ——. (2024) Profil bank indonesia. Accessed: Jan. 01, 2024. [Online]. Available: https://www.bi.go.id/id/tentang-bi/profil/Default.aspx
[10] ——. (2024) Metadata bank indonesia. Accessed: Feb. 01, 2024. [Online]. Available: https://www.bi.go.id/id/statistik/metadata/default.aspx
[11] L. Benaddi, C. Ouaddi, A. Jakimi, and B. Ouchao, "A systematic review of chatbots: Classification, development, and their impact on tourism," *IEEE access*, vol. 12, pp. 78 799–78 810, 2024.
[12] A. T. Neumann, Y. Yin, S. Sowe, S. Decker, and M. Jarke, "An llm-driven chatbot in higher education for databases and information systems," *IEEE Transactions on Education*, 2024.
[13] C. Borek, "Comparative evaluation of llm-based approaches to chatbot creation," 2024.
[14] A. A. Search. (2024) Azure ai search documentation. Accessed: Jul. 16, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/search/
[15] S. Purnama, C. S. Bangun, and E. P. Mahadewi, "Predicting consumer purchase intention in personal shopper services using big data analytics and sem," *International Journal of Cyber and IT Service Management*, vol. 5, no. 1, pp. 105–119, 2025.
[16] R. Hartati and E. B. Manullang, "Implementation of telegram chatbot ai with natural language processing (nlp) in learning creative entrepreneurship to develop students' creative and innovative competence," in *Talenta Conference Series: Local Wisdom, Social, and Arts (LWSA)*, vol. 7, no. 2, 2024, pp. 72–79.
[17] A. O. Service. (2024) Azure openai service documentation. Accessed: Jul. 16, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/openai/
[18] A. Giuliani, R. Savona, S. Carta, G. Addari, and A. S. Podda, "Corporate risk stratification through an interpretable autoencoder-based model," *Computers & Operations Research*, vol. 174, p. 106884, 2025.
[19] S. Abror, M. Mutrofin, E. Hardinanto, M. Mintarsih *et al.*, "Reimagining teacher professional development to link theory and practice," *Journal of Teaching and Learning*, vol. 1, no. 1, pp. 22–36, 2024.
[20] R. A. Sekarwati, A. Sururi, R. Rakhmat, M. Arifin, and A. Wibowo, "Survey of chatbot testing methods on social media to measure accuracy," *Sisfotenika*, vol. 11, no. 2, pp. 172–182, 2021.
[21] E. J. Gordon, J. Gacki-Smith, M. J. Gooden, P. Waite, R. Yacat, Z. R. Abubakari, D. Duquette, A. Agrawal, J. Friedewald, S. K. Savage *et al.*, "Development of a culturally targeted chatbot to inform living kidney donor candidates of african ancestry about apol1 genetic testing: a mixed methods study," *Journal of Community Genetics*, vol. 15, no. 2, pp. 205–216, 2024.
[22] A. C. Griffin, S. Khairat, S. C. Bailey, and A. E. Chung, "A chatbot for hypertension self-management support: user-centered design, development, and usability testing," *JAMIA open*, vol. 6, no. 3, p. ooad073, 2023.
[23] E. A. Yam, E. Namukonda, T. McClair, S. Souidi, N. Chelwa, N. Muntalima, M. Mbizvo, and B. Bellows, "Developing and testing a chatbot to integrate hiv education into family planning clinic waiting areas in lusaka, zambia," *Global Health: Science and Practice*, vol. 10, no. 5, 2022.
[24] K. Boyd, C. Potts, R. Bond, M. Mulvenna, T. Broderick, C. Burns, A. Bickerdike, M. Mctear, C. Kostenius, A. Vakaloudis *et al.*, "Usability testing and trust analysis of a mental health and wellbeing chatbot," in *Proceedings of the 33rd European Conference on Cognitive Ergonomics*, 2022, pp. 1–8.
[25] N. P. L. Santoso, R. Nurmala, and U. Rahardja, "Corporate leadership in the digital business era and its impact on economic development across global markets," *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, vol. 6, no. 2, pp. 188–195, 2025.
[26] K. Mageira, D. Pittou, A. Papasalouros, K. Kotis, P. Zangogianni, and A. Daradoumis, "Educational ai chatbots for content and language integrated learning," *Applied Sciences*, vol. 12, no. 7, p. 3239, 2022.
[27] A. BAIAI and N. F. BENNACER, "Developing a retrieval-augmented generation (rag) chatbot enhanced by knowledge graphs," Ph.D. dissertation, 2024.
[28] M. A. Quidwai and A. Lagana, "A rag chatbot for precision medicine of multiple myeloma," *MedRxiv*, pp. 2024–03, 2024.
[29] J. Siswanto, V. A. Goeltom, I. N. Hikam, E. A. Lisangan, and A. Fitriani, "Market trend analysis and data-based decision making in increasing business competitiveness," *Sundara Advanced Research on Artificial Intelligence*, vol. 1, no. 1, pp. 1–8, 2025.