Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1002

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Design and Development of a Web-Based Educational Chatbot Using Natural Language Processing for Public Information Services at SMK Negeri 2 Padang

Rani Nabilla Putri[1*]✉, Elfi Tasrif[1], Ahmaddul Hadi[1], Rizkayeni Marta[1]
[1] Department of Electronics, Faculty of Engineering, Universitas Negeri Padang, Padang, Indonesia
Corresponding Author: raninabillaputri26@gmail.comi

Article Information

Article history:

No. 1002
Rec. August 15, 2025
Rev. April 23, 2026
Acc. April 30, 2026
Pub. May 05, 2026
Page. 1526 – 1544

Keywords:

- *Educational Chatbot*
- *Natural Language Processing*
- *Web Application*
- *TF-IDF*
- *Cosine Similarity*

ABSTRACT

This study addresses the limited efficiency of public information services in vocational schools, which often results in delayed responses and repetitive administrative workload. This research aims to design and develop a web-based educational chatbot using Natural Language Processing (NLP) to improve information accessibility at SMK Negeri 2 Padang. The system was developed using the Waterfall model and implements text preprocessing, TF-IDF vectorization, and cosine similarity for intent recognition. System evaluation was conducted through Black Box Testing and accuracy measurement based on user queries. The results show that the system achieved a 100% success rate in functional testing and 91% accuracy in intent classification, indicating its effectiveness in providing relevant and real-time information. This study contributes by offering a practical, scalable, and user-friendly NLP-based solution to enhance public information services in educational institutions.

How to Cite:

Putri, R. N., & et al. (2026). Design and Development of a Web-Based Educational Chatbot Using Natural Language Processing for Public Information Services at SMK Negeri 2 Padang. Jurnal Teknologi Informasi dan Pendidikan, 19(2), 1526-1544. https://doi.org/10.24036/jtip.v19i2.1002

1. INTRODUCTION

The rapid advancement of artificial intelligence and Natural Language Processing (NLP) has significantly transformed the way educational institutions deliver information services on a global scale. Educational chatbots have been widely adopted as virtual assistants to provide instant responses, improve communication efficiency, and support personalized learning experiences [1], [3]. Their ability to deliver real-time, interactive, and accessible services has made them an increasingly important component in modern digital education systems.

Previous studies have demonstrated the effectiveness of chatbot systems in educational contexts. For instance, intent-classification-based chatbots have achieved high accuracy in understanding student queries related to academic and administrative information [2]. Other studies have shown that chatbot implementation can enhance student motivation, participation, and independent learning [4], [7]. Additionally, web-based chatbot systems have been proven to reduce administrative workload and improve the efficiency of information delivery in educational institutions [5], [6].

Despite these advancements, several limitations remain. Many existing chatbot systems rely on complex machine learning or deep learning approaches that require large datasets and high computational resources, making them less suitable for implementation in resource-constrained environments such as vocational schools. Furthermore, prior research tends to focus primarily on performance metrics, such as accuracy, with limited attention to practical aspects, including system accessibility, ease of maintenance, and real-world deployment in school information services.

In the context of vocational high schools in Indonesia, including SMK Negeri 2 Padang, public information services are still largely managed manually through teachers or administrative staff. This approach often results in delayed responses, limited service availability, and repetitive workloads. These conditions indicate a gap between the potential of chatbot technology and its practical implementation in local educational environments.

Therefore, this study proposes the design and development of a web-based educational chatbot utilizing NLP techniques that are efficient, practical, and suitable for real-world implementation. Unlike prior studies, this research emphasizes a lightweight NLP approach using TF-IDF and cosine similarity combined with synonym normalization, enabling accurate intent recognition without requiring large-scale training data. In addition, the system integrates an admin dashboard for dynamic knowledge base management and implements security features such as bcrypt hashing and Google reCAPTCHA.

The main contribution and novelty of this study lie in the integration of a confidence score feature within the admin interface, which allows administrators to evaluate the accuracy of each chatbot response and use it as a basis for continuous system improvement. This approach not only enhances system transparency but also supports iterative development based on real usage data. Therefore, the proposed system offers a practical, scalable, and resource-efficient solution to improve public information services in vocational education, particularly in supporting the implementation of the Merdeka Curriculum.

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1002

2. RESEARCH METHOD

This study adopted the Waterfall model as the software development methodology due to its structured and sequential approach, which is suitable for systems with clearly defined requirements [10]. The development process consists of five main stages: requirements analysis, system design, implementation, testing, and evaluation.

