

## Chatbot-Based Library Service Optimization: Implementing Levenshtein Distance

Bonifasius Dandy Krisnanda<sup>1\*</sup>, Ary Prabowo<sup>2</sup>

<sup>1,2</sup>Informatics Engineering Study Program, Faculty of Computer Science, Esa Unggul University

<sup>1\*</sup>bonidandi@gmail.com, <sup>2</sup>ary.prabowo@esaunggul.ac.id

### Abstract

Library information services at Esa Unggul Citra Raya University are still manual, requiring students to come directly during operational hours to get information. This condition causes long queues, long waiting times, and limited access to information outside of operational hours. This research aims to develop a fuzzy matching-based chatbot application to improve the efficiency of library information services. The development method uses Extreme Programming (XP) with stages of planning, design, coding, and testing. The system is built using Python with the Flask framework as the backend, MySQL database for data storage, and the Levenshtein Distance algorithm for fuzzy matching that is capable of handling typos. Data collection was carried out through observations, interviews with library staff, and questionnaires to 168 students. Testing using the Blackbox Testing method shows that all system functions are running according to specifications. Performance evaluation was carried out through two approaches: objective evaluation using a test set of 250 labeled questions which resulted in an accuracy of 84.4%, precision of 85.2%, recall of 83.7%, and an F1-score of 84.4%; As well as perception evaluation through a questionnaire to 53 respondents who showed positive results with 81.1% of respondents stating that the chatbot was able to understand questions despite typing errors, 88.7% rated the answers as very appropriate, 90.6% stated that the response was very fast, and 86.8% felt very satisfied. The optimal threshold value is set at 75% based on the tuning results that balance the wrong answer and the fallback. The system successfully reduced the number of direct-to-staff inquiries from an average of 20 questions/day to 6 questions/post-implementation day, saved an estimated 1.17–2.33 hours of staff work time per day, reduced queues at staff desks, and provided 24/7 access to information. The implementation of the chatbot successfully improves the efficiency of library services and provides an optimal user experience.

**Keywords:** chatbot, fuzzy matching, library services

© 2026 Journal of DINDA

### 1. Introduction

Libraries have a vital role as a center for providing a variety of information resources that can be accessed according to the needs of their users [1]. Along with the acceleration of the development of science and technology, the library paradigm has undergone a significant transformation from just a place to a place to an institution that provides services and collections that continue to grow according to the needs of users [2]. The application of technology in library management can improve the efficiency and effectiveness of services, so that libraries become more responsive to students' literacy needs [3]. However, at Esa Unggul Citra Raya University, library services still require students to be physically present during operational hours to obtain the

necessary information. This condition poses obstacles for students who need access to information outside of operational hours or do not have time to come in person. The process of physically searching for books on library shelves often takes an inefficient amount of time, especially when students do not know the location or status of the availability of the books they are looking for. This situation is exacerbated by the occurrence of queues at the library staff desks, especially when students scan their Student ID Cards and ask questions, which ultimately creates a less conducive atmosphere and hinders students from obtaining the information and reference resources needed to complete academic tasks efficiently.

The usage of *Chatbot* has expanded to various sectors, including libraries, to support service enhancements and strengthen interactions with users [4]. *Chatbot* is a computer program designed to understand feedback from users, both in text and voice form, and provide intelligent responses [5]. Implementation *Chatbot* allows students to access library services remotely without the need to be physically present, thereby increasing efficiency in supporting library service operations [6]. Research conducted by Abel Uzoka et al. shows that *Chatbot* Able to answer up to 70% of routine questions from users, allowing staff to focus more on dealing with more complex issues and ultimately improve overall efficiency [7]. Similar research conducted by Taufik and Supriyanto on implementation *Chatbot* for services *Frequently Asked Question Academic* use *Dialogflow* shows that *Chatbot* Able to provide quick and accurate responses to user queries with above-average scores in terms of appeal, clarity, efficiency, regularity, stimulation, and novelty [8]. Approach *Fuzzy Matching* with algorithms *Levenshtein Distance* has been shown to be effective in correcting typos and detecting text similarities with minor differences [9].

This research was conducted to overcome the problem of library information services which are still manual and limited to operational hours. In contrast to previous studies that used the *Dialogflow* limiting development flexibility, this study uses a Python-based approach with *Flask framework* to support freer and more customized development. The developed system is not only capable of answering common questions, but also equipped with *Dashboard Admin* which allows the data to be dynamically updated by the manager. The development methods used are *Extreme Programming (XP)*, an iterative approach that provides greater flexibility than the *Waterfall* which tend to be rigid and difficult to adapt to changing needs during development [10].

