Development of a Natural Language Processing-Based Chatbot "GoLotim" for Tourism Information Services in East Lombok Regency

[1]Dwinita Arwidiyarti
Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Kecerdasan Artifisial, Universitas Teknologi Mataram
dwinita@utmmataram.ac.id

[2]Lalu M. Takdir Darmawan Bakti
Program Studi Rekayasa Perangkat Lunak, Fakultas Ilmu Komputer dan Kecerdasan Artifisial , Universitas Teknologi Mataram
laludarmawanbakti@utmmataram.ac.id

Received: 20 Juni 2026 | Revised: 10 Juli 2026 | Accepted: 10 Agustus 2026

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: 10.32767/jti.v18i3.3276

ABSTRACT

East Lombok Regency has diverse tourism destinations, but information is fragmented across various platforms. This study aims to develop "GoLotim," a custom-built Natural Language Processing (NLP) chatbot integrated into a web-based tourism information system. Unlike pre-built frameworks (e.g., Dialogflow, Rasa), this approach ensures data sovereignty, lightweight deployment, and full control over localized domain terminology. The system was developed using the Waterfall SDLC with ReactJS, Flask, and MongoDB. The knowledge base comprises a manually compiled JSON dataset consisting of 6 distinct intents (tags) and 42 training question patterns. User queries undergo tokenization, case folding, and stemming, followed by text representation via a 128-dimensional Bag of Words (BoW). The feature vector is classified using a Neural Network with a 64-neuron hidden layer (ReLU) and a Softmax output layer. Evaluation was conducted via Black Box Testing and 30 predefined chatbot question scenarios (5 per intent). An "appropriate response" is operationally defined as the system correctly predicting the intended tag and retrieving factually matching information from the knowledge base. The chatbot achieved a scenario-based success rate of 93.3% (28 appropriate, 2 inappropriate). Technical classification metrics yielded a Macro-Average Precision of 0.94, Recall of 0.93, and F1-Score of 0.93, supported by Confusion Matrix analysis. The average response time was 1.20 seconds (Min: 0.85s, Max: 1.65s, Std. Dev: 0.22s). The two misclassifications were identified as False Negatives caused by unseen vocabulary variations outside the 128-feature BoW limit. These findings demonstrate the technical feasibility of the custom NLP implementation for regional tourism. However, the study is limited by the small dataset size and the absence of direct user satisfaction evaluation, highlighting the need for future research on larger independent test sets and user-based assessments.

Keywords-Chatbot; Natural Language Processing; Neural Network; Bag of Words; Tourism Information System; East Lombok.

I. INTRODUCTION

East Lombok Regency is one of the regions in the West Nusa Tenggara Province that boasts diverse tourism potential, encompassing natural, marine, and cultural tourism. Various destinations, including beaches, waterfalls, mountainous areas, tourist villages, and local cultural traditions, serve as attractions for both domestic and international tourists. This potential must be supported by the provision of easily accessible tourism information, enabling tourists to obtain details about their intended destinations. Information regarding tourism destinations in East Lombok Regency can currently be found across various media, such as websites, social media, blogs, and other digital platforms. Tourists may need to consult multiple information sources to obtain details regarding locations, facilities, travel access, ticket prices, operating hours, and other supporting information. This condition indicates a need for an information medium capable of integrating tourism information and providing a simpler interaction mechanism for users. The issue of providing tourism information across fragmented platforms has also been addressed in previous studies [1]. The development of Artificial

Intelligence (AI), particularly Natural Language Processing (NLP), enables computer systems to process questions conveyed in natural language. The application of NLP in chatbots allows users to submit questions in sentence form without needing to use specific command formats. The chatbot then processes these questions to determine the category or intent of the inquiry and provides a response based on available information.

"Various studies have developed chatbots for the tourism sector using different approaches and platforms. Some studies utilize Dialogflow, rule-based approaches, or no-code platforms [1][2][3][4]. These approaches demonstrate that chatbots can serve as effective tourism information providers. However, there is room for development regarding the integration of chatbots with tourism information systems and the management of the knowledge base used to generate responses.

While pre-built frameworks such as Dialogflow or Rasa are widely adopted due to their advanced natural language understanding capabilities, this study deliberately employs a custom-built Natural Language Processing (NLP) approach using Bag of Words and Neural Networks. This architectural decision is driven by several strategic considerations. **First**, developing a proprietary NLP model eliminates dependency on third-party APIs, thereby ensuring data privacy and sovereignty, which is crucial for regional tourism information systems. **Second**, a custom implementation offers a significantly more lightweight deployment with no licensing costs, making it highly suitable for local tourism platforms with constrained computational resources. **Third**, this approach provides full control over the domain-specific terminology and knowledge base structure unique to East Lombok tourism. Unlike pre-built frameworks that often impose rigid architectural constraints such as predefined slot-filling mechanisms, the custom JSON-based dataset allows developers to tailor the question patterns, tags, and response retrieval mechanisms precisely to the localized context without being bound by the limitations of external frameworks. Ultimately, this design prioritizes seamless integration with the web-based tourism information system while maintaining complete autonomy over the underlying data and classification logic.

The contribution of this study lies in the implementation of a question classification mechanism based on a Neural Network within the East Lombok Regency tourism information chatbot and its integration into a web-based tourism information system. Evaluation is conducted through functional testing and chatbot question scenario testing to determine response appropriateness and system response time. Thus, this study is not intended to compare the statistical superiority of the Neural Network model against other algorithms, but rather to evaluate the system's implementation capability in executing tourism information service functions based on predetermined testing scenarios.