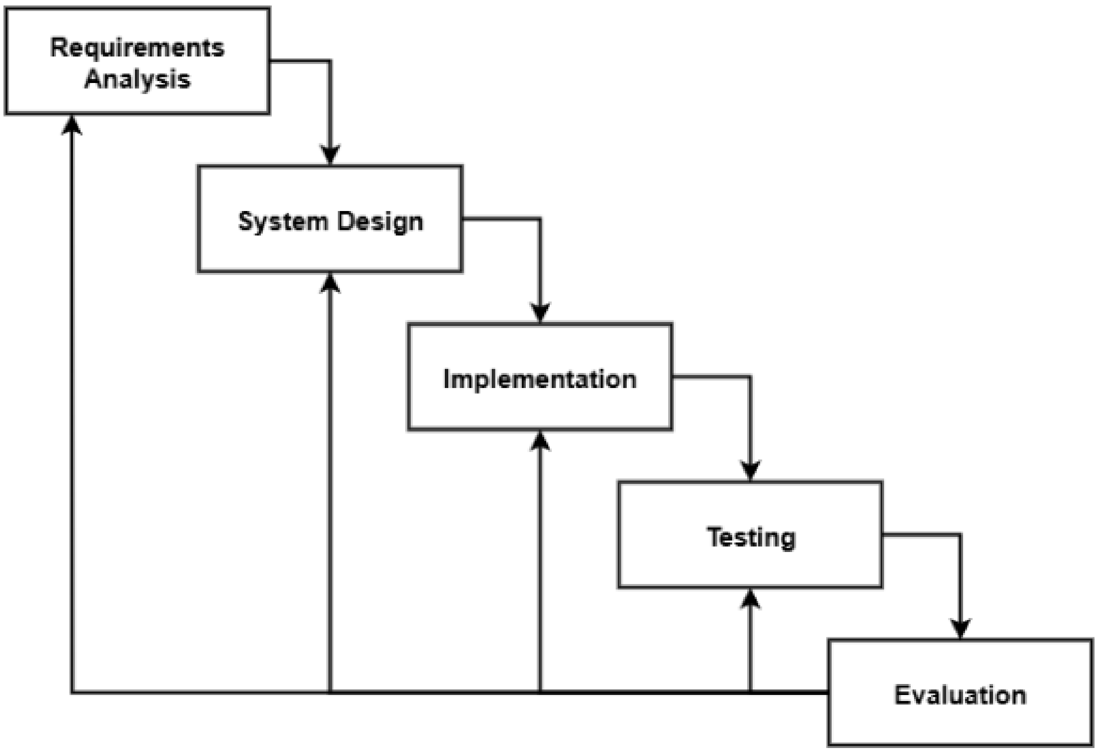

Figure 1. Waterfall Model

To enhance the clarity of the system workflow, a flowchart of the proposed chatbot system is presented in Figure 2. The flowchart illustrates the overall process of the system, including user interaction, Natural Language Processing (NLP), intent recognition, and response generation.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

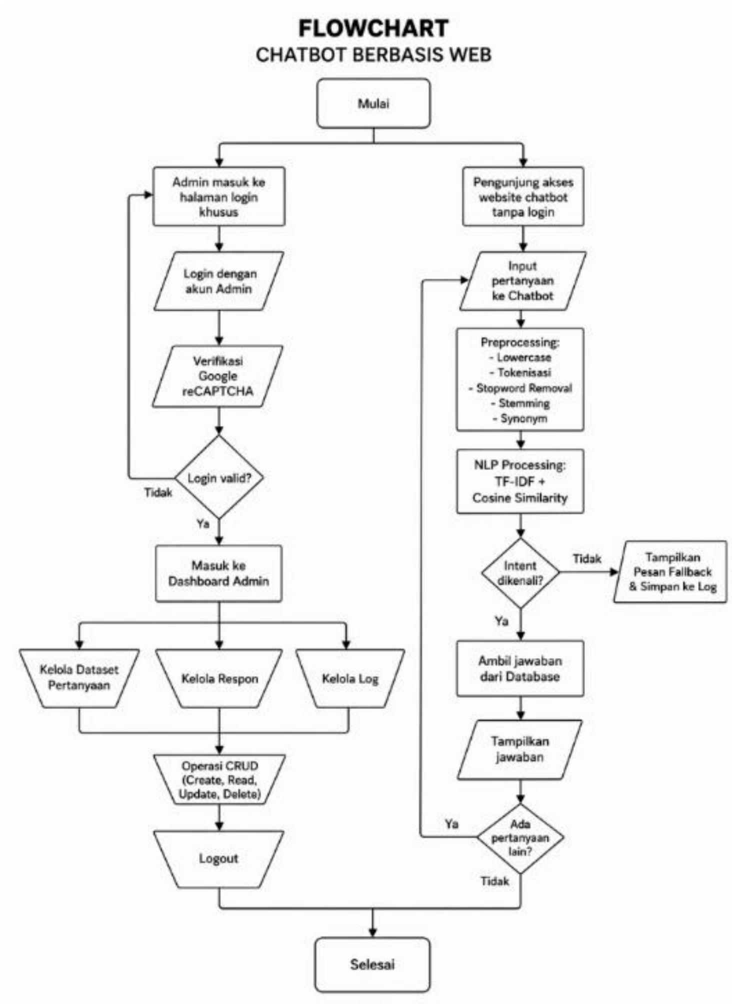

Figure 2. Flowchart of the Chatbot System

2.1. Requirements Analysis

The requirements analysis stage was conducted through observation and interviews with administrative staff at SMK Negeri 2 Padang. This stage aimed to identify both functional and non-functional requirements of the system.

Functional requirements include processing user queries in natural language, performing intent classification using NLP techniques, managing chatbot data (intents, patterns, responses, and synonyms) through an admin dashboard, and generating automated responses. Non-functional requirements include system security using bcrypt password hashing and Google reCAPTCHA, system responsiveness, and public accessibility without requiring user authentication [11].

2.2. System Design

The system architecture consists of three main components: (1) a web-based user interface, (2) an NLP processing module, and (3) a MySQL database. The user interface facilitates interaction between users and the chatbot, while the NLP module processes input text and identifies user intent. The database stores patterns, intents, responses, and interaction logs.

To represent system structure and workflow, Unified Modeling Language (UML) diagrams were used, including use case, activity, sequence, and class diagrams [12]. Additionally, the NLP pipeline was designed to include preprocessing steps such as case folding, tokenization, stopword removal, stemming, and synonym normalization [13].

2.3. Implementation

The system was implemented using PHP with the CodeIgniter 4 framework for backend development and Bootstrap for frontend design. The chatbot utilizes TF-IDF vectorization to represent text data and cosine similarity to measure the similarity between user queries and stored patterns.

Text preprocessing was implemented using the Sastrawi library for stemming and a custom synonym normalization module. This approach ensures that variations in user input can still be matched effectively with the knowledge base [14], [15].

2.4. Testing

System testing was carried out using Black Box Testing to ensure that all system functionalities operated according to the specified requirements. In addition, usability testing was conducted using the GMetrix platform to evaluate system performance and user experience. Performance indicators measured include page load speed, interface responsiveness, and interaction efficiency, using metrics such as Largest Contentful Paint (LCP) and Total Blocking Time (TBT) [16].

