Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 7 No. 1 Januari 2025 Hal. 143-149
http://jurnal.unidha.ac.id/index.php/jteksis

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

Developing a Chatbot System for PT. NG Tech Supplies based on the Python Flask Framework

Nimatul Mamuriyah[a], Haeruddin[b],Jackson[c]
[a]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, nimatul@uib.ac.id
[b]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, haeruddin@uib.ac.id
[c]Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, 2132013.jackson@uib.edu

Submitted: 13-01-2025, Reviewed: 16-01-2025, Accepted 22-01-2025
https://doi.org/10.47233/jteksis.v7i1.1821

Abstract
This study focuses on developing and implementing a chatbot system using the Python Flask framework to enhance customer service efficiency at PT. NG Tech Supplies. The chatbot is designed to provide fast and accurate responses to customer inquiries, leveraging artificial intelligence and natural language processing technologies. The development follows a systematic approach using the Waterfall methodology, encompassing requirements analysis, design, implementation, verification, and maintenance. Key features include seamless integration with order management and CRM systems, enabling real-time updates on product and order status. Black-box testing was conducted to evaluate system performance, demonstrating an average response time of less than 2 seconds and high accuracy in handling queries. The chatbot also supports a well-organized folder structure in cPanel, ensuring efficient management and scalability. This research highlights the chatbot's potential to automate repetitive tasks, reduce workload, and improve customer satisfaction through enhanced service delivery.
Keywords: *Chatbot, Python Flask, CRM Integration, AI, Customer Service Automation*

INTRODUCTION

In the midst of the rapidly advancing digital era, efficiency in customer service has become a key to business success. Companies across various industries are striving to innovate and enhance their services to meet the evolving needs and expectations of customers. PT. NG Tech Supplies, a company engaged in the distribution of industrial components for the maritime and heavy industry sectors, recognizes the importance of innovation in maintaining competitiveness and providing the best possible service to its customers [1]. However, the company faces challenges in providing fast and efficient customer service due to the increasing volume of inquiries and requests. To address this challenge, PT. NG Tech Supplies has chosen to implement a chatbot as an innovative solution.

A chatbot is a computer program capable of communicating with humans using natural language, similar to human interaction. Chatbots can replace humans in various tasks, such as answering customer questions, providing product information, and processing orders [2]. Chatbots work by mimicking human conversation through text or voice, utilizing artificial intelligence (AI) and natural language processing (NLP) technologies [3], [4]. Some chatbot applications, such as customer service, call centers, and others, use Artificial Intelligence Markup Language (AIML) to communicate with users [5].

To create a reliable and efficient chatbot, PT. NG Tech Supplies has opted to use the Python Flask framework. Flask is known for its simplicity and flexibility in web application development [6]. By utilizing Flask, the chatbot development process becomes more efficient and allows for easy integration with the company's existing systems. Additionally, Flask supports various extensions that can add features to the application, such as object-relational mapping, form validation, and upload handling [7], [8], [9].

In a similar study titled "Pengembangan Virtual Assistant Chatbot Berbasis WhatsApp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan" (Development of a WhatsApp-Based Virtual Assistant Chatbot at the Student Information Service Center of the Institute of Education), the aim was to develop a chatbot system implemented on WhatsApp. The results showed that users felt greatly assisted by this system [10], leading the author to believe that implementing a chatbot for PT. NG Tech Supplies would be efficient. Another study titled "Prediksi Nilai Ujian Sekolah Siswa SMK Plus Padjadjaran Berbasis Web Menggunakan Jaringan Syaraf Tiruan Backpropagation" (Prediction of School Exam Scores for SMK Plus Padjadjaran Students Based on the Web Using Backpropagation Neural Networks) aimed to predict school exam scores based on existing sample scores [11], The results of this study showed that the web-based system worked well and successfully achieved its objectives, leading the author to choose a website as the platform for implementing the chatbot. In the next study titled "Impelementasi AI

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 1 Januari 2023 Hal. 36-41
http://jurnal.unidha.ac.id/index.php/jteksis