Based on this background, this study aims to answer the question: how to develop a *fuzzy matching-based chatbot* application that can provide library information accurately and responsively to students of Esa Unggul Citra Raya University? How to design and implement a *chatbot system* with the ability to understand text in *fuzzy matching* and provide automatic answers to routine questions related to library services? How to reduce the manual workload of library staff by automating answers to repetitive questions thereby improving the operational efficiency of the service? And how to integrate the *admin dashboard* feature to enable dynamic and *real-time* updates of library service data?

This research is expected to make a significant contribution to various parties. For students, this *chatbot* application makes it easier for them to get library information quickly and efficiently without being bound

by time and place, thus supporting a more effective learning process. For library staff, the system can improve service efficiency by reducing the workload of answering students' routine questions, allowing them to focus more on tasks that require complex handling. Institutionally, this research contributes to the application of *fuzzy matching-based chatbot technology* in the library environment of Esa Unggul Citra Raya University, which can be used as a reference for similar implementations in other educational institutions in the future.

## 2. Research Methods

This study uses a system development approach with the *Extreme Programming (XP)*, which is part of the methodology *Agile*. The XP method was chosen because it prioritizes the creation of rapidly functioning program code through short iterations to adapt to user needs, as well as providing high flexibility in responding to changing needs [11]. The research stages are carried out systematically starting from literature study, data collection, system needs analysis, system design, implementation, and system testing.

The data collection technique is carried out through three approaches. First, direct observation of the information service process at the library of Esa Unggul Citra Raya University to understand the current workflow, including the obstacles faced by staff and users. Second, in-depth interviews with library staff to obtain information related to frequently asked questions by students and expectations of the system *Chatbot* developed. Third, the distribution of questionnaires to 168 students as respondents to identify information needs and interest in the use of *Chatbot*. The collected data is then analyzed using the PIECES method (*Performance, Information, Economy, Control, Efficiency, Service*) to identify problems and provide appropriate solutions in system development [12].

The design of the system is done using diagrams *Unified Modeling Language (UML)* which includes *Use Case Diagram, Class Diagram, and Activity Diagram* to model the functional needs and relationships between system components. System implementation using the Python programming language with *Flask framework* As *Backend* to set up application logic and data communication. MySQL databases are used to store library service information, question-and-answer data, and book collections. Algorithm *Levenshtein Distance* implemented to perform matching *Fuzzy Matching* between user input and question patterns stored in the database, so that the system is able to recognize the question even if there are typos [13]. Similarity calculations are converted to *Similarity Score* in the range of 0–100% using the following formula:

$$\text{Similarity (\%)} = (1 - (\text{Lifetime Distance} / \max(\text{len}(s1), \text{len}(s2)))) \times 100 \quad (1)$$

The *threshold* value is determined through a *tuning* process by testing four candidate scores, namely 70%, 75%, 80%, and 85%, against a *labeled* test set to observe the *trade-off* between *false answers* and *fallbacks*. An answer will be given if the *similarity score* meets or exceeds the selected *threshold* value; instead, the system displays a *fallback* message directing the user to contact the library staff directly. For objective evaluation, a *test set* of 250 labeled questions was constructed which included normal questions, questions with *light typos* (1–2 characters), questions with heavy *typos* (3+ characters), synonym variations, and word order variations. Each question is labeled with a *ground truth tag*. From the *test set*, *accuracy*, *precision*, *recall*, and *F1-score metrics* were calculated to objectively assess the system's performance.

To validate the algorithm choice *Levenshtein Distance*, a comparison was made with two methods *baseline*, namely *cosine similarity* based on TF-IDF and *Jaccard similarity* based *Character N-gram* (n=2). All three methods tested on *Test set* which is the same as *Threshold* optimal for each so that the comparison is fair. Functional testing of the system is carried out using the *Blackbox Testing* to ensure that all functions run according to the specifications that have been set without checking the internal structure of the program code [14]. User perception evaluation was carried out through the distribution of a structured questionnaire to 53 respondents using a scale *Likert* 1–5 which covers six aspects: ease of use, ability to understand questions with typos, responsiveness, response speed, time-saving usefulness, and user satisfaction levels. The questionnaire instrument was tested for reliability using *Cronbach's Alpha* with a value of 0.847, which indicates good reliability. The data from the questionnaire results were analyzed descriptively to determine the percentage of user acceptance of the developed system.

