Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

Designing a Chatbot Application Using the Flask Framework and Rule-Based Algorithm

Haeruddin[a], Sabariman[b], Vincent Su[c]
[a]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, haeruddin@uib.ac.id
[b]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, sabariman@uib.ac.id
[c]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, 2132016.vincent@uib.edu

Submitted: 13-01-2025, Reviewed: 16-01-2025, Accepted 22-01-2025
https://doi.org/10.47233/jteksis.v5i1.1820

Abstract
This research aims to develop a website based chatbot application implemented at PT. International Hardware Indo using the Flask framework and Rule-Based algorithm to improve customer service efficiency. The application is designed to providing quick answers to frequently asked questions, reducing the workload of customer service, and addressing operational time constraints. The chatbot development process follows the Waterfall methodology, which includes requirements analysis, design, implementation, testing, and maintenance. The Rule-Based algorithm is employed to map user input to the company dataset through logical "IF-THEN" rules. The research results demonstrate a chatbot accuracy rate of 90%, as measured through black box testing, despite limitations in handling input outside the trained dataset. The chatbot is designed with hybrid navigation, a simple interface, and cross-platform device compatibility. This research contributes to the development of chatbot based information systems that are widely accessible to customers, thereby enhancing user experience and improving operational efficiency for the company.
Keywords: *Chatbot, Flask, Rule-Based, Waterfall, Customer Service*

INTRODUCTION

With the advancement of time, technology has undergone rapid development due to the ease and speed of access to information [1]. One aspect that has seen significant growth in the modern era is artificial intelligence. Artificial intelligence enables machines to think and make decisions independently, including in the form of chatbot technology. The advancement of chatbot technology has been widely applied by various companies as a support for customer information services. A chatbot is an application designed to facilitate interaction between humans and machines [2].

Chatbots, or conversations with robots, are artificial intelligence applications capable of simulating intelligent conversations between humans based on their knowledge. As intelligent agents, chatbots can mimic human communication capabilities through text messages [3]. Chatbots are now widely used in various companies, such as in the banking sector. They function to assist in answering related questions, including banking information and handling customer complaints.

In practice, chatbots can be built manually or using conversation applications that provide services to develop chatbots. To understand and answer questions in a conversation, chatbots require specific algorithms, commonly referred to as Natural Language Processing (NLP) methods [4]. NLP, which is often abbreviated, is a field of study combining computing, artificial intelligence, and linguistics, focusing on the interaction between computers and natural human languages. Natural language refers to languages that humans can understand, including understanding implied meanings or semantics [5]. NLP can also handle algorithms to analyze, model, and understand human language and the semantics of their sentences [6]. Algorithms used to implement semantics include Rule-Based algorithms, which summarize questions and determine answers [7].

Rule-Based is an algorithm that applies artificial intelligence techniques to solve problems based on a series of predefined rules in the form of IF-THEN statements [8]. This algorithm requires gathering relevant facts or datasets. The IF rule describes a specific condition or state, while the THEN rule describes the action or result that must be accepted [9]. With these two rules, chatbots become easier and faster to build, as they do not require the training process typical of NLP algorithm approaches.

There have been previous studies related to the development of chatbot applications, such as chatbots for library services based on User-Centered Design using the Google Design Sprint method [10]. This research implemented Artificial Intelligence features in libraries by installing chatbots for customer service tasks using the Manychat platform. The features were adjusted to Frequently Asked Questions (FAQs) documented in the library.

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis
E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

However, Manychat had platform limitations, and the resulting FAQ accuracy was relatively basic.

Previous research has also explored chatbots in the context of academic services [11]. Chatbots were designed using several AWS services, such as AWS Lambda, Amazon Lex, Amazon DynamoDB, Amazon API Gateway, Amazon CloudFront, and Amazon S3. The chatbot implementation involved configurations in Amazon Lex and integration with Twilio. However, users became dependent on third-party services, namely AWS and Twilio, and faced relatively high operational costs due to the use of AWS services.

Further research explored machine learning-based chatbot development using the StarSpace algorithm and Count Vectorizer [12]. Chatbots were built using PHP and Python programming languages through the Telegram application. The chatbots were designed to answer simple questions from prospective students, while complex questions were forwarded to customer support. However, these chatbots required large datasets, consuming significant time, and their accuracy depended heavily on data training.

Some previous studies have implemented chatbot applications for customer service, such as the Milki Bot, a chatbot for small and medium enterprises marketing dairy products [13], and Alitta, a chatbot used as an automated response service at the Sweetener and Fiber Plants Research Center [14]. These applications used a forward-chaining method with numerous rules for every possible user action. This made the chatbots less flexible for more complex business models, as additional rules extended the system's complexity. Besides forward-chaining methods, chatbots can also be developed using Artificial Intelligence Markup Language (AIML). AIML is a type of XML designed specifically for storing question-and-answer patterns. The question patterns are collected and paired with possible answers [15]. Based on previous studies [16], [17], [18], the use of AIML has limitations, such as difficulty in handling complex questions due to the need for users to manually build the conversation flow.