II. LITERATURE REVIEW

2.1 Chatbot

A chatbot is a computer program designed to interact with humans through text-based conversations [5]. Chatbots can be used to provide automated information services, allowing users to obtain information without direct communication with human agents. The advancement of AI enables chatbots to utilize natural language processing to handle user inquiries and determine appropriate responses [6] [7]. In the context of information services, a chatbot can serve as an alternative interaction interface between users and systems. Users do not need to understand the database structure or available information menus; instead, they can submit questions using sentences understood by the system. Chatbot implementation has expanded across various fields, including customer service, education, healthcare, and tourism [8].

2.10 Natural Language Processing

Natural Language Processing (NLP) is a field within AI focused on processing natural language so that computers can interpret the text or language used by humans [9]. In chatbot applications, NLP is used to process user input through several stages, such as text preprocessing, text representation, classification, and response determination [10][11]. In this study, NLP is applied using a classification approach. User questions are first processed through tokenization, case folding, and stemming. The processed results are then represented using the Bag of Words. This representation serves as input for the Neural Network model to determine the tag or category of the question.

2.2 Bag of Words

Bag of Words (BoW) is a text representation method that builds a vocabulary from a collection of texts and converts each text into a numerical vector based on the presence of words within the text. In presence-based representation, a value of 1 indicates that the word is present in the text, while a value of 0

indicates its absence [12]. In the implementation of the GoLotim chatbot, every word in the vocabulary is represented by a value of 1 if it appears in the question, and 0 if it does not. This representation allows text to be converted into a numerical form that can be processed by the Neural Network model. Thus, BoW acts as a bridge between the text preprocessing results and the classification process.

2.6 Neural Network

A Neural Network is a machine learning model composed of numerous interconnected neurons or nodes, organized into several layers: the input layer, hidden layer, and output layer [13]. In this study, the model is used to classify user question patterns. The architecture consists of three main layers: the input layer, hidden layer, and output layerThe input layer uses 128 nodes (linear activation), the hidden layer uses 64 nodes with the ReLU activation function, with the number of outputs corresponding to the number of classes (tags) in the training data. The learning process utilizes feedforward and backpropagation approaches. During the feedforward process, the BoW vector is passed through the model layers to produce probabilities for each class. Subsequently, the backpropagation process is used to update weights based on prediction errors during training.

2.5 Tourism Information System

A tourism information system is an information technology-based system used to provide and manage information regarding destinations, facilities, transportation, and supporting tourism services [14]. The development of digital technology encourages the utilization of AI, IoT, cloud computing, and other technologies within the concept of smart tourism [15]. Chatbots can be a component of tourism information systems by providing a conversational interface to access destination information. The use of chatbots in the tourism sector has been developed in various studies [16] [17].

2.6 Previous Studies

Previous studies indicate that chatbots have been used to provide tourism information through various approaches. Sanjaya and Winarno [1] developed a tourism information chatbot for Pati Regency using NLP based on Dialogflow. Other studies developed chatbots on tourism websites [18], chatbot rule-based [2], tourism information systems based on Rasa [3], and no-code chatbots utilizing a user-centered design approach [4].

These studies demonstrate the utilization of chatbots as a medium for tourism information interaction. The GoLotim study takes a different approach in system implementation by developing a chatbot with a JSON dataset containing tags, patterns, and responses, and subsequently using Bag of Words and a Neural Network to classify question patterns. The chatbot is integrated into a web-based tourism information system using ReactJS, Flask, and MongoDB. These differences form the basis for developing GoLotim as an NLP chatbot implementation for East Lombok Regency tourism information. The research evaluation focuses on the appropriateness of chatbot responses and system response time based on available testing scenarios.

III. RESEARCH METHODOLOGY

3.1 System Development Method

The system development method used is the System Development Life Cycle (SDLC) with the Waterfall model. This model consists of the requirement, analysis, design, coding/implementation, testing, and maintenance stages [19].

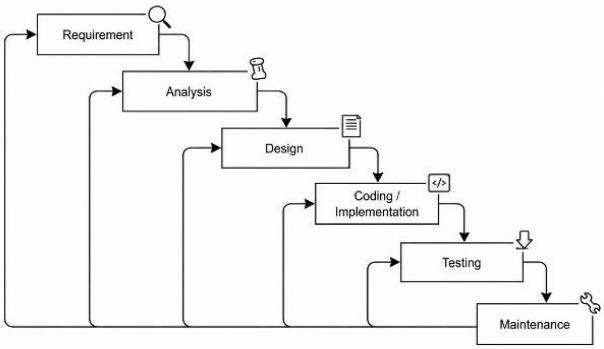


Figure 1. Software Development Life Cycle Phases.

1. Requirement
   Identifies system needs, including East Lombok tourism destination information, user needs, chatbot functions, and administrator information management needs
2. Analysis
   Determines business processes, data requirements, user-system interaction flows, and the NLP question processing mechanism
3. Design
   The design stage focuses on developing the system architecture, designing the database and user interface (UI), and modeling the system using

appropriate diagrams, including the Use Case Diagram, as illustrated in Figure 2.