To measure the impact of implementation more objectively, a *before-after evaluation* was also carried out based on: (1) the number of direct questions to staff per day before and after implementation based on staff records, (2) estimated average service time per question before and after implementation, and (3) the number of daily *fallbacks* recorded in the system *log* as a proxy for questions that the *chatbot* could not answer. The complete stages of the research can be seen in Figure 1.

### 3. Results and Discussion

#### 3.1 Improved Library Staff Time and Labor Efficiency and Student Service Experience

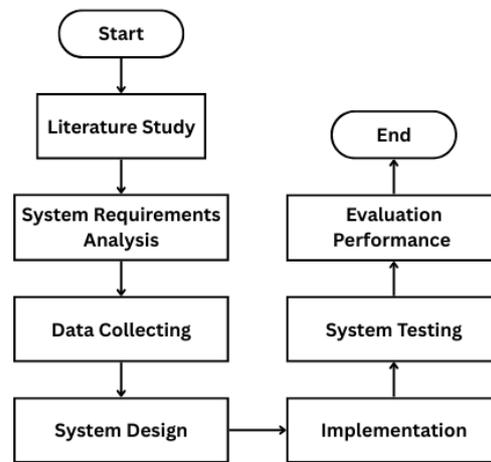


Figure 1. Flow of research stages  
 Source : Researcher's Processed Results 2025

The questionnaire instrument was compiled using a scale *Likert* 1–5 (1 = Strongly Disagree/Strongly Dissatisfied, 5 = Strongly Agree/Very Satisfied) with six question items that each measured: ease of use, comprehension *Typo*, answer suitability, response speed, time-saving benefits, and overall satisfaction. Prior to deployment, the instrument was tested for reliability using *Cronbach's Alpha* and obtained a value of  $\alpha = 0.847$ , which indicates the reliability of the instrument in the good category [15].

Based on the results of observations and interviews with library staff of Esa Unggul Citra Raya University, a fundamental problem was identified in manual library information services. The library staff of two had to deal with a variety of recurring routine questions, such as book availability information, borrowing rules, repository access, and discussion room borrowing. The estimated time it takes to answer one routine question ranges from 5-10 minutes. With an average of 20 students asking questions in a day, the time wasted reaches 100-200 minutes or about 1.67-3.33 hours per day just to answer questions that can actually be automated. This condition causes staff's workload to increase and reduce their focus on tasks that require more complex handling.

Application implementation *Chatbot* based *Fuzzy Matching* with algorithms *Levenshtein Distance* designed to automate the process of answering students' routine questions. System *Chatbot* Able to respond in less than 1 second, which is much faster compared to manual services that require 5–10 minutes per question. Based on *Log* system collected over two post-implementation weeks, an average of 20 questions per day came in via *Chatbot*, where 14 questions (70%) were answered automatically and 6 questions (30%) resulted

in *Fallback* and passed on to staff. Staff records data prior to implementation recorded an average of 20 direct inquiries per day; Post-implementation the number was reduced to an average of 6 direct questions per day. With an estimated time of 5–10 minutes per question, staff time savings are measurably 30–60 minutes per day, or when combined with the reduction of previously unrecorded questions, total savings are estimated at 1.17–2.33 hours per day. Research conducted by Abel Uzoka et al. also reinforces these findings by showing that *Chatbot* In general, it is able to answer up to 70% of routine questions from users, allowing staff to focus more on dealing with more complex problems [7].

The results of the performance evaluation questionnaire distributed to 53 student respondents showed a positive impact on the service experience. As many as 79.2% of respondents stated that *Chatbot* It is very easy to use, while 13.2% rate it quite easy, and only 7.5% find it difficult. In terms of response speed, 90.6% of respondents stated that *Chatbot* responded very quickly, 5.7% rated it fairly fast, and only 3.8% rated it slow. This high response speed provides a satisfactory user experience and supports the effectiveness of library information services. From the aspect of usefulness in saving time, 81.1% of respondents stated that *Chatbot* helps them save time when searching for information, while 18.9% think otherwise. These results prove that the existence of *Chatbot* able to speed up the process of searching for information compared to the conventional way that requires students to come in person and queue to ask questions to staff [5].

In terms of user satisfaction, 86.8% of respondents stated that they were very satisfied with using *chatbots*, 7.5% were quite satisfied, and 5.7% were dissatisfied. This high level of satisfaction indicates that the *chatbot* has successfully met the needs of users and provided a positive experience in accessing library information services. Interviews with library staff also confirmed that the use of *chatbots* optimizes existing services in the library and is the choice of library communication media to users, especially related to library service information. Thus, the implementation of *chatbots* not only improves the efficiency of staff time and manpower, but also provides a better service experience for students through ease of access, speed of response, and consistent availability of information.