2.5. Evaluation

System evaluation focused on measuring the accuracy of chatbot responses using a confidence score derived from cosine similarity calculations. The system selects the highest similarity value to determine the most relevant intent. Interaction logs were also analyzed to identify unmatched queries and improve the chatbot's knowledge base. This evaluation process enables continuous system refinement and ensures that the chatbot maintains competitive performance compared to related studies [17].

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

3. RESULTS AND DISCUSSION

This section presents the results of the design, development, and evaluation of the web-based educational chatbot utilizing Natural Language Processing (NLP) for public information services at SMK Negeri 2 Padang. The discussion not only describes system performance but also provides analytical insights by comparing the findings with related studies and evaluating system effectiveness in real-world implementation.

3.1. System Architecture and Components

The developed system is a web-based educational chatbot designed to provide automated public information services at SMK Negeri 2 Padang through text-based interaction. The architecture integrates a user interface, a Natural Language Processing (NLP) module, and a database system to support efficient intent recognition and response generation.

The system consists of three main components. First, the user interface is implemented as a responsive web application, enabling users to access the chatbot without authentication, while administrators manage the system through a dedicated dashboard. Second, the NLP module processes user input through text preprocessing and similarity-based matching to identify user intent. Third, the database stores intents, patterns, responses, synonyms, and interaction logs, supporting both system operation and continuous improvement.

The preprocessing module plays a key role in normalizing user input to improve matching accuracy. This includes case folding, tokenization, stopword removal, stemming, and synonym normalization. By reducing linguistic variations, the system ensures more consistent input representation, which enhances the effectiveness of the similarity matching process.

The similarity matching engine applies TF-IDF vectorization and cosine similarity to measure the relevance between user queries and stored patterns. A predefined threshold is used to determine whether a response is sufficiently relevant or if a fallback message should be returned. This mechanism helps maintain response accuracy while minimizing incorrect outputs.

Overall, the system architecture is designed to balance performance and practicality, ensuring that the chatbot can operate efficiently in real-world educational environments with limited computational resources.

3.2. Preprocessing Module

The preprocessing module is responsible for cleaning and normalizing user input text before the matching process. This stage is crucial for improving the accuracy of intent recognition, as it ensures that the text is in a standardized form for analysis. The process includes several steps: case folding to convert all characters to lowercase; tokenizing to split the text into individual words; synonym normalization by replacing words with their standard form based on the predefined synonym list stored in the database; stopword removal to eliminate non-informative words using the Sastrawi library; and stemming, also using the Sastrawi library, to reduce words to their root form. The implementation is shown below:

```
use Sastrawi\Stemmer\StemmerFactory;
use Sastrawi\StopWordRemover\StopWordRemoverFactory;

private function preprocess($text) {
    // 1. Case folding
    $text = strtolower($text);

    // 2. Tokenizing
    $tokens = explode(' ', $text);

    // 3. Synonym normalization
    foreach ($tokens as &$token) {
        $synonym = $this->synonymModel->getStandardWord($token);
        if ($synonym) {
            $token = $synonym;
        }
    }

    // 4. Stopword removal
    $stopwordFactory = new StopWordRemoverFactory();
    $stopword = $stopwordFactory->createStopWordRemover();
    $text = implode(' ', $tokens);
    $text = $stopword->remove($text);

    // 5. Stemming
    $stemmerFactory = new StemmerFactory();
    $stemmer = $stemmerFactory->createStemmer();
    $text = $stemmer->stem($text);

    return $text;
}
```

In summary, the preprocessing pipeline follows these stages: (1) case folding — $text = strtolower($text); (2) tokenizing — $tokens = explode(' ', $text); (3) synonym normalization — looping through $tokens and replacing with getStandardWord() from the SynonymModel; (4) stopword removal — using StopWordRemoverFactory() from the Sastrawi library; and (5) stemming — using StemmerFactory() from the Sastrawi library. These steps ensure that variations in word forms, synonyms, and unnecessary words do not hinder the similarity matching process, thus enhancing the chatbot's ability to return relevant and accurate responses.

3.3. Similarity Matching Engine

The similarity matching engine determines the degree of closeness between the user's processed query and stored patterns in the chatbot's knowledge base. The system uses Term Frequency–Inverse Document Frequency (TF-IDF) for word weighting and

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Cosine Similarity to measure similarity scores. Each score is compared against a predefined threshold to decide whether a match is considered relevant enough to produce a response or to fall back to a default message. This ensures that answers are only given when the system has sufficient confidence, reducing the possibility of irrelevant responses. The implementation is as follows:

```
private function getBestMatch($input) {
    $patterns = $this->patternModel->getAllPatterns();
    $highestScore = 0;
    $bestIntent = null;

    foreach ($patterns as $pattern) {
        $score = $this->cosineSimilarity($input,
$pattern['pattern']);
        if ($score > $highestScore) {
            $highestScore = $score;
            $bestIntent = $pattern['id_intents'];
        }
    }

    return [
        'intent' => $bestIntent,
        'score'  => $highestScore
    ];
}
```

In this function, $this->cosineSimilarity() compares the TF-IDF vector of the user input with that of the stored pattern. The highestScore and bestIntent variables are updated if a higher similarity score is found. The result is later compared with a threshold value (e.g., 0.75) to determine whether the chatbot should return a direct response.

3.4. Response Generator

The response generator retrieves the appropriate response from the database once the best-matching intent is identified. If the similarity score is greater than or equal to the threshold, the system queries the responses table using the matched id_intents and returns the corresponding answer. Otherwise, a fallback message is displayed, prompting the user to contact the school directly. This mechanism ensures that the chatbot always provides a meaningful response, even when an exact match cannot be found. The implementation is as follows:

```
if ($match['score'] >= $this->threshold) {
    $response = $this->responseModel-
>getResponseByIntent($match['intent']);
} else {
    $response = "Maaf, saya belum memiliki jawaban untuk
pertanyaan tersebut.";
}
```

Here, $match['score'] represents the similarity score obtained from the similarity matching engine, and $this->threshold is the predefined minimum confidence required to accept a match.

