Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 2 (2025) 393 - 403 e-ISSN: 2580-0760

Improving Government Helpdesk Service With an AI-Powered Chatbot Built on the Rasa Framework

Wirat Moko Hadi Sasmita[1*], Surya Sumpeno[2], Reza Fuad Rachmadi[3]
[1,2,3]Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2,3]Department of Computer Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[1]6022231052@student.its.ac.id, [2]surya@its.ac.id, [3]fuad@its.ac.id

Abstract

Helpdesk services are an important component in supporting Information Technology (IT) services. The helpdesk operates as the initial interface for managing and resolving concerns. Helpdesk helps users to get solutions when facing problems while using an IT service. This research focuses on the impact of artificial intelligence (AI)-powered chatbots on the performance of the initial response of government helpdesk services. The chatbot is designed to improve service performance by quickly identifying and classifying reported issues and automatically responding to messages, enabling faster responses. The research proposed a new System Design of a helpdesk system with an AI-based chatbot. The data used comes from Telegram group chat logs, exported in JSON format. We find that the Rasa NLU model with DIET Classifier successfully achieved an accuracy rate of 0.825 in classifying intents, with the precision value of 0.838, recall of 0.829, and F1 score of 0.821 using a Rasa model with cross-validation, where folds is 5 in evaluation. And initial response time was highly improved after using chatbot artificial intelligence from more than 3 hours on the telegram group helpdesk based to an average of 2.15 seconds. These research results suggest AI-Chatbot-based ability to assist the helpdesk team in handling user queries and reports, and improving initial time response.

Keywords: Chatbot; Helpdesk; Natural Language Rasa; Understanding

How to Cite: W. M. H. Sasmita, S. Sumpeno, and R. F. Rachmadi, “Improving Government Helpdesk Service with AI-Powered Chatbot Built on Rasa Framework”, *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 2, pp. 393 - 403, Apr. 2025.
DOI: https://doi.org/10.29207/resti.v9i2.6293

1. Introduction

Helpdesk services are a vital component of Information Technology (IT) services in an organization. Helpdesk helps customers get solutions when facing problems while using a service. Good customer support will enable an organization to sustain its customers to keep them using their service. In general, helpdesk work involves standing in front of the desk to serve customer complaints, problems, and questions via phone or chat [1]. But now, many organisations create a helpdesk system using technology. Helpdesk system builds a user application to make it easier for users to report their faced problems. In general, the helpdesk workflow is divided into several levels. Level 1 is Initial Response; at this stage, an agent receives the user's report and provides a ticket number to confirm that the complaint has been recorded. Level 2 is advanced support. If the report requires further handling related to technical issues, the agent will escalate the case to the technical team. Level 3 is Expert Support. If the technical team requires assistance from specialists with more specific expertise, the case will be escalated to an advanced technical team or experts.

The IT Helpdesk was established by the Department of Communication and Informatics of Mojokerto Regency in compliance with the Government and President Regulation [2], [3] on Electronic-Based Government Systems (SPBE). This regulation requires all IT services to provide a help desk to ensure efficient and uninterrupted operations. To fulfill this requirement, the Department of Communication and Informatics of Mojokerto Regency has currently adopted a helpdesk system using Telegram groups. An evaluation of the helpdesk's performance from August 2023 until June 2024 revealed significant shortcomings. The average initial response time to user reports was 3 hours, 10

Received: 09-01-2025 | Accepted: 06-04-2025 | Published Online: 20-04-2025

minutes, and 4 seconds, indicating considerable delays. Furthermore, only 581 out of 1054 initial user-submitted reports received a response. This level of performance highlights a serious lack in the helpdesk's ability to deliver effective user support.

With the faced problem, we need a solution to improve the performance of the helpdesk. Artificial Intelligence (AI) offers a potential solution to improve the performance of helpdesk services by quickly identifying and classifying reported problems, making it possible to provide a faster and more accurate response [4]. Various helpdesk systems are available to be adopted by organization, one of them is a chatbot system. Chatbot is a technology that allows users to interact with Artificial Intelligence (AI) applications via chat. In general, chatbots can be categorized as Rule-based, AI-powered chatbots, and Hybrid chatbots. A rule-based chatbot is the simplest type of chatbot available today. These chatbots operate by having users choose from predefined options to address their queries. They ask questions and allow users to respond, meaning the bot analyzes the provided information and replies accordingly. In certain situations, these types of chatbots may not be the best choice, as their predefined responses can lead to longer response times in guiding users to their desired outcomes. In contrast to rule-based chatbots, AI-driven chatbots are more efficient when it comes to response time.

Based on research conducted by Abdellatif A, et al[5], the research compares Natural Language Understanding (NLU) platforms for chatbots. Several NLU frameworks can be used to build AI chatbots. Evaluation of four popular NLU platforms: IBM Watson, Google Dialogflow, Rasa, and Microsoft LUIS, focused on their performance in intent classification, confidence scoring, and entity extraction in software engineering tasks. These tasks include queries to software repositories and technical questions, such as those on Stack Overflow. Rasa showed the best confidence score, with a median of 0.91 for correctly classified intents.

Rasa is an open-source Natural Language Understanding (NLU) framework for building AI chatbots contextually. Rasa can be used to build text-based or voice-based conversational chatbots [6]. Rasa have the capability to understand user input and have a conversation with the user. This NLU framework works on two main components, namely Rasa NLU and Rasa Core. Rasa NLU is a natural language processing tool, used for intent classification and entity extraction in text conversations, as well as using machine learning to recognize patterns and make generalization for previously unseen sentences [7]. Rasa Core is an open-source chatbot framework for handling contextual conversations, used for machine learning-based conversation management [8].

There are some researchers conducted research on chatbot development (see Table 1). Research by Farid A et al.[9], uses Rasa open source to create a chatbot that delivers a question-and-answer service to give recommendations to users for the Semarang city tourism destination. This research uses Telegram from a user interface of the chatbot. The result of other research by Fauzia L et al.[10], build a chatbot using Rasa for new student admission at “Politeknik Siber dan Sandi Negara”, Indonesia. This chatbot focuses on addressing questions about new student admission. The chatbot is built with Docker and put as a Chat Widget on the website. And other research by Paranjape A, et al, Using the RASA framework, researchers [11] create a voice-based smart assistant system for a vehicle using Rasa. Their research aims to reduce attention away from the task of driving while engaging in other activities like calling, playing music, navigation, and getting updates on the weather forecast and latest news, and road safety.