### 3.2 Reducing Queues at Library Staff Desks to Create a More Conducive Service Atmosphere

Direct observation conducted at the Esa Unggul Citra Raya University Library shows that long queues often occur at the desk of library staff, especially when students have to scan their Student ID Cards to access services as well as ask questions. This condition creates an atmosphere that is less conducive for other students

who are using library facilities to study or read. Based on interviews with library staff, one of the main challenges is when many users who want to enter and exit the library at the same time at the circulation desk, where librarians need to be careful in looking for one by one Student Identity Cards accompanied by attending for users who will enter the library. This situation is exacerbated by the presence of routine questions asked by students at the same time, which makes the service process slower and creates queue buildup.

The implementation of *the chatbot* application provides an alternative solution for students to get information without having to come directly to the library staff desk. The *chatbot* system can be accessed through students' personal devices from anywhere and at any time, reducing the need for students to come in person just to ask general information questions. The results of the respondents' background questionnaire distributed to 168 students showed that the most dominant problem was uncertainty about the availability of books experienced by 87.3% of respondents, as well as ignorance about when the borrowed books would be returned by 56.3%. In addition, students also experience other obstacles such as difficulty in extending book borrowing (26.8%), limited access to services when the library is closed (28.2%), and lack of information about borrowing rules, how to register, access to campus repositories, and the use of discussion rooms. These questions must be answered directly by the staff in advance, which contributes to the queue buildup.

With the presence of *Chatbot*, students can ask these questions through the app without having to come to the library, thus reducing the number of students waiting in line at the staff desk. Based on the data that *Chatbot* able to answer up to 70% of routine questions, so from the 20 students who usually ask questions every day, about 14 students no longer need to come to the staff desk just to ask informational questions. This reduction significantly reduces queues and creates a more conducive atmosphere in the library area. Library staff can focus more on serving students who really need direct assistance, such as borrowing and returning physical books, as well as handling more complex issues [6].

The results of post-implementation observations showed that queues at staff desks were reduced especially for general information questions. Students who previously had to wait for their turn to ask questions can now directly interact with *Chatbot* to get an answer. This not only improves service efficiency, but also creates a quieter and conducive library environment for learning activities. Interviews with library staff confirm that *Chatbot* helps reduce stress during rush hours, especially when many students are coming at the same time. Thus, the implementation of *Chatbot* successfully overcome the queue problem and create a more conducive service

atmosphere at the Esa Unggul Citra Raya University Library [4].

### 3.3 Provision of an Interactive and Responsive Communication Platform for Information Access Outside Operating Hours

One of the main limitations of conventional library services is limited access to information during operational hours which take place from Monday to Saturday at 09.00-17.00 WIB. The results of the questionnaire distributed to 168 students showed that 28.2% of respondents experienced problems not being able to access services when the library was closed or on holidays. This limitation is a problem for students who have a busy lecture schedule or who need information outside of operational hours to complete their academic assignments. This condition indicates the need for a communication platform that can be accessed flexibly without being bound by time and place restrictions.

Application *Chatbot* developed as an interactive communication platform that can be accessed 24 hours a day and 7 days a week (24/7), so that students can obtain library information at any time according to their needs. System *Chatbot* Implemented with a simple and intuitive user interface, it allows students to interact through text and voice input. Home *Chatbot* displays the conversation area that separates the user's message on the right and the reply *Chatbot* on the left. At the bottom of the interface is a text input field with a button *Send* to send messages and buttons *Speak* which allows the user to provide input through voice recognition (*speech-to-text*). Features *speech-to-text* implemented to improve accessibility for users who are more comfortable using voice commands, making the information search process more flexible and efficient [16].

System *Chatbot* Designed with a Architecture-Based *Web* using the Python programming language and *Flask framework* As *Backend* to set up application logic and data communication. MySQL database is integrated to store library service information, question-answer data (*Question and Answer/QnA*), as well as the available book collection. The platform can be accessed through various devices such as *Smartphone*, tablet, or computer, thus providing maximum flexibility for students to access information from anywhere. System responsiveness is maintained by ensuring response times *Chatbot* less than 1 second, which is evident from the results of the questionnaire in which 90.6% of respondents stated that *Chatbot* Responds very quickly [17].

The availability of this interactive and responsive communication platform provides an effective solution to the limited operating hours of the library. Students who need information on evenings, weekends, or holidays can still access the service through *Chatbot*.