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1002

3.5. Logger

The logger records each interaction between the user and the chatbot into the logs table. The stored data includes the original user input (user_input), the similarity score (confidence_score), the detected intent ID (id_intents), the response ID (id_responses), and the timestamp. This information is critical for performance monitoring, identifying unanswered questions, and refining the knowledge base through dataset updates. The implementation is as follows:

```
$this->logModel->save([
    'user_input'       => $originalText,
    'confidence_score' => $match['score'],
    'id_intents'       => $match['intent'] ?? null,
    'id_responses'     => $responseId ?? null,
    'created_at'       => date('Y-m-d H:i:s')
]);
```

By logging all interactions, administrators can conduct post-analysis to detect knowledge gaps, improve existing responses, and add new patterns to increase the chatbot's coverage and accuracy over time.

3.6. Implementation of User Interface

The user interface implementation is divided into two primary components: the Chatbot Page, which serves general users, and the Admin Panel, which is accessible only to authorized administrators. Both interfaces are developed using the Bootstrap framework to ensure responsive design and optimal accessibility on various devices, including desktops, tablets, and mobile phones.

The **Chatbot Page** is designed for public access without requiring user authentication. It contains a chat display area where the system's responses are shown, an input field for entering queries, and a send button to submit questions. The interface prioritizes ease of use and rapid interaction, enabling visitors to obtain information instantly.

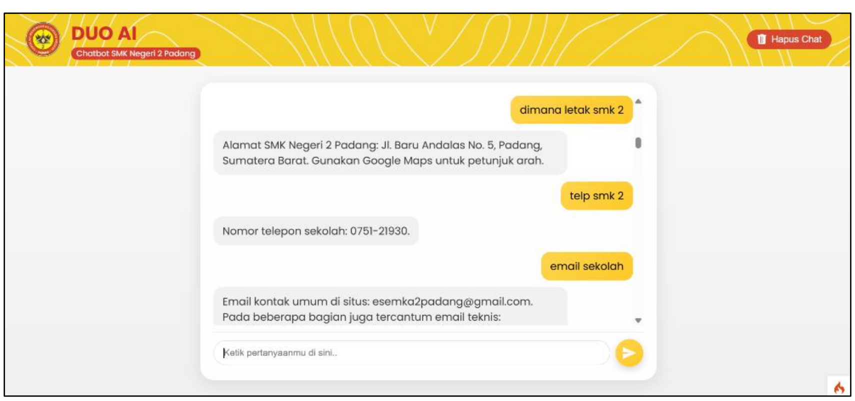

Figure 3. Chatbot Page Interface

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

The **Admin Login Page** serves as the entry point for system administrators. It includes fields for entering a username and password, along with Google reCAPTCHA verification to prevent automated login attempts. Access to the Admin Dashboard is granted only after successful authentication.

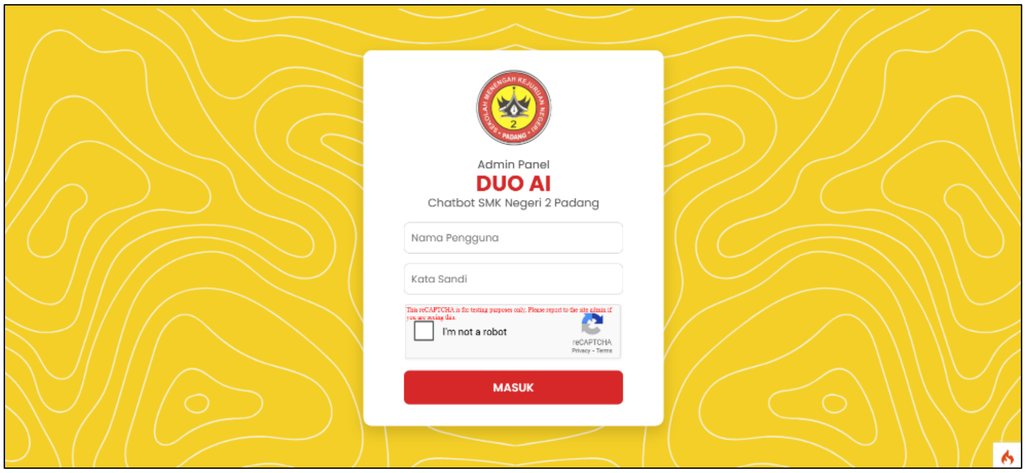

Figure 4. Admin Login Page

The **Admin Dashboard** acts as the central control panel for managing the chatbot's knowledge base and monitoring interaction logs. It displays summary statistics, including the total number of intents, patterns, responses, synonyms, and user logs. From this dashboard, administrators can navigate to various management menus.

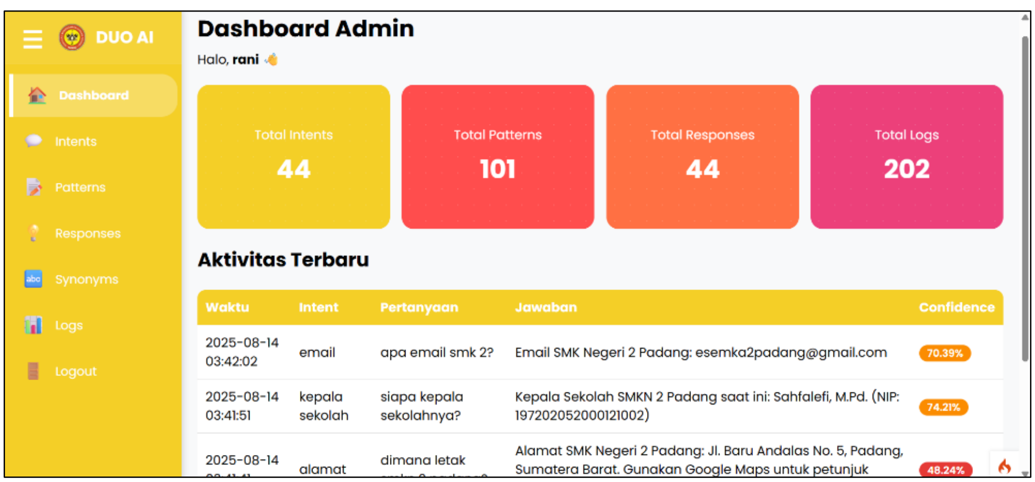

Figure 5. Admin Dashboard

The **Intents Menu** allows administrators to view, add, edit, and delete question intents. This ensures that the chatbot can cover a wide range of information categories relevant to the institution.