This study makes several contributions. First, it evaluates the performance of the Rasa Model using the Rasa model for initial response helpdesk using the helpdesk dataset filling a gap in the literature where this Rasa model has not been implemented in other research. Secondly researcher proposed a new system design for a helpdesk system using an AI chatbot. This study will propose a helpdesk system based on AI chatbot automation built using Rasa to automate initial response tasks in the helpdesk system at the Department of Communication and Information Technology of Mojokerto Regency.

Table 1. Performance Comparison of the Rasa model on variant contexts

Study	Model	Dataset	Task	Evaluation
[7]	Rasa Pipeline - SVM, CRF and LSTM	441 dataset	Question Answering about College in Vietnamese	Higher accuracy is 94,33%.
[8]	Rasa Default Pipeline - WhitespaceTokenizer - RegexFeaturizer - LexicalSyntax Featurizer - CountVectorsFeaturizer - CountVectorsFeaturizer, analyzer : char_wb, min_ngram: 1, max_ngram: 4 - DIETClassifier, epoch 100 - Entity SynonymMapper - ResponseSelector, epochs: 100	96 dataset answers questions about toddler stunting.	Toddler Stunting Consultation	precision, accuracy, and F1 score of 0.928, 0.932 and 0.930 with 80 / 20 data split

Study	Model	Dataset	Task	Evaluation
	- FallbackClassifier, threshold: 0.3, ambiguity_threshold: 0.1			
[9]	Rasa Default Pipeline - WhitespaceTokenizer - RegexFeaturizer - LexicalSyntax Featurizer - CountVectorsFeaturizer - CountVectorsFeaturizer, analyzer : char_wb, min_ngram: 1, max_ngram: 4 - DIETClassifier, epoch 100 - Entity SynomimMapper - ResponseSelector, epochs: 100 - FallbackClassifier, threshold: 0.3, ambiguity_threshold: 0.1	77 tourism FAQ datasets	Question Answer Semarang city tourism destination	Model accuracy 62%
[11]	Rasa Custom Pipeline - WhitespaceTokenizer - RegexFeaturizer - LexicalSyntax Featurizer - CountVectorsFeaturizer - CountVectorsFeaturizer, analyzer : char_wb, min_ngram: 1, max_ngram: 4 - DIETClassifier, epoch 500 - Entity SynomimMapper - ResponseSelector, retrieval_intent: out_of_scope, scale_loss: True, epochs: 100 - FallbackClassifier, threshold: 0.3, ambiguity_threshold: 0.1	300 datasets of play music command, news headline, get weather command, get navigation command	Voice-Based Smart Assistant System for Vehicles	Model accuracy is 93.67%
[12]	Rasa Default Pipeline - WhitespaceTokenizer - RegexFeaturizer - LexicalSyntax Featurizer - CountVectorsFeaturizer - CountVectorsFeaturizer, analyzer : char_wb, min_ngram: 1, max_ngram: 4 - DIETClassifier, epoch 100 - Entity SynomimMapper - ResponseSelector, epochs: 100 - FallbackClassifier, threshold: 0.3, ambiguity_threshold: 0.1	488 college enquiry datasets	Question Answer about College Enquiry	Precision, Accuracy & F1 0.628, 0.725 and 0.669 with cross-validation k=5
[13]	Rasa Custom Pipeline - WhitespaceTokenizer, token_pattern (?u)\b\w+\b) - RegexFeaturizer - LexicalSyntax Featurizer - CountVectorsFeaturizer - CountVectorsFeaturizer, analyzer : char_wb, min_ngram: 1, max_ngram: 4 - DIETClassifier - DIETClassifier, epoch 100 - ResponseSelector, retrieval_intent: out_of_scope, scale_loss: True, epochs: 100 - FallbackClassifier, threshold: 0.5, ambiguity_threshold: 0.3	291 datasets about Prambanan Temple	Question Answering about Prambanan Temple Tourism Object.	Model accuracy is 0.91 precision of 0.97, a recall of 0.94, and an F1-score of 0.95

2. Research Methods

2.1 Proposed System Design

The chatbot system uses Telegram and Rasa is an AI chatbot solution that combines the capabilities of the Telegram platform with Rasa. The following is a description of the workflow of this system with reference to Figure 1. Users start by sending a message via the Telegram application. A Telegram bot that receives the message and processes it using the Telegram Bot API. Once the message is received by Telegram, it is forwarded to the helpdesk application. In a helpdesk application, messages are saved, and tickets are created automatically, if not been created yet. The message was then forwarded to Rasa via the configured webhook.

Rasa could perform natural language understanding to detect user intent and extract entities from messages. After processing the message, Rasa determines the appropriate response based on the ongoing dialogue. Rasa uses a trained model to decide the most appropriate response, whether in the form of text or a specific action such as calling an external API or querying a database. Responses from Rasa are then sent back to the user via the Telegram Bot API via the helpdesk application to save each response from Rasa. These replies will appear in the user's Telegram app, providing answers to the user's questions or requests.