Differing from previous research, this research aims to develop a chatbot application service using the Flask framework and a Rule-Based algorithm applied to the PT. IHI website, which specializes in distributing high-quality industrial equipment for marine, mining, oil and gas, and technical tools [19]. This chatbot is expected to accelerate the information service process but address the issue of human delay caused by limited customer service availability. When multiple customers inquire about products simultaneously during office hours, the time to address questions is constrained. The chatbot is envisioned to offer quick responses without requiring information skimming or scanning [20], thereby reducing the workload of PT. IHI customer service team.

RESEARCH METHODOLOGY

2.1 Research Stages

The Waterfall diagram method is used to develop an application requiring clear and systematic planning [21], with minimal changes during the design process [22]. This method is divided into sequential stages, where each stage must be completed before proceeding to the next one, without skipping any steps. The Waterfall diagram for this research methodology is illustrated in Figure 1.

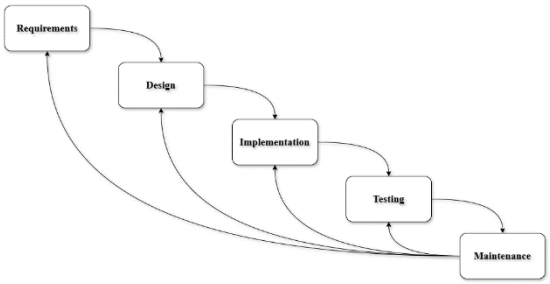


Figure 1. Waterfall Diagram

2.1.1 Requirements

The requirements stage involves analyzing the needs for creating a chatbot system for a company. The first step is identifying the domain and hosting of the website to connect the frontend chatbot system with the backend system. The second step involves gathering information through literature reviews related to chatbot case studies and similar topics, as well as observing websites that have integrated chatbots [23]. Lastly, data collection from the company is conducted to determine the type of information to be trained into the chatbot.

2.1.2 Design

The design stage involves creating a schematic or architecture and designing the features of the chatbot based on the requirements identified in the previous stage. This process involves several critical aspects. First aspect is designing the conversation flow between the chatbot and the user, including how the chatbot responds and provides answers based on the trained data. Second aspect is selecting the language for the chatbot, such as Indonesian, to make it easier for users to operate, and choosing font styles and colors to enhance the user experience. Third aspect is designing a user-friendly and informative chatbot interface that is easy to navigate. The resulting interface serves as a reference for coding the frontend and backend components of the chatbot application [24].

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

2.1.3 Implementation

The implementation stage involves developing the chatbot based on the design created, including designing the user interface and the chatbot's functionality [25]. The system is coded using the Flask framework as the foundation. The chatbot program contains datasets related to PT. IHI to provide quick responses to queries. The chatbot employs a hybrid navigation system, allowing users to interact by typing or clicking. This hybrid navigation system enhances flexibility during user interactions with the chatbot. The chatbot is implemented into the website using HTML, CSS, and Python programming.

2.1.4 Testing

The testing phase is the stage where chatbot application testing is carried out. The testing is conducted using black-box testing and system testing methods. Black-box testing is a method that focuses on the external functionality of the program without examining its internal structure [26]. This testing helps identify how effectively the chatbot interacts with users and delivers responses that meet expectations.

Additionally, system testing is performed to evaluate the performance of completed features and identify shortcomings in newly implemented features. This testing ensures that the completed chatbot is evaluated for its functionality. This stage examines how well the chatbot understands user questions or input and how accurately it provides responses. If errors are identified in the chatbot system during the testing process, the next step is to revert to the previous stage, namely the implementation stage, for evaluation and rectification of the failed processes. Conversely, if no errors are identified during the testing phase, the designed chatbot features will be implemented on the company's website [27].

2.1.5 Maintenance

The maintenance stage ensures the developed and deployed system undergoes continuous maintenance. This includes addressing bugs detected during black box testing or reported by users. The goal is to ensure the chatbot features remain updated and function well over time, providing real-time and valid data [28].

2.2 Framework Flask

Flask is a Python microframework designed for building websites. It does not require special tools or libraries, as it already provides libraries and code collections that allow users to avoid starting from scratch [29]. Flask is also flexible and supports custom extension development on top of its core framework, enabling seamless integration of additional features that feel native to Flask [30].

2.3 Python

Python is a high-level programming language designed to translate code instructions into executable application functions. Unlike many other programming languages, Python has a syntax resembling human language, making it easier to understand and use in programming [31].

2.4 Website

A website is a collection of web pages published on the internet with a domain or URL (Uniform Resource Locator) accessible to all internet users by typing the address. Websites are made possible by World Wide Web (WWW) technology. Typically, web pages are documents written in Hyper Text Markup Language (HTML) format and can be accessed via HTTP or HTTPS protocols. These protocols transmit information from the website server for display in web browsers [32].

