Design and Development of a Web-Based Laravel Application for Library Automation at SMP Muhammadiyah 2 Kartasura

Mochamad Raihan Mukhorobun[1], Ihsan Cahyo Utomo[2]
[1,2]Program Studi Teknik Informatika
Universitas Muhammadiyah Surakarta
[1]l200220028@student.ums.ac.id,[2]icu886@ums.ac.id

Submitted: February 11, 2026 ; accepted: October 11, 2026

Abstract

The library at SMP Muhammadiyah 2 Kartasura currently relies on inefficient manual management that is prone to data errors. This research aims to develop a web-based library information system using the Laravel framework and MySQL database, employing the Agile development method. The system is designed to automate the management of book data and lending transactions, featuring automatic reminder notifications via WhatsApp API to prevent book loss. Black Box testing on 13 main scenarios demonstrated a 100% success rate in functionality, while an evaluation using the System Usability Scale (SUS) with 36 respondents revealed a positive perception of the system's usability. These results prove that the digital transformation effectively enhances operational efficiency, simplifies librarian tasks, and provides an accurate, user-friendly lending management solution.

Keywords*: Library, Agile, Laravel, WhatsApp API, Automation*

1. Introduction

Libraries are vital facilities that provide a variety of information services to support the learning process. In general, libraries are divided into public libraries that serve the broad community, and school libraries that focus on the needs of students, teachers, and staff within the educational environment. The existence of libraries plays a crucial role in achieving national educational goals, which is to produce intelligent and highly knowledgeable human resources [1]. Since the past, libraries have been known as bridges of knowledge. However, the rapid advancement of information technology in the digital era presents both challenges and new opportunities. This development changes individual behavior in obtaining and sharing information, thus libraries are required to adapt through digital transformation [2].

Even though digital transformation is becoming increasingly urgent, the operational system of the library at SMP Muhammadiyah 2 Kartasura is currently still conducted manually. Administrative management and lending rely entirely on physical recording and conventional shelf arrangement [3]. This manual process impacts the speed of information access and leads to a high risk of data errors (human error). Real constraints are evident when staff face difficulties in managing book data, monitoring collection availability, and tracking students who are late in returning books. This condition has the potential to cause the loss of library book assets, resulting in a decrease in the number of available collections [4].

In today's digital era, the existence of an informative and easily accessible online library system has become a crucial necessity [5]. Given that SMP Muhammadiyah 2 Kartasura has not yet implemented digitalization, the development of a web-based library application is the right solution to improve service efficiency [6]. In this development, Laravel was chosen as the primary framework. Laravel is known as one of the best PHP frameworks that offers efficient architecture, ease of authentication, simple routing management, and a robust security structure [7], [8]. These advantages are expected to accelerate the development process while simplifying the complexity of the application code.

Although numerous studies regarding library information systems have been conducted, several aspects distinguish this research from previous references. Earlier studies generally focused on catalog digitalization and basic circulation management without an active reminder mechanism for users. The primary advantage of the "SiMuda" system lies in the integration of an automated notification feature via the WhatsApp API, which operates in real-time to send due date reminders to students. This differs from the use of email, which tends to have a lower readability rate. The implementation of the WhatsApp API for notifications is crucial because many individuals currently prefer WhatsApp as a primary social media platform for communication via text or voice messages, both individually and in groups [9]. Furthermore, the application of the Agile method in this research allows for feature development that is more adaptive to the specific needs of SMP Muhammadiyah 2 Kartasura.

This study aims to design and develop a web-based library application named "SiMuda" using the Laravel framework to overcome the limitations of the current manual system. Through this system, the processes of managing book data, membership, and borrowing/returning circulation are expected to become more accurate, rapid, and efficient. Practically, this application contributes to improving the quality of school services, facilitating administrative tasks for librarians, and expanding access to information for students and teachers. Furthermore, this research provides a theoretical contribution as a reference for the development of digital library systems in the future.

