Journal of Hypermedia & Technology-Enhanced Learning
Vol. 3, No. 1, February (2025), Pages 37–63
e-ISSN: 2985–9166 | 10.58536/j-hytel.165

RESEARCH ARTICLE

Automated Academic Supervisor Allocation Using the C4.5 Decision Tree Algorithm: A Scalable Web-Based Solution

Nike Maulani[1]*, Geovanne Farell[2], Ahmaddul Hadi[2], Dedy Irfan[2], Mazen Alzyoud[3], Sharshova Regina Nikolaevna[4]

[1] Informatics Engineering Education, Faculty of Engineering, Universitas Negeri Padang, **Indonesia**
[2] Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang, **Indonesia**
[3] Department of Computer Science, Faculty of Information Technology, Al al-Bayt University, **Jordan**
[4] International Information Technology University, Almaty, **Kazakhstan**

**Corresponding Author: nikemaulani@student.unp.ac.id*

This article contributes to:

4 QUALITY EDUCATION

SUSTAINABLE DEVELOPMENT GOALS

ABSTRACT

Efficient allocation of academic supervisors is a critical yet challenging process in higher education, often hindered by mismatches in expertise and uneven workload distribution. This study introduces a web-based recommendation system leveraging the C4.5 decision tree algorithm to address these issues. The system generates data-driven, accurate recommendations by assessing supervisor expertise, workload, and alignment with student research topics. The system emphasizes scalability and modularity and was developed using the Laravel framework and following the waterfall development model. Functional testing demonstrated a 96.7% accuracy rate for recommendations, while usability testing reflected high user satisfaction, with an average score of 92% for ease of use and relevance. These results highlight the system's effectiveness in optimizing supervisor assignments, enhancing administrative efficiency, and providing a scalable solution for educational institutions. Future work will further integrate diverse data sources and AI-driven features to improve adaptability and responsiveness to evolving academic demands.

KEYWORDS

Recommendation system; C4.5 algorithm; academic supervision; higher education

Received: Nov. 03, 2024; *Revised:* Dec. 10, 2024; *Accepted*: Jan. 04, 2025; *Published Online:* Jan. 15, 2025

How to Cite: Maulani, N., Farell, G., Hadi, A., Irfan, D., Alzyoud, M., & Nikolaevna, S. R. (2025). Automated Academic Supervisor Allocation Using the C4.5 Decision Tree Algorithm: A Scalable Web-Based Solution. *Journal of Hypermedia & Technology-Enhanced Learning*, *3*(1), 37–63. https://doi.org/10.58536/j-hytel.165

Published by Sagamedia Teknologi Nusantara

1. INTRODUCTION

The rapid advancement of information technology has profoundly transformed the educational landscape, enabling the global sharing and exchange of academic resources without temporal or spatial limitations [1]. Beyond its influence on social and business activities, this technological evolution has reshaped educational paradigms by providing unprecedented access to information, facilitating cross-border academic collaboration, and accelerating innovation across disciplines [2]. Among its most significant applications in higher education is the deployment of web-based information systems, which offer students and educators continuous, location-independent access to essential academic resources and institutional information [3], [4].

Academic information systems have become indispensable tools for enhancing operational efficiency within higher education institutions. They play a critical role in managing complex administrative processes such as course scheduling, assessment management, and student supervision for final projects [5], [6], [7]. These systems streamline administrative tasks and align with the objectives outlined in Indonesian Law Number 12 of 2012 on Higher Education, which emphasizes fostering students' skills, shaping their character, and advancing national development. A cornerstone of undergraduate education in Indonesia is the final project—a critical requirement for degree conferral. To support this, universities must assign faculty members with appropriate expertise to guide students effectively through their research and development processes [8].

The selection of academic supervisors is a critical determinant of the final project's success. Supervisors are pivotal in ensuring students complete their research projects[9]. However, inadequate assignment processes can result in mismatches between supervisors' expertise and students' research topics and unequal workload distribution among faculty members. These inefficiencies may compromise the quality of academic guidance [10]. To address this, the Rector's Decree of Universitas Negeri Padang No. 02 of 2020 mandates that final project supervisors possess expertise relevant to the student's research areas and hold at least the rank of Assistant Professor, ensuring specialized and qualified guidance.

Despite these regulations, the Faculty of Engineering at Universitas Negeri Padang faces challenges in achieving accurate and equitable supervisor assignments for final projects. Although a web-based information management system supports project administration, the supervisor selection process remains manual and is managed by the Head of Study Program (HSP). This traditional approach often results in suboptimal distribution of supervisory workloads and imprecise alignment between faculty expertise and student project topics, leading to imbalances that hinder the effectiveness of academic guidance.

To address these challenges, this study proposes a web-based recommendation system for selecting final project supervisors, employing the C4.5 decision tree algorithm to ensure balanced and accurate supervisory assignments [11]. The C4.5 algorithm is particularly well-suited for this application, as it handles datasets with complex attributes and generates interpretable and reliable recommendations. Prior studies, such as that by Harryanto et al. [12], have demonstrated the algorithm's effectiveness, achieving an accuracy exceeding 71% in similar applications. By leveraging the C4.5 algorithm, this study seeks to minimize existing imbalances, optimize workload distribution, and enhance the efficiency of final project management.

This study contributes to the development of a novel recommendation system designed to assist the HSP in making objective and efficient decisions for assigning final project supervisors. The proposed system promotes a fair and balanced distribution of advising responsibilities by incorporating key factors such as faculty expertise, alignment with project topics, and current supervisory workloads. Furthermore, the system has substantial potential for broader application in higher education institutions, where it could improve faculty workload management, enhance academic services, and support the creation of a more efficient and effective educational environment.

2. METHODS

2.1. Research Design

This study employs a developmental research approach, focusing on the design and implementation of a web-based recommendation system for assigning final project supervisors. The research addresses persistent challenges such as imbalanced advising workloads and inefficiencies in managing final project supervision by automating the supervisor selection process.

The system leverages the C4.5 decision tree algorithm to provide precise recommendations based on critical parameters, including faculty expertise, current advising workloads, and the alignment between students' project topics and faculty members' areas of specialization. The C4.5 algorithm was selected for its robustness and ability to handle large datasets while producing interpretable and reliable classification rules [13], [14].

The algorithm calculates entropy to measure data uncertainty and employs information gain to identify the attribute with the highest discriminatory power for partitioning the dataset [15]. This ensures the decision tree effectively represents the relationships among attributes, yielding reliable and data-driven recommendations for supervisor assignments. The entropy calculation is expressed by the following formula (1):

$$Entropy\ (S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{1}$$

Explanation:

S	Entropy of the dataset
n	Number of classes in the dataset
p_i	The probability of each class i in the dataset, calculated as the ratio of the number of samples in class i to the total number of samples

The algorithm calculates information gain for each attribute to determine its effectiveness in reducing dataset uncertainty. Information gain is computed as the difference between the initial entropy of the dataset and the entropy after partitioning based on a specific attribute. This metric identifies the attribute that most effectively minimizes uncertainty, serving as the optimal criterion for further partitioning the dataset within the decision tree. The calculation for information gain is expressed by the following formula (2):

$$Information\ Gain\ (S, A) = Entropy\ (S) - \sum_{v \in A} \frac{|S_v|}{|S|} \times Entropy\ (S_v) \tag{2}$$

Explanation:

S	Initial dataset
A	Attribute to be tested for information gain
S_v	Subset of S where attribute $A = v$
$\|S_v\|$	Number of samples in the subset S_v

$\|S\|$	Total number of samples in dataset S
$Entropy\ (S)$	Entropy of the initial dataset
$Entropy\ (S_v)$	The entropy of the subset S_v

In this study, the variable "umbrella research" demonstrated the highest information gain, with a value of 0.116, indicating its substantial influence on the supervisor recommendation process. This result underscores the critical role of this variable in reducing uncertainty and enhancing the accuracy of the recommendations.

The decision tree derived from these calculations established a hierarchical structure for decision-making. It prioritized key variables such as field of expertise, balanced supervisory load, and involvement in umbrella research. The structure of the decision tree is illustrated in Figure 1, which demonstrates the attribute selection process at each branch. The figure visually captures how the algorithm optimally partitions data to deliver precise, data-driven supervisor recommendations.