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

Chatbot Sebagai Support Assistant Website Universitas Nurul Jadid Menggunakan Algoritma BiLSTM" (Implementation of an AI Chatbot as a Support Assistant for the Nurul Jadid University Website Using the BiLSTM Algorithm), the aim was to create a virtual assistant to answer simple questions. The results of this study showed that the system worked well in reducing the workload of administrative staff [12]. Based on this research, the author chose a web-based virtual assistant chatbot with the expectation that it would help serve customers. The next related study is titled "Implementasi Natural Language Processing pada Chatbot Untuk Helpdesk Informasi Wisata (Studi kasus: Tangerang Raya)" (Implementation of Natural Language Processing on a Chatbot for Tourism Information Helpdesk (Case Study: Tangerang Raya)). This study used the Flask framework with the aim of improving business efficiency and productivity [13]. Therefore, the author used the Flask framework as a platform for developing a web-based chatbot. The following study titled "Perancangan Aplikasi Chatbot FAQ Berbasiskan Aplikasi Android" (Design of an Android-Based FAQ Chatbot Application) had a shortcoming in its UI design, which still needed to be developed [14]. Therefore, in this research, the author created a UI based on the desires and needs of PT. NG Tech Supplies.

This research aims to develop and implement a chatbot using the Python Flask framework to enhance customer service efficiency at PT. NG Tech Supplies. The chatbot is expected to automate responses to common inquiries, reduce the workload of the customer service team, and improve service speed and accuracy. By automating these tasks, the chatbot can provide a more responsive service and free human agents to focus on more complex and personalized customer interactions.

RESEARCH METHODOLOGY

2.1 Research Stages

This research employs the Waterfall diagram as the methodology for developing a chatbot. The Waterfall diagram is used to develop an application requiring sequential and systematic planning [15]. The research is divided into five sequential stages, where each stage must be completed before proceeding to the next without skipping any prior process [16]. The resulting Waterfall diagram for these research stages can be seen in Figure 1, as follows.

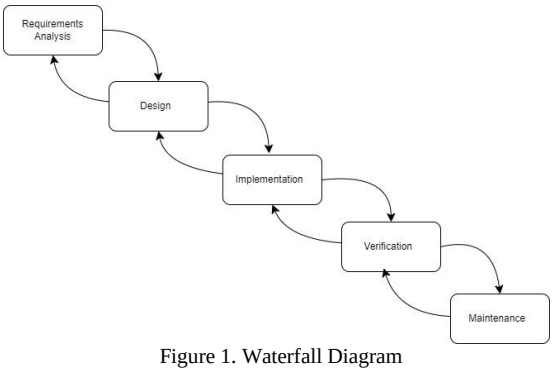

Figure 1. Waterfall Diagram

2.1.1 Requirements Analysis

The first stage in this research is the requirements analysis phase, which aims to understand the primary issues faced by PT. NG Tech Supplies in managing customer interactions. This analysis involves identifying specific needs related to customer interactions that can be improved through the use of chatbots, such as the most frequently asked questions by customers, desired response times, and customer preferences for interacting with customer service.

2.1.2 Design

The second stage begins with developing the chatbot using the Flask Python framework. The development process involves several steps, including designing conversation flows, developing the backend, and integrating with existing company systems such as Customer Relationship Management (CRM) and order management systems. Conversation flows are designed to ensure the chatbot can handle various types of customer inquiries with quick and accurate responses. The conversation flow includes common scenarios customers might face, such as inquiries about order status, product information, and technical support. The CRM and order management systems enable access to the latest information regarding order status and product details. This integration allows the chatbot to provide accurate and up-to-date information to customers in a short time. To understand the entire chatbot system, a flowchart diagram of the chatbot system design is presented below.