2. Research Method

This chapter outlines the systematic framework used in conducting the research to achieve the formulated objectives. The explanation includes the type and object of the research, the timeline of implementation, and relevant data collection techniques. In addition, this chapter details the applied software development workflow, ranging from requirements analysis, system design, and code implementation to testing and evaluation, to ensure that the developed library information system functions optimally and meets the user requirements at SMP Muhammadiyah 2 Kartasura.

2.1 Type, Object, and Time of Research

This research is classified as applied research focusing on software development. The object of the research is the library information system at SMP Muhammadiyah 2 Kartasura. This research was conducted from October 2025 to January 2026.

2.2 Data Collection

Data collection techniques were carried out through two primary methods:

1. Direct observation of the manual library circulation workflow at SMP Muhammadiyah 2 Kartasura to understand operational constraints.
2. Interviews conducted with librarians and school staff to map the system's functional requirements, borrowing procedures, and applicable fine regulations [10].

2.3 System Development Methods

The system development applies the Agile Software Development method, which focuses on short development cycles (sprints) and continuous improvement [11]. As shown in Figure 1, the research workflow consists of six iterative stages: planning, design, development, testing, review, and deployment.

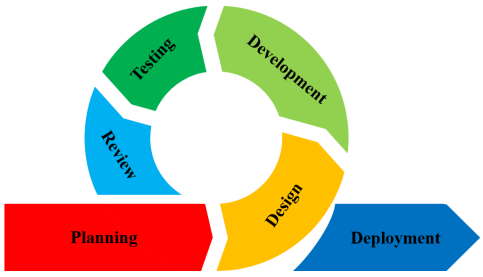


Figure 1. Agile method flow

2.3.1 *Planning*

This stage focuses on the preparation of strategic plans and system development steps to transform library management from manual to digital [12].

1. Needs Analysis

Based on the interview results, the system requirement specifications are summarized in Table 1 to ensure that the developed features are relevant to user needs.

Table 1. Needs Analysis

Functional Requirements Analysis	Non-Functional Requirements Analysis
• Librarians can manage book data. • Librarians can record book borrowing and returns. • Librarians can manage member data. • The system can automatically calculate late returns. • The system can send overdue notifications via WhatsApp.	• The system must be compatible with various modern web browsers and feature a responsive interface design. • The system must possess robust security with authentication (login) and role-based access control (RBAC) using Laravel Breeze equipped with middleware. • The system must demonstrate high-speed

• Users can view books and submit book loan requests.	performance when processing borrowing transactions. • Integration of the WhatsApp API for sending overdue warning notifications and loan approval messages. • The system interface is designed to be simple and user-friendly. • API Security: Communication between the application server and the WhatsApp service provider (Fonnte) is secured using a confidential API Token to prevent unauthorized access by third parties.

2. Development Plan (Sprint)

The development process is divided into several sprints to ensure a structured approach, as detailed in Table 2.

Table 2. Sprint

Sprint	Work
1	Perform Laravel and database setup.
2	CRUD for book management and member management.
3	Development of book borrowing and return features.
4	WhatsApp API integration for notifications.
5	System testing and deployment.

2.3.2 Design (Planning)

System design includes the database architecture design, system logic, and user interface [13]. Modeling is performed using the Unified Modeling Language (UML).

1. Use Case Diagram

A Use case diagram is a diagram that describes the relationship between actors and the system, where this diagram illustrates the activities performed by the actors [14]. Figure 2 visualizes the interaction between the actors and the system. There are two main actors: the Admin (Librarian), who has full access rights for data management, and the User (Student), who has access to searching and submitting loan requests.



Figure 2. Use Case Diagram

2. Entity-Relationship Diagram (ERD)

The database structure is designed by centering the loan transaction table as the link between the student and book entities (Figure 3). A One-to-Many relationship is implemented so that the system can record detailed transaction history and accommodate the automatic overdue notification feature [15].