2.5 Black Box Testing

Black box testing evaluates test cases designed based on specifications and focuses on the output generated as a response to selected inputs and execution conditions [33]. Using black box testing only requires identifying program errors by testing system functionality without needing to understand the system’s internal mechanisms or structure.

RESULT AND DISCUSSION

In this stage, the results obtained based on the previous methodology steps are explained, aiming to provide accurate answers and accelerate the overall service process, allowing users to save significant time in obtaining product specifications.

3.1 Requirements

At the requirements stage, as shown in Table 1, several essential files are needed to build a chatbot and a website. These files support the distribution of the collected data. Table 2 lists seven commands that have been trained with different functions. These commands were derived from data training or rules permitted by the company to be displayed by the chatbot.

Table 1. File Manager

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

No	Name	Description
1	app.py	Company bearing data file
2	ihi_project	Python, HTML, and CSS files
3	index.html	Company website data file
4	public_html	Collection of all company data files
5	style.css	File for font and color usage on the company website

Table 2. Rule-Based System

No	Command	Rules	Function
1	bot	-	Displaying greeting message by the bot
2	ihi	User input about company	Displaying information about the company
3	brand	User input bearing brands	Displaying available bearing brands in the company
4	bearing	User input bearing types	Displaying available bearing types in the company
5	size	User input bearing codes	Displaying the bearing catalog available in the company
6	contact	-	Displaying contact information in the company
7	bye	-	Displaying closing message

3.2 Design

In the design stage, the target usage of this chatbot is identified as the customer service of PT. IHI. The primary goal is to enhance response speed to customers and reduce unanswered questions. The chatbot uses a website that allows users to type or select available question options. The design includes a simple interface featuring the hex code #B22222 (firebrick) and #F1F1F1 (white smoke). This chatbot is trained and designed in Indonesian.

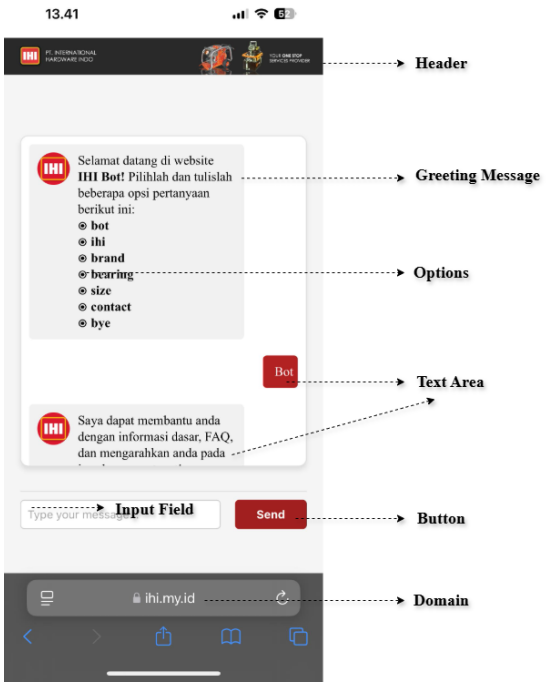


Figure 2. Design Interface of Chatbot

As depicted in Figure 2, the initial interface design when entering the website begins with a **Header** at the top, featuring the company's name or logo, "PT. International Hardware Indo (IHI)," and the company tagline for primary identification. Below that is a **Greeting Message** displayed by the chatbot to users. Then, a list of question **Options** is provided to help users navigate. Users can select one of the available categories for more information, such as bot, ihi, brand, bearing, size, contact, and bye. The interface also includes a **Text Area** showing the chatbot responses to user input and a history of the user's conversation with the chatbot. The **Input Field** allows users to type questions or commands for the chatbot, and a **Button** is provided to send input. Finally, the bottom of the screen displays the website **Domain**, "ihi.my.id," indicating the site where the chatbot can be accessed.

To understand the chatbot system, a flowchart explains how the chatbot operates from the user's initial interaction to the final response.

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis
E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