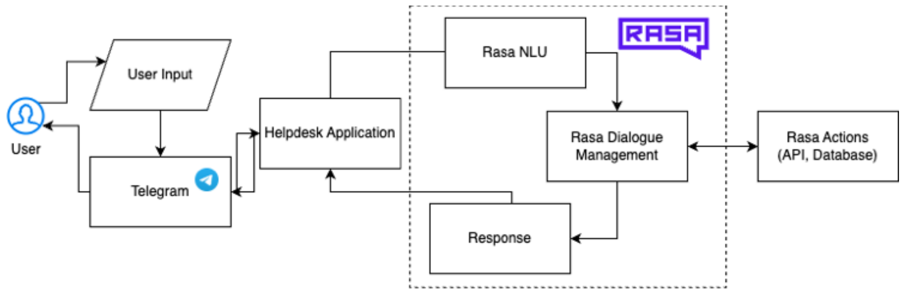
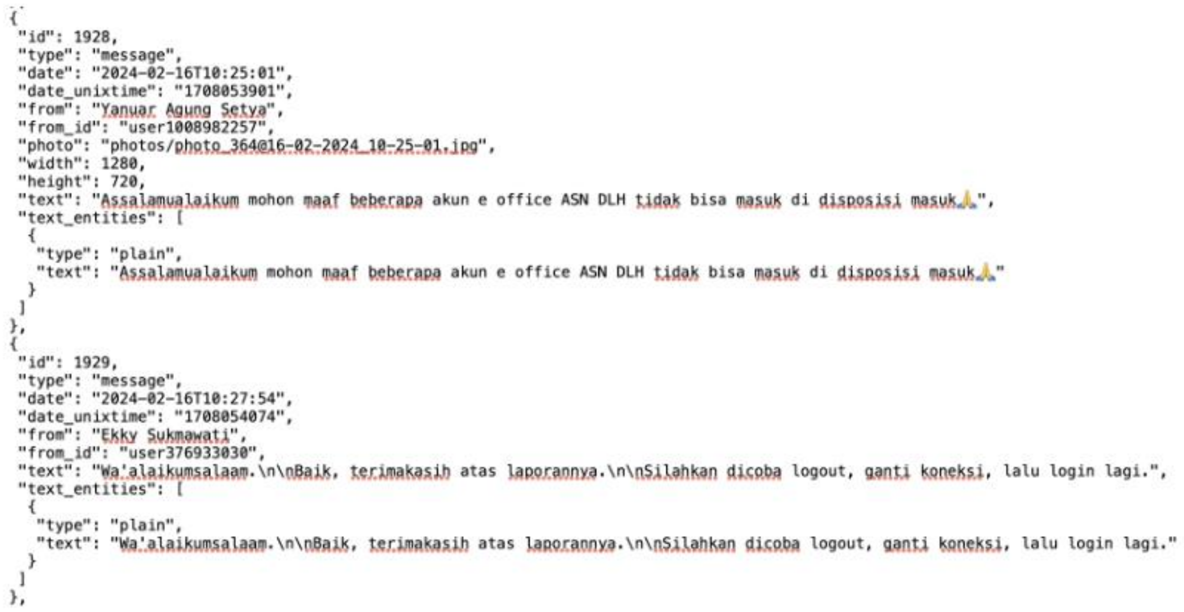


Figure 1. Proposed System Design

2.2 Dataset



```
{
 "id": 1928,
 "type": "message",
 "date": "2024-02-16T10:25:01",
 "date_unixtime": "1708053901",
 "from": "Yanuar Agung Setya",
 "from_id": "user1008982257",
 "photo": "photos/photo_364@16-02-2024_10-25-01.jpg",
 "width": 1280,
 "height": 720,
 "text": "Assalamualaikum mohon maaf beberapa akun e office ASN DLH tidak bisa masuk di disposisi masuk🙏",
 "text_entities": [
  {
   "type": "plain",
   "text": "Assalamualaikum mohon maaf beberapa akun e office ASN DLH tidak bisa masuk di disposisi masuk🙏"
  }
 ]
},
{
 "id": 1929,
 "type": "message",
 "date": "2024-02-16T10:27:54",
 "date_unixtime": "1708054074",
 "from": "Ekky Sukmawati",
 "from_id": "user376933030",
 "text": "Wa'alaikumsalaam.\n\nBaik, terimakasih atas laporannya.\n\nSilahkan dicoba logout, ganti koneksi, lalu login lagi.",
 "text_entities": [
  {
   "type": "plain",
   "text": "Wa'alaikumsalaam.\n\nBaik, terimakasih atas laporannya.\n\nSilahkan dicoba logout, ganti koneksi, lalu login lagi."
  }
 ]
},
```

Figure 2. Raw Sample of Telegram Exported Chat

In the JSON structure shown in Figure 2, several key elements are defined to represent the properties of a message. The id serves as a unique identifier for each message, while type indicates the nature of the message, such as text, image, or video. The date records the time the message was posted by the user in a human-readable format, and date_unixtime provides the same timestamp in Unix format. The from field displays the name of the user who posted the message, and from_id contains the corresponding user identifier. Lastly, text_entities refers to portions of the message text that are styled or enriched with additional elements, such as bold or italic formatting, hyperlinks, mentions, and other special entities.

In this study, researchers focused on collecting conversations that were exported as raw data in JSON format. This format not only preserves the context of each interaction but also allows for structured analysis of the dialogue at hand. An example of raw data exported in JSON format is shown in Figure 2. The structured nature of JSON makes it easy to extract key elements from a conversation, such as user questions, responses, and the flow of dialogue between various participants. After raw data is collected, it is imported into a database using tools designed to simplify the process. This makes it possible to organize and analyze data effectively to find more value in helpdesk interactions. By utilizing this large amount of conversation data, this research aims to deepen understanding of the problem reporting and resolution process in helpdesk services. For this research, the data contains mixed text in Bahasa and English. Based on collected data from telegram, researchers conducted analyzes and found the top 4 most asked and reported topics as shown in Table 2.

Table 2. Most asked and reported topics

No.	Topics
1	Application state
2	Application guide
3	Report of application bug
4	Report of internet connection issue

Data cleaning is an important process in data analysis that aims to improve the quality and accuracy of data. Based on research conducted by Widiari et al [14], the process involves removing distractions such as unnecessary characters, standardizing formatting (e.g. lowercase text), correcting spelling and grammatical errors, handling synonyms and stop words, and balancing the data set to avoid bias. It also includes resolving ambiguities, removing duplicate or irrelevant data, and handling missing information. Clean and organized data is essential for effective analysis, as errors in the data can lead to incorrect conclusions and poor decision-making.

2.4 Rasa NLU Pre-Processing

Preprocessing aims to convert the cleaned data to the format desired by Rasa, specifically in YAML format. This step is crucial in the development of AI models and chatbot applications using Rasa, as it ensures that the data is structured in a way that can be interpreted and utilized effectively by Rasa to train the model. In Rasa, intent represents the user's intent of message send. During the preprocessing stage, user messages are grouped into specific intents, which helps the model understand what the user wants. This process involves mapping each text to the appropriate intent label based on the context of the conversation. Responses in Rasa are pre-defined messages that the chatbot returns to the user based on a recognized intent. During the preprocessing stage, the researcher creates a series of possible responses according to each intent. For example, if the user intent is identified as get_weather, the response could be "The weather in Surabaya today is sunny with a maximum temperature of 36°C." These responses can be set to be informative, interesting, or specific, depending on the user's needs. The list of intentions and responses in this study is shown in Tables 3 and 4.