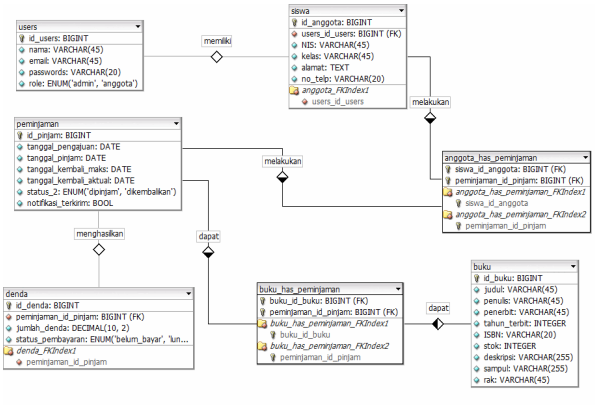


Figure 3. Entity Relationship Diagram

3. Activity Diagram

The system workflow is mapped in Figure 4. This diagram details the sequence of activities starting from the login process, loan validation by the admin, to the book search mechanism by the user.

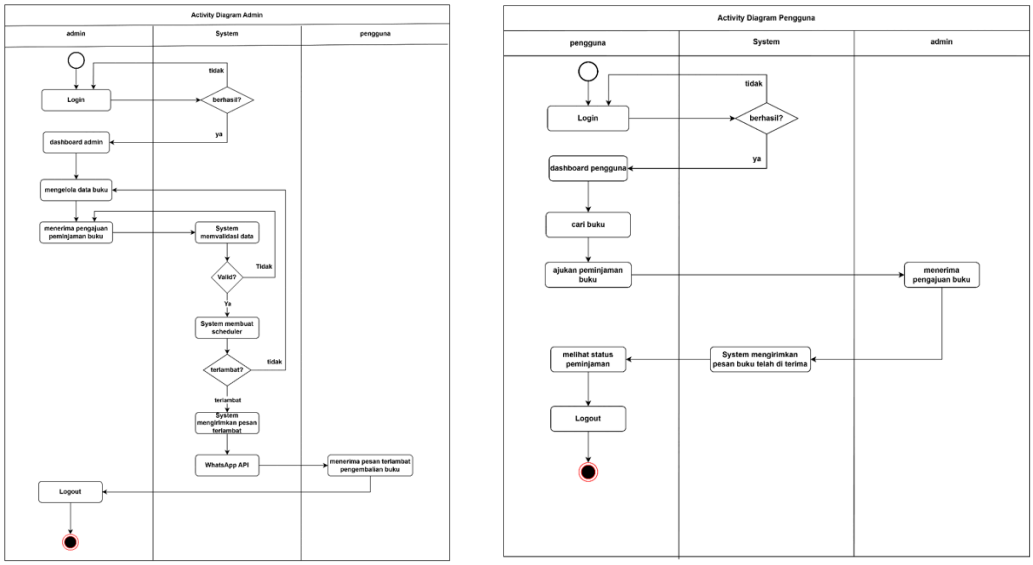
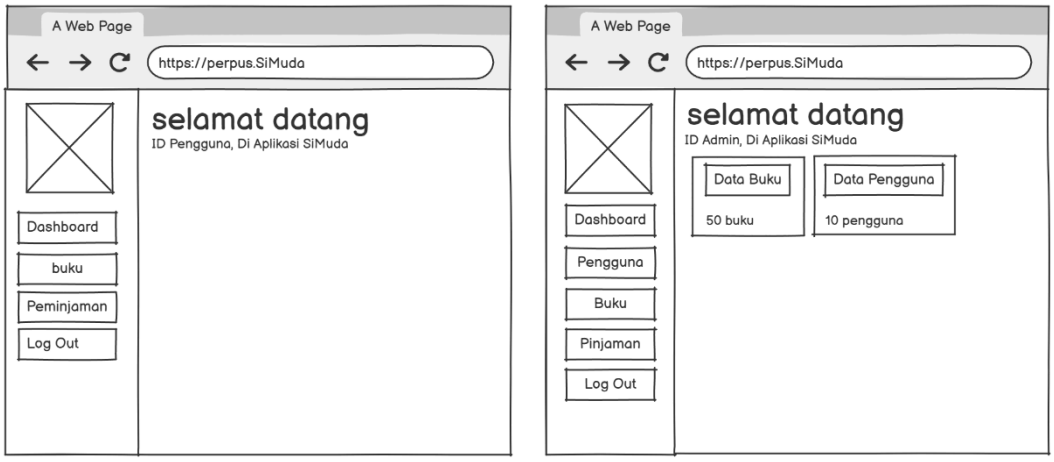


Figure 4. Activity Diagram

4. Wireframe

The interface framework is designed as the application blueprint (Figure 5). The design includes a user dashboard for accessing primary features, an admin dashboard for data management, and a detailed book view.