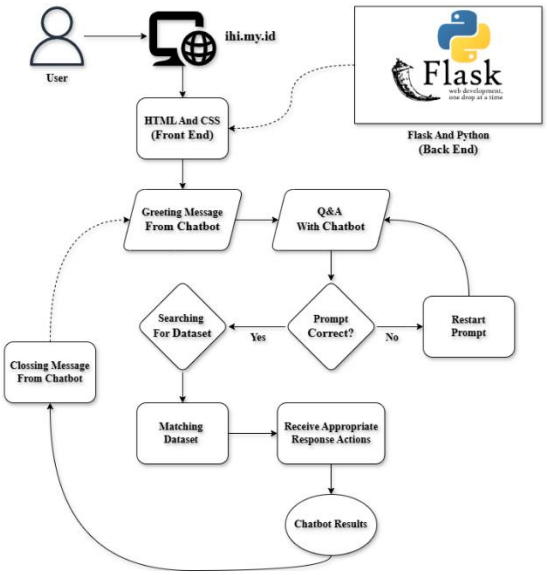


Figure 3. Flowchart of Chatbot

Based on Figure 3, the chatbot system flowchart begins with user interaction through the "ihi.my.id" website. Users will see the chatbot visual elements and interface built using frontend technologies like HTML and CSS, while the backend is developed with Flask to handle logic, data processing, and response management. Flask enables the chatbot to interact with datasets and generate answers based on user input. When users access the page, the chatbot greets them with a greeting message to initiate interaction. Users can type questions or commands in the chat field. The chatbot then verifies whether the input is valid. If no matching keywords are found, the chatbot prompts users to re-enter their query. If keywords are found, the chatbot searches for relevant data in the dataset to match the user query with the available answers. The chatbot then generates an appropriate response based on the input and retrieved data, displaying it to the user. Once the interaction ends, the chatbot provides a closing message to conclude the conversation.

3.3 Implementation

During implementation, the chatbot is integrated into the website through three main components such as the frontend user interface, visual styles (CSS), and the backend powered by Flask. Each component plays a specific role in supporting the overall system. The frontend is designed using HTML to structure the chatbot interface, with JavaScript managing dynamic message sending. Figure 4 shows the onsubmit event handler used in the form. This function employs fetch with the POST method to send data to the backend. After the message is sent, the function updates the interface with the latest messages and clears the input box.

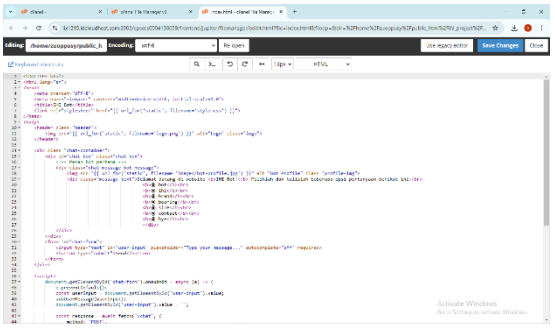


Figure 4. HTML Script

The backend system is implemented using the Flask framework to manage chatbot logic and data. Key elements of the system include:

1. FAQ Data Management
   The chatbot question and answer data are stored in a Python dictionary called faq. This data includes information ranging from chatbot introductions to technical details such as brand, type, and size of bearing products.
2. Communication Endpoint
   Flask provides an endpoint (/chat) to receive and process requests from the frontend. Each message received is matched with the data in the faq dictionary. Appropriate answers are then returned in JSON format for display on the user interface. Figure 5 demonstrates how chatbot datasets are written using the Rule-Based algorithm.

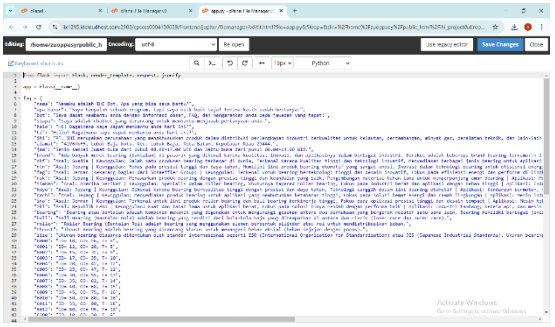


Figure 5. Dataset Training

To create a professional and user-friendly appearance, the chatbot interface style is defined using CSS. Layout adjustments are made with flexbox to organize elements vertically. Padding is added between the header and chatbot to improve readability. Aesthetic enhancements include chat boxes styled with white background, shadows, and borders to highlight the elements. Rounded corners are applied using the border-radius property to achieve a modern look. The design differentiates the visual style of user and chatbot responses to make them easily distinguishable. Figure 6 shows the elements used by the chatbot.