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 1 Januari 2023 Hal. 36-41
http://jurnal.unidha.ac.id/index.php/jteksis

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

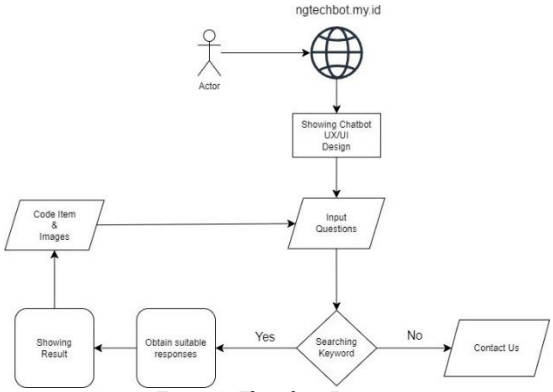

Figure 2. Flowchart Diagram

Based on Figure 2, the chatbot flow starts with users accessing the website via the internet. Users are directed to the UX/UI design interface of the chatbot provided on the website. After viewing the interface, users can input questions or queries into the chatbot. If keywords are found, the system searches for a corresponding response in the chatbot database or algorithm; otherwise, users are directed to the "Contact Us" option to directly contact the support team. Once the chatbot successfully responds, the system displays relevant answers or results to the users based on their input.

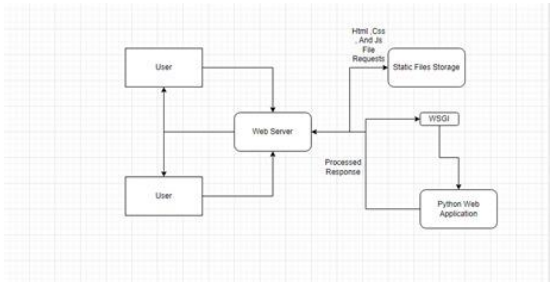

Figure 3. Architecture Diagram

Based on Figure 3, this diagram explains the interaction between users and the web-based application. The web server acts as an intermediary between users and the Python web application, receiving user requests and directing them to the web application for processing. Static files such as HTML, CSS, and JavaScript are stored here to support the user interface. Static file requests are served directly by the server without involving the Python web application. Additionally, WSGI is a protocol that connects the web server with the Python web application. WSGI ensures that user requests are properly forwarded to the Python application, and the generated response is returned to the web server. Then, the Python web application processes the business logic and generates a response based on the user's request. The result is sent back to the web server via WSGI. Once the web application completes the process, the response is processed and returned to the user through the web server.

2.1.3 Implementation

The third stage is the implementation phase, where the backend is developed using Flask to handle business logic and database interactions. PostgreSQL is used as the database to store customer data, conversation history, and other relevant information for chatbot interactions.

2.1.4 Verification

The fourth stage is the verification phase, which involves testing the developed chatbot to ensure its functionality aligns with the requirements. Testing is conducted using black box testing to identify any system errors or bugs. Black-box testing is a method that focuses on the software's functional requirements. Its purpose is to identify deficiencies in categories such as missing chatbot features, user interface errors, database structure issues, and performance problems, such as how the chatbot responds to user inquiries [17].

2.1.5 Maintenance

The final stage involves evaluating the chatbot's performance by measuring its response time, the accuracy of the information provided, customer satisfaction, and the reduction of workload on human operators. Customer satisfaction surveys are also conducted to gather feedback about their experience interacting with the chatbot.

2.2 Web Server

A web server is a software running on a server computer that enables stored web documents to be accessed by users via the internet. The web server also receives requests for web pages via HTTP or HTTPS from users, referred to as web browsers, and returns the results in the form of web pages, usually in HTML format [18].

2.3 Flask

Flask is a web framework written in Python that can be classified as a micro-framework because it does not require specific tools or libraries and has a built-in database [19]. This framework provides libraries and a collection of code that can be used to build applications without starting from scratch [20].

2.4 Python

Python is a high-level programming language used to convert commands in code into executable functions for applications. Unlike most other programming languages, Python has syntax similar to everyday human language, making it simpler to understand and write [21].

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 1 Januari 2023 Hal. 36-41
http://jurnal.unidha.ac.id/index.php/jteksis
E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

2.5 Black Box Testing

Black-box testing focuses on the functional aspects of software. This testing aims to demonstrate how the software operates by verifying whether the input data is processed correctly and ensuring that the stored information remains up-to-date [22].

RESULT AND DISCUSSION

The implementation results show that the developed chatbot is capable of providing fast and accurate responses to various customer inquiries, with an average response time of less than 2 seconds. Additionally, integration with order management and CRM systems enables the chatbot to provide up-to-date information regarding order status and product details.