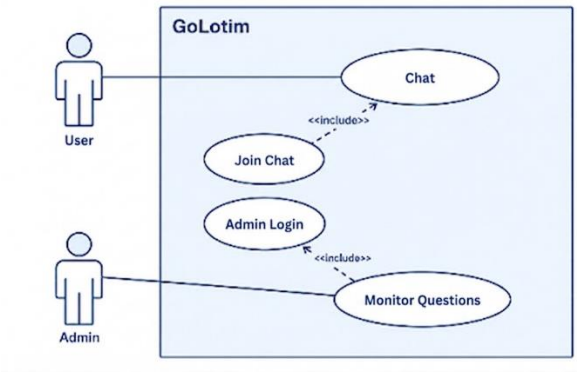


Figure 2. Usecase Diagram

In addition, the chatbot workflow is designed to illustrate the process from receiving user input to retrieving the corresponding answers from the knowledge base, as illustrated in Figure 3.

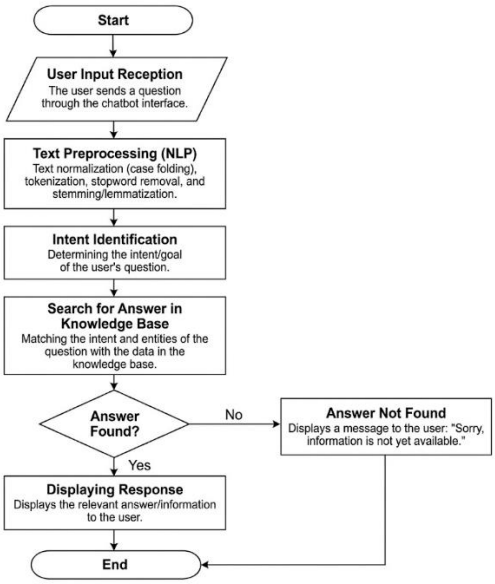


Figure 3. Chatbot Workflow

4. Coding/Implementation
   Implemented using ReactJS for the frontend, Flask for the backend, and MongoDB for the database. The chatbot module is implemented to receive questions, perform NLP processing, determine the tag, and provide a response.
5. Testing
   Conducted using Black Box Testing to verify system functions against established requirements. Chatbot testing involves 30 tourism information question scenarios. Parameters include response appropriateness and response time.
6. Maintenance
   Involves updating destination information, adding question patterns and responses to the knowledge base, fixing errors, and developing system functions.

3.2 Chatbot Dataset and Knowledge Base

The chatbot dataset was created manually and stored in JSON format. It provides question patterns as training input and responses to be delivered based on the question class. The dataset structure consists of three main components:

1. Tag, the label or category used by the model to determine the question class and response..
2. Patterns, a collection of question patterns representing user input variations for a specific tag
3. Responses, answers provided by the system based on the successfully predicted tag.

The data relationship can be described as: **Patterns → Tag → Responses**. The JSON dataset is used in model formation, while the tourism information displayed by the system is managed as part of the application database.

The chatbot dataset was manually compiled by the authors based on frequently asked questions (FAQs) from official East Lombok Regency tourism websites and local tourist information centers. The knowledge base is structured in JSON format, comprising 6 distinct intents (tags): *Destinasi* (Destination), *Lokasi* (Location), *Fasilitas* (Facilities), *Rute* (Travel Route), *Harga* (Ticket Price), and *Jam Operasional* (Operating Hours). Each intent contains an average of 7 training question patterns, resulting in a total of 42 training patterns. For evaluation, a separate test set of 30 scenarios was designed, strictly balanced with 5 unique, unseen natural language questions per intent to ensure unbiased performance measurement.

3.3 Text Preprocessing

User questions are processed through three main stages to ensure text consistency and readiness for numerical representation:

1. Tokenization: Breaks sentences down into individual word units or tokens, forming the basis for subsequent processing.
2. Case Folding: Converts all characters to lowercase to ensure text uniformity, preventing capitalization differences from creating distinct word representations.
3. Stemming: Converts words into their base form or lemma to normalize vocabulary variations. Since the GoLotim chatbot operates in Indonesian, this

study utilizes the Sastrawi library, which is specifically designed for Indonesian text normalization. Sastrawi systematically removes prefixes and suffixes (affixes) to retrieve the root word (e.g., converting "*berwisata*" to "*wisata*" or "*pantai-pantai*" to "*pantai*"). This normalization is crucial as it ensures that different morphological variations of the same word are recognized as identical tokens, thereby improving the accuracy of the subsequent Bag of Words representation and Neural Network classification.

3.10 Bag of Words

In this study, the Bag of Words (BoW) method is implemented to convert the preprocessed text from the JSON dataset into a numerical matrix. The process begins by extracting all unique words from the patterns array across all tags to build a fixed vocabulary. Each user question is then transformed into a binary vector of the same length as the vocabulary. A value of 1 is assigned if a specific vocabulary word exists in the user's question, and 0 if it is absent. To optimize computational efficiency and mitigate the risk of overfitting due to sparse vocabulary, this study applies a feature selection mechanism by retaining only the 128 most frequent words based on term frequency across the entire dataset. Less frequent words are excluded from the final vocabulary. This resulting numerical vector directly serves as the input for the Neural Network's input layer (which is designed with 128 nodes, corresponding to the selected vocabulary size). This implementation ensures that the unstructured natural language input is successfully mapped into a structured format that the Neural Network can compute during the feedforward process.

3.5 Model Neural Network

The classification model used is a Neural Network, consisting of three layers as shown in Table 1.

Table 1. Neural Network Architecture

Layer	Number of Nodes	Activation Function
Input Layer	128	-
Hidden Layer	64	ReLU
Output Layer	Follows number of classes (tags)	Softmax

The model uses feedforward and backpropagation mechanisms. During feedforward, the BoW vector is passed through the input and hidden layers to produce an output at the output layer. The Softmax function generates the probability of each class. The class with the highest probability is used as the predicted tag, which subsequently determines the appropriate response.