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

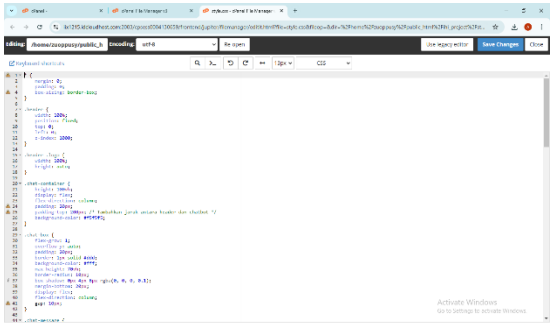


Figure 6. CSS Script

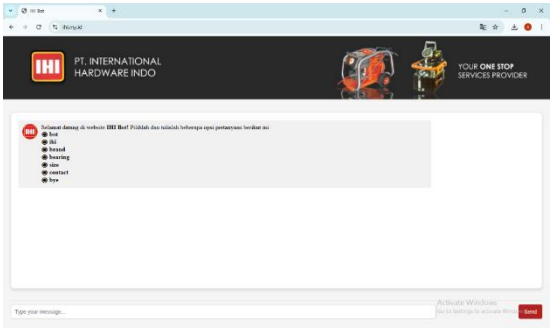
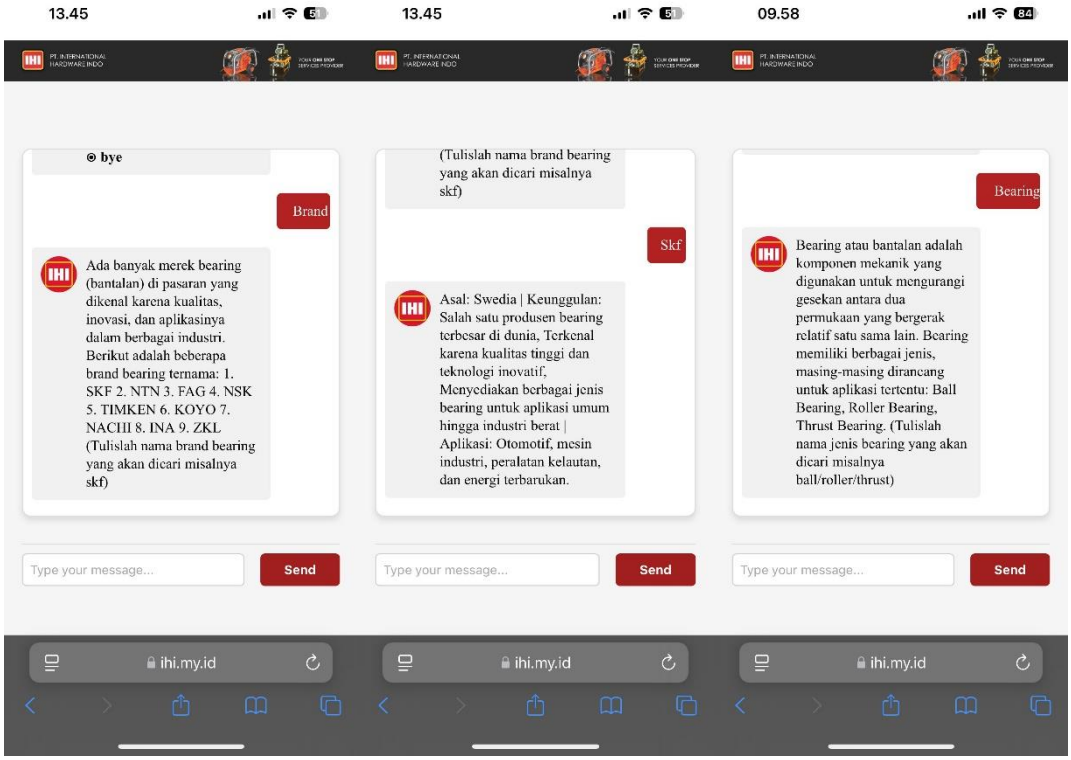


Figure 7. Chatbot Interface Screen

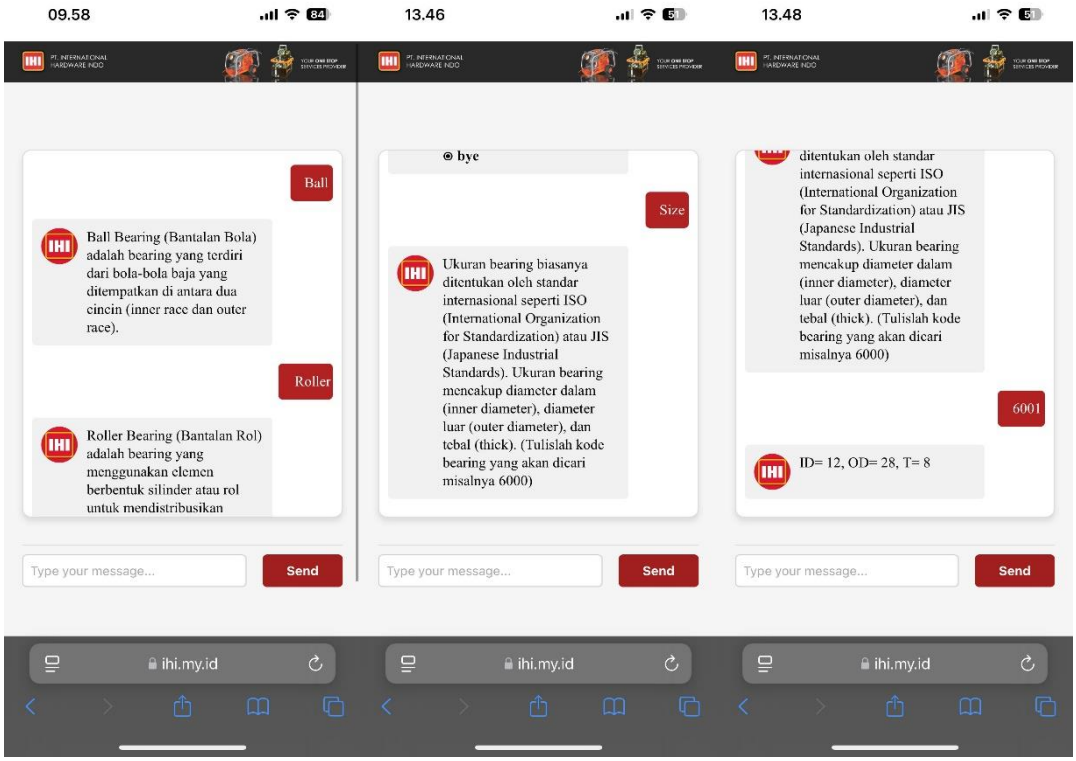
Figure 7 illustrates the chatbot initial message when users access the PT. IHI website. The chatbot greets users with, "Selamat datang di website IHI Bot! Pilihlah dan tulislah beberapa opsi pertanyaan berikut ini:". The options are equipped with keywords, and the chatbot responds based on the trained data or rules.