3.1 Requirements Analysis

The requirements analysis phase highlights the importance of an organized folder structure in cPanel to support the efficient and manageable development of the chatbot. This structure is designed to ensure that all system components are neatly stored and quickly accessible. The static folder is used to store interface support files such as CSS, HTML, and JavaScript, aimed at creating an attractive and responsive appearance. Meanwhile, the chatbot folder contains main scripts such as app.py and log files to monitor system activities and record errors that may occur. In addition, the image folder is utilized to store product images that will support visual interactions in the chatbot's responses. With this arrangement, the chatbot system is not only designed to meet functional requirements but also to provide flexibility in future development and maintenance.

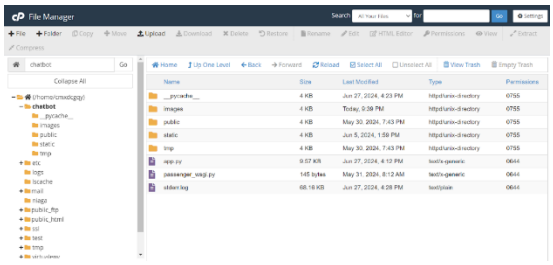

Figure 4. Folder Structure in cPanel for Chatbot

Figure 4 shows the folder structure in cPanel used to organize chatbot system components systematically. This folder includes static which stores interface design files such as CSS, HTML, and JavaScript to support responsive and dynamic displays. The chatbot folder contains the main Python script such as app.py which functions as the core logic of the system, as well as log files to monitor system activity and errors in real-time. In addition, the image folder is used to store product images needed as part of a more interactive and informative chatbot response. With this organization, the chatbot is designed to be easy to manage, flexible, and able to grow according to user needs.

3.2 Design

The chatbot interface design focuses on convenience and ease of use for customers. Using HTML for structure, CSS for styling, and JavaScript for dynamic interactions, the interface is designed to be responsive and accessible across devices ranging from desktops to smartphones. The design also takes into account a structured conversation flow, so the chatbot can handle various types of questions quickly and accurately. In addition, integration with Customer Relationship Management (CRM) and order management systems allows the chatbot to provide real-time information on order status and product details. This approach ensures that the chatbot is able to meet user expectations in providing relevant and accurate information, while increasing efficiency in customer service.

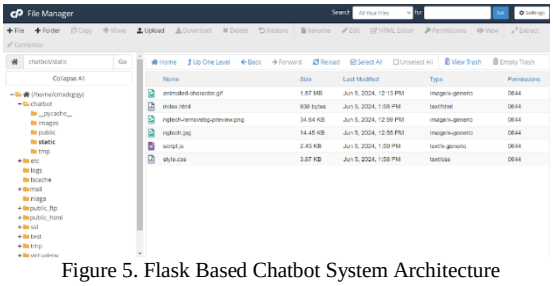

Figure 5. Flask Based Chatbot System Architecture

Figure 5 shows the chatbot interface design that is designed with a combination of HTML, CSS, and JavaScript to create a friendly and responsive user experience. This interface is designed to provide easy navigation and support a variety of devices, from desktops to smartphones. The design structure also includes a logical conversation flow, allowing the chatbot to provide fast and accurate responses to various customer inquiries. The figure also shows integration with CRM and order management systems, which ensure the chatbot always displays up-to-date and relevant information. With this approach, the chatbot design focuses not only on aesthetics, but also on functionality and user convenience.

3.3 Implementation

The chatbot implementation was done using the Python Flask framework, known for its simplicity and flexibility. This process involved developing a backend designed to handle business logic and interact with a PostgreSQL database that stores customer data, conversation history, and other related information. The system is also integrated with various third-party APIs to extend the chatbot's functionality, such as providing real-time product

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 1 Januari 2023 Hal. 36-41
http://jurnal.unidha.ac.id/index.php/jteksis
E-ISSN :**2655-8238**
P-ISSN : **2964-2132**

stock information or order status. With this combination of technologies, the chatbot can respond to customer questions quickly and accurately, while also providing convenience in updating information periodically. This implementation positions the chatbot as an innovative solution that is able to automate customer service effectively and professionally.