3.6 Response Selection Mechanism

The chatbot process begins when the user inputs a question. The question undergoes preprocessing and is converted into a BoW vector. This vector is fed to the Neural Network model to predict the tag. The predicted tag is used to select the corresponding response from the dataset, which is then sent back to the user via the chatbot interface. The overall flow is: **User Input → Tokenization → Case Folding → Stemming → Bag of Words → Neural Network → Tag Prediction → Response Selection → Chatbot Output.**

3.7 Software Implementation

The system is developed using ReactJS as the frontend, Flask as the backend, and MongoDB as the database. ReactJS builds the application interface, Flask handles application logic and requests, and MongoDB stores system data.

3.1 Testing and Evaluation Parameters

Testing is conducted in two forms: Black Box Testing for application functions and chatbot testing using question scenarios. The chatbot testing uses 30 scenarios covering: (1) destination names; (2) locations; (3) facilities; (4) travel routes; (5) ticket prices; and (6) operating hours. A response is deemed appropriate if the output matches the expected information based on the question category (tag) and dataset responses. It is inappropriate if it fails to provide the expected information. The response success rate is calculated based on the 30 scenarios, where 28 yielded appropriate responses.

An 'appropriate response' is operationally defined as a two-step success: (1) the Neural Network correctly predicts the user's intended tag (intent classification accuracy), and (2) the system retrieves a response from the JSON knowledge base that factually answers the user's query. Response time was measured programmatically using Python's time.time() function within the Flask backend, capturing the cumulative duration from the moment the HTTP request is received until the final JSON response is dispatched. Measurements were recorded over 30 consecutive trials

under stable local network conditions to calculate the average, minimum, maximum, and standard deviation.

IV. RESULTS AND DISCUSSION

4.1 System Implementation

The GoLotim system was developed as a web-based tourism information system integrating the tourism information interface with chatbot services. The backend uses Flask, while the interface is built using ReactJS supported by Tailwind CSS. MongoDB serves as the application database. This integration allows users to access tourism information via web pages and submit questions through the chatbot.

4.2 NLP Implementation

The NLP implementation in GoLotim follows the flow: **User Question → preprocessing → Bag of Words → Neural Network → tag prediction → response.** Initially, the user inputs a question via the chatbot interface. The question is processed through tokenization, case folding, and stemming. The results are converted into a vector using Bag of Words. The BoW vector inputs the Neural Network model, which predicts the class or tag. The system then uses this tag to determine the appropriate response. This approach enables the system to handle various tourism information categories present in the dataset, such as destination locations, facilities, ticket prices, operating hours, and travel routes.

4.3 System Interface

1. Main Page

The main page functions as the initial landing page of the tourism information system, providing navigation to destination information and chatbot services. The interface is designed to grant users direct access to tourism information and conversational features.

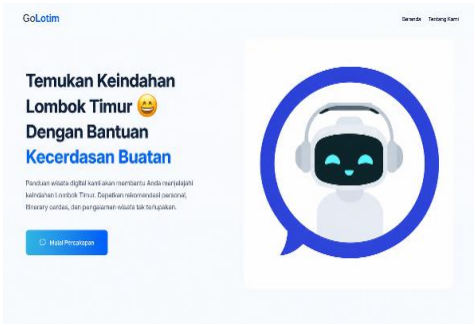


Figure 4. Main Page Display

2. Chatbot Interface

The chatbot page provides an interaction space between the user and the system. Users can input questions in natural language regarding available tourism information. The questions are processed by the NLP module, and responses are displayed on the conversational interface.

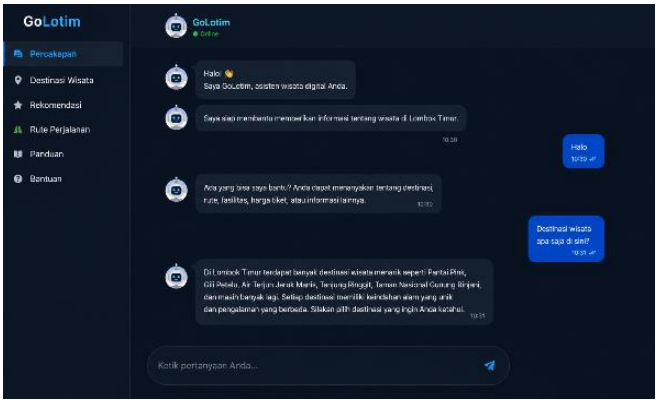


Figure 5. Chatbot Interface Display

4.4 Black Box Testing Results

Black Box Testing was conducted to verify whether the system's functional features operate in accordance with the specified requirements. Unlike chatbot testing, which focuses on the quality of NLP responses, Black Box Testing evaluates the overall functionality of the GoLotim web-based tourism information system, including authentication, tourism destination data management (CRUD operations), page navigation, and the integration of the chatbot module with the main system. A total of **25 functional testing scenarios** were designed, covering aspects such as: (1) admin login and logout functionality; (2) adding new tourism destination data; (3) editing existing destination information; (4) deleting destination data; (5) displaying destination details on the main page; (6) navigating between different pages (home, destinations, about, contact); (7) opening the chatbot interface from the main page; (8) submitting queries through the chat widget; and (9) displaying chatbot responses correctly. A scenario is considered successful if the system feature functions as expected based on the predefined requirements (e.g., data is saved correctly, pages are displayed properly, and buttons are responsive).