The results of the questionnaire showed that 81.1% of respondents stated that *Chatbot* helps them save time when searching for information, which indicates that the platform has successfully met the need for unlimited information accessibility. The system is also equipped with *Dashboard Admin* which allows library staff to dynamically update information, so that the content available in the *Chatbot* always accurate and *up-to-date*. Thus, *Chatbot* successfully provides an interactive, responsive, and anytime-accessible communication platform to meet the information needs of students outside of library operating hours [18].

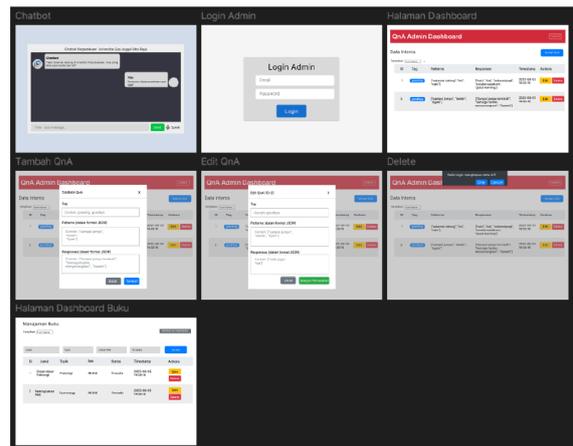


Figure 2. Chatbot interface and admin dashboard view for qna data management and book collections

### 3.4 Intelligent Technology Integration to Simplify the Information Search Process

The process of searching for information in the library often experiences obstacles because students do not know the exact location of books, availability status, or other service information. The results of the questionnaire showed that 87.3% of respondents experienced uncertainty about the availability of books, and 42.3% of respondents stated that information about the availability of books was their main need. The process of physically searching for books on library shelves is time-consuming, especially when students do not know the location or availability status of the books they are looking for. This condition requires intelligent technological solutions that can simplify the process of searching for information and provide accurate results in a short time.

Integration of intelligent technology in the application *Chatbot* carried out through the implementation of algorithms *Fuzzy Matching Using Levenshtein Distance* to calculate the edit distance between user input and query patterns stored in the database. This algorithm works by calculating the minimum number of character

insertion, deletion, and substitution operations required to change one *string* become *string* others. Value *Levenshtein Distance* then converted to *Similarity Score* in the range of 0-100% using a formula that considers length *string* longest as a divider [13]. The smaller the edit distance required, the higher the similarity value obtained. The system will provide an answer if the *Similarity Score* exceeds the threshold (*Threshold*) that has been set, whereas if the value is below the threshold, then the system will display the message *Fallback* which directs the user to contact the library staff directly.

The matching process begins with the pre-processing stage of text (*Text Preprocessing*) which includes the conversion of all inputs to lowercase (*Lowercase*) and punctuation to make the matching process easier. Once the text is processed, the system retrieves all the data *Pattern* and *Response* from the MySQL database. Then, the system calculates the similarity between the user's input and each *Pattern* Using algorithms *Levenshtein Distance*. The result of the calculation is converted to *Similarity Score*, and the system selects *Pattern* with the highest score. If the similarity score meets or exceeds *Threshold*, then the system displays the appropriate response based on *Tags* that fits. On the other hand, if the score is below *Threshold*, the system displays a message that *Chatbot* does not understand the questions and advises users to contact the library staff [19].

Advantages of the algorithm *Levenshtein Distance* lies in its ability to handle typing errors (*Typo*) that often happens when the user types a question. As an improved illustration, the calculation process *Levenshtein Distance* can be explained through the example of a comparison between the word "borrow" as *Target* and "Loan" as *Source* (Simulation *Typo* the letter "m" replaced by "rn"). Second *string* is six characters long, so a character comparison matrix is compiled measuring 7×7 (including *string* empty). The final result of the

calculation shows that the edit distance between "borrow" and "borrow" is 1, which means that only one character replacement operation is required. With the formula  $Similarity\ Score = (1 - 1/6) \times 100 = 83,3\%$ , this value exceeds *Threshold* 75% up to *Chatbot* can still give the right answer even if there is a *Typo* [9].

**Tuning Threshold.** To determine the optimal *threshold value*, the system was tested on a *test set* of 250 labeled questions with four candidate threshold scores. The test results are presented in Table 1.

Based on the Table 1, the 70% threshold produces the highest *false answer* rate (12.8%) because the system is too permissive in accepting matches. In contrast, the 85% threshold results in a *very high fallback* (20%), meaning that the system is so restrictive that many valid questions go unanswered. The 75% threshold was chosen as the optimal value because it provides the highest accuracy (84.4%) with the *best trade-off* between incorrect answers and *fallbacks*.