(a) (b) (c)

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**



(d) (e) (f)

Figure 8. Chat Session with Chatbot

Based on Figure 8, images a and b show an example conversation where the user selects the "brand" option. The chatbot displays nine bearing brands based on the company sales data. Images c and d show the user selecting the "bearing" option, explaining what a bearing is and its general types. Images e and f depict the user selecting the "size" option, where the chatbot provides information on bearing codes available in the company's catalog, detailing the three main components: inner diameter, outer diameter, and thickness.

3.4 Testing

In the testing stage, which employs black box testing, several steps are observed. First, prepare the testing requirements, namely the IHI Bot chatbot application website. Second, use predefined inputs for testing, consisting of keyword rules designed for the chatbot. Third, define the expected responses as benchmarks to evaluate the chatbot performance. The test results are obtained as a success rate percentage.

Table 3. Results of Black Box Testing

No	Scenario	Result	Output
1	The user accesses the website using a domain *ihi.my.id*	The chatbot responds by displaying a greeting message	Success
2	The user types a question with the keyword *bot*	The chatbot responds by explaining the chatbot's purpose	Success
3	The user enters a question about the company with the keyword *ihi*	The chatbot responds by displaying several question options	Success
4	The user enters a question about bearing brands with the keyword *brand*	The chatbot responds by displaying options for related questions	Success
5	The user enters a question about bearing types with the keyword *bearing*	The chatbot responds by displaying options for related questions	Success
6	The user enters a question about bearing codes with the keyword *size*	The chatbot responds by displaying options for related questions	Success

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

7	The user enters a bearing code that is not in the catalog	The chatbot responds with a message indicating it doesn't understand	Failed
8	The user types a question with the keyword *contact*	The chatbot responds by displaying the company contact information	Success
9	The user types a question with the keyword *bye*	The chatbot responds by displaying a farewell message	Success
10	The user attempts to reset the chatbot by refreshing the website	The chatbot responds by displaying the greeting message again	Success

As shown in Table 3, testing was conducted 10 times with different scenarios. The results showed 9 successful scenarios and 1 failed scenario, meaning the chatbot achieved a 90% accuracy rate. It was noted that answer accuracy heavily depends on the data training or rules created. Using keywords outside the rules causes the chatbot to fail to provide an answer. Adding more rules is highly recommended to broaden the chatbot response capabilities. Nonetheless, the chatbot performance is excellent, responding in under 3 seconds, and its website-based implementation ensures accessibility across any device.

3.5 Maintenance

In the maintenance stage, improvements are made based on issues detected during testing and user interactions with the chatbot. When issues arise, developers thoroughly analyze and identify the root cause. However, one limitation remains: failure scenarios when users input item codes not registered in the system cannot be addressed during maintenance. This is because the chatbot is designed using trained data or rules based on available company items. Another issue was found where questions could only be processed if typed in uppercase letters. This problem was immediately fixed, and the system now accepts lowercase inputs. Figure 9 shows an example of user input.