Table 2. Black Box Testing Recapitulation

Parameter	Result
Scope of testing	System functional features (authentication, data management, navigation, chatbot integration)
Number of	25

testing scenarios	
Functions Appropriate	23
Functions Inappropriate	2
Success rate	92%

The results indicate that 23 out of 25 functional scenarios operated as required, while 2 scenarios encountered minor issues (e.g., UI alignment inconsistencies on certain screen resolutions and delayed page loading on slow network connections). These results confirm that the system's functional features are largely ready for deployment. However, it is important to note that Black Box Testing does not measure the quality of NLP classification; it only verifies whether the chatbot interface can be opened and whether user input can be submitted and processed by the system.

4.5 Chatbot Testing Results

In contrast to Black Box Testing, Chatbot Testing specifically evaluates the quality of the NLP module in producing appropriate responses to user queries. This testing focuses on the accuracy of the Neural Network model in predicting the correct tag based on the user's input, and whether the selected response matches the expected tourism information. A total of 30 question scenarios were designed, covering six main categories: (1) destination names; (2) destination locations; (3) facilities; (4) travel routes; (5) ticket prices; and (6) operating hours. Each scenario consists of a natural language question that is intentionally varied in structure and vocabulary to simulate real user queries. A response is considered appropriate if the chatbot output provides information matching the expected answer for the tested question category (tag).

Table 3. Chatbot Testing Recapitulation

Parameter	Result
Scope of testing	NLP response quality (tag prediction accuracy and response appropriateness
Number of quastion scenarios	30
Appropriate responses	28
Inappropriate responses	2
Response success rate	93,3%
Average response time	1.2 second

Out of 30 question scenarios, 28 yielded appropriate responses, and 2 did not, resulting in a scenario-based response success rate of 93.3%. The average response time recorded was 1.2 seconds, representing the cumulative time for request reception, NLP preprocessing, model prediction, response selection, and result delivery. The two inappropriate responses indicate that the system still has limitations in handling question variations that fall outside the patterns available in the JSON dataset.

4.6 Chatbot Response Analysis

The two scenarios yielding inappropriate responses indicate that the system still has limitations in handling question variations. To provide a transparent evaluation, Table 5 presents a representative sample of the 30 testing scenarios, explicitly illustrating the two misclassified cases alongside successful predictions.

Table 4. Representative Chatbot Testing Scenarios and Error Classification

No	Intent (Actual)	Test Question (Scenario)	Predicted Intent	Status	Error Type & Cause Analysis
1	Destination	"Pantai apa saja yang ada di Lombok Timur?"	Destination	Correct	-
2	Location	"Di desa mana letak Air Terjun Jeruk Manis?"	Location	Correct	
3	Facilities	"Apakah tersedia kamar mandi di Pantai Pink?"	Facilities	Correct	
4	Route	"Bagaimana cara pergi ke Gunung Rinjani via Sembalun?"	Route	Correct	

5	Price	"Berapa tiket masuk untuk wisatawan domestik?"	Price	Correct	
6	Operating Hours	"Pukul berapa wisata Desa Adat Sasak buka?"	Operating Hours	Correct	
-	...	*(Scenarios 7-27 yielded correct responses)*	...	Correct	
28	Facilities	"Apakah ada tempat istirahat di sekitar pantai?"	Destination	Incorrect	False Negative: The word "pantai" (beach) has a high term frequency in the *Destination* training patterns, overpowering the less frequent word "tempat istirahat" (rest area) in the BoW feature vector.
29	Route	"Jalan menuju air terjun dari arah timur lewat mana?"	Location	Incorrect	False Negative: The phrase "air terjun" (waterfall) strongly activates the *Location* node. The model failed to recognize the directional phrase "Jalan menuju... lewat mana" as a distinct *Route* indicator, as this syntactic variation was absent from the 42 training patterns.
30	Price	"Apakah ada diskon untuk rombongan pelajar?"	Price	Correct	

A deeper analysis of the two misclassified scenarios (Table 5, No. 28 and 29) reveals the inherent limitations of the Bag of Words (BoW) representation when faced with unseen vocabulary variations. In scenario 28, the high term frequency of the word "pantai" in the *Destination* training data overpowered the less frequent word "tempat istirahat," causing a False Negative for the *Facilities* intent. Similarly, in scenario 29, the strong lexical match for "air terjun" triggered the *Location* intent, while the model failed to recognize the directional phrase "Jalan menuju... lewat mana" as a distinct *Route* indicator. This confirms that while the Neural Network architecture functions correctly, the model's robustness is strictly bounded by the lexical coverage of the 128-feature vocabulary. Expanding the training patterns with diverse synonyms and syntactic structures is the most viable mitigation strategy.

4.7 Response Time Analysis

The response time obtained from testing represents the cumulative duration of system processes, including request reception, NLP preprocessing, model prediction, response selection, and result delivery to the user. To provide a comprehensive statistical evaluation, response time measurements were recorded for all 30 testing scenarios. As shown in Table 5, the average response time is 1.20 seconds. The minimum recorded time was 0.85 seconds, while the maximum was 1.65 seconds, with a standard deviation of 0.22 seconds. The relatively low standard deviation indicates that the system's response time is stable and consistent across different types of queries and intents. These statistical metrics confirm that the custom-built NLP architecture is lightweight and efficient enough to support real-time interactive communication without causing significant latency for the user.

Table 5. Chatbot Response Time Statistics (n=30)

Metric	Value (seconds)
Average (Mean)	1.20
Minimum	0.85
Maximum	1.65
Standard Deviation	0.22

4.8 Discussion on NLP Implementation

To provide a rigorous technical evaluation of the Neural Network's classification performance, this study goes beyond scenario-based success rates and calculates standard NLP classification metrics: Precision, Recall, F1-Score, and the Confusion Matrix. These metrics are derived from the 30 testing scenarios to measure how accurately the model predicts the correct intent (tag) across all classes.