Table 3. Intent List

Intent	Description	Number of Samples
report_app_error	Report errors in the application	7
ask_app_status_with_name	Ask application conditions	11
provide_app_version	Delivers the application version	5
provide_app_screenshot	Send the application screenshot	5
provide_app_name	Sends the application name	6
report_app_error_and_provide_app_name	Reports an error in the application	15
ask_tutorial_self_sign	Provides a tutorial for requesting a self-signature	5
doesnt_have_sign_certificate	User does not have a digital certificate	5
request_eoffice_tutorial	Request a tutorial from e-office	6
ask_delete_employee	Asks how to delete employee data	5
greet	Greetings	6
thank	Acknowledgments	9
ask_permission	Request permission	6
ask_ticket_status	Ask status report	5
report_internet_problem	Report internet problems	9
slow_connection	Reports that the internet is slow	8
inform_internet_speed	Specifies the speed of the internet	9
report_initial_problem	Report an initial problem with Telegram	6
report_user_action	Reports user actions on the Telegram	5
report_reinstall_issue	Reports that the user performed a reinstall	5
	total samples	143

Table 4. Response List

Response	Description
utter_ask_error_type	Asks about the type of error
utter_ask_app_version	Asks for the application version
utter_ask_screenshot	Asks for a screenshot of the error
utter_confirm_ticket	Ask for related confirmation
utter_ask_more_details	Request details of the error
utter_ask_app_name	Asks for the name of the application
utter_tutorial_self_sign	Sends tutorials related to requesting a self-employed electronic signature
utter_contact_dana	Instructions to contact Mr. Dana
utter_tutorial_eoffice	Sends the E-Office application tutorial link
utter_delete_employee	Information related to deleting employee data
utter_greet	Reply Greeting
utter_goodbye	Saying goodbye
utter_ask_image_description	Asks for a description of the sent image
utter_ask_problem_type	Asks about the type of problem
utter_ask_location	Asks about the location of the problem
utter_ask_duration	Asks about the duration of the problem
utter_ticket_status	Sends the ticket status
utter_check_slow_internet	Instructions to check the internet speed
utter_connecting_human	Connecting to the Technical Team
utter_welcome	Greetings, you're welcome

Dialogues are structured using data from intents and responses that describe conversations. The dialogue is structured based on the main scenarios of the services provided. The list of dialogues displayed is in Table 5.

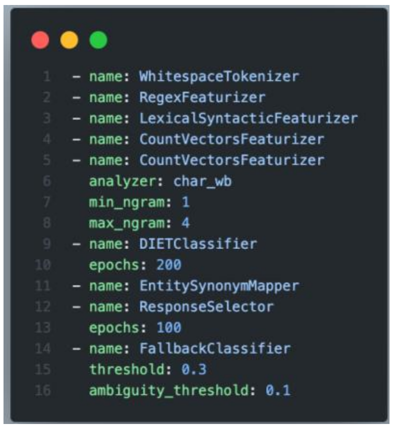
Table 5. Dialogue and Rule List

Rules / Story	Description
Error report, ask the app name	A dialogue about reporting application errors, by asking for the name of the application
Ask app status	A dialogue asks for the status of the application
Ask how to use self-sign	A dialogue asks about the use of a self-signature
doesn’t have a certificate	A dialogue if the user does not have a digital certificate
E-Office tutorial	A dialogue asks for e-office application tutorials
Ask delete employee	A dialogue asks how to delete employee data
Greet	Greeting dialogue

Rules / Story	Description
Solve Problem	A dialogue if the problem has been successfully resolved
Start Chatbot	A dialogue for starting a chatbot
Restart conversation	A dialogue for restarting the chatbot
report slow internet	A dialogue to report slow internet
How to register a signing certificate	A dialogue on how to register a digital certificate
Ask Image Description	A dialogue asks for a description of the image

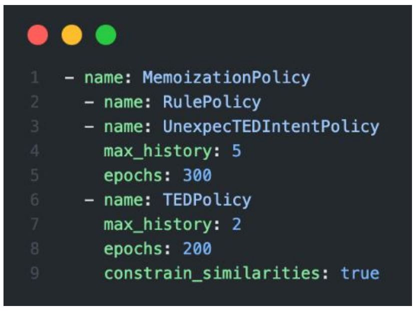
2.5 Rasa Data Training

Rasa has provided configurations for training intent and dialogue datasets called pipelines and policies [13]. The Rasa pipeline is a set of model components used to process user input during training and implementation. This pipeline helps the model understand user input, extract entities, and classify intents. This pipeline is defined in the config.yml file shown in Figure 3 and Figure 4 consists of various components such as tokenization, feature generation, and classification.



```
- name: WhitespaceTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 200
- name: EntitySynonymMapper
- name: ResponseSelector
  epochs: 100
- name: FallbackClassifier
  threshold: 0.3
  ambiguity_threshold: 0.1
```

Figure 3. Rasa Pipelines Configuration



```
- name: MemoizationPolicy
  - name: RulePolicy
  - name: UnexpecTEDIntentPolicy
    max_history: 5
    epochs: 300
  - name: TEDPolicy
    max_history: 2
    epochs: 200
    constrain_similarities: true
```

Figure 4. Rasa Policies Configuration

2.6 Rasa Data Testing and Evaluation

At this stage, system testing will be carried out to determine the accuracy of the model that has been created. Testing will use the cross-validation method. Cross-validation is a method used to assess the generalization and robustness of a model. This method is very common in machine learning and predictive modeling. The main concept of cross-validation is to divide data into several batches, train the model on some of these batches, and validate the model on the remaining batches [12]. This helps ensure that the model performs well on data it has never seen before, reducing problems such as overfitting [15]. Rasa provides automated testing using the rasa test script. The chatbot test results using Rasa produce a report on the NLU model test results [13], [16].

The NLU model is tested based on the model training dataset in handling user input. This process includes tasks such as entity extraction and clarification of user message intent according to predefined categories. Model testing will produce a confusion matrix, which provides a general description of the model's performance.

2.7 Experimental Setup

This research utilizes a local server to test the Rasa model integrated with the helpdesk system. To make the local server accessible over the internet, Ngrok is used. Ngrok is a tool that creates a secure tunnel to local applications, enabling them to be publicly accessible via a URL, even when the server is running locally [17], [18]. The specifications of the local server used in this research are shown in Table 6.