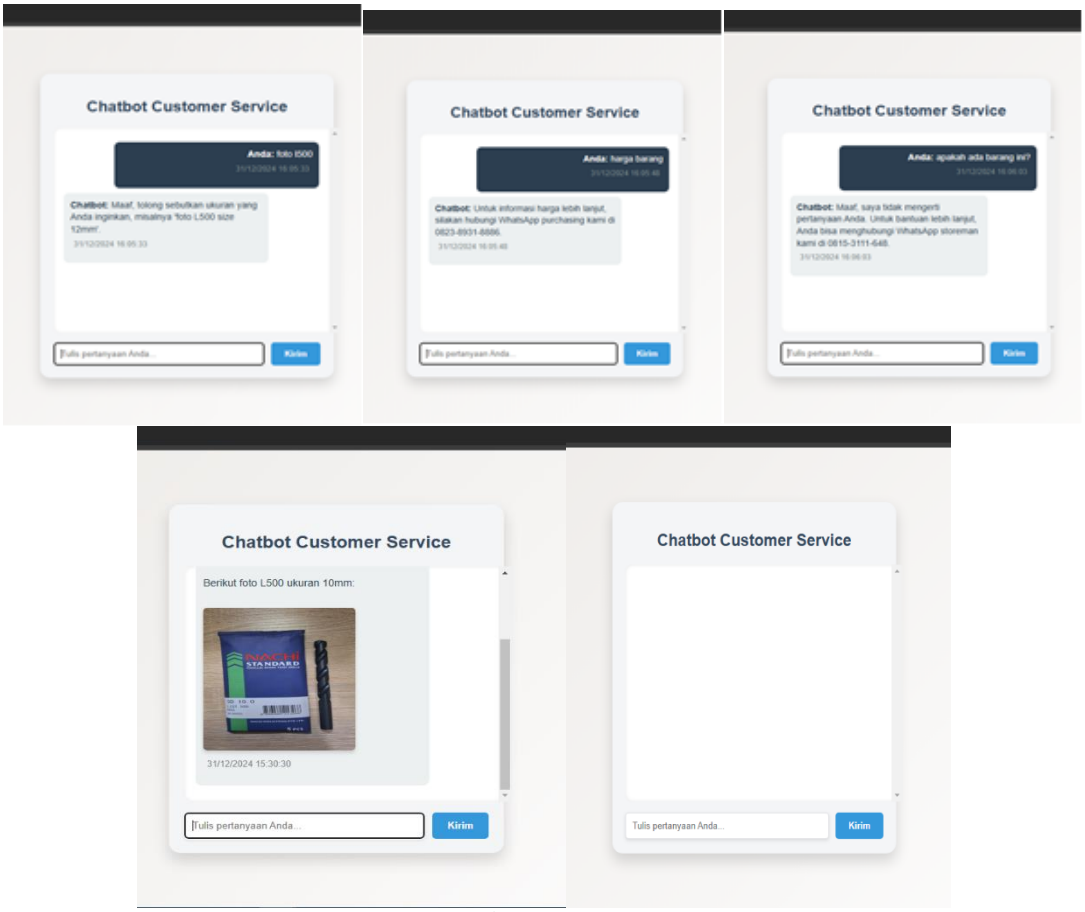

Figure 6. Chatbot Interface Design and Conversation Flow

Figure 6 illustrates the architecture of a chatbot system built using the Python Flask framework. The architecture shows how the backend components are organized to handle business logic and interact with a PostgreSQL database. The backend manages user input, processes requests, and accesses relevant data to provide fast and accurate responses. The diagram also shows integrations with third-party APIs that allow the chatbot to extend its capabilities, such as displaying order status or product information in real-time. The architecture is designed to be flexible and easy to integrate with existing systems, ensuring the chatbot can grow with business needs.

3.4 Verification

The verification process is carried out to ensure that the chatbot functions according to needs and is able to provide optimal service to users. This test uses the blackbox testing method which focuses on testing system functionality based on the input and output produced. One of the test results showed a failure in handling invalid input formats.