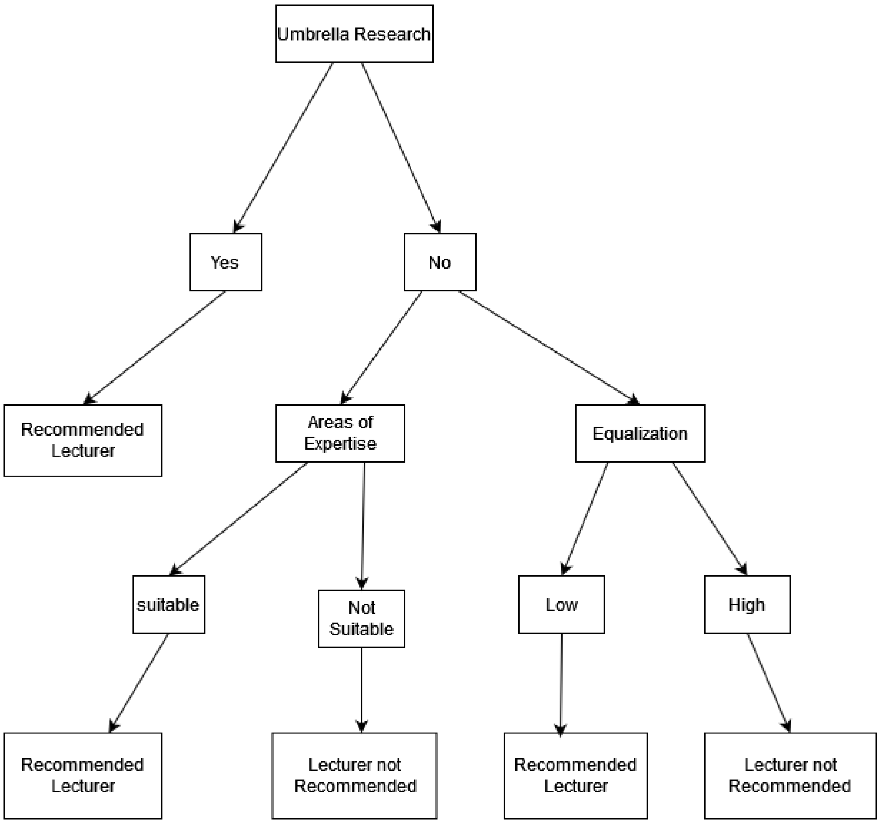


Figure 1. Structure of the C4.5 decision tree for academic supervisor allocation

2.2 System Development Method

The development of the web-based recommendation system for final project supervisor selection utilized the Waterfall methodology, as shown in Figure 2. This methodology consisted of sequential phases: requirements analysis, system design, implementation, testing, and maintenance [16], [17]. The Waterfall approach was chosen for this study because it enabled a systematic and structured development process, ensuring that each phase was thoroughly completed before moving on to the next [18], [19].

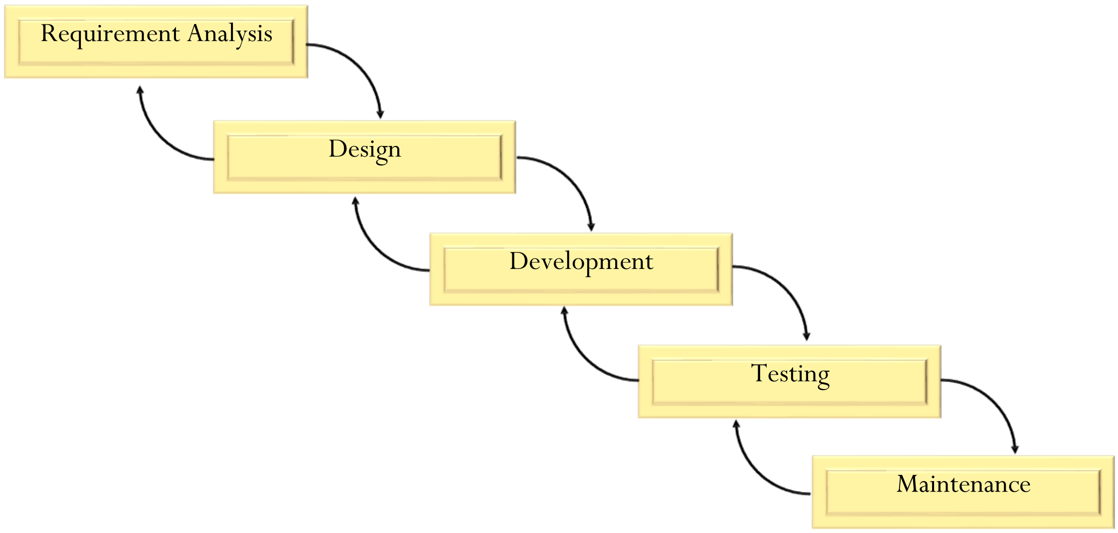


Figure 2. Waterfall methodology

2.2.1 Requirement Analysis

The requirements analysis phase focused on identifying the specifications necessary to address the challenges in selecting final project supervisors. At the time, the conventional supervisor selection process often overlooked optimal workload distribution, potentially resulting in imbalanced faculty advising responsibilities. The proposed system was designed to offer an efficient and objective solution through automation. The requirements analysis included the following key steps:

a) Identification of Key Users

The primary user of this system was the Head of Study Program (HSP), who held responsibility for assigning academic supervisors to students undertaking final projects. In addition to the HSP, faculty members could access the system to review supervisory workload distribution, and students could use it during the supervisor selection process. However, the HSP's role remained central, as they held the authority for final decision-making in supervisor assignments.

b) Functional Requirements

The system supported data management for both faculty and students, including the input of faculty competencies, supervisory loads, and student project topics. Additionally, the system provided an automated recommendation feature for supervisor assignments using the C4.5 decision tree algorithm.

c) Non-Functional Requirements

The system featured a user-friendly interface that allowed for intuitive navigation, making it accessible for users with varying levels of technical proficiency. Furthermore, it was accessible online to ensure users could access the system from any location without requiring specialized software. Fast processing times were also essential to support operational efficiency, enabling real-time recommendations and minimizing delays in the supervisor selection process.

d) Algorithmic Requirements

The C4.5 decision tree algorithm was chosen as the foundation for generating supervisor recommendations due to its ability to process data with several complex attributes and produce accurate classifications. The algorithm received input data from faculty and student records, encompassing faculty competencies, supervisory workloads, and student project topics. It then calculated information gain values to identify the most influential variables, such as faculty expertise and workload distribution, ensuring optimal supervisor recommendations. The algorithm constructed a decision tree that categorized faculty based on predefined criteria, aiming to balance supervisory responsibilities and align expertise with project topics. This decision tree was updated periodically to accommodate faculty and student data changes.The C4.5 decision tree algorithm was chosen as the foundation for generating supervisor recommendations due to its ability to process data with several complex attributes and produce accurate classifications. Key requirements for implementing this algorithm in the system included:

2.2.2 System Design

The system design phase aimed to translate identified requirements into a suitable system architecture and user interface. The developed design included:

a) System Architecture

The system was built using a web-based architecture supported by the Laravel framework, which enabled modular and efficient application development. This architecture separated the front-end and back-end components, each crucial in ensuring optimal system functionality and performance.

a. Front-end

The front end was responsible for providing an intuitive and responsive user interface (UI), allowing users—namely the Administrator, HSP, and faculty members—to interact with the system seamlessly [20], [21]. Developed with technologies such as HTML, CSS, and JavaScript, the interface was designed to be user-friendly, featuring straightforward navigation for accessing core functionalities such as data input, recommendation management, and report visualization.

b. Back-end

The back-end handled data processing, application logic, and database management. Leveraging Laravel as the primary framework, the back end followed the Model-View-Controller (MVC) structure, which separated business logic from data presentation, supporting modularity and flexibility for maintenance [22]. The back-end processed user inputs, ran the C4.5 decision tree algorithm for supervisor recommendations, and stored both recommendation results and related data in the database.

c. Database

The system utilized MySQL as the database management system to store and manage structured data, including faculty details, student information, final project topics, and recommendation results. The database was designed with a relational model, allowing efficient data retrieval and simplified data maintenance over time.

b) User Interface (UI) Design

The UI design of this system was created to be intuitive and user-friendly, focusing on the needs of the primary user, the HSP, as well as other users such as the Administrator and faculty members. The interface was developed to facilitate easy navigation and quick access to key system features, thus supporting efficiency in daily use.

a. Dashboard UI

The dashboard UI centralizes access to essential recommendation system data, including faculty members, students, faculty competencies, and student project topics. Its user-friendly design enables efficient navigation and quick data retrieval, minimizing input and navigation errors. Figure 3(a) illustrates the dashboard design.

b. Recommendation Criteria Settings

The interface also includes a criteria settings panel, allowing the HSP to adjust recommendation criteria according to specific needs. These settings include options for selecting faculty areas of expertise, current supervisory loads, and specific preferences related to student project topics. The feature is presented in dropdown and checkbox formats to facilitate easy selection and increase flexibility in settings, as shown in Figure 3(b).

(a) (b)

Figure 3. (a) The design of the dashboard UI; (b) The design of the recommendation criteria settings

c. Recommendation Results Display

The UI displays recommendation results in an informative and easy-to-read format. Each recommendation for faculty advisors includes indicators of suitability based on set criteria, such as expertise matching level and supervisory load. This display aids the HSP in understanding the rationale behind each recommendation and making informed decisions, as presented in Figure 4(a).

d. Report Visualization

The UI includes a report visualization feature to support analysis and evaluation, such as graphs or tables displaying the distribution of supervisory loads and recommendation history data. This visualization is designed to help users understand overall supervisory load patterns and make adjustments if necessary, as depicted in Figure 4(b).

(a) (b)

Figure 4. (a) The design of the recommendation results in UI; (b) The design of the report visualization

c) Process Design

In the process design phase, the data flow from students and faculty members was visualized in a flowchart to illustrate how the C4.5 decision tree algorithm processes data to generate supervisor recommendations. In addition to the flowchart, Unified Modeling Language (UML) diagrams were used to represent the system's interactions, activities, process sequences, and class structures. The following diagrams were designed to support this recommendation system:

a. Use Case Diagram

The Use Case Diagram, as shown in Figure 5, illustrates the interactions between actors (users) and the system. This diagram shows the main functionalities available within the system and the roles of users such as the HSP, faculty members, and students. The primary actor in this system is the HSP, who is responsible for selecting supervisors based on the system's recommendations.