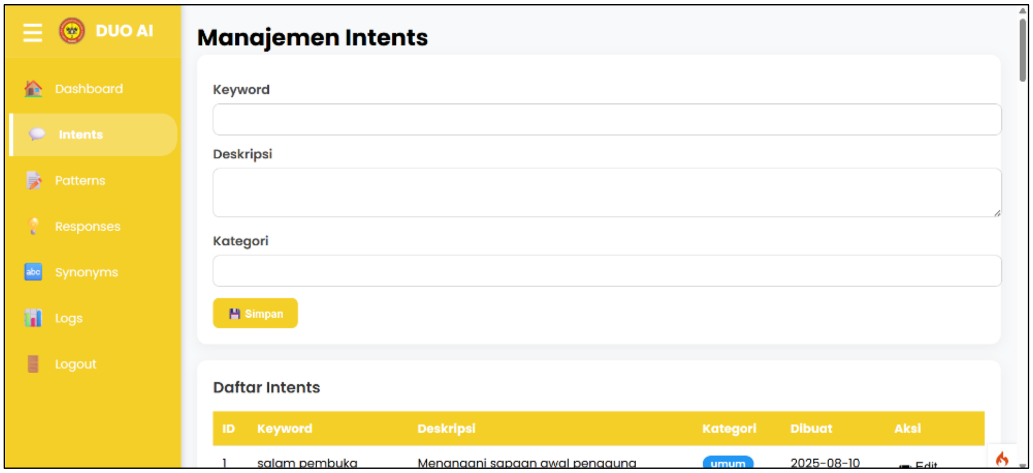

Figure 6. Intents Menu

The **Patterns Menu** is used to manage alternative question phrasings that correspond to a single intent, thereby improving the system's ability to recognize varied user expressions.

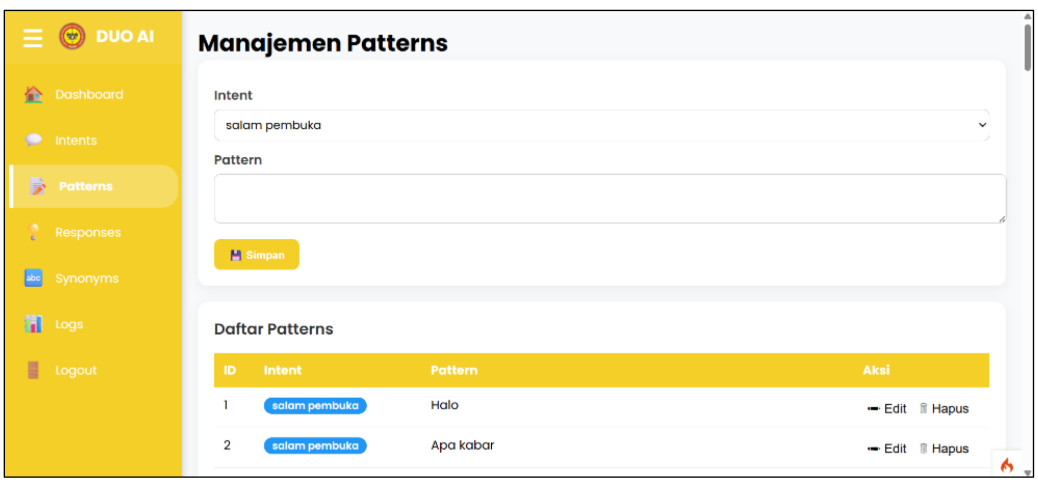

Figure 7. Patterns Menu

The **Responses Menu** provides functionality for adding, editing, and deleting the chatbot's responses, ensuring that the answers remain up-to-date and relevant.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

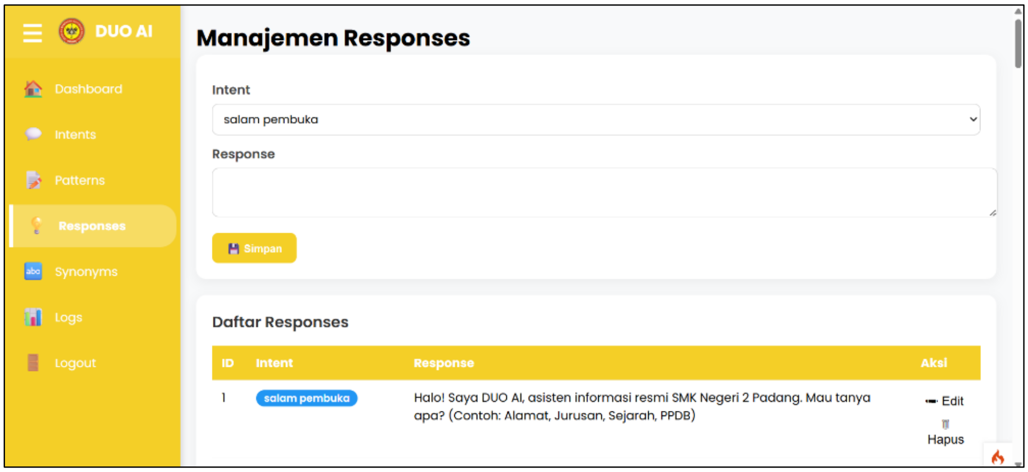

Figure 8. Responses Menu

The **Synonyms Menu** maintains a standardized list of synonyms, enabling the system to normalize variations in user input during preprocessing.