Table 1. Blackbox Testing Results

Test Case	Input	Expected Output	Actual Output	Result
Response Time Test	User query about product availability	Response within 2 seconds	Response within 1.8 seconds	Pass
Accuracy Test	Specific query about order status	Correct order status information displayed	Correct order status information displayed	Pass
Error Handling Test	Empty input or special characters	Error message explaining the correct input format	Chatbot not responding or showing generic error message	False

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 1 Januari 2023 Hal. 36-41
http://jurnal.unidha.ac.id/index.php/jteksis

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

Stress Test	100 concurrent queries	Responses without crashing or significant delays	Responses without crashing or significant delays	Pass
Database Query Test	Request for stored conversation logs	Logs retrieved successfully	Logs retrieved successfully	Pass

Table 1 shows the results of black-box testing focused on evaluating the functionality of the chatbot. The diagram shows the test scenario, including the expected input and output, as well as the actual results of the test. One of the results shown is the chatbot's failure to handle invalid input, such as blank characters or special symbols. The figure shows how the error was identified and became the main focus for the maintenance process in the next stage. By displaying the test results visually, this figure provides a clear picture of the chatbot's performance during the test and shows areas that need further improvement.

3.5 Maintenance

Maintenance steps are focused on fixing deficiencies found during testing, particularly related to the chatbot's failure to handle invalid input. This system maintenance includes several important aspects:

1. Input Validation Improvements
   The validation system will be updated to detect invalid input formats, such as blank characters, special symbols, or unrecognized data. After this update, the chatbot is expected to provide more informative error messages and guide users to provide correct input.
2. Retesting
   After the improvements are made, retesting using the black-box method will be applied to ensure that the chatbot is able to handle error scenarios correctly. This process involves simulating input that previously caused the system to fail.
3. Continuous Monitoring
   The system will be monitored periodically to detect potential similar errors in the future. The chatbot's activity logs will be examined to identify new problem patterns that may arise from user interactions.
4. Database Updates
   The chatbot's database will be updated to include additional responses relevant to the error scenario. This will enrich the chatbot's ability to provide more appropriate solutions to users.With these maintenance steps, the chatbot is expected to function better and provide a more optimal user experience.

With these maintenance steps, it is hoped that the chatbot can function better and provide a more optimal user experience.

CONCLUSION

The chatbot system maintenance process focuses on improving performance and refining features to address weaknesses found during testing. One of the main issues identified during the verification stage was the chatbot's failure to handle invalid input, such as blank characters or special symbols. Therefore, the first step in this maintenance is to fix the input validation system so that the chatbot can detect these errors and provide more informative messages to users. In addition, retesting is carried out after the update to ensure that the system can respond to each scenario correctly and accurately.

As part of ongoing maintenance, the chatbot is also equipped with a real-time performance monitoring mechanism to monitor responses and activity logs. This monitoring helps identify potential issues that may arise in the future so that they can be fixed immediately before they affect the user experience. In addition, the chatbot database is updated regularly to add new, more relevant responses and expand the scope of questions that the system can answer.

This maintenance process also includes improving features by updating the natural language processing (NLP) algorithm to improve understanding of more complex user input. With this series of updates, the chatbot is expected to be able to provide better, more accurate, and more efficient services, while supporting the ever-growing business needs.

REFERENCES

[1] I. NTS, "Company Profile Page."
[2] Malvin, C. Dylan, and A. H. Rangkuti, "WhatsApp Chatbot Customer Service Using Natural Language Processing and Support Vector Machine," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 12, no. 3, pp. 130–136, 2022, doi: 10.46338/ijetae0322_15.
[3] Alissyah Putri, Dani Azka Faz, Anshari Rusmeniar R.A, Yuni nur fari'ah, Falah Arfani, and Faisal Dharma Adhinata, "Jurnal Pencegahan dan Penanganan Kekerasan Seksual menggunakan Natural Languange Process dan Data Science," *El-Mujtama J. Pengabdi. Masyarakat* , vol. 4, no. 3, pp. 1833–1849, 2024, doi: 10.47467/elmujtama.v4i3.2428.
[4] F. Firdaus, R. A. Rajagede, A. Sari, S. Hanifah, and D. A. Perwitasari, "Digital Assistant for Pharmacists Using Indonesian Language Based on Rules and Artificial Intelligence," *Int. J. Eng. Trans. B Appl.*, vol. 37, no. 9, pp. 1746–1754, 2024, doi: 10.5829/IJE.2024.37.09C.04.
[5] M. K. Lubis, "Pengembangan Sistem Otomatisasi dengan Python : Meningkatkan Efisiensi Operasional," pp. 1–8.