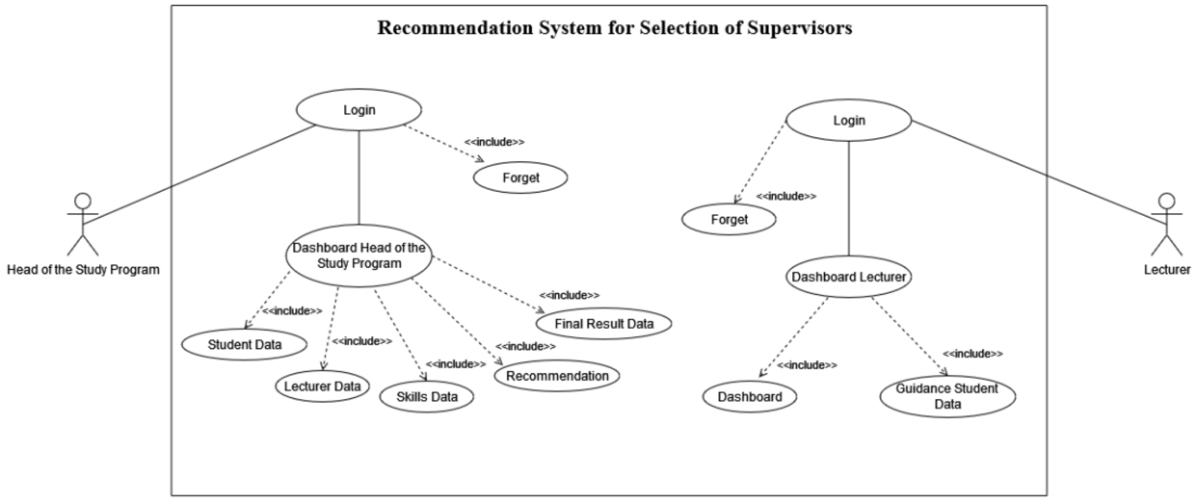


Figure 5. Use case diagram

b. Activity Diagram

The Activity Diagram represents the workflow and activities within the system across several scenarios. Each user's primary activity has a diagram illustrating how users interact with the system at each stage. These activity diagrams include:

1. Login Activity Diagram

This diagram illustrates the process users go through to log in to the system. The login activity is the initial step to ensure secure access and user authentication. The detailed login flow is visualized in Figure 6.

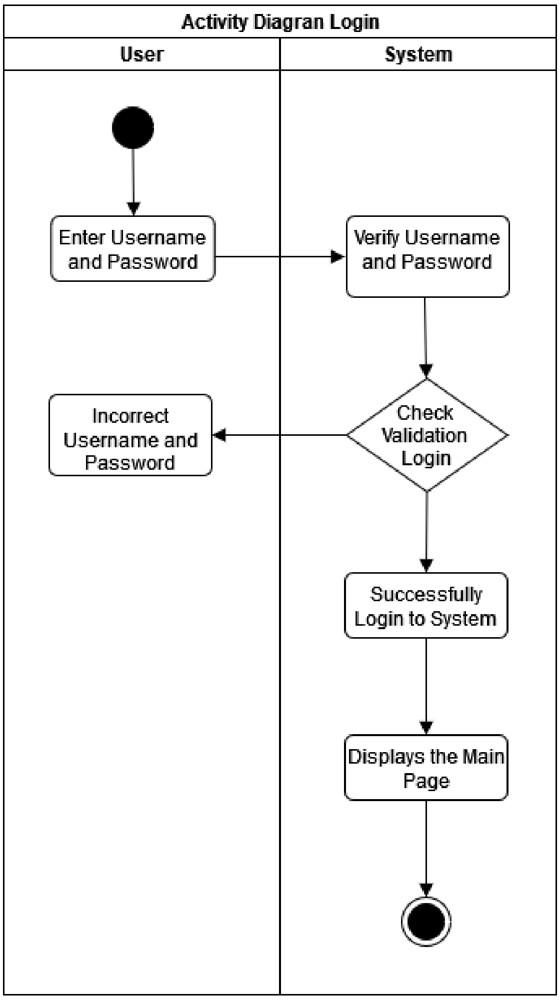


Figure 6. Activity diagram for the user login process

2. Head of Study Program (HSP) Activity Diagram

This diagram illustrates the HSP's activities in executing the system's primary functions, such as managing student data, setting recommendation criteria, and selecting academic supervisors. The diagram provides an overview of the HSP's workflow from the beginning to the end of the supervisor selection process, as shown in Figure 7.

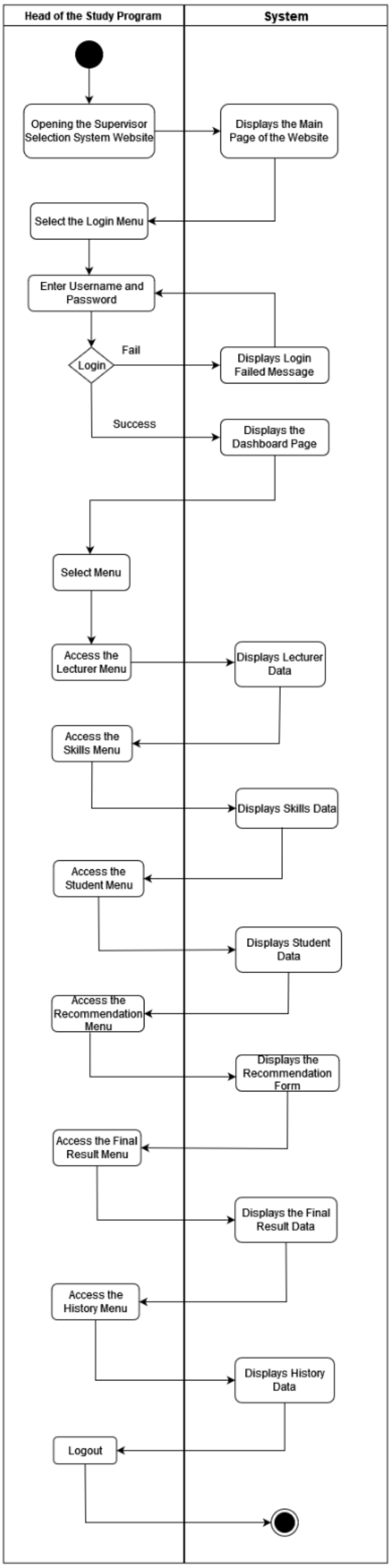


Figure 7. Activity diagram for the head of study program's workflow in the supervisor selection process

3. Faculty Activity Diagram

This diagram shows the activities of faculty members about the system, such as viewing student final projects or updating supervision status. These activities facilitate faculty members in accessing relevant information regarding the final projects they supervise, as illustrated in Figure 8(a).

4. Recommendation Activity Diagram

This diagram illustrates the HSP's activity flow when utilizing the recommendation feature. The system processes student and faculty data according to the established criteria and generates supervisor recommendations. The recommendation workflow is shown in Figure 8(b).

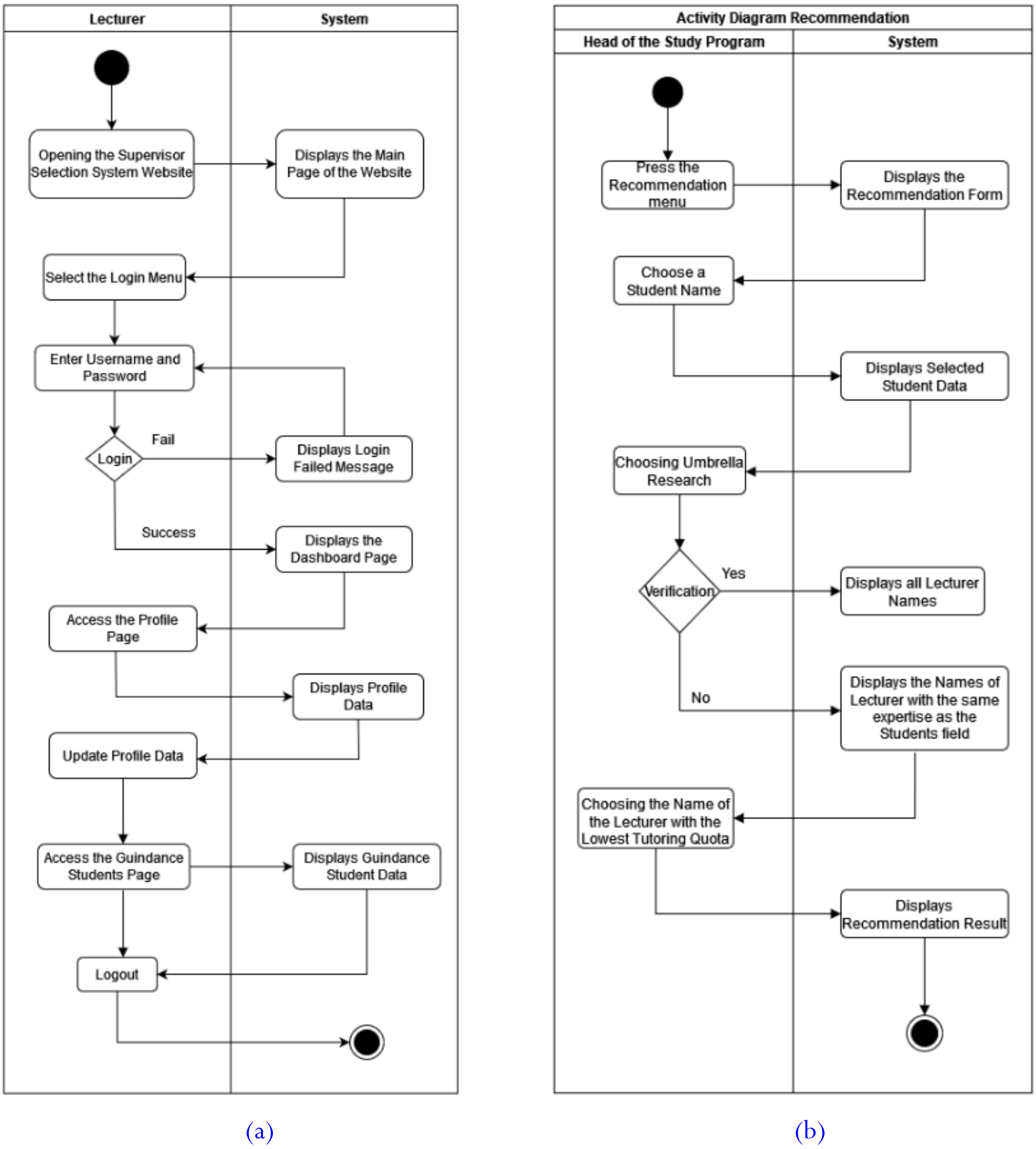


Figure 8. (a) Activity diagram for faculty member activities in supervising final projects; (b) Activity diagram for the recommendation process

c. Sequence Diagram

The Sequence Diagram illustrates the interactions between objects within the system in a specific sequence over time. This diagram shows how messages and commands are sent between users and the system to complete specific processes. The sequence diagrams designed for this system include:

1. Login Sequence Diagram

This diagram describes the login process, starting from the user entering credentials into the system, verifying the data, and granting access to the user. Each step performed by the user and the system during the login phase is visualized in Figure 9.