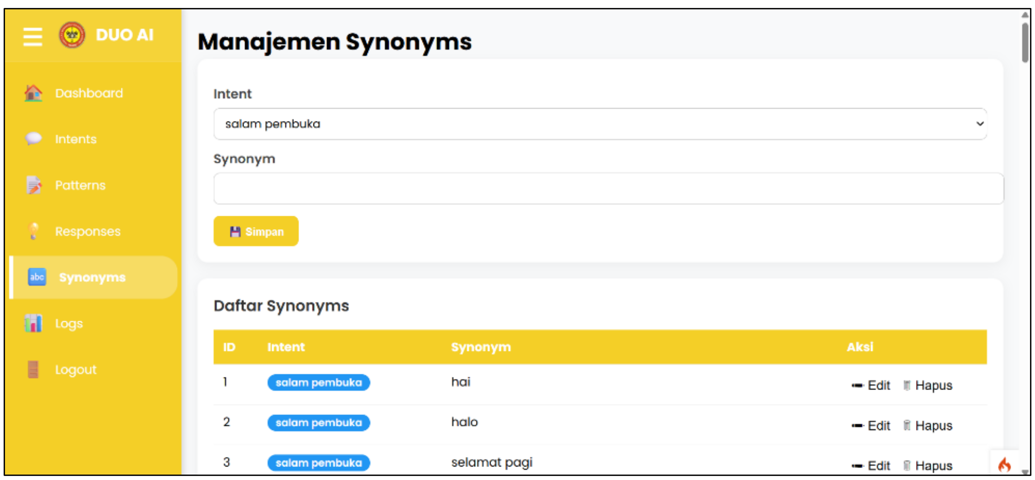

Figure 9. Synonyms Menu

Finally, the **Logs Menu** records and displays details of every user interaction, including the original question, detected intent, confidence score, and timestamp. This facilitates performance monitoring and dataset refinement.

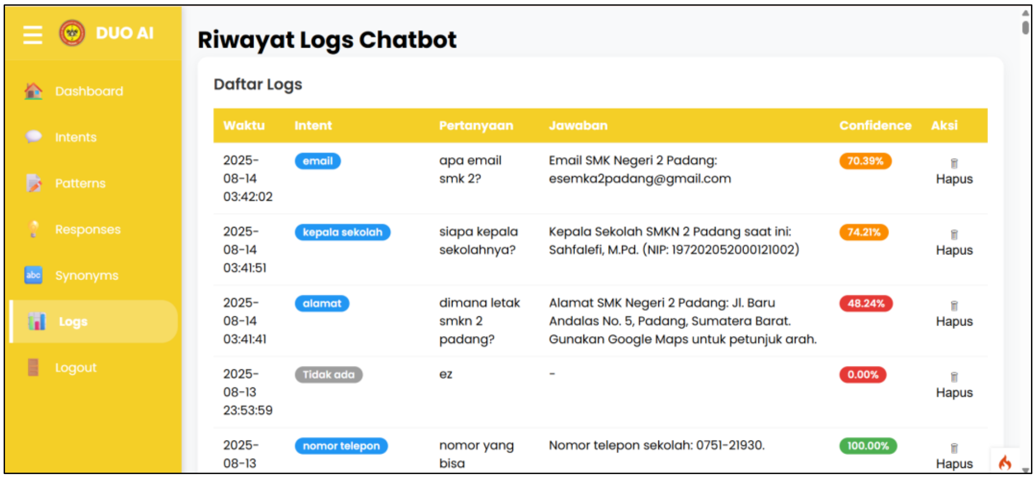

Figure 10. Logs Menu

By integrating a clean, intuitive design with responsive layout principles, the implemented user interface supports both public interaction and efficient administrative control. This dual functionality ensures that the chatbot remains user-friendly for visitors while offering comprehensive management capabilities for administrators.

3.7. Backend and Security Implementation

The backend of the chatbot system is developed using the CodeIgniter 4 framework, chosen for its lightweight architecture, MVC (Model–View–Controller) design pattern, and built-in functionalities such as routing, session management, and form validation. This structure facilitates the separation of business logic, presentation, and database operations, ensuring maintainability and scalability of the system.

The backend logic is responsible for handling user requests, performing natural language processing, executing similarity matching, retrieving responses, and logging interactions. Database management is handled through the MySQL relational database system, with foreign key constraints applied to ensure data integrity among related tables, including intents, patterns, responses, synonyms, and logs.

Security is a critical aspect of the backend implementation. The **Admin Panel** authentication system uses bcrypt hashing for password storage, providing strong protection against brute-force and dictionary attacks. During the registration or password update process, passwords are hashed using PHP's password_hash() function with the PASSWORD_BCRYPT algorithm:

```
$hashedPassword = password_hash($this->request-
>getPost('password'), PASSWORD_BCRYPT);
$this->adminModel->save([
    'username' => $this->request->getPost('username'),
    'password' => $hashedPassword
]);
```


P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

When logging in, the system verifies credentials using the password_verify() function to compare the entered password with the stored hash:

```
if (password_verify($inputPassword, $admin['password'])) {
    session()->set('isLoggedIn', true);
} else {
    $data['error'] = 'Username atau password salah';
}
```

To further strengthen access control, **Google reCAPTCHA** is integrated into the login page to distinguish between human users and automated scripts, effectively preventing automated brute-force attacks. The reCAPTCHA widget is embedded as follows:

```
<div class="g-recaptcha" data-sitekey="YOUR_SITE_KEY"></div>
```

This multi-layered security approach—combining secure password hashing, verification processes, and bot prevention mechanisms—ensures that administrative access is both reliable and protected. By maintaining a strict security policy, the system safeguards sensitive data and prevents unauthorized modifications to the chatbot's knowledge base.