Table 6 illustrates the Confusion Matrix of the chatbot's intent classification. The values on the main diagonal (bolded) represent the True Positives (correct predictions), while the off-diagonal values represent misclassifications (False Positives and False Negatives). As shown, the model perfectly classified the *Destination*, *Location*, *Price*, and *Operating Hours* intents. However, misclassifications occurred between *Facilities* and *Destination*, as well as between *Route* and *Location*, which directly corresponds to the two inappropriate responses analyzed in Section IV.F.

Table 6. Confusion Matrix of Chatbot Intent Classification (n=30)

Actual/ Predicted	Destination	Location	Facilities
Destination	5	0	0
Location	0	5	0
Facilities	1	0	4
Route	0	1	0
Price	0	0	0
Operating Hours	0	0	0

Based on the Confusion Matrix, the detailed classification metrics per class are computed and presented in Table 7. Precision measures the model's exactness (when it predicts a class, how often is it correct?), while Recall measures the model's completeness (how many actual instances of a class did it find?). The F1-Score provides the harmonic mean of both.

Table 7**.** Classification Report (Precision, Recall, F1-Score)

Intent Class	Precision	Recall	F1-Score
Destination	0.83	1.00	0.91
Location	0.83	1.00	0.91
Facilities	1.00	0.80	0.89
Route	1.00	0.80	0.89
Price	1.00	1.00	1.00
Operating Hours	1.00	1.00	1.00
Macro Average	0.94	0.93	0.93
Overall Acuracy	0.933 (93.3%)	30

The evaluation yields an Overall Accuracy of 93.3%, aligning with the scenario-based success rate. More importantly, the model achieved a Macro-Average Precision of 0.94, Recall of 0.93, and F1-Score of 0.93. The slight decrease in Recall for the *Facilities* and *Route* classes (0.80) is a direct mathematical reflection of the two False Negative errors caused by the Bag of Words (BoW) vocabulary limitations, as previously discussed. Meanwhile, the Precision for *Destination* and *Location* dropped to 0.83 because they absorbed the False Positives from the misclassified *Facilities* and *Route* queries.

Overall, these quantitative metrics confirm that the custom-built Neural Network architecture is highly effective for this specific domain. The results validate that while the NLP pipeline (preprocessing → BoW → Neural Network) functions robustly, its generalization capability is strictly bounded by the lexical diversity of the 42 training patterns. Expanding the dataset with more syntactic variations remains the primary strategy to push the Recall and F1-Score closer to 1.0 in future deployments.

4.9 Comparison with Previous Studies

Revious studies have demonstrated various approaches to developing tourism information chatbots. Sanjaya and Winarno [1], for instance, developed a tourism information chatbot using NLP based on Dialogflow. Other studies used rule-based approaches [2], Rasa [3], or no-code platforms [4]. While these studies confirm the effectiveness of chatbots in the tourism domain, each approach carries distinct technical and operational trade-offs. To contextualize the contribution of GoLotim, **Table 8** presents a theoretical comparison across six critical dimensions: data privacy, deployment cost, contextual understanding, dataset control, maintenance effort, and suitability for local tourism.

Table 8. Theoretical Comparison of Chatbot Implementation Approaches in Tourism

Feature /Aspect	**GoLotim (Custom BoW + Neural Network)**	**Dialogflow/ Rasa (Pre-built Framework)**	**Rule-Based/ Keyword Matching**
Data Privacy & Sovereignty	High (Fully local; no third-party API dependency)	Low to Medium (Depends on cloud provider and data policy)	High (Fully local)
Deployment Cost	Low (Open-source stack; no licensing fees)	Medium to High (Potential API quotas, hosting, or subscription costs)	Low
Contextual Understanding	Medium (Learns question patterns via Neural Network)	High (Advanced NLU, entity extraction, context management)	Low (Strict exact-match or simple pattern rules)
Dataset Control & Flexibility	High (Custom JSON structure; fully tailored to local terminology)	Medium (Constrained by platform schema, intents, and entities)	High
Maintenance Effort	Medium (Requires manual pattern and vocabulary expansion)	Low (Platform handles underlying NLP and model updates)	High (Requires manual rule updates for every variation)
Suitability for Regional Tourism	High (Tailored, cost-effective, privacy-preserving, and lightweight)	Medium (Often overkill for simple FAQ; dependent on internet connectivity)	Medium (Fails on natural language variation)

As shown in **Table 8**, pre-built frameworks such as Dialogflow and Rasa offer superior out-of-the-box Natural Language Understanding (NLU) capabilities, including advanced entity extraction and context management. However, they introduce dependencies on third-party services, potential licensing costs, and constraints imposed by platform-specific schemas. In contrast, rule-based systems provide full data sovereignty but lack the flexibility to handle natural language variations, making them unsuitable for diverse user queries.

The custom-built GoLotim implementation occupies a strategic middle ground. While its contextual understanding is bounded by the 128-dimensional BoW vocabulary and the 42 training patterns, it provides three decisive advantages for regional tourism systems like East Lombok Regency: **(1) absolute data sovereignty**, as no user query is transmitted to external servers; **(2) zero licensing costs**, making it sustainable for local government platforms with limited budgets; and **(3) complete flexibility** to tailor the knowledge base to highly localized terminology (e.g., Sasak cultural terms, local

destination names) without being constrained by the rigid architectural limitations of external frameworks.