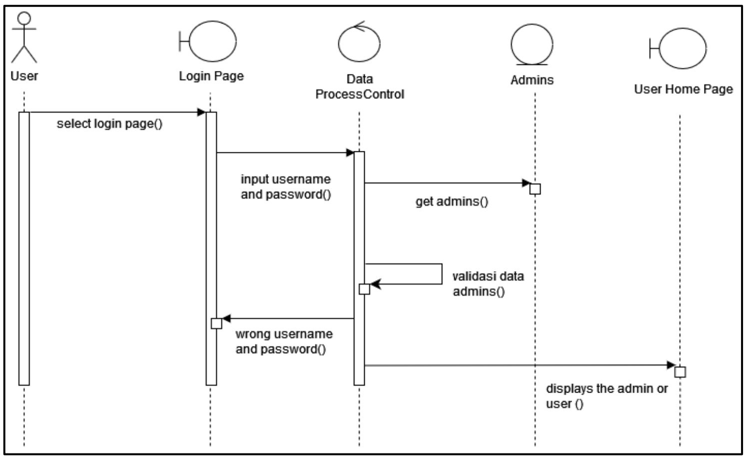


Figure 9. Sequence diagram for user login process

2. Recommendation Sequence Diagram

This diagram illustrates the supervisor recommendation feature in the HSP. The process includes submitting student and faculty data, processing it using the C4.5 algorithm, and returning the recommendation results to the system. This diagram is shown in Figure 10.

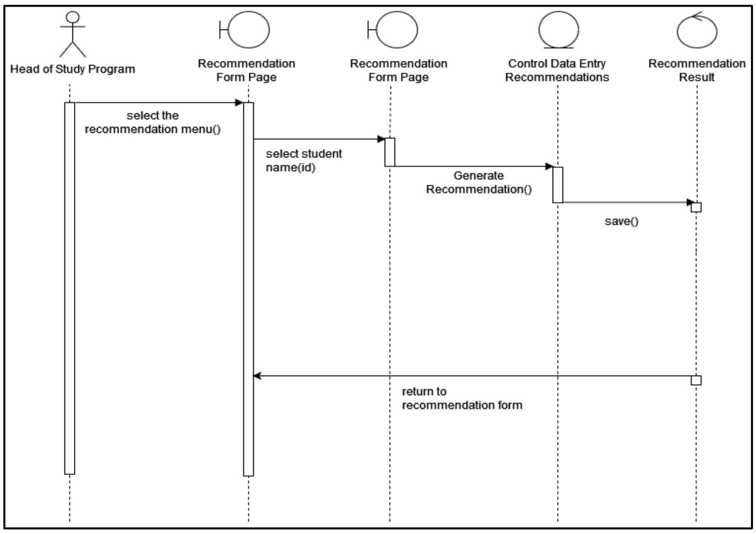


Figure 10. Sequence diagram for the supervisor recommendation process

2.2.3 Implementation

The implementation phase involved translating the designed architecture into program code and database structure to develop an effective recommendation system for selecting supervisors based on specific criteria. The process began with data preparation, where information regarding faculty members and students was input into the system as initial data for the recommendation process. This data included faculty competencies, current supervisory workloads, and student final project topics.

Next, the C4.5 decision tree algorithm was implemented in PHP using the Laravel framework to generate supervisor recommendations. Functional integration combined data input, recommendation processing, and result display, preparing the system for testing.

2.2.4 Testing

The testing phase aimed to verify that each system feature functioned according to the specified requirements, ensuring the system could reliably provide accurate and relevant recommendations. Testing was conducted using the black box testing method, which focuses on testing functionality without considering the internal structure of the code [23], [24]. This testing covered several key aspects:

a) Functional Testing of Input and Output

This testing ensured that the system could correctly receive and process input data while generating outputs meeting user requirements. In this stage, relevant data, such as faculty competencies, supervisory workloads, and student project topics, were entered into the system to verify that the data were stored, accessed, and managed correctly according to the specified functional requirements. The resulting output, including supervisor recommendations and reports on workload distribution, was then reviewed to confirm its alignment with the input data.

A specially designed test sheet based on the black box testing method with a checklist approach was used to record the success of each system feature. This approach allowed testers to assess whether each function operated as expected by observing the input and output without examining the internal code. Each feature was evaluated with “Yes/Y” (functional) or “No/N” (non-functional) as success indicators. Additionally, testing was conducted based on the roles of the primary users (Administrator, HSP, Faculty) to assess the functionality relevant to each role. If any feature was non-functional, this analysis helped identify specific user roles that might require further system adjustments.

All test results and analyses were documented in detail. Specific recommendations for improvement were provided for any feature that failed during testing to address any identified issues, thereby enhancing system quality in subsequent maintenance phases. This documentation also served as a foundation for further system development and maintaining system performance consistency.

b) Algorithm Testing

The algorithm testing aimed to ensure that the C4.5 decision tree algorithm implemented in the system could generate relevant and accurate supervisor recommendations based on predefined parameters. These parameters included faculty competencies, current supervisory workloads, and the alignment between the student’s project topic and faculty expertise. Testing was conducted using several test data scenarios to observe whether the algorithm could consistently produce relevant recommendations that meet the specified criteria.

The steps in algorithm testing involved processing test data with diverse variable values to assess the algorithm's sensitivity and flexibility across different scenarios. The results of each test were analyzed to verify that the generated recommendations met system requirements and reflected the prioritized criteria as designed. Any inconsistencies or anomalies in the recommendations were documented and analyzed to allow for adjustments or improvements to the algorithm as necessary.

c) User Practicality Testing

User practicality testing ensured the system was easy to use and understand, providing an efficient and satisfactory user experience. This testing evaluated aspects such as ease of navigation, system speed, and user comprehension of the recommendations generated by the system. The three main user roles involved in this testing—Administrator, HSP, and Faculty—were assessed to ensure that each role could effectively use the relevant features according to their needs. A combination of direct observation and user satisfaction surveys was used in this testing:

a. Direct Observation

Testers observed users interacting with the system to identify any barriers or difficulties encountered when using specific features.

b. User Satisfaction Survey

A Likert-scale-based survey was conducted to collect user responses regarding ease of navigation, system speed, clarity of recommendation interpretation, and willingness to use the system in the future. Users rated each item on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree).

The results of the user practicality testing were recorded and analyzed to obtain the average score and percentage for each evaluated aspect. System improvement recommendations were included in the test results documentation if any practical aspects were rated as insufficient. Practicality categories were evaluated based on Table 1.

Table 1. Practicality categories

Average Score (1-5)	Percentage (%)	Practicality Category
4.6 - 5.0	92% - 100%	Highly Practical
4.0 - 4.5	80% - 91%	Practical
3.0 - 3.9	60% - 79%	Moderately Practical
2.0 - 2.9	40% - 59%	Less Practical
1.0 - 1.9	20% - 39%	Not Practical

2.2.5 Maintenance

After the implementation and testing phases were completed, the system entered the maintenance phase to ensure it continues functioning effectively and meets user needs over the long term [17]. Maintenance activities included data updates involving periodic adjustments to the system in response to changes in faculty or student data that could affect supervisor recommendations. Additionally, system modifications were made based on user feedback, such as adding or refining features to enhance system functionality and usability. Security management also became a crucial part of this maintenance phase to safeguard privacy and protect faculty and student data from unauthorized access.

3. Results

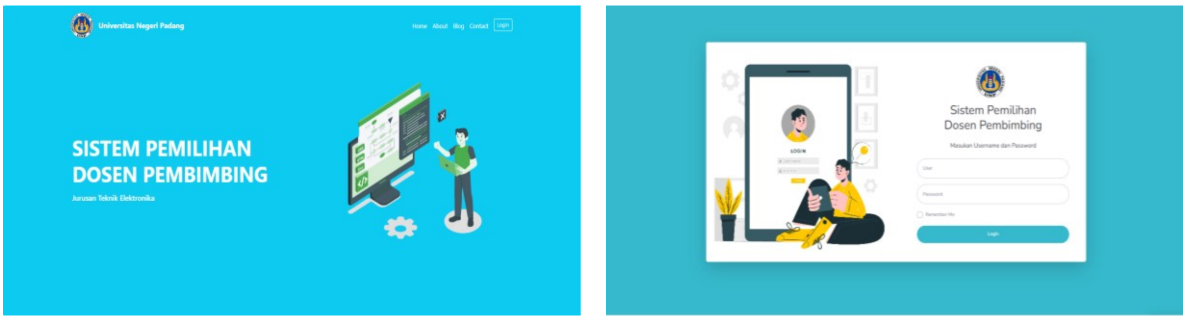
The recommendation system for selecting final project supervisors was successfully designed with a web-based architecture using the Laravel framework, which separates the user interface (front-end) from data processing (back-end). The system includes key features such as faculty and student data input, recommendation criteria settings, and the display of supervisor recommendation results.