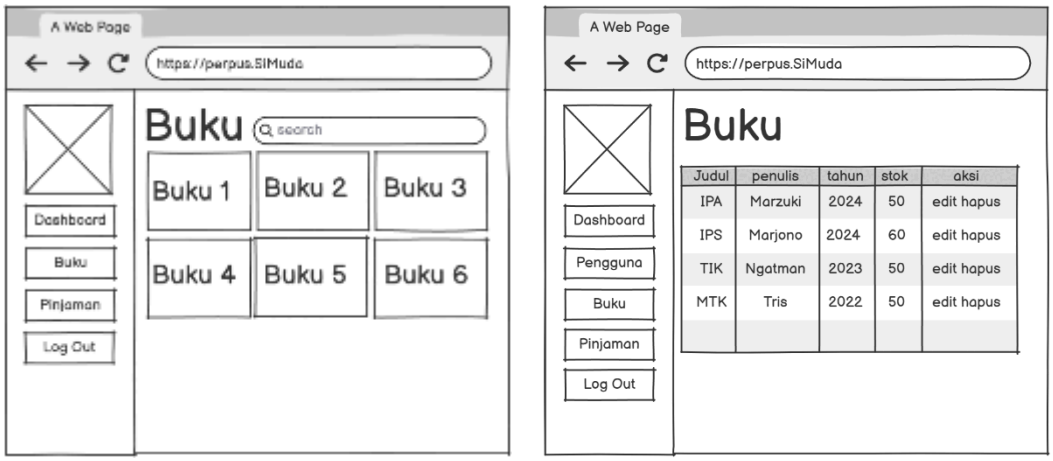


Figure 5. Wireframe

2.3.3 Development

Program code implementation is carried out based on the prepared design. The system is built using the Laravel framework with a MySQL database [16]. Specific features, such as overdue notifications, are integrated using the WhatsApp API. This stage implements an automatic notification system using WhatsApp via Fonnte [17]. Technically, the message delivery mechanism is managed through a Laravel Console Command named SendOverdueNotification, designed to execute tasks on a scheduled basis. The message delivery mechanism utilizes the HTTP POST method to the official Fonnte endpoint by sending a JSON format payload containing parameters for the destination number (target), message content (message), and country code (countryCode). To maintain reliability, the system implements error handling using try-catch blocks; if an API response failure occurs, the system records the error message in the logs for technical audit purposes. Furthermore, to avoid number blocking due to rapid bulk sending (rate limiting), the system is configured to provide a random time delay between 4 to 7 seconds in each delivery iteration using the sleep() function. Development is conducted in stages according to the sprint sequence.

2.3.4 Testing

System verification is carried out through two approaches to ensure software quality:

1. Black Box Testing
   Functional testing is performed without examining the internal code to ensure that every input-output feature operates according to the specifications [18].
2. System Usability Scale (SUS)
   Usability evaluation is conducted by distributing questionnaires to respondents to measure the level of satisfaction and the ease of system use [19].

2.3.5 Review

A comprehensive evaluation is conducted with users after the testing phase. Feedback from librarians and students is used as a basis for improving features, security, and the interface before the final launch. If bugs or workflow inconsistencies are found, improvements are immediately implemented during this cycle.

2.3.6 Deployment

The final stage includes cloud server configuration, database installation, and source code uploading. Once the system is online, production environment trials (including real-time WhatsApp notification tests) are conducted, along with periodic monitoring to maintain system performance stability.

3. Results and Discussion

This chapter presents the implementation results of the developed web-based library management system, "SiMuda," as well as the results of the system's functional and usability testing. The system is designed to be responsive and accessible via various devices, including laptops and mobile devices.