Table 1. Tuning Threshold Results in Test Set 250 Questions

<i>Threshold (%)</i>	<i>Accuracy (%)</i>	<i>Incorrect Answer (%)</i>	<i>Fallback (%)</i>
70	80,4	12,8	6,8
<b>75</b>	<b>84,4</b>	<b>8,4</b>	<b>7,2</b>
80	81,2	5,6	13,2
85	76,8	3,2	20,0

**Comparison with Baseline Method.** In order to validate the choice of the *Levenshtein Distance* algorithm, a comparison was made with the *TF-IDF-based cosine similarity* and the *Jaccard similarity* based on *character bigram* (n=2) in the *same test set*. The results of the comparison are presented in Table 2.

Table 2. Performance Comparison of Fuzzy Matching Method

<i>Method</i>	<i>Optimal Treshold</i>	<i>Accuracy (%)</i>	<i>Accuracy (%)</i>	<i>Recall (%)</i>	<i>F1 Score (%)</i>
<i>Jaccard</i> (bigram)	0,40	78,0	77,4	76,8	77,1
<i>TF-IDF Cosine</i>	0,60	81,6	82,0	80,9	81,4
<b><i>Levenshtein Distance</i></b>	<b>75%</b>	<b>84,4</b>	<b>85,2</b>	<b>83,7</b>	<b>84,4</b>

*Levenshtein Distance* superior to the second *baseline* on all metrics. This advantage can be explained by the characteristics of the input data: *Chatbot* Libraries tend to be short (3–8 words) with the main variations in the form of *Typo* single character, where *Levenshtein Distance* directly calculates the distance of the character so that it is more sensitive to these variations than *TF-IDF* which is based on the frequency of the token or *Jaccard* that is slice-based *Bigram* [9].

**Objective Evaluation of Chatbots.** The results of the evaluation using *Test set* 250 labeled questions showed that the system achieved 84.4% accuracy, *Precision* 85.2%, *Recall* 83.7%, and *F1 Score* 84.4%. Analysis *Error* shows that the most failures occur in: (1) questions with *Typo* Heavy (3+ characters) by 8.8%, (2) questions using synonyms not listed in the QnA pattern by 4.4%, and (3) long sentence questions (>10 words) by 2.4%. These findings indicate that future development needs to

integrate the *Word Embedding* or *Machine Learning* to handle more complex semantic variations. The results of the perception evaluation questionnaire are also in line with the objective findings, where 81.1% of respondents stated that *Chatbot* Be able to understand questions despite the existence of *Typo* [9].

The system is equipped with *Dashboard Admin* which allows library staff to dynamically manage QnA data and book collections. On *Book Dashboard*, admins can manage book collection information that includes ID, Title, Topic, Shelf, Status, *Timestamp*, and Action. The system also provides search and filter features based on Title, Topic, Shelf Location, and Status to make it easier for admins to find specific data. All changes made through *Dashboard Admin* will instantly update the MySQL database and be automatically reflected in the response *Chatbot* when students ask questions related to the availability of books. With this integration, students can easily search for book information by simply typing in the topic or title of the book, and *Chatbot* will provide complete information about the availability and location of the bookshelf in a short time [20].

System testing is carried out using the *Blackbox Testing* to verify that all functions are running according to the specifications that have been set. Testing covers a variety of scenarios, including home page access *Chatbot*, send a message using the *Send*, feature usage *speech-to-text* via the button *Speak*, ability *Chatbot* Answering questions as per the pattern in the database, handling unrecognized inputs with responses *Fallback*, authentication of admin logins with true and false credentials, and CRUD operations on QnA data and book collections [14].

The test results in Table 3 show that all scenarios provide results as expected with a status of "Successful", which proves that the integration of intelligent technology in the system *Chatbot* works well. In terms of the suitability of answers, 88.7% of respondents considered that the answers given were very appropriate to the questions asked, while only 5.7% considered it quite appropriate and 3.8% considered it inappropriate. These findings indicate that the integration of algorithms *Fuzzy Matching* successfully simplify the information search process by providing relevant and accurate responses. Comparison with previous research shows that the Python-based approach and *Flask framework* provides greater flexibility than the platform *Dialogflow* that are tied to a specific ecosystem. Thus, the integration of intelligent technology in the form of algorithms *Levenshtein Distance*, *Dashboard Admin* dynamic, and features *speech-to-text* successfully simplifies the process of searching library information and provides an optimal user experience [21].