3.1. Main Website Page Display

The main page is the first page users see when accessing the recommendation system application. This page serves as the gateway to all features available within the system. The main display includes a primary navigation menu with options: Home, About, Blogs, Contact, and Login. The Home menu directs users back to the main page, while the About section provides information about the system. The Blogs and Contact sections offer additional resources to help users better understand the application's functions and objectives. The Login option allows authorized users to access the system, as depicted in Figure 11(a).

3.2. Login Page Display

The login page is displayed after the user clicks the Login menu on the main page. This page is the primary access point for registered users to enter the system, whether as an Administrator, HSP, or Faculty member. Users enter their assigned username and password, and upon successful login, they are directed to their role-specific page within the system. This page helps maintain secure access to the data, ensuring that only authorized users can access certain features, as shown in Figure 11(b).



(a) (b)

Figure 11. (a) The main website page display; (b) Login page display

3.3. Faculty Page

The faculty page provides specific access for the HSP to manage and view faculty data and expertise information. Faculty data includes details about faculty members within the Informatics Engineering Education Program under the Department of Electronics Engineering, such as name, position, and available supervision quota. Meanwhile, the expertise data contains information on each faculty member's area of expertise, which is instrumental in the final project supervisor selection process. This page enables the HSP to obtain necessary information quickly, facilitating decision-making related to the distribution of supervisory workloads, as shown in Figure 12. The Expertise Data page also displays detailed information about each faculty member's specialization areas, which supports the final project supervisors' selection process, as illustrated in Figure 13.

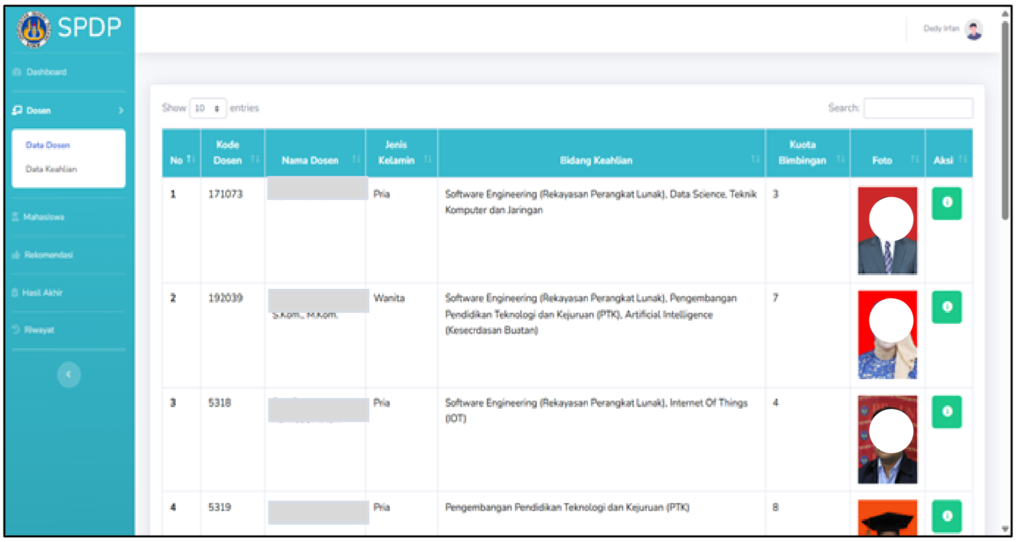


Figure 12. Faculty page display

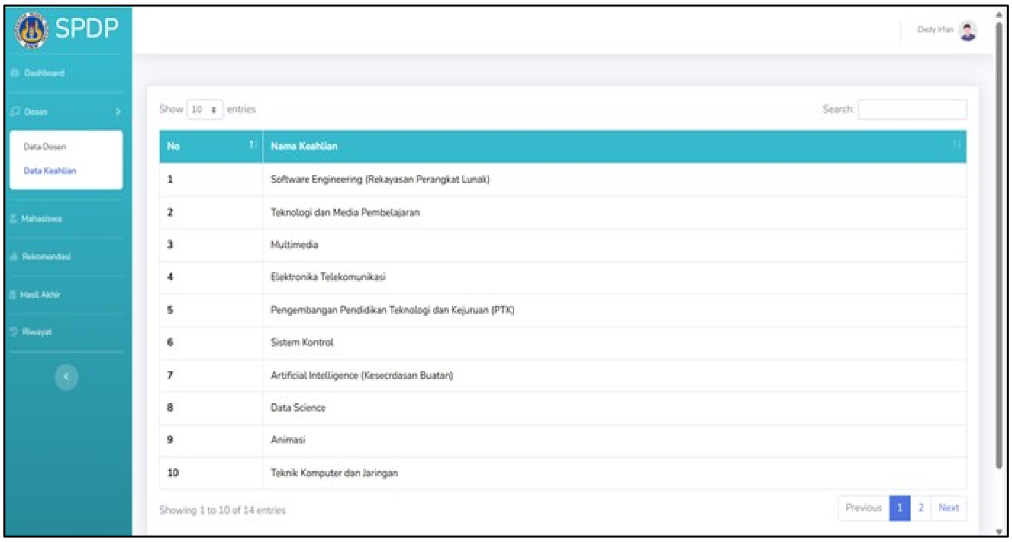


Figure 13. Expertise data page display

3.4. Student Page

As shown in Figure 14, the student page provides the HSP with access to view and manage student data within the Informatics Engineering Education Program. Available student data includes names, student identification numbers, and proposed final project topics. The HSP can update student information as needed to ensure that data remains current before initiating the supervisor recommendation process. This page assists the HSP in ensuring that each student receives guidance aligned with their field of study.

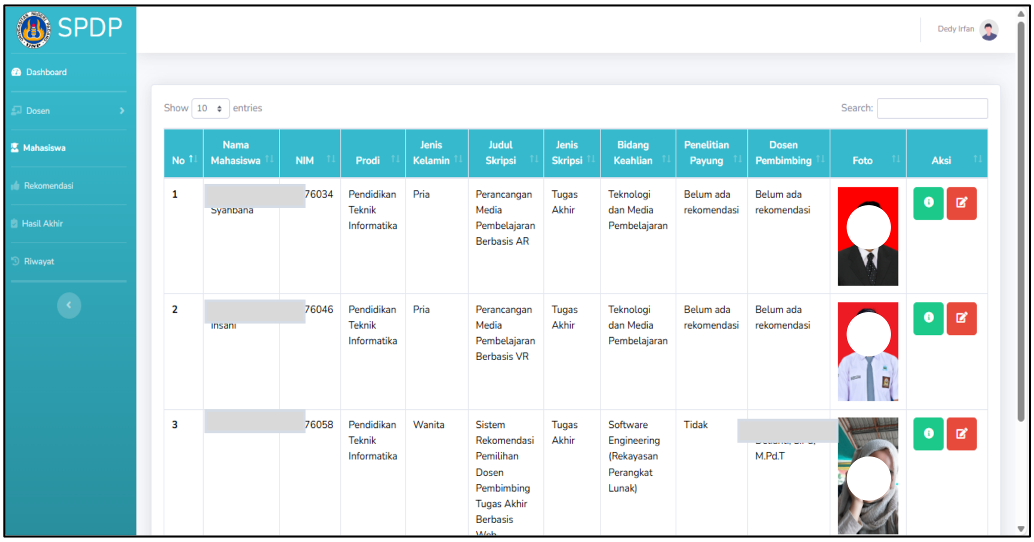


Figure 14. Student page display

3.5. Recommendation Page

The recommendation page, as shown in Figure 15, is the system's core, where the Head of Study Program (HSP) can assign academic supervisors for students requiring final project guidance. On this page, the HSP can select students from the available list and fill out a recommendation form based on umbrella research variables, field of expertise, and supervisory workload distribution (supervision quota). Once all criteria are set, the HSP can click the "Generate Recommendation" button to process the data and produce a list of recommended supervisors. This page also displays recommendation results, showing the names of recommended supervisors, their field of expertise, and supervision quota information to support informed decision-making, as illustrated in Figure 16.