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

[6] N. Idris, C. F. Mohd Foozy, and P. Shamala, "A Generic Review of Web Technology: DJango and Flask," *Int. J. Adv. Sci. Comput. Eng.*, vol. 2, no. 1, pp. 34–40, 2021, doi: 10.62527/ijasce.2.1.29.

[7] G. P. M. Putra, A. Tenriawaru, and Gunawan, "Rancang Bangun Virtual Assistant Chatbot Menggunakan Node.Js pada Layanan Sistem Informasi Akademik," *Pros. Semin. Nas. Pemanfaat. Sains Dan Teknol. Inf.*, vol. 1, no. 1, pp. 345–352, 2023.

[8] B. Setiaji and F. W. Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling," *Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS*, vol. 0, pp. 72–77, 2016, doi: 10.1109/ISMS.2016.53.

[9] C. S. IEEE, "2012 IEEE Sixth International Conference on Semantic Computing," 2011, p. xxxvi. doi: 10.1109/ICSC.2011.102.

[10] S. H. Bariyah and K. A. N. Imania, "Pengembangan Virtual Assistant Chatbot Berbasis Whatsapp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan Indonesia - Garut," *J. Petik*, vol. 8, no. 1, pp. 66–79, 2022, doi: 10.31980/jpetik.v8i1.1575.

[11] A. Sujjada, Somantri, A. R. Ramdani, K. Kibtiyah, M. P. Utami, and M. Ridwan Nullah, "Prediksi Nilai Ujian Sekolah Siswa SMK Plus Padjadjaran Berbasis Web Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Inf. dan Teknol.*, vol. 5, no. 2, pp. 151–158, 2023, doi: 10.37034/jidt.v5i2.370.

[12] M. E. Rianto and A. Furqon, "Impelementasi AI Chatbot Sebagai Support Assistant Website Universitas Nurul Jadid Menggunakan Algoritma BiLSTM".

[13] C. Wijayanto and Y. A. Susetyo, "Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH)," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 3, pp. 858–868, 2022, doi: 10.29100/jipi.v7i3.3161.

[14] A. Matalarens and D. Setyowatie, "Perancangan Aplikasi Chatbot FAQ Berbasiskan Aplikasi Android," *Pros. Semin. Nas. Pemanfaat. Sains dan Teknol. Inf.*, vol. 1, no. 1, pp. 281–286, 2023.

[15] 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.

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

[17] R. Hidayat, "Aplikasi Penjualan Jam Tangan Secara Online Studi Kasus: Toko JAMBORESHOP," *J. Tek. Komput.*, vol. III, no. 2, pp. 90–96, 2017.

[18] L. F. Jara and M. R. Putra, "Implementasi Digital Marketing dalam Membangun Brand Awareness Menggunakan Metode Object Oriented Analysis and Design pada Umkm Tekstil Kota Padang," *J. KomtekInfo*, vol. 8, no. 2, pp. 110–117, 2021, doi: 10.35134/komtekinfo.v8i2.105.

[19] S. Saladdin, I. Arwani, and B. Rahayudi, "Pengembangan Chatbot Yanies Cookies Untuk Pemesanan Kue Kering Berbasis Dialogflow," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 7, pp. 2121–2129, 2020.

[20] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.

[21] 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.

[22] Luqman, "APLIKASI WEB SISTEM INFORMASI PENJUALAN PADA KHAZANAH PONSEL YOGYKARTA," *Jur. Tek. Inform.*, p. 32, 2012.