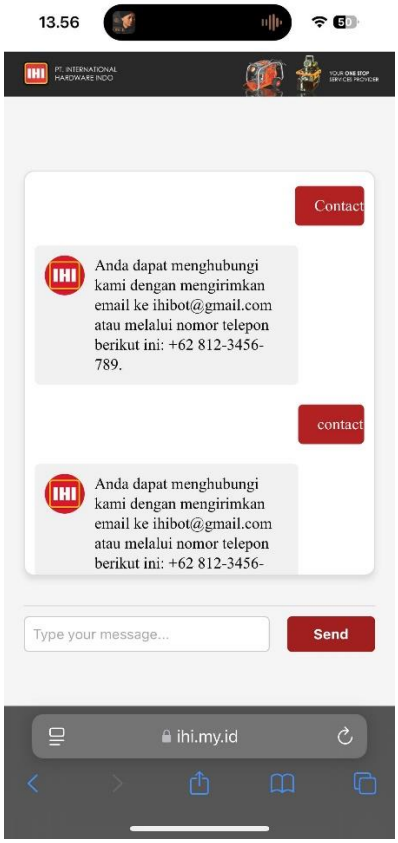


Figure 9. Uppercase and Lowercase Display

CONCLUSION

This research successfully developed a Flask-based chatbot using the Rule-Based algorithm, providing quick responses to customer queries. The application is designed to address the limitations of PT. IHI customer service, particularly in handling inquiries outside of business hours. By implementing a user-friendly website interface with hybrid navigation, the chatbot effectively improves the company's information service efficiency. Black box testing demonstrated a 90% success rate, indicating adequate performance with an average response time of under 3 seconds. However, the chatbot has limitations in processing inputs outside the trained rules and requires improvements to handle user input flexibility better.

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 133-142
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

In conclusion, this Flask Python and Rule-Based chatbot significantly contributes to supporting PT. IHI operations. Continuous maintenance and dataset expansion will enhance the chatbot service scope, accommodating more complex customer inquiries. This research also demonstrates the potential for further development and application in other companies with similar needs.

REFERENCES

[1] R. Parlika, S. I. Pradika, A. M. Hakim, and K. R. N. Manab, "Bot Whatsapp Sebagai Pemberi Data Statistik COVID-19 Menggunakan PHP, Flask, Dan MySQL," *J. Inform. dan Sist. Inf.*, vol. 1, no. 2 SE-Articles, pp. 282–293, 2020, [Online]. Available: http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/101

[2] R. Z. Ramadhani, H. Rusdianto, and V. Yahya, "Rancang Bangun Aplikasi Pusat Informasi Sekolah Dengan Penerapan Chatbot Menggunakan Aiml Berbasis Android Pada Smk Otomotif Al Husna Tangerang," *JIKA (Jurnal Inform.*, vol. 3, no. 2, pp. 27–33, 2019, doi: 10.31000/jika.v3i2.2076.

[3] S. Nurhayati and M. A. H, "Pembangunan Aplikasi Chatbot Midwify sebagai Media Pendukung Pembelajaran Ilmu Kebidanan Berbasis Android di Stikes Bhakti Kencana Bandung," *Komputika J. Sist. Komput.*, vol. 8, no. 1, pp. 45–52, 2019, doi: 10.34010/komputika.v8i1.1630.

[4] A. Kedia and M. Rasu, *Hands-On Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications*. Packt Publishing Ltd, 2020.

[5] N. K. Wangsanegara and B. Subaeki, "Implementasi Natural Language Processing Dalam Pengukuran Ketepatan Ejaan Yang Disempurnakan (Eyd) Pada Abstrak Skripsi Menggunakan Algoritma Fuzzy Logic," *J. Tek. Inform.*, vol. 8, no. 2, 2015, doi: 10.15408/jti.v8i2.3185.

[6] S. Vajjala, B. Majumder, A. Gupta, and H. Surana, *Practical natural language processing: a comprehensive guide to building real-world NLP systems*. O'Reilly Media, 2020.

[7] F. Ishlakhuddin and A. SN, "Ontology-based Chatbot to Support Monitoring of Server Performance and Security By Rule-base," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 2, p. 131, 2021, doi: 10.22146/ijccs.58588.

[8] F. Saputra, R. M. Handoko, W. Putra, R. Priskila, and V. H. Pranatawijaya, "Chatbot Berbasis Whatsapp Teknik Informatika Universitas Palangkaraya: Rules Based System," *J. Teknol. Inform. dan Komput.*, vol. 10, no. 1, pp. 296–308, 2024, doi: 10.37012/jtik.v10i1.2111.

[9] M. I. Ghozali, A. A. Riadi, D. A. Putra, and W. H. Sugiharto, "Pengembangan Sistem Sortir Otomatis untuk Jeruk Citrus: Integrasi Teknologi Sensor dan Algoritma Rule-Based," vol. 4, no. 3, pp. 241–248, 2024, [Online]. Available: https://djournals.com/resolusi

[10] N. I. Wijayanti, R. Yulianti, and B. Wijaya, "Perancangan Chat Bot Messenger Dengan Pendekatan User Centered Design (Studi Kasus : Perpustakaan Fakultas Teknik Universitas UGM)," *Media Pustak.*, vol. 26, no. 4, pp. 254–267, 2019.

[11] A. Lubis and I. Sumartono, "Implementasi Layanan Akademik Berbasis Chatbot untuk Meningkatkan Interaksi Mahasiswa," *Media Online*, vol. 3, no. 5, pp. 397–403, 2023.

[12] L. Hakim, S. Gustina, S. F. Putri, and S. U. Faudiah, "Perancangan Chatbot di Universitas Proklamasi 45," *Edumatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 91–100, 2020, doi: 10.29408/edumatic.v4i1.2157.