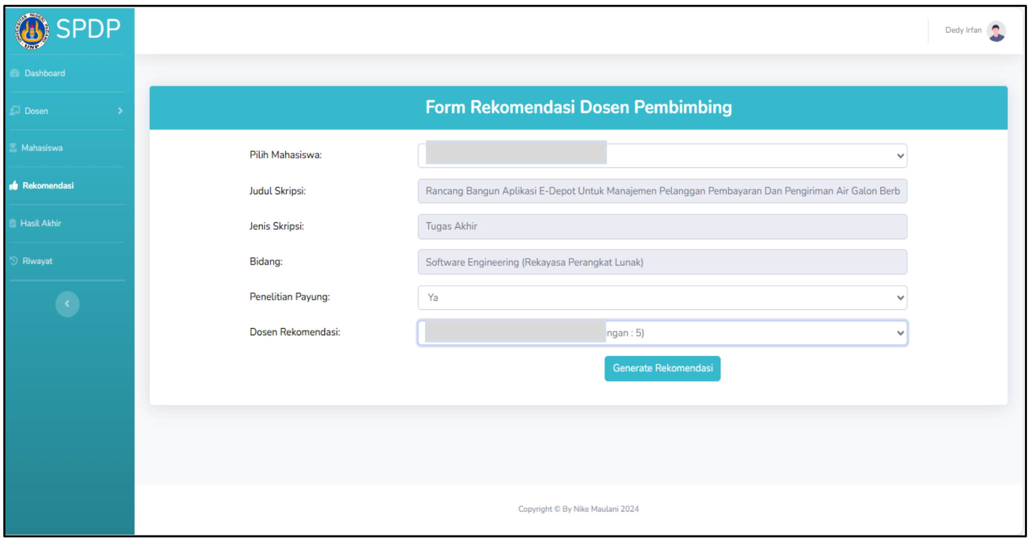


Figure 15. Recommendation form page display

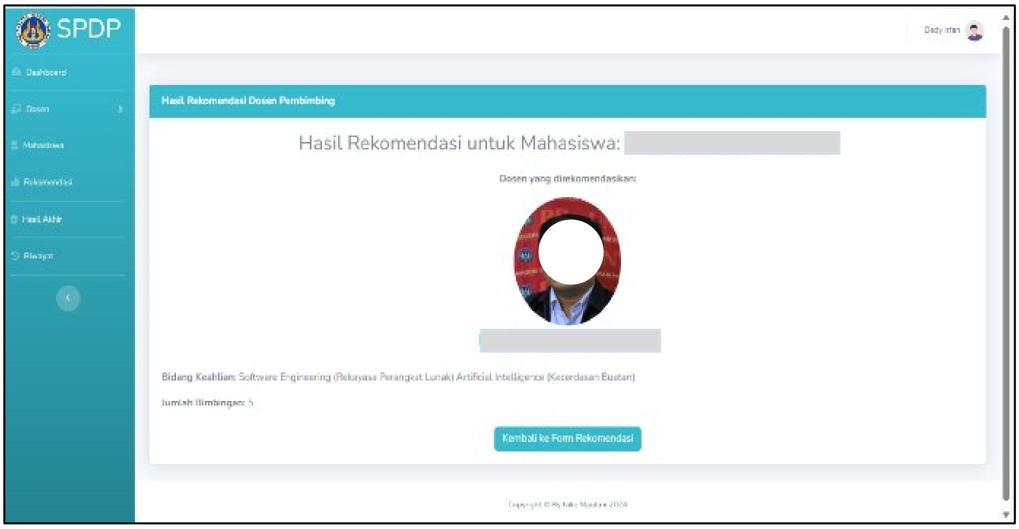
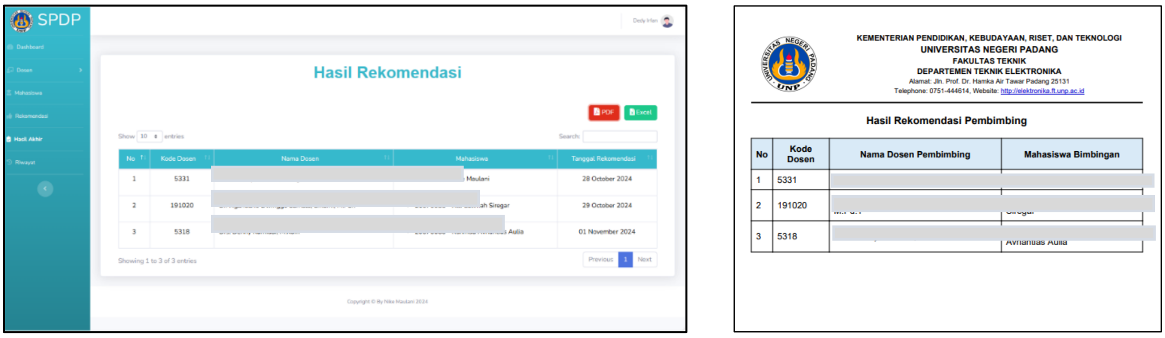


Figure 16. The recommendation results page display

3.6. Recommendation Results Report

Once the supervisor's recommendations have been established, the system provides the Recommendation Results Report feature to document the recommendations in a printable format. This report includes the names of the recommended supervisors and a list of students assigned to them, along with faculty codes for easier filing and identification. This page assists the Head of Study Program (HSP) in archiving the recommendation results as a reference or official document for further administrative processes, as shown in Figure 17. Specifically, Figure 17(a) illustrates the PDF download page, while Figure 17(b) displays the report view.



(a) (b)

Figure 17. Recommendation Display: (a) PDF download page (b) Report display

3.7. Results of System Functionality Testing

Functionality testing was conducted for each main feature within the system using the black box testing method, and three experts evaluated it. This testing focused on whether each system feature operated according to the specified requirements and met user needs. The recommendation system for selecting final project supervisors was tested based on user groups: Administrator, HSP, and Faculty (lecturer), to ensure each user could access and utilize the relevant features. Table 2 presents the results of the testing on the system's functionality. Each feature was evaluated based on the criteria of Yes/Y (functioning correctly) and No/N (not functioning according to specifications).

Table 2. System functionality testing results (black box)

No.	User	System Functionality	E1	E2	E3
1	Admin, HSP, and Lecturer	All navigation features within the supervisor recommendation system function properly.	Y	Y	Y
2	Admin, HSP, and Lecturer	The login menu functions properly.	Y	Y	Y
3	Admin, HSP, and Lecturer	The Logout menu functions properly.	Y	Y	Y
4	Admin, HSP, and Lecturer	The user profile editing menu functions properly.	Y	Y	Y
5	Admin and HSP	Menu for managing lecturer data functions properly.	Y	Y	Y
6	Admin and HSP	Menu for managing expertise data functions properly.	Y	Y	Y
7	Admin and HSP	Menu for managing student data functions properly.	Y	Y	Y
8	Admin and HSP	Menu for managing recommendations functions properly.	Y	Y	Y
9	Admin and HSP	Menu for managing recommendation results functions properly.	Y	Y	Y
10	Admin and HSP	Menu for managing history functions properly.	Y	Y	Y
11	Admin and HSP	Menu for exporting results data to PDF functions properly.	Y	Y	Y
12	Admin and HSP	Menu for exporting results data to Excel functions properly.	Y	Y	Y
13	Admin and HSP	Menu for exporting history data to PDF functions properly.	Y	Y	Y
14	Lecturer	Menu for managing supervised student data functions properly.	Y	Y	Y

Note: E1, E2, and E3 represent Expert 1, Expert 2, and Expert 3 respectively.

The testing results indicated that all system functionalities operated effectively per the design requirements. All tested features were confirmed to function correctly, including system navigation, login/logout menus, faculty and student data management, and data export to various formats (PDF and Excel). Faculty members' management of student supervision data also proceeded smoothly without issues. These results demonstrate that the system functions technically and has the potential for improvement in user experience (UX), which can be implemented in subsequent development phases to enhance user comfort.

3.8. Algorithm Testing Results

The C4.5 decision tree algorithm was tested by inputting various test data scenarios to ensure that the algorithm produced recommendations based on the parameters of faculty competencies, supervisory workloads, and student final project topics. The testing results showed that the algorithm generated accurate and relevant recommendations, achieving a match rate of 96.7% against the test data, as presented in Table 3.

Table 3. Algorithm testing results

No.	Test Data	Recommendation Result	Suitability		
			5 = Excellent	3 = Fair	1 = Poor
1	Student A, Topic 1	Lecturer Y	5		
2	Student B, Topic 2	Lecturer D	5		
3	Student C, Topic 3	Lecturer R	3		
4	Student D, Topic 4	Lecturer A	5		
5	Student E, Topic 5	Lecturer D	5		
6	Student F, Topic 6	Lecturer V	5		
7	Student G, Topic 7	Lecturer Y	3		
8	Student H, Topic 8	Lecturer R	5		
9	Student I, Topic 9	Lecturer R	5		
10	Student J, Topic 10	Lecturer G	3		
11	Student K, Topic 11	Lecturer A	5		
12	Student L, Topic 12	Lecturer H	1		
13	Student M, Topic 13	Lecturer H	3		
14	Student N, Topic 14	Lecturer G	5		
15	Student O, Topic 15	Lecturer S	3		
Total			**58**		
Persentase			**96.7%**		

The results of the algorithm testing demonstrated that the system could provide consistent and optimal supervisor recommendations. The high level of recommendation accuracy indicated that the algorithm effectively considered each criterion, including faculty competencies and workload distribution. The recommendations showed that the algorithm functioned as expected for this system.

3.9. User Practicality Testing Results

The results of user practicality testing aimed to assess the ease of use, system speed, and user understanding of the supervisor recommendation system based on the C4.5 decision tree algorithm. This testing involved three main user roles: Administrator, HSP, and Faculty (lecturer), who interacted directly with the system. The testing measured five key aspects: ease of navigation, system speed, clarity of recommendations, satisfaction with the accuracy of recommendations, and willingness to use the system again. Details of the testing results can be found in Figure 19. With an average score of 4.59 or 91.73%, as shown in Table 4, the system is categorized as Highly Practical. This category reflects a generally positive user experience, indicating that the system is efficient and easy to use. These results affirm the practicality and effectiveness of the system in supporting the supervisor selection process within academic institutions. The high scores across various aspects, such as ease of navigation and satisfaction with recommendation accuracy, further underscore the system's capability to meet user needs effectively.