Table 6 .Hardware Specification

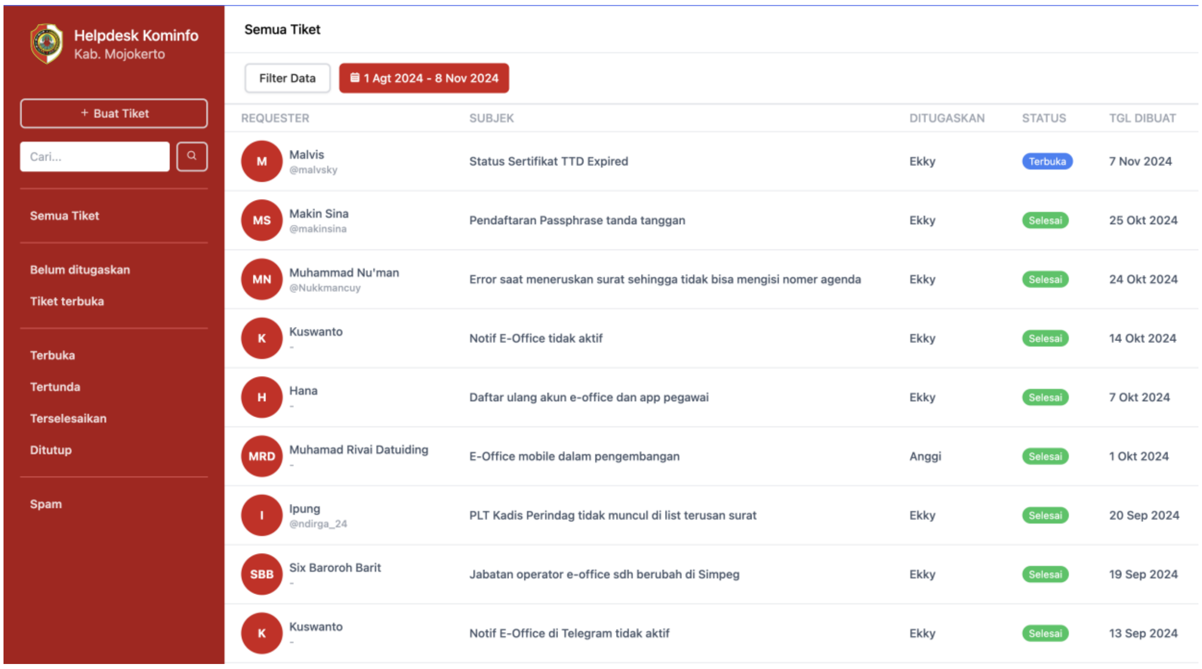
Hardware	Specification
CPU	Apple M2 8-Core
GPU	Apple Metal 10-Core
RAM	8 GB
Operating System	MacOS

Users interact with Rasa using the Telegram application. Telegram is a messaging application that emphasizes speed and security. This app is fast, easy to use, and free. Telegram can be accessed on several devices simultaneously, and all messages will be automatically synchronized across phones, tablets or computers [17].

3. Results and Discussions

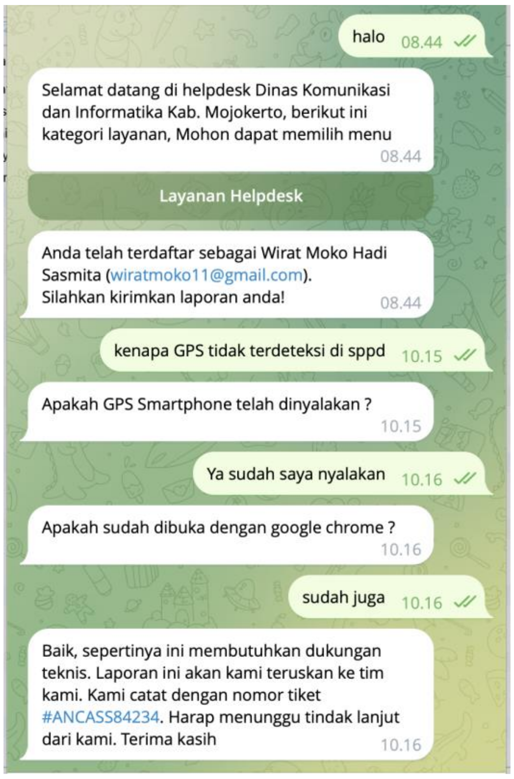
3.1 Prototype of Helpdesk System Using Chatbot AI

The chatbot in this research runs on a server, meaning that all conversations are handled via HTTP requests between the web application and the chatbot server. In addition, Rasa can be integrated with Telegram to manage user interactions, using the Telegram API to receive messages and send responses in real-time. To enable this communication from local servers, Ngrok is used. This setting ensures that user messages from Telegram are forwarded via Ngrok to Rasa, enabling efficient two-way communication. Figure 5 shows the implementation results using Telegram.



REQUESTER	SUBJEK	DITUGASKAN	STATUS	TGL DIBUAT
M Malvis @malvsky	Status Sertifikat TTD Expired	Ekky	Terbuka	7 Nov 2024
MS Makin Sina @makinsina	Pendaftaran Passphrase tanda tanggan	Ekky	Selesai	25 Okt 2024
MN Muhammad Nu'man @Nukkmancuy	Error saat meneruskan surat sehingga tidak bisa mengisi nomer agenda	Ekky	Selesai	24 Okt 2024
K Kuswanto -	Notif E-Office tidak aktif	Ekky	Selesai	14 Okt 2024
H Hana -	Daftar ulang akun e-office dan app pegawai	Ekky	Selesai	7 Okt 2024
MRD Muhamad Rivai Datuiding -	E-Office mobile dalam pengembangan	Anggi	Selesai	1 Okt 2024
I Ipung @ndirga_24	PLT Kadis Perindag tidak muncul di list terusan surat	Ekky	Selesai	20 Sep 2024
SBB Six Baroroh Barit -	Jabatan operator e-office sdh berubah di Simpeg	Ekky	Selesai	19 Sep 2024
K Kuswanto -	Notif E-Office di Telegram tidak aktif	Ekky	Selesai	13 Sep 2024

Figure 5. Helpdesk Application



halo 08.44

Selamat datang di helpdesk Dinas Komunikasi dan Informatika Kab. Mojokerto, berikut ini kategori layanan, Mohon dapat memilih menu 08.44

Layanan Helpdesk

Anda telah terdaftar sebagai Wirat Moko Hadi Sasmita (wiratmoko11@gmail.com). Silahkan kirimkan laporan anda! 08.44

kenapa GPS tidak terdeteksi di sppd 10.15

Apakah GPS Smartphone telah dinyalakan ? 10.15

Ya sudah saya nyalakan 10.16

Apakah sudah dibuka dengan google chrome ? 10.16

sudah juga 10.16

Baik, sepertinya ini membutuhkan dukungan teknis. Laporan ini akan kami teruskan ke tim kami. Kami catat dengan nomor tiket #ANCASS84234. Harap menunggu tindak lanjut dari kami. Terima kasih 10.16

Figure 6. Test using Telegram

From Telegram, messages are forwarded to the helpdesk application (shown in Figure 6) before being processed by Rasa. This step is taken to store every conversation received by the helpdesk, making it easier to monitor incoming reports. In cases where Rasa fails to recognize or respond to an incoming message, an agent can act manually. this flow makes the agent work with efficiency, focusing on important and urgent reports that need more effort to resolve the problem.