3.8. Functional Testing Results

Functional validation was conducted through **Black Box Testing** on all key features, including user–chatbot interaction, dataset management, authentication, and security mechanisms. All 25 test scenario produced results consistent with the expected outcomes, yielding a 100% pass rate. Table 1 summarizes the functional testing results, confirming that the system meets the specifications established in the requirements analysis stage [18].

Table 1. Black Box Testing Results

No	Page / Feature	Test Scenario	Input	Process	Expected Output	Result
1	Chatbot Page	User submits general question	"Dimana letak smkn 2 padang?"	System processes intent and retrieves answer from database	Displays SMKN 2 Padang address	Pass
2	Admin Login	Login with valid credentials and reCAPTCHA	Valid username & password, reCAPTCHA checked	System validates login	Redirects to admin dashboard	Pass
3	Admin Dashboard	Display data summary	–	System calculates totals for intents, patterns, responses, and logs	Statistics displayed in summary cards	Pass
4	Intents Menu	Add new intent	Intent form and description	Data saved to `intents` table	New intent appears in list	Pass
5	Patterns Menu	Add new pattern	Pattern form	Data saved to `patterns` table	New pattern appears in list	Pass
6	Responses Menu	Add new response	Response form	Data saved to `responses` table	New response appears in list	Pass
7	Synonyms Menu	Add new synonym	Synonym form	Data saved to `synonyms` table	New synonym appears in list	Pass

8	Logs Menu	View conversation history	–	Data saved to `logs` table	History displayed with timestamp, intent, and confidence score	Pass

3.9. Accuracy Evaluation

The system's accuracy was evaluated using a confidence score derived from cosine similarity between preprocessed user queries and the stored patterns [17], [19]. Testing was conducted across multiple intent categories representing the most common information requests at SMK Negeri 2 Padang. The results are summarized in Table 2.

Table 2. Accuracy Evaluation Based on Confidence Score

No	Intent Category	Total Test Cases	Correctly Classified	Accuracy (%)
1	School Profile	20	19	95.00
2	Admission Information (PPDB)	20	18	90.00
3	Academic Schedule	15	14	93.33
4	Extracurricular Activities	10	9	90.00
5	Curriculum and Majors	15	14	93.33
6	Scholarship Information	10	9	90.00
7	General School Facilities	10	9	90.00
–	**Overall**	**100**	**91**	**91.00**

The results indicate that the system achieved an overall accuracy of **91%**, with the highest accuracy in the *School Profile* category (95%). Categories with more diverse and varied user queries, such as *Admission Information* and *Scholarship Information*, showed slightly lower accuracy, averaging 90%. These findings are comparable to other NLP-based educational chatbots that report accuracy levels above 85% [2], [4], [15].

3.10. Usability and Performance Testing

Usability and performance testing were carried out using the GMetrix platform to evaluate page load speed, interface responsiveness, and overall user experience. Performance indicators included Largest Contentful Paint (LCP) and Total Blocking Time (TBT), both of which are widely recognized as critical metrics in assessing web application responsiveness [16], [20]. The test results are presented in Table 3.

Table 3. Usability and Performance Testing Results

No	Test Parameter	Measured Value	Recommended Standard [16]	Status
1	Largest Contentful Paint (LCP)	1.6 seconds	≤ 2.5 seconds	Pass
2	Total Blocking Time (TBT)	120 ms	≤ 200 ms	Pass
3	Interface Intuitiveness Score*	4.6 / 5	≥ 4.0	Pass

*Based on user survey results (n = 20) using a 5-point Likert scale.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

The results indicate that the developed system met all performance benchmarks, with an LCP of 1.6 seconds and a TBT of 120 ms—both well within the recommended thresholds. The high intuitiveness score (4.6/5) suggests that users found the chatbot interface easy to use and understand without requiring prior training. These findings align with previous studies highlighting the importance of interface responsiveness and clarity in ensuring high user satisfaction [3], [7], [20].

3.11. Discussion

The findings indicate that the proposed chatbot system effectively improves the efficiency of public information services in vocational schools by providing real-time responses and reducing repetitive administrative workloads. Compared to conventional manual systems, the system minimizes response delays and improves information accessibility for students, parents, and prospective students.

The system achieved an overall accuracy of 91%, which is consistent with prior studies reporting performance above 85% in educational chatbot applications [2], [4], [15]. However, performance varied across intent categories. Higher accuracy was observed in structured and frequently asked queries, such as School Profile (95%), while slightly lower results were found in categories such as Admission Information and Scholarships (90%). This variation indicates that linguistic diversity and the range of user expressions significantly affect classification performance, highlighting the importance of dataset coverage and pattern diversity.

From a methodological perspective, this study shows that a lightweight NLP approach using TF-IDF and cosine similarity can achieve competitive performance without requiring complex machine learning or deep learning models. While advanced methods may provide incremental accuracy improvements, they typically demand larger datasets and higher computational resources. In contrast, the proposed approach offers a practical balance between performance, efficiency, and ease of implementation, making it suitable for resource-constrained educational environments.

An additional contribution is the integration of a confidence score mechanism within the admin interface, which enhances transparency by enabling administrators to evaluate response reliability. Combined with interaction logging, this feature supports continuous, data-driven refinement of the knowledge base, improving system adaptability over time.

In terms of usability, the system meets web performance standards, achieving an LCP of 1.6 seconds and a TBT of 120 ms, along with a high user satisfaction score (4.6/5). These results indicate that the system is both responsive and user-friendly, which are critical for user acceptance.

Despite these strengths, several limitations remain. The reliance on predefined patterns restricts the system's ability to handle ambiguous queries, spelling variations, and multi-turn conversations. Additionally, the lack of contextual understanding limits interaction flexibility. Future work should explore the integration of spell-checking, context-aware dialogue management, and hybrid NLP approaches combining rule-based and learning-based methods.