It is important to note that this study does not claim statistical superiority of the GoLotim model over Dialogflow or Rasa in terms of large-scale NLU benchmarks. Rather, the contribution lies in demonstrating that a lightweight, privacy-preserving, and fully customizable NLP chatbot can be successfully implemented for regional tourism information services using open-source technologies, offering a viable alternative for contexts where data sovereignty, cost efficiency, and localized domain control are prioritized over advanced conversational capabilities.

4.10 Research Limitations

This study has several limitations that should be acknowledged:

1. **Dataset Size and Coverage.** The chatbot knowledge base was manually compiled, consisting of only 6 intents and 42 training question patterns. Although classification metrics (Precision, Recall, F1-Score) were successfully computed for the 30 testing scenarios, this relatively small dataset limits the model's ability to generalize to a broader population of real-world user queries. The two inappropriate responses observed during testing directly reflect this lexical coverage limitation, where unseen vocabulary variations outside the 128-feature Bag of Words (BoW) vocabulary led to misclassification. Future research should expand the training dataset with diverse synonyms, syntactic variations, and localized East Lombok terminology to improve the model's robustness.
2. **Absence of User-Based Evaluation.** The study evaluated the chatbot exclusively through scenario-based functional testing and technical classification metrics. It did not conduct direct user evaluation using established instruments such as the System Usability Scale (SUS) or User Experience Questionnaire (UEQ). Therefore, this study cannot draw quantitative conclusions regarding user satisfaction, perceived ease of use, or actual improvement in user experience. User-centered evaluation is recommended as a critical next step before large-scale deployment.
3. **Response Time Measurement Conditions.** The reported response time statistics (Mean: 1.20s, Min: 0.85s, Max: 1.65s, Std. Dev: 0.22s) were measured under stable local network conditions over 30 consecutive trials using Python's time.time() function. Real-world performance may vary depending on server load, concurrent user traffic, client device specifications, and network latency. Future studies should conduct stress testing and measurement under heterogeneous deployment conditions.
4. **Scope of Testing Scenarios.** Black Box Testing employed 25 functional scenarios to verify overall system functionality (authentication, data management, navigation, and chatbot integration), while Chatbot Testing employed 30 question scenarios specifically designed to evaluate NLP response quality. The difference in scenario count reflects the distinct objectives of each testing phase rather than an inconsistency in methodology.
5. **Architectural Constraints of the BoW Representation.** The Bag of Words approach, while computationally efficient and interpretable, does not capture word order, semantic relationships, or contextual meaning. This inherent limitation explains why structurally different questions with overlapping keywords (e.g., "tempat istirahat di pantai" being misclassified as *Destination* instead of *Facilities*) may yield suboptimal predictions. Exploring more advanced text representation methods such as Word Embeddings (Word2Vec, FastText) or transformer-based models in future research could address this limitation at the cost of increased computational complexity.

These limitations form the basis for future research directions, particularly in expanding the dataset, conducting user-based evaluation, deploying the system under real-world conditions, and exploring more context-aware NLP architectures.

V. CONCLUSION

This study successfully developed "GoLotim," a custom-built Natural Language Processing (NLP) chatbot integrated into a web-based tourism information system for East Lombok Regency. The system utilizes a manually compiled JSON knowledge base and processes user queries through tokenization, case folding, and stemming (using the Sastrawi library), followed by a 128-dimensional Bag of Words (BoW) representation. The intent classification is performed by a Neural Network architecture consisting of a 128-node input layer, a 64-node hidden layer with ReLU activation, and a Softmax output layer corresponding to the number of intent classes.

The evaluation results demonstrate the technical feasibility of this custom NLP implementation. Out of 30 predefined testing scenarios covering six intents

(destination, location, facilities, route, price, and operating hours), the chatbot achieved a scenario-based success rate of 93.3% (28 appropriate responses, 2 inappropriate). Technical classification metrics yielded a Macro-Average Precision of 0.94, Recall of 0.93, and F1-Score of 0.93, supported by Confusion Matrix analysis. The system also exhibited stable performance with an average response time of 1.20 seconds (Min: 0.85s, Max: 1.65s, Std. Dev: 0.22s). The two misclassifications were identified as False Negatives caused by unseen vocabulary variations outside the limited training patterns, highlighting that the model's robustness is currently bounded by the lexical coverage of the dataset.

The primary contribution of this study lies in demonstrating that a lightweight, privacy-preserving, and fully customizable NLP chatbot can be effectively implemented for regional tourism without relying on third-party APIs (e.g., Dialogflow or Rasa). This approach ensures absolute data sovereignty, zero licensing costs, and complete flexibility in tailoring the knowledge base to highly localized East Lombok terminology.

However, this study acknowledges limitations regarding the relatively small dataset size (6 intents, 42 training patterns) and the absence of direct user-based evaluation (e.g., System Usability Scale or User Experience Questionnaire). Therefore, the current results primarily validate the system's functional and technical classification performance under controlled testing scenarios, rather than claiming broad improvements in user satisfaction. Future research should focus on expanding the training dataset with diverse syntactic variations, exploring more advanced text representation methods (e.g., Word Embeddings), and conducting comprehensive user-centered evaluations to assess real-world usability prior to large-scale deployment.

REFERENCE

[1] R. A. Sanjaya and E. Winarno, "Pengembangan Chatbot Informasi Pariwisata di Kabupaten Pati Menggunakan Metode Natural Language Processing Berbasis Dialogflow," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 13, no. 1, p. 368, 2024, doi: 10.35889/jutisi.v13i1.1828.