[13] A. Dwi R, F. Imamah, Y. M. Andre S, and Andriansyah, "Aplikasi chatbot (milki bot) yang terintegrasi dengan web CMS untuk customer service pada UKM minsu," *J. Cendikia*, vol. XVI, pp. 2–2, 2018.

[14] Eka Yuniar and Heri Purnomo, "Implementasi Chatbot 'Alitta' Asisten Virtual Dari Balittas Sebagai Pusat Informasi Di Balittas," *Antivirus J. Ilm. Tek. Inform.*, vol. 13, no. 1, pp. 24–35, 2019, doi: 10.35457/antivirus.v13i1.714.

[15] R. S. Wallace, "El programa Artificial Linguistic Internet Computer Entity (A.L.I.C.E.).," *Parsing Turing Test*, pp. 181–210, 2009.

[16] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-Janua, no. September 2017, pp. 1525–1530, 2017, doi: 10.1109/ICACCI.2017.8126057.

[17] B. Rusmarasy, B. Priyambadha, and F. Pradana, "Pengembangan Chat Bot pada CoMa untuk Memberikan Motivasi Kepada Pengguna Menggunakan AIML," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 5, pp. 4484–4490, 2019.

[18] D. Suryani and E. L. Amalia, "Aplikasi Chatbot Objek Wisata Jawa Timur," *Smartics J.*, vol. 3, no. 2, pp. 47–54, 2017.

[19] I. IHI, "Home Page." Accessed: Nov. 25, 2024. [Online]. Available: https://www.ihi.co.id/

[20] V. R. Prasetyo, N. Benarkah, and V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya," *Teknika*, vol. 10, no. 2, pp. 114–121, 2021, doi: 10.34148/teknika.v10i2.370.

[21] Elita Natalia Sugianto, Jessica Aurelia Sujangga, N. Delvia, Verdiana Ayustika, and Agus Cahyo Nugroho, "Pengembangan Chatbot 'Ciovita' Virtual Assistant Cioccolato Brownie Semarang Dengan Metode Waterfall," *J. Appl. Comput. Sci. Technol.*, vol. 3, no. 2, pp. 179–185, 2022, doi: 10.52158/jacost.v3i2.348.

[22] F. Supandi and M. Sudir, "Analisis Resiko Pada Pengembangan Perangkat Lunak Yang Menggunakan Metode Waterfall Dan Prototyping," in *Seri Prosiding Seminar Nasional Dinamika Informatika*, 2019.

[23] A. Yulianto *et al.*, "Development of an Integrated Chatbot on the Website Using IBM Watson Assistant," pp. 217–232.

[24] A. El Hayat Soumiya and M. Bahaj, "Converting UML class diagrams into temporal object relational database," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 5, pp. 2823–2832, 2017, doi: 10.11591/ijece.v7i5.pp2823-2832.

[25] D. Christianto, E. Siswanto, and R. Chaniago, "Penggunaan Named Entity Recognition dan Artificial Intelligence Markup Language untuk Penerapan Chatbot Berbasis Teks," *J. Telemat.*, vol. 10, no. 2, pp. 61–68, 2016, doi: 10.61769/telematika.v10i2.130.

[26] B. Gunawan and M. Sidik, "Pembuatan Aplikasi Chatbot Kolektor Dengan Metode Extreme Programming Dan Strategi Forward Chaining," *J. Int.*, vol. 8, no. 2, pp. 293–302, 2021, doi: 10.25126/jtiik.202184298.

[27] D. R. Lakshmi and S. S. Mallika, "A review on web application testing and its current research directions," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 4, pp. 2132–2141, 2017, doi: 10.11591/ijece.v7i4.pp2132-2141.

[28] D. K. Kim, "Development of mobile cloud applications using UML," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 1,

E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

pp. 596–604, 2018, doi: 10.11591/ijece.v8i1.pp596-604.

[29] P. Vogel, T. Klooster, V. Andrikopoulos, and M. Lungu, "A Low-Effort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services," *Proc. - 2017 IEEE Work. Conf. Softw. Vis. Viss. 2017*, vol. 2017-Octob, pp. 109–113, 2017, doi: 10.1109/VISSOFT.2017.13.

[30] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," *Metrop. Univ. Appl. Sci.*, no. May, pp. 13–33, 2020.

[31] A. Jolie, D. Dedrick, R. K. Sugeng, W. A. Lee, and A. Yulianto, "Aplikasi Sistem Manajemen Perpustakaan dengan Penerapan Pemrograman Berorientasi Objek," *Telcomatics*, vol. 7, no. 2, pp. 61–69, 2022, doi: 10.37253/telcomatics.v7i2.7349.

[32] . N., A. Ibrahim, and A. Ambarita, "Sistem Informasi Pengaduan Pelanggan Air Berbasis Website Pada Pdam Kota Ternate," *IJIS - Indones. J. Inf. Syst.*, vol. 3, no. 1, p. 10, 2018, doi: 10.36549/ijis.v3i1.37.

[33] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012, doi: 10.5121/ijesa.2012.2204.