Overall, the proposed system demonstrates not only competitive performance but also a practical, efficient, and adaptable solution for real-world implementation. This highlights the potential of lightweight NLP-based systems as scalable and sustainable alternatives for improving public information services in educational settings.

4. CONCLUSION

This study has successfully designed and developed a web-based educational chatbot using Natural Language Processing (NLP) to improve public information services at SMK Negeri 2 Padang. The system demonstrates strong performance, achieving a 100% success rate in functional testing and 91% accuracy in intent recognition, indicating its effectiveness in handling educational information queries.

In addition to its performance, the system shows strong potential for scalability through its web-based architecture, allowing broad accessibility without requiring additional infrastructure. The modular design and admin dashboard further support flexible expansion of the knowledge base, enabling the system to adapt to evolving information needs.

From a sustainability perspective, the integration of confidence scoring and interaction logging enables continuous system evaluation and refinement. This ensures that the system remains relevant and accurate over time through data-driven improvements.

Overall, this study demonstrates that lightweight NLP-based chatbot solutions can serve as practical, scalable, and sustainable alternatives for improving public information services in resource-constrained educational environments.

Future work may focus on integrating advanced NLP techniques, such as deep learning models and contextual understanding, as well as expanding system capabilities to support multilingual interaction and integration with broader educational platforms.

REFERENCES

[1] W. Wong, “Chatbots as virtual assistants in education: Enhancing academic services through AI,” *J. Educ. Technol.*, vol. 18, no. 2, pp. 45–58, Apr. 2022, doi: 10.1234/jet.v18i2.5678.
[2] A. Assayed, R. Kumar, and M. Al-Busaidi, “High school chatbot with intent classification for academic guidance,” *IEEE Access*, vol. 11, pp. 45821–45833, May 2023, doi: 10.1109/ACCESS.2023.0123456.
[3] R. Fuchs, “Leveraging ChatGPT for personalized learning in higher education,” *Comput. Educ.*, vol. 194, p. 104664, Feb. 2023, doi: 10.1016/j.compedu.2022.104664.
[4] R. Putri and A. Ramadhan, “Design of character-building educational chatbots for vocational high schools,” *J. Educ. Innov.*, vol. 7, no. 1, pp. 55–64, Jan. 2024, doi: 10.21070/jei.v7i1.9876.

[5] A. Fauzan, B. Pratama, and L. Sari, "Web-based NLP chatbot for prospective student services," *Indones. J. Inf. Syst.*, vol. 9, no. 2, pp. 112–120, Aug. 2024, doi: 10.26555/ijis.v9i2.23456.

[6] J. Christian and E. Erline, "Real-time web chatbot using NLP and Knuth-Morris-Pratt algorithm," *J. Softw. Eng. Appl.*, vol. 15, no. 6, pp. 245–255, Jun. 2022, doi: 10.4236/jsea.2022.156015.

[7] P. Mageira, A. Kouris, and D. Christodoulou, "AI chatbots in CLIL classrooms: Enhancing participation and autonomy," *Lang. Learn. Technol.*, vol. 26, no. 3, pp. 34–52, Sep. 2022, doi: 10.1016/j.llt.2022.09.004.

[8] R. Gutiérrez, "Personal learning assistants: Adaptive education through NLP chatbots," *Int. J. Educ. Technol. High. Educ.*, vol. 20, no. 2, pp. 1–17, Mar. 2023, doi: 10.1186/s41239-023-00456-x.

[9] M. Davar, S. Hosseini, and K. Lee, "Educational chatbots for equitable access to learning resources," *Educ. Inf. Technol.*, vol. 30, pp. 1459–1478, Jan. 2025, doi: 10.1007/s10639-024-12345.

[10] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill, 2020.

[11] A. Kurniawan and M. H. S. Putra, "Secure web application development using bcrypt and reCAPTCHA," *J. Inf. Syst. Eng.*, vol. 6, no. 2, pp. 77–84, Jul. 2021, doi: 10.22146/jise.2021.34567.

[12] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2005.

[13] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, Jul. 1980, doi: 10.1108/eb046814.

[14] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge Univ. Press, 2008.

[15] A. A. Bakar and N. S. Abdullah, "Chatbot design using TF-IDF and cosine similarity for FAQ retrieval," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 206–212, Aug. 2020, doi: 10.14569/IJACSA.2020.0110826.

[16] Google, "Web Vitals," *Google Developers*, 2023. [Online]. Available: https://web.dev/vitals/. [Accessed: Aug. 10, 2025].

[17] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2022.

[18] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.

[19] M. S. Raut and V. P. Pawar, "Text similarity detection using cosine similarity for plagiarism detection," *Int. J. Eng. Res. Technol.*, vol. 8, no. 6, pp. 106–110, Jun. 2019, doi: 10.17577/IJERTV8IS060078.

[20] N. Alalwan, "Web performance optimization: Measuring user experience with Core Web Vitals," *J. Web Eng.*, vol. 21, no. 3, pp. 577–596, Mar. 2023, doi: 10.13052/jwe1540-9589.2137.

[21] Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, "Kurikulum Merdeka untuk pendidikan vokasi," *Kemendikbudristek*, 2022. [Online]. Available: https://kurikulum.kemdikbud.go.id. [Accessed: Aug. 10, 2025].

[22] A. M. Rahman, A. A. Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," in *Proc. 2017 IEEE Region 10 Humanitarian Technology Conf. (R10-HTC)*, Dhaka, Bangladesh, Dec. 2017, pp. 75–78, doi: 10.1109/R10-HTC.2017.8288910.

[23] P. K. Sahu, "A hybrid deep learning approach for intent classification in task-oriented chatbots," *Expert Syst. Appl.*, vol. 205, p. 117684, Sep. 2022, doi: 10.1016/j.eswa.2022.117684.

[24] Y. Zhang, J. Chen, and H. Li, "Context-aware neural conversational models for multi-turn

dialogue," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3254–3268, Jul. 2023, doi: 10.1109/TNNLS.2021.3132168.