3.1 System Implementation

The system implements a multi-layered security protocol to ensure the privacy and protection of students' personal data. This security is realized through the implementation of Role-Based Access Control (RBAC) and password encryption using the Bcrypt algorithm to maintain the confidentiality of account information. The use of this encryption technique is crucial in a digital library ecosystem to protect various

sensitive information—ranging from identity data to borrowing history—against the risk of unauthorized access. With these protection mechanisms in place, the integrity of user data is maintained while minimizing the potential for information misuse by unauthorized external parties [20]. The security of students' WhatsApp numbers is safeguarded through a restricted-use policy solely for circulation notifications, where messages are sent ethically through routine scheduling to avoid disruption (spam). Furthermore, the WhatsApp API integration utilizes authentication tokens stored securely within the system's environment variables to minimize the risk of unauthorized access by third parties.

3.1.1 Login Page

The login page serves as an authentication gateway to verify user identity before accessing the system. As shown in Figure 6, this page provides two primary input fields: username and password. The system implements strict data validation; if the credentials match the database, the user will be directed to the dashboard according to their assigned role. This validation aims to maintain data security and prevent unauthorized access.

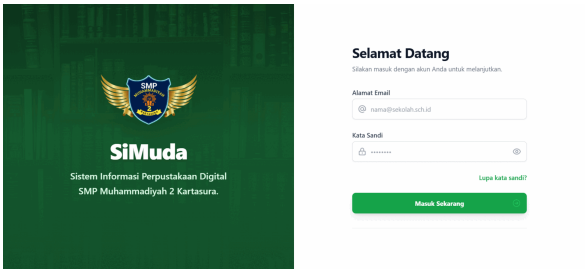


Figure 6. Login Page

3.1.2 Dashboard Page

The dashboard page is an information center tailored to the user's role (Figure 7).

1. Admin Dashboard: Presents a summary of operational statistics, including total book data, student data, the number of books currently on loan, and a real-time list of overdue returns.
2. Student Dashboard: Displays information on the latest book collections and a "My History" feature that contains the student's personal borrowing history. This design aims to facilitate navigation and provide a quick overview of the library status.

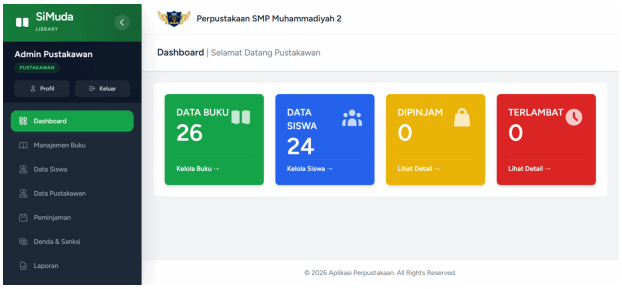


Figure 7. Dashboard Page

3.1.3 Book Management Page

This page is specifically dedicated to the admin for managing the library collection (Figure 8). Available features include Create (add book), Read (book list), Update (edit data), and Delete (remove data). When the admin adds a new book, the system stores the book's metadata details into the database. The book list table is equipped with interactive action buttons to facilitate the librarian in performing inventory data updates.

.

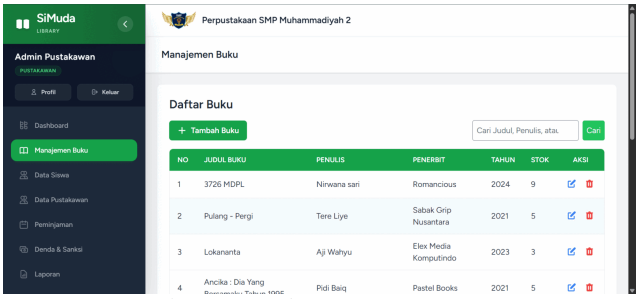


Figure 8. Book Management Page

3.1.4 Book Catalog Page

The catalog page (Figure 9) is the primary interface for students to browse the library collection. Books are displayed in a card format containing information such as the cover, title, author, description, and remaining stock. A search feature is provided to expedite book discovery. The borrowing process is designed to be simple: students only need to click the "Borrow" button, and the system will validate the student's personal data completeness before sending the loan request to the admin for verification.