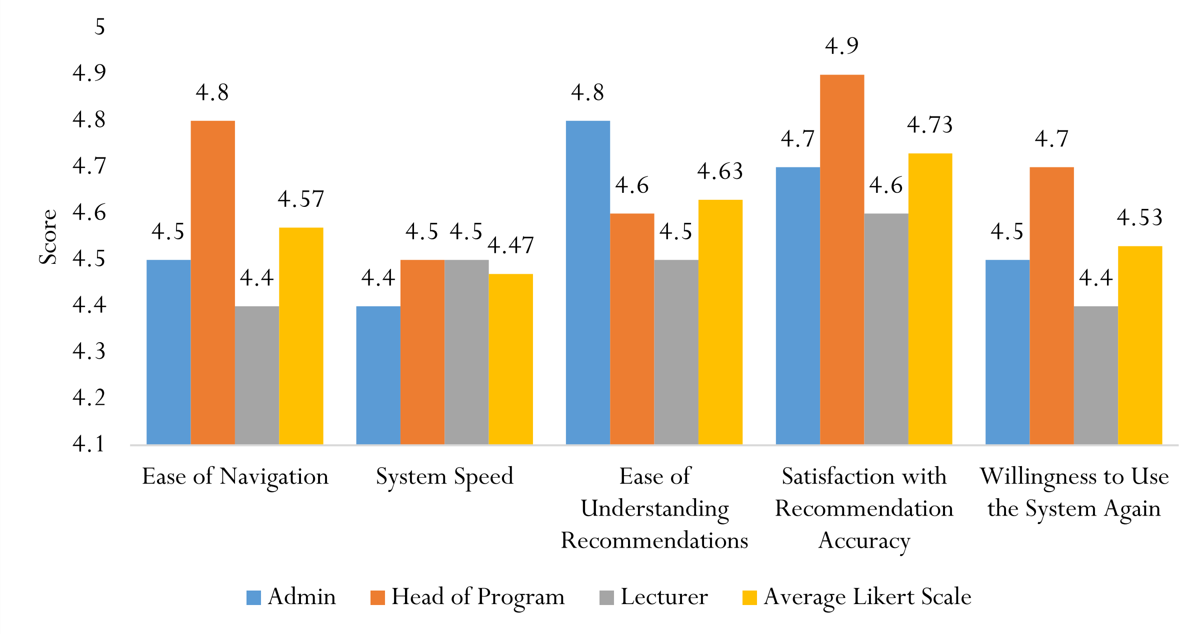


Figure 19. User practicality testing results

Table 4. Average scores and percentages of user practicality testing results

No.	Aspect Measured	Average Likert Scale
1	Ease of Navigation	4.57
2	System Speed	4.47
3	Ease of Understanding Recommendations	4.63
4	Satisfaction with Recommendation Accuracy	4.73
5	Willingness to Use the System Again	4.53
Total		**4.59**
Percentage		**91.73% ≈ 92%**
Category		**Highly Practical**

4. DISCUSSION

The recommendation system for selecting final project supervisors utilizing the C4.5 decision tree algorithm successfully achieved an accuracy rate of 96.7% in generating relevant and appropriate recommendations. The algorithm considered key factors such as faculty competencies, workload distribution, and the alignment between student final project topics and faculty expertise. These results indicate that the C4.5 decision tree algorithm is an effective choice for classification and recommendation tasks in an academic context, particularly in objectively distributing supervisory workloads.

The findings align with previous research by Wang et al. [25], who discovered that the C4.5 decision tree algorithm provides high accuracy in applications involving complex data classification, such as academic recommendation systems. This conclusion is also consistent with studies by Harryanto et al. [12], which applied decision trees in selecting instructors and found that this algorithm could effectively consider various parameters, including competency and availability, with an accuracy exceeding 70%. A significant difference in this study is that the developed system achieved a higher accuracy with optimal recommendation suitability, primarily due to the integration of specific variables in the model, such as workload distribution. Thus, this research contributes to existing literature, providing additional evidence that the C4.5 decision tree algorithm is effective for recommendations that consider multiple important attributes simultaneously.

These findings have significant practical implications for educational institutions, especially in managing final project supervision. This system provides a more objective and efficient recommendation process, reducing reliance on manual selection that often leads to imbalances in faculty workloads. With its intuitive features and responsive interface, the system can be easily integrated into academic and administrative processes to enhance service quality and improve student experiences in academic guidance. These implications align with the results of Ngurah et al. [26], which state that implementing web-based systems in academic environments can enhance operational efficiency and user satisfaction.

The methodology used in this study, particularly the selection of the C4.5 decision tree algorithm, proved effective in meeting the needs of the recommendation system with a high accuracy level. This algorithm operates optimally in scenarios where specific attributes must be considered to generate objective recommendations. Additionally, the web-based architecture allows for a clear separation between the front-end and back-end, providing greater system maintenance and development flexibility. This separation also enables future system enhancements and integration with other academic systems.

Usability testing indicated that the system has a high level of usability, with an average score of 4.59 or 91.87%. Users from various roles (Administrator, HSP, Faculty) found the system intuitive and efficient, with the highest score on satisfaction with recommendation accuracy, rated at 4.7. This indicates that the system meets functional requirements and provides a satisfactory user experience. Suggestions provided by users, such as adding usage guides for specific features, highlight potential improvements that could further enhance user experience.

Despite the high accuracy and success rates observed during testing, some limitations exist. First, the testing was conducted on a limited scale of data. Although the algorithm demonstrated good performance with this data, additional testing with larger and more diverse datasets is necessary to ensure the consistency of algorithm performance under various conditions. Second, the system's parameter settings, such as supervisory loads, still require manual intervention. This can be a limitation, especially if faculty data or student final project topics frequently change. In such cases, the system requires manual updates, affecting efficiency.

For future developments, this research recommends testing the system with larger datasets to evaluate algorithm performance on a broader scale. Additionally, incorporating features such as report previews before export and change confirmations could enhance user experience, as suggested by three experts in UX testing [27]. Further development may consider integration with academic information systems (SIAKAD) for real-time access to student data. Moreover, adding AI components for automated adjustments to recommendation criteria parameters could make the system more adaptive and responsive to changes in faculty data or student final project topics.

5. Conclusion

This study successfully developed a web-based recommendation system for selecting final project supervisors, leveraging the C4.5 decision tree algorithm. The system achieved a high % accuracy rate of

96.7% in generating relevant and appropriate recommendations by evaluating critical factors such as faculty competencies, supervisory workloads, and the alignment between students' final project topics and faculty expertise. Additionally, the system met functional requirements and was well-received by users, achieving an average satisfaction score of 4.59 (91.87%). This reflects its user-friendly design, efficient processing speed, and ability to deliver clear and actionable recommendations.

The application of the C4.5 decision tree algorithm proved highly effective in addressing complex classification challenges in an academic context. The system's key features—including faculty and student data management, configurable recommendation criteria, and detailed reporting of recommendation outcomes—significantly support the Head of Study Program (HSP) in achieving equitable and optimal supervisory load distribution.

This system demonstrates the considerable potential to improve efficiency in academic management by automating the traditionally manual and often imbalanced supervisor selection process. Its web-based architecture ensures scalability and flexibility, enabling seamless integration with existing academic information systems and supporting future enhancements.

Given these promising outcomes, the system holds potential for broader implementation as a sustainable solution for enhancing academic administration. Future research should focus on expanding the system's functionalities, such as incorporating advanced AI-based features to enhance adaptability and performance in dynamic academic environments. Furthermore, testing the system with larger and more diverse datasets will be essential to validate its robustness and ensure consistent performance across various academic institutions.

DECLARATIONS

Author Contributions

Nike Maulani: Conceptualization, Methodology, Software, Data curation, Writing - Original Draft, Writing - Review & Editing. **Geovanne Farell:** Supervision, Validation, Writing - Review & Editing. **Ahmaddul Hadi:** Supervision, Validation. **Dedy Irfan:** Supervision, Validation, Writing - Review & Editing. **Mazen Alzyoud:** Formal analysis, Writing - Review & Editing. **Sharshova Regina Nikolaevna:** Writing - Review & Editing. All authors have read and approved the final version of this manuscript.

Acknowledgments

We thank Universitas Negeri Padang for its support and our international collaborators for their invaluable expertise in system design and testing. Their contributions enriched the quality and global relevance of this study.

Funding

The authors declare that this research received no external funding.

Competing Interests

The authors declare that they have no competing interests.

Generative AI and AI-Assisted Technologies Statement

While preparing this work, the author(s) used Grammarly and Quillbot to improve the manuscript's readability, language, and overall structure. Following the use of this tool, the author(s) thoroughly reviewed and edited the content, ensuring its accuracy and integrity. The author(s) take full responsibility for the content and conclusions presented in the published article.

REFERENCES

[1] K. E. Elfaki and E. M. Ahmed, "Digital technology adoption and globalization innovation implications on Asian Pacific green sustainable economic growth," *J. Open Innov. Technol. Mark. Complex.*, vol. 10, no. 1, p. 100221, Mar. 2024, doi: 10.1016/J.JOITMC.2024.100221.

[2] S. J. Oks *et al.*, "Cyber-physical systems in the context of industry 4.0: A review, categorization, and outlook," *Inf. Syst. Front.*, vol. 1, pp. 1–42, Apr. 2022, doi: 10.1007/S10796-022-10252-X.

[3] M. Abu Talib, A. M. Bettayeb, and R. I. Omer, "Analytical study on the impact of technology in higher education during the age of COVID-19: Systematic literature review," *Educ. Inf. Technol. (Dordr)*, vol. 26, no. 6, pp. 6719–6746, Nov. 2021, doi: 10.1007/S10639-021-10507-1.