Table 3. Blackbox Testing Results of Library Chatbot Application Testing

Yes	Testing Scenarios	Test Results	Conclusion
1	Access the chatbot's homepage	Conform	Successful
2	Send a message with the Send button	Conform	Successful
3	Voice input with Speak button	Conform	Successful
4	Questions with typos	Conform	Successful
5	Unrecognized inputs of the system	Conform	Successful
6	Admin login with incorrect credentials	Conform	Successful
7	Admin login with the correct credentials	Conform	Successful
8	Managing QnA data (CRUD)	Conform	Successful
9	Managing book data (CRUD)	Conform	Successful
10	Confirm data deletion	Conform	Successful

#### 4. Conclusion

This research successfully developed the application *Chatbot* based *Fuzzy Matching* with algorithms *Levenshtein Distance* to improve the efficiency of library information services at Esa Unggul Citra Raya University. Objective evaluation using *Test set* 250 labeled questions showed that the system achieved 84.4% accuracy, *Precision* 85.2%, *Recall* 83.7%, and *F1 Score* 84.4%, with *Threshold* optimal 75% set through the process *Tuning* to balance *False Answer* (8.4%) and *Fallback* (7.2%). Comparison with the *baseline* prove that *Levenshtein Distance* superior to *TF-IDF cosine similarity* (F1 81.4%) and *Jaccard bigram* (F1 77.1%), justifying the selection of the algorithm for the case *Chatbot* based on short questions with variations *Typo* character. System *Chatbot* Able to respond in less than 1 second and successfully reduce direct questions to staff from an average of 20 questions/day to 6 questions/day based on data *Log* and post-implementation staff records, with estimated savings of 1.17–2.33 hours of staff working time per day. The system has succeeded in reducing queues at library staff desks by providing an alternative access to information that students can use from their personal devices, thus creating a more conducive library atmosphere. The interactive communication platform available 24/7 allows students to access information outside of operating hours, overcoming the constraints experienced by 28.2% of respondents. Evaluate user perceptions through a scaled questionnaire *Likert* with reliability *Cronbach's Alpha*  $\alpha = 0.847$  indicates a

satisfaction rate of 86.8%, with 88.7% of respondents rating the answers to be very appropriate and 81.1% confirming the system's handling ability. *Typo*. This application can be implemented in the libraries of other educational institutions as a solution for digitizing information services. Future research is suggested integrating technology *Machine Learning* and *Word Embedding* to handle semantic and synonym variations that the algorithm has not yet been able to handle *Levenshtein Distance*, as an approach *Machine Learning* has been shown to be effective in a wide range of classification and pattern recognition tasks [22], as well as expanding the scope of integration with regional or national catalog systems.

### Acknowledgments

The author would like to thank Esa Unggul Citra Raya University for the support of the research facility, Mr. Ary Prabowo, S.Komp., M.Kom as a supervisor, as well as library staff and students who have participated in this research.