3.2 Performance of Model

During the training process of the model using Rasa pipeline, it achieved an accuracy rate of 0.967. This result indicates that the training process well-worked. But to ensure the model, researcher need to test and validate the model. The chatbot model is tested through simulation using test data to assess its performance during operation. The testing process, which is conducted with the Rasa test script, automatically produces system reports in the form of images, JSON files and YAML files. Testing was carried out using the cross-validation method with 5 folds, which means the data was divided into 5 batches. Each batch is used to iteratively train and validate the model, with 4 batches used for training and 1 batch for testing in each iteration. This ensures that model performance is tested on different parts of the data, thereby increasing model reliability. For the NLU model testing results, Rasa produces several documents consisting of the intent error report, intent histogram, intent confusion matrix, and intent report.

The Intent Confusion matrix shown in Figure 7 provides information about intents that were predicted by the model. If all intents are predicted accurately, the matrix displays a diagonal pattern, reflecting the corresponding predictions. In the case of the trained model, there are several intents that do not match predictions. Based on the confusion matrix for intent prediction as in Figure 7, accuracy, precision, recall, and f1-score values can be obtained as shown in Equation 1.

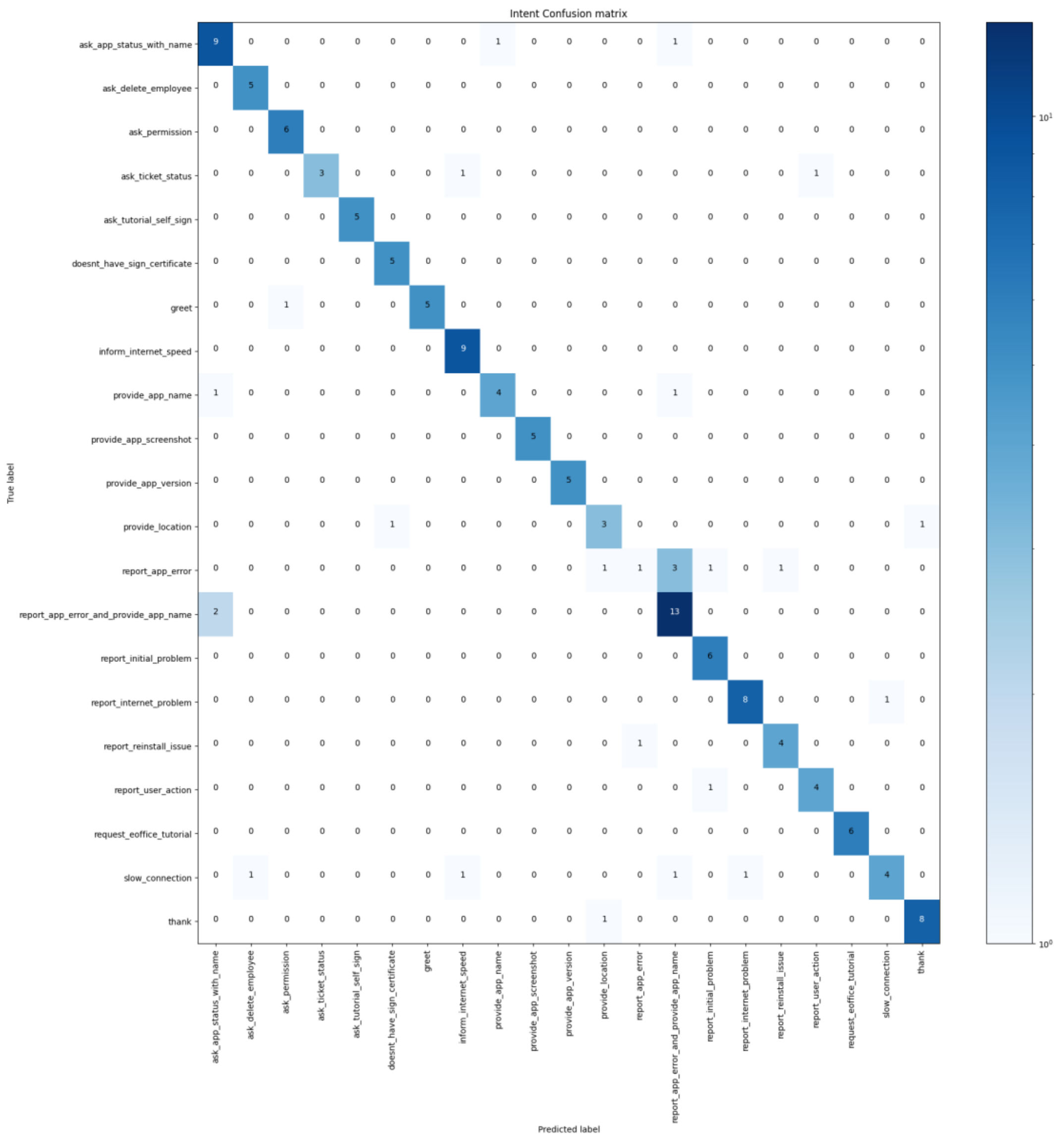


Figure 7. Intent Confusion Matrix

$$accuracy = \frac{true\ prediction}{number\ of\ samples} = \frac{118}{143} = 0.825 \qquad (1)$$

Accuracy is obtained by dividing true predictions by a number of samples. With the accuracy of the chatbot model being 0.825, this shows that the model has been able to identify the intent of the input text well enough, but this also implies that the model produces incorrect predictions in approximately 0.175 of the cases. To ensure appropriate accuracy, the precision, recall and f1 score values will also be calculated for each intent as in Table 7.