[4] R. Chugh, D. Turnbull, M. A. Cowling, R. Vanderburg, and M. A. Vanderburg, "Implementing educational technology in Higher Education Institutions: A review of technologies, stakeholder perceptions, frameworks and metrics," *Educ. Inf. Technol. (Dordr)*, vol. 28, no. 12, pp. 16403–16429, Dec. 2023, doi: 10.1007/S10639-023-11846-X.

[5] H. Oemar, S. Sumarto, A. Komariah, M. M. Ibrahim, F. R. I. L., and M. K. B. M., "Information system design for final project in higher education," *J. Inform. Educ. Manage.*, vol. 6, no. 3, pp. 196–201, 2021. https://doi.org/10.33536/jiem.v6i3.1055.

[6] A. M. Hassan, Usman, Bala, Mannir, and Karfe, "Web-based intelligent tutoring system on students achievement and interest in technical drawing in Niger State," *J. Inf. Syst. Technol. Res.*, vol. 2, no. 1, pp. 13–18, Jan. 2023, doi: 10.55537/JISTR.V2I1.527.

[7] D. Florell, "Web-based training and supervision," *Comput. -Assist. Web-Based Innov. Psychol. Spec. Educ. Health*, pp. 313–337, Jan. 2016, doi: 10.1016/B978-0-12-802075-3.00012-7.

[8] N. Fitria, F. H. Nasution, and A. Aldimas, "Perancangan sistem informasi pengelolaan judul proposal tugas akhir mahasiswa program studi sistem informasi berbasis web," *J. Ilmu Komput. Sist. Inf. (JIKOMSI)*, vol. 7, no. 1, pp. 55–65, Jan. 2024, doi: 10.55338/JIKOMSI.V7I1.2501.

[9] N. Satyahadewi, "Sistem informasi monitoring tugas akhir (SIMTA) berbasis web fakultas MIPA Universitas Tanjungpura," *CESS (J. Comput. Eng. Syst. Sci.)*, vol. 4, no. 1, pp. 83–87, Feb. 2019, doi: 10.24114/CESS.V4I1.11796.

[10] D. Yunda, Syaifudin, and S. Sari, "Sistem informasi tugas akhir berbasis web di Jurusan Teknik Informatika Universitas Trisakti," *Intelmatics*, vol. 2, no. 2, pp. 73–86, Jul. 2022, doi: 10.25105/ITM.V2I2.14311.

[11] A. Pariddudin and F. S. Warsa, "Penerapan algoritma c4.5 untuk rekomendasi mentor santri baru," *TeknoIS: J. Ilm. Teknol. Inf. Sain*, vol. 13, no. 1, pp. 44–49, Feb. 2023, doi: 10.36350/JBS.V13I1.169.

[12] F. F. Harryanto and S. Hansun, "Penerapan algoritma c4.5 untuk memprediksi penerimaan calon pegawai baru di PT WISE," *JATISI (J. Tek. Inform. Sist. Inf.)*, vol. 3, no. 2, pp. 95–103, Mar. 2017, doi: 10.35957/JATISI.V3I2.71.

[13] D. A. Mukhsinin, M. Rafliansyah, S. A. Ibrahim, R. Rahmaddeni, and D. Wulandari,

"Implementasi algoritma decision tree untuk rekomendasi film dan klasifikasi rating pada platform netflix," *MALCOM: Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 570–579, Mar. 2024, doi: 10.57152/MALCOM.V4I2.1255.

[14] J. Shanthi, D. G. N. Rani, and S. Rajaram, "A C4.5 decision tree classifier based floorplanning algorithm for System-on-Chip design," *Microelectron. J.*, vol. 121, p. 105361, Mar. 2022, doi: 10.1016/J.MEJO.2022.105361.

[15] B. Rawal and R. Agarwal, "Improving the accuracy of classification based on c4.5 decision tree algorithm using big data analytics," *Adv. Intell. Syst. Comput.*, vol. 711, pp. 203–211, 2019, doi: 10.1007/978-981-10-8055-5_19.

[16] N. Fajriati and K. Budiman, "Web-based employee attendance system development using the waterfall method," *J. Adv. Inf. Syst. Technol.*, vol. 3, no. 2, pp. 8–20, Sep. 2021, doi: 10.15294/JAIST.V3I2.52942.

[17] A. Saravanos and M. X. Curinga, "Simulating the software development lifecycle: The waterfall model," *Appl. Syst. Innov.*, vol. 6, no. 6, p. 108, Nov. 2023, doi: 10.3390/ASI6060108.

[18] J. Molina-Ríos and N. Pedreira-Souto, "Comparison of development methodologies in web applications," *Inf Softw Technol*, vol. 119, p. 106238, Mar. 2020, doi: 10.1016/J.INFSOF.2019.106238.

[19] T. Thesing, C. Feldmann, and M. Burchardt, "Agile versus waterfall project management: decision model for selecting the appropriate approach to a project," *Procedia Comput. Sci.*, vol. 181, pp. 746–756, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.227.

[20] T. Kaluarachchi and M. Wickramasinghe, "A systematic literature review on automatic website generation," *J. Comput. Lang.*, vol. 75, p. 101202, Jun. 2023, doi: 10.1016/J.COLA.2023.101202.

[21] J. A. Lara, D. Lizcano, M. A. Martínez, and J. Pazos, "Developing front-end Web 2.0 technologies to access services, content, and things in the future Internet," *Future Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1184–1195, Jul. 2013, doi: 10.1016/J.FUTURE.2013.01.006.

[22] Z. Lei, H. Zhou, W. Hu, and G. P. Liu, "Toward an international platform: A web-based multi-language system for remote and virtual laboratories using react framework," *Heliyon*, vol. 8, no. 10, p. e10780, Oct. 2022, doi: 10.1016/J.HELIYON.2022.E10780.

[23] S. Supriyono, "Software testing with the approach of blackbox testing on the academic information system," *IJISTECH (Int. J. Inf. Syst. Technol.)*, vol. 3, no. 2, pp. 227–233, May 2020, doi: 10.30645/IJISTECH.V3I2.54.

[24] M. Last, S. Eyal, and A. Kandel, "Effective black-box testing with genetic algorithms," *Lect. Notes Comput. Sci. (LNCS)*, vol. 3875, pp. 134–148, 2006, doi: 10.1007/11678779_10.

[25] H. Bin Wang and Y. J. Gao, "Research on C4.5 algorithm improvement strategy based on MapReduce," *Procedia Comput. Sci.*, vol. 183, pp. 160–165, Jan. 2021, doi: 10.1016/J.PROCS.2021.02.045.

[26] G. Ngurah *et al.*, "Kualitas dan kepuasan pengguna terhadap sistem informasi akademik berbasis web," *J. Ilm. Akunt. Bisnis*, vol. 11, no. 2, pp. 84–90, Mar. 2017, doi: 10.24843/JIAB.2016.V11.I02.P03.

[27] Y. Lu, C. Zhang, I. Zhang, and T. J. J. Li, "Bridging the gap between UX practitioners' work practices and AI-enabled design support tools," *Conf. Hum. Factors Comput. Syst. - Proc.*, Apr. 2022, doi: 10.1145/3491101.3519809.

AUTHOR BIOGRAPHIES

Nike Maulani is a faculty member in the Faculty of Engineering, Department of Informatics Engineering Education at Universitas Negeri Padang. She specializes in software engineering, information systems, and informatics education. Her research focuses on developing innovative web-based applications and website technologies to enhance learning experiences and improve educational practices in information technology.

Geovanne Farell is a senior lecturer and practitioner at Universitas Negeri Padang, with expertise in Information Technology, Software Engineering, and Machine Learning. His work spans research and practice in vocational education and TVET (Technical and Vocational Education and Training), where he leverages technology to drive innovation in educational practices.

Ahmaddul Hadi is a senior lecturer at Universitas Negeri Padang, specializing in Informatics, Information Technology, and vocational applications of technology in education. His research spans information systems, online examination frameworks, and expert systems.

Dedy Irfan is a senior lecturer at Universitas Negeri Padang, specializing in vocational education and technology. His research focuses on innovative learning models, such as blended learning and digital instructional tools, and spans diverse applications from educational software to online queuing systems in healthcare. His work is highly cited, reflecting his influence in academia and applied vocational education.

Mazen Alzyoud is an assistant professor in the computer science department at Al al-Bayt University in Jordan. Alzyoud received His B.Sc. and M.Sc. from Al al-Bayt University in computer science and his Ph.D. from Kent State University, USA in computer science. His main interests include but are not limited to machine learning applications in medical diagnosis, machine learning applications in education, data mining, and data science.

Sharshova Regina Nikolaevna is a senior lecturer in the Language Department at International Information Technology University (Almaty, Kazakhstan). She has published in the field of computer linguistics, ICT, and AI tools in education. Her current Ph.D. research is focused on multidisciplinary studies using digital technologies and AI tools in philological education and linguistics.

Publisher's and Journal's Note *Sagamedia Teknologi Nusantara, as the publisher and editor of the Journal of Hypermedia & Technology-Enhanced Learning (J-HyTEL), upholds the highest ethical standards in academic publishing. The journal remains neutral with regard to jurisdictional claims in published maps and institutional affiliations. Authors are fully responsible for the originality, accuracy, and integrity of their work. Post-publication ethical concerns will be addressed through corrections, clarifications, or retractions as necessary. The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States.* https://www.un.org/sustainabledevelopment