### References

- [1] V. C. Aini, "Developing the Function of Recreation as an Effort to Improve Services in the Library," *IQRA' J. Perpust. and Inf.*, vol. 16, no. 1, pp. 57–68, 2022, doi: :10.30829/iqra.v16i1.10313.
- [2] S. Endarti, "Libraries as a Place of Information Recreation," *ABDI PUSTAKA J. Perpust. and Archives*, vol. 2, no. 1, pp. 23–28, 2022, doi: 10.24821/jap.v2i1.6990.
- [3] D. Hidayah, Widodo, and E. Hasanah, "Optimizing the Implementation of Library Services to Improve Student Literacy," *Acad. Educ. J.*, vol. 15, no. 2, pp. 1504–1514, 2024, doi: 10.47200/aoej.v15i2.2512.
- [4] I. Permadi, "Criteria Selection and Comparative Analysis of Popular Chatbot Frameworks (Dialog flow, Microsoft Bot Framework, IBM Watson Assistant and Rasa) For Implementation in Libraries: a Systematic Literature Review," *JPUA J. Perpust. Univ. Airlangga Media Inf. and Commun. Librarianship*, vol. 13, no. 2, pp. 94–103, 2023, doi: 10.20473/jpua.v13i2.2023.94-103.
- [5] B. Alsafari, E. Atwell, A. Walker, and M. Callaghan, "Towards effective teaching assistants: From intent-based chatbots to LLM-powered teaching assistants," *Nat. Lang. Process. J.*, vol. 8, no. August, p. 100101, 2024, doi: 10.1016/j.nlp.2024.100101.
- [6] S. Sugiono, "The Role of Chatbots in Supporting Smart Services in Smart Libraries," *VISION LIBRARY Bul. Jar. Inf. Inter Library.*, vol. 23, no. 3, pp. 207–220, 2021.
- [7] Abel Uzoka, Emmanuel Cadet, and Pascal Ugochukwu Ojukwu, "Leveraging AI-Powered chatbots to enhance customer service efficiency and future opportunities in automated support," *Comput. Sci. IT Res. J.*, vol. 5, no. 10, pp. 2485–2510, 2024, doi: 10.51594/csitrj.v5i10.1676.
- [8] Z. A. Taufik and Supriyanto, "Implementation of Chatbot for Academic Frequently Asked Question Services with the Use of Dialogflow," *J. SAINTEKOM*, vol. 13, no. 1, pp. 1–10, 2023, doi: 10.33020/saintekom.v13i1.337.
- [9] A. R. Kaufman and A. Klevs, "Adaptive Fuzzy String Matching: How to Merge Datasets with Only One (Messy) Identifying Field," *Polit. Anal.*, vol. 30, no. 4, pp. 590–596, 2022, doi: 10.1017/pan.2021.38.
- [10] R. W. Saputra, C. Q. Pirera, and V. V. Verdana, "Risk Analysis of the Use of Waterfall and Prototyping Methods in Website Development," *JATI (Journal of Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 4405–4410, 2024.
- [11] C. P. Krishna and K. Bhargavi, "High Integrity Systems Using Extreme Programming (XP)," *Neuroquantology*, vol. 20, no. 22, pp. 3990–3996, 2022, doi: 10.48047/NQ.2022.20.22.NQ10398.
- [12] R. Prayogi, K. Ramanda, C. Budihartanti, and Arief Rusman, "Application of the PIECES framework method in the analysis and evaluation of M-BCA applications," *J. Infortech*, vol. 3, no. 1, pp. 1–10, 2021.
- [13] Khalidah, "Implementation of Levenshtein Distance Algorithm in the Digital Biology Dictionary Search Function," *Indonesian. J. Artif. Intell. Data Min.*, vol. 4, no. 2, pp. 89–96, 2021.
- [14] H. Zhang, J. Wang, Y. Wang, M. Li, J. Song, and Z. Liu, "ICVTest: A Practical Black-Box Penetration Testing Framework for Evaluating Cybersecurity of Intelligent Connected Vehicles," *Appl. Sci.*, vol. 14, no. 1, 2024, doi: 10.3390/app14010204.
- [15] U. Sekaran and R. Bougie, "Research Methods for Business," *Wiley*, 2016.
- [16] N. Lubis, M. Z. Siambaton, and R. Aulia, "Implementation of Deep Learning Algorithms in Online Speech to Text Applications with the Recurrent Neural Network (RNN) Method," *sudo J. Tek. Inform.*, vol. 3, no. 3, pp. 113–126, 2024, doi: 10.56211/sudo.v3i3.583.
- [17] A. Morchid *et al.*, "IoT-enabled fire detection for sustainable agriculture: A real-time system using flask and embedded technologies," *Results Eng.*, vol. 23, no. July, p. 102705, 2024, doi: 10.1016/j.rineng.2024.102705.
- [18] C. I. Nicola, M. Nicola, D. Sacerdotianu, and I. Pătru, "Real-Time Monitoring of Cable Sag and Overhead Power Line Parameters Based on a Distributed Sensor Network and Implementation in a Web Server and IoT," *Sensors*, vol. 24, no. 13, 2024, doi: 10.3390/s24134283.
- [19] M. A. A. Mosleh, A. Assiri, A. H. Gumaeci, B. F. Alkhamees, and M. Al-Qahtani, "A Bidirectional Arabic Sign Language Framework Using Deep Learning and Fuzzy Matching Score," *Mathematics*, vol. 12, no. 8, 2024, doi: 10.3390/math12081155.
- [20] A. Fu'adi and A. Prianggono, "Analysis and Design of the Academic Information System of the Pacitan State Community Academy Using UML and EER Diagrams," *J. Ilm. Technology. Inf. Asia*, vol. 16, no. 1, pp. 45–54, 2022, doi: 10.32815/jitika.v16i1.650.
- [21] U. Nugraha and T. Sianturi, "Blackbox Testing On E-Commerce System Web-Based Evermos (Feature: Registration Experiment & Revamp)," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 8, pp. 1026–1037, 2021.
- [22] A. Prabowo, W. A. Kusuma, M. Rafi, and others, "Identification of Java Tea Adulteration by Babadotan and Tekelan using Machine Learning," *J. Jamu Indones.*, vol. 7, no. 3, pp. 86–92, 2022.