[2] M. Bengi and C. R. Gunawan, "Pengembangan Chatbot Rule-Based untuk Rekomendasi Wisata Berbasis Content-Based Filtering," *Algoritm. J. Ilmu Komput. Inform.*, vol. 9, no. 2, pp. 146–155, 2025, doi: 10.30829/algoritma.v9i2.27904.

[3] S. Villasari and J. S. Wibowo, "Sistem Informasi Wisata Kab. Semarang melalui Chatbot dengan Framework Rasa," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 1, pp. 208–215, 2024, doi: 10.29100/jipi.v9i1.4453.

[4] R. T. R. . Bau, H. A, S. Suhada, M. R. P. F. Idris, and R. K. Batalipu, "Rancang Bangun Chatbot Budaya Berbasis Web Menggunakan Platform No-Code dengan Pendekatan User Centered Design," *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 8, no. 2, pp. 236–249, 2025, doi: 10.57093/jisti.v8i2.343.

[5] W. N. Akirini *et al.*, "Perancangan Aplikasi Layanan Customer Service Menggunakan Chatbot Berbasis Website Pada Pt. Telekomunikasi Indonesia Witel Cirebon," *J. Digit*, vol. 14, no. 1, p. 79, 2024, doi: 10.51920/jd.v14i1.374.

[6] G. Guntoro, Loneli Costaner, and L. Lisnawita, "Aplikasi Chatbot untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML)," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 291–300, 2020, doi: 10.31849/digitalzone.v11i2.5049.

[7] R. Riannada and S. Mardliyah, "Peran Kelompok Sadar Wisata (POKDARWIS) Kencana Dalam Pengembangan Desa Wisata Adat Osing Kemiren," *J+Plus Unesa J. Mhs. Pendidik. Luar Sekol.*, vol. 10, no. 1, pp. 195–204, 2021, [Online]. Available: https://www.academia.edu/download/71404072/33103.pdf

[8] R. A. Azzah, Z. F. Mardhiah, A. Kusma, and R. Mubarok, "Literature Review Chatbot E-Commerce," *Socius J. Penelit. Ilmu-Ilmu Sos.*, vol. 3, no. 6, pp. 13–30, 2026.

[9] A. Puspitasari, A. N. Paradhita, Y. W. Tineka, V. Sulistyowati, N. K. S. Noriska, and Haryanto, "Natural Language Processing (NLP) Technology for Chatbot Website," *J. Penelit. Pendidik. IPA*, vol. 10, no. SpecialIssue, pp. 319–324, 2024, doi: 10.29303/jppipa.v10ispecialissue.8241.

[10] Robet Robet, Johanes Terang Kita Perangin Angin, and Randy Wilson, "Perancangan Aplikasi Web Chatbot Multi-Bahasa Berbasis NPL Translator API Dengan Multibahasa Terjemahan," *J. Ilm. Tek. Mesin, Elektro dan Komput.*, vol. 5, no. 1, pp. 108–119, 2025, doi: 10.51903/juritek.v5i1.4150.

[11] F. L. D. Cahyanti and R. D. A. Raya, "Perancangan Sistem Informasi Chatbot Retrieval Augmented Generation Berbasis Website Pada PT. Revolusi Cita Edukasi," *Indones. J. Comput. Sci.*, vol. 4, no. 1, pp. 15–21, 2025, doi: 10.31294/m75d4782.

[12] H. Shih, K. Juluru, K. Murthy, and P. Elnajjar, "Bag-of-Words Technique in Natural Language Processing : A Primer for Radiologists," *RadioGraphics*, vol. 41, no. 5, pp. 1420–1426, 2021.

[13] P. N. Zivich and A. I. Naimi, "A Primer on Neural Networks," *Am. J. Epidemiol.*, vol. 194, pp. 1473–1475, 2025, doi: https://doi.org/10.1093/aje/kwae380.

[14] M. A. N. Rachmad and N. D. Saputro, "Sistem Informasi Manajemen Pariwisata Berbasis Web Di

Kabupaten Kendal," *In-Fest 2023 Semin. Nas. Inform. UPGRIS*, vol. 2023, pp. 803–815, 2023.

[15] B. H. Ye, H. Ye, and R. Law, "Systematic review of smart tourism research," *Sustain.*, vol. 12, no. 8, pp. 1–15, 2020, doi: 10.3390/SU12083401.

[16] M. A. Camilleri and C. Troise, *Chatbot recommender systems in tourism: A systematic review and a benefit-cost analysis*, vol. 1, no. 1. Association for Computing Machinery, 2023. doi: 10.1145/3589883.3589906.

[17] D.-H. Kim, H.-S. Im, J.-H. Hyeon, and J.-W. Jwa, "Development of the Rule-based Smart Tourism Chatbot using Neo4J graph database," *Int. J. Internet, Broadcast. Commun.*, vol. 13, no. 2, pp. 179–186, 2021, [Online]. Available: http://dx.doi.org/10.7236/IJIBC.2021.13.2.179

[18] S. Suryayusra and W. Maula, "Penerapan Chatbot pada Website Pariwisata Palembang untuk Membantu Masyarakat dalam Mengakses Informasi Pariwisata Kota Palembang," *J. Pengabdi. Masy. Bangsa*, vol. 3, no. 4, p. 1251, 2025, [Online]. Available: http://jurnalpengabdianmasyarakatbangsa.com/index.php/jpmba/article/view/2386/1783

[19] M. Ridwan, I. Fitri, and B. Benrahman, "Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 5, no. 2, pp. 173–184, 2021, doi: 10.35870/jtik.v5i2.209.