Overall, although the trained model shows good accuracy and between precision and recall, there are opportunities for improvement. By improving the parts represented by the confusion matrix and the metrics calculated, the model can be refined to provide more reliable predictions in the future.

Result of evaluation of the chatbot's intent prediction model, several misclassifications were identified that provide valuable insights for future improvement. Table 8 shows 5 examples from the test data highlighting these advantages. One notable issue was the overlap in intents such as report_internet_problem and slow_connection, where user queries reporting slow internet speeds were sometimes classified incorrectly. This error likely appears due to the similarity in language patterns between these intents. For example, a query like “Koneksi internet di kecamatan jetis lambat sekali” was sometimes misclassified as report_internet_problem, even though the correct intent should be slow_connection. Such errors indicate the need for additional training samples that emphasize distinguishing features between these intents. Another challenge was observed with the intent report_app_error_and_provide_app_name, which had a lower F1-score compared to others. This could be attributed to the complexity of multi-intent sentences, where users combined error reporting with additional details, such as the application's name.

Table 7 Precision, Recall, F1-Score for each intent

Intent	TP	FP	FN	Precision	Recall	F1-Score
report_internet_problem	8	1	1	0.889	0.889	0.889
ask_ticket_status	3	0	2	1.000	0.600	0.750
slow_connection	4	1	4	0.800	0.500	0.615
report_reinstall_issue	4	1	1	0.800	0.800	0.800
ask_app_status_with_name	9	3	2	0.750	0.818	0.783
request_eoffice_tutorial	6	0	0	1.000	1.000	1.000
report_app_error_and_provide_app_name	13	6	2	0.684	0.867	0.765
provide_app_screenshot	5	0	0	1.000	1.000	1.000
ask_tutorial_self_sign	5	0	0	1.000	1.000	1.000
provide_app_name	4	1	2	0.800	0.667	0.727
doesnt_have_sign_certificate	5	1	0	0.833	1.000	0.909
report_user_action	4	1	1	0.800	0.800	0.800
report_app_error	1	1	6	0.500	0.143	0.222
greet	5	0	1	1.000	0.833	0.909
provide_location	3	2	2	0.600	0.600	0.600
inform_internet_speed	9	2	0	0.818	1.000	0.900
thank	8	1	1	0.889	0.889	0.889
provide_app_version	5	0	0	1.000	1.000	1.000
ask_delete_employee	5	1	0	0.833	1.000	0.909
report_initial_problem	6	2	0	0.750	1.000	0.857
ask_permission	6	1	0	0.857	1.000	0.923
			Average	0.838	0.829	0.821

Table 8 Examples of Errors Intent Prediction

Text	Intent Prediction	True Intent
Koneksi internet saya sangat lambat sekali (My Internet Connection is slow)	report_internet_problem	slow_connection
aplikasi E-Office error (E-Office application error)	ask_app_status _with_name	report_app_error_and_provide_app_name
E-office	report_app_error_and_provide_app_name	provide_app_name
Koneksi internet di kecamatan jetis lambat sekali (The internet connection in Jetis sub-district is very slow)	report_internet_problem	slow_connection
e-sppd masih masalah (E-SPPD is still a problem)	provide_app_name	report_app_error_and_provide_app_name
minta tutorial terima surat masuk sampai dengan distribusi di e-office (ask for a tutorial on receiving incoming mail through to distribution in e-office)	report_app_error_and_provide_app_name	request_eoffice_tutorial

3.3 Initial Response Time

The initial response time is calculated by subtracting the timestamp of the starting point from that of the endpoint (see Figure 8). This study involves two scenarios. The first scenario includes other services in the process, following the sequence: Start Point → 1 → 2 → 3 → 4 → End Point. And second scenario uses the Rasa model to reply without involving other services, following the sequence: Start Point → 1 → 4 → End Point.

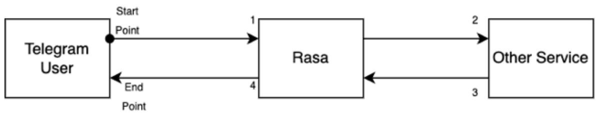


Figure 8. Flow to calculate Initial Response Time

Refer to Figure 9, the simulation with 40 messages resulted initial response. The chatbot demonstrates an average response time of just 2.15 seconds to the user's initial message, highlighting significantly greater efficiency compared to the typically slower response times observed with the Telegram group method. This response time was measured by capturing the timestamp of the start point and end point of the chatbot's response. It is worth noting that the recorded time includes potential delays introduced by the Telegram API and external services. Can been seen that there are a few initial responses has a long time. This happens because the service is not available or is having an issue. This makes the action wait for a timeout. Other factors can affect response time, such as server performance and internet connection speed. Enhanced server capabilities, such as the use of GPUs, can greatly improve processing speed, resulting in faster and more efficient responses from the Rasa framework. By comparing the performance with the Telegram group-based method, it is evident that the AI chatbot significantly improves initial response times, provided that the topics in the user's messages have been included in the model's training data.

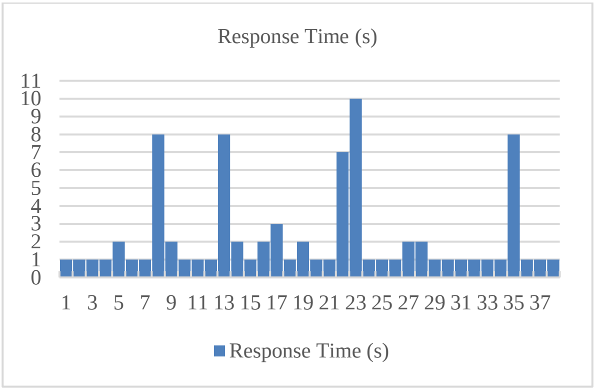


Figure 9. Initial Response Time

3.4 Comparing Telegram Group vs Chatbot AI Helpdesk System

The results of the research comparison of the previous method using a Telegram group-based system with the proposed AI chatbot-based system (shown in Table 9) reveal that the helpdesk system with the AI chatbot offers several advantages over the group-based method. The chatbot excels in providing fast responses, measured in seconds, and allows for 24-hour accessibility for submitting reports. However, the chatbot has a limitation in that it can only handle topics that have been specifically trained into its model. Additionally, another advantage of the chatbot-based helpdesk system is its ability to simplify the monitoring of incoming reports, making the process more efficient and organized.

Table 9 Comparison of Telegram Group vs AI Chatbot based

Feature	Telegram Group	Chatbot AI with Helpdesk System
Response Time	Depends on the technical team; sometimes delayed	Fast and consistent responses
Availability	Not available 24/7	Operates 24/7
Response Speed	Generally slow	Very fast response times
Scope	Unlimited but unfocused	Limited to trained topics, ensuring precision
Monitoring	No report monitoring	Enables easy monitoring of all incoming reports

4. Conclusions

In this research, the IT helpdesk system using AI chatbot base, developed using the open-source framework Rasa version 3.6.2 with the Python programming language functions as an NLU-based to improve initial response. The process to conduct this research include several stages : identification problem, literature survey, proposed system design, organize dataset, training model, testing and evaluation. AI chatbot process requires several dataset configurations as a basis for model training, including NLU, Stories, Rules, Domain, and Config. Several critical challenges were identified in the current helpdesk system workflow, such as slow initial response, inability to monitor incoming reports, helpdesk can't respond 24 hours. By implementing the proposed IT helpdesk system. The current system design, while in a prototype stage, has limitations as it has not yet been developed into a full scope of helpdesk and functional system. Testing result of the chatbot model for intent classification using 5-fold cross-validation achieved an accuracy of 0.825. Using the Confusion Matrix as a performance metric, the model recorded a precision of 0.838, a recall of 0.829, and an F1-score of 0.821. The test results showed that the chatbot successfully responded to 118 out of 143 messages, indicating its ability to assist the helpdesk team in handling user queries and reports. Future efforts should prioritize creating a fully AI scope of helpdesk model, and integrate comprehensive usability testing to validate the chatbot's performance in the real world. Additionally, the initial response time improved significantly with the implementation of automation using the AI chatbot. The initial response time, which previously relied on the Telegram group method and was measured in hours, has now been reduced to seconds with an AI chatbot. This can happen as long as the scope has been trained on the model. This research demonstrates that the adoption of AI chatbots can significantly improve initial response times in government helpdesk services.

Acknowledgments

Researchers would like to express their gratitude to the Ministry of Communication and Digital (KOMDIGI) Indonesia for the very significant support and funding, which has made this research possible. Their commitment to advancing technology and innovation in the field of information technology has been an invaluable part of this research.

References

[1] C. Cassandra, S. Hartono, and M. Karsen, “Online Helpdesk Support System for Handling Complaints and Service,” in *2019 International Conference on Information Management and Technology (ICIMTech)*, 2019, pp. 314–319. doi: 10.1109/ICIMTech.2019.8843726.
[2] Presiden Republik Indonesia, *Peraturan Presiden Nomor 82 Tahun 2023 Percepatan Transformasi Digital dan Keterpaduan Layanan Digital Nasional*. Peraturan Presiden, 2023.
[3] Presiden Republik Indonesia, *Nomor 95 Tahun 2018 Tentang Sistem Pemerintahan Berbasis Elektronik*. Peraturan Presiden, 2018.
[4] A. Følstad *et al.*, “Future directions for chatbot research: an interdisciplinary research agenda,” *Computing*, vol. 103, no. 12, pp. 2915–2942, Dec. 2021, doi: 10.1007/s00607-021-01016-7.
[5] A. Abdellatif, K. Badran, D. E. Costa, and E. Shihab, “A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering,” *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 3087–3102, Aug. 2022, doi: 10.1109/TSE.2021.3078384.

[6] Rasa Technologies GmbH, “Introduction to Rasa Open Source & Rasa Pro.” Accessed: Aug. 06, 2024. [Online]. Available: https://rasa.com/docs/rasa/

[7] K. N. Lam, N. N. Le, and J. Kalita, “Building a Chatbot on a Closed Domain using RASA,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2020, pp. 144–148. doi: 10.1145/3443279.3443308.

[8] W. Hadikurniawati, S. Wijono, D. Manongga, I. Sembiring, and K. D. Hartomo, “Toddler Stunting Consulting Chatbot using Rasa Framework,” *Jurnal Rekayasa Elektrika*, vol. 19, no. 4, Dec. 2023, doi: 10.17529/jre.v19i4.33014.

[9] F. A. Anam and S. N. Anwar, “Telegram Chatbot Implementation Using Rasa Framework to Recommend Tourism in Semarang City,” Mar. 2024.

[10] L. Fauzia, R. B. Hadiprakoso, and Girinoto, “Implementation of Chatbot on University Website Using RASA Framework,” in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2021, pp. 373–378. doi: 10.1109/ISRITI54043.2021.9702821.

[11] A. Paranjape, Y. Patwardhan, V. Deshpande, A. Darp, and J. Jagdale, “Voice-Based Smart Assistant System for Vehicles Using RASA,” in *2023 International Conference on Computational Intelligence, Networks and Security, ICCINS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICCINS58907.2023.10450143.

[12] S. Meshram, N. Naik, V. R. Megha, T. More, and S. Kharche, “College Enquiry Chatbot using Rasa Framework,” in *2021 Asian Conference on Innovation in Technology, ASIANCON 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021. doi: 10.1109/ASIANCON51346.2021.9544650.

[13] Zein Hanni Pradana, Hanin Nafi’ah, and Raditya Artha Rochmanto, “in Chatbot-based Information Service using RASA Open-SourceFrameworkin Prambanan Temple Tourism Object,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 656–662, Aug. 2022, doi: 10.29207/resti.v6i4.3913.

[14] N. Widiari, I. M. A. D. Suarjaya, and D. Githa, “Teknik Data Cleaning Menggunakan Snowflake untuk Studi Kasus Objek Pariwisata di Bali,” *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, p. 137, Jul. 2020, doi: 10.24843/JIM.2020.v08.i02.p07.

[15] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, “Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier],” *IEEE Comput Intell Mag*, vol. 13, no. 4, pp. 59–76, 2018, doi: 10.1109/MCI.2018.2866730.

[16] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[17] Telegram, “Telegram FAQ.” Accessed: Oct. 01, 2024. [Online]. Available: https://telegram.org/faq?

[18] R. Vineeth, S. Maskey, U. S. Vishakan, and Y. Singh, “A Proposed Chatbot Psykh Your Personal Therapist and Stress Buster Using RASA Open-Source Framework,” in *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development, OTCON 2022*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/OTCON56053.2023.10114025.