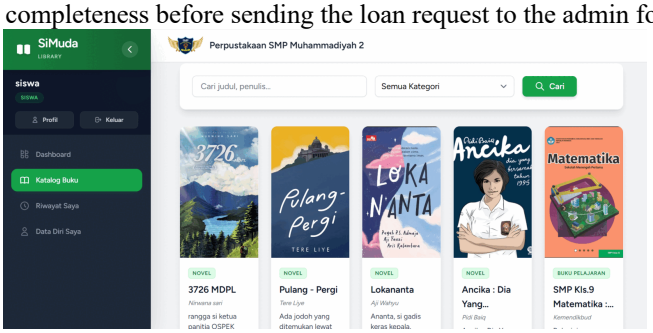


Figure 9. Book Catalog Page

3.1.5 Borrowing Page

This page is used by the admin to manage circulation (Figure 10). The admin can verify student loan requests by reviewing the details of the applicant and the requested book. In addition to approvals, this page contains a list of ongoing book loans. The return feature allows the admin to record returned books, and the system will automatically calculate any delays along with the amount of the fine to be paid.

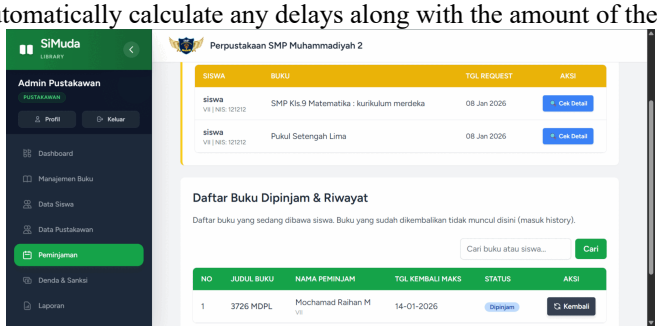


Figure 10. Borrowing Page

3.1.6 Fines and Sanctions Page

The system automatically calculates late returns. The Fines Page (Figure 11) displays a list of students with overdue books along with the accumulated fines that must be paid. This feature assists the admin in monitoring student discipline. There is a "Pay" action button used to update the payment status to "Paid" once the student has fulfilled their obligation.



Figure 11. Fines and Sanctions Page

3.1.7 Report Page

For administrative and archival purposes, the system provides a report page (Figure 12) that records all borrowing transactions. The report table includes details such as the student's name, class, book title, borrow date, return date, and status. The system is also equipped with a report export feature to PDF format, allowing the admin to download monthly borrowing recapitulations for school reporting purposes.



Figure 12. Report Page

3.1.8 WhatsApp Notification Automation Mechanism

The implementation results demonstrate that the system is capable of accurately processing overdue student data. Delivery is executed by performing a query on the borrowing table to identify records with a "borrowed" status that have exceeded the maximum return date (tgl_kembali_maksimal).

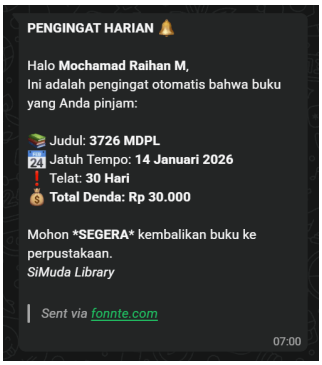


Figure 13. Message Template Example

3.2 System Testing

The testing phase is conducted to ensure the system operates according to the requirement specifications and provides a good user experience.

3.2.1 Black Box Testing

Black Box Testing focuses on validating the system's input and output functionality without examining the internal code. Based on Table 3, all testing scenarios ranging from login, data management, and borrowing transactions to WhatsApp notification integration and reporting showed a "Success" status. No logic errors or functional failures were found, thus the system is declared functionally feasible.

Table 3. Black Box Testing

No.	Feature Name	Test Scenario	Test Data	Expected Result	Status
1.	User Login	Admin enters a valid username and password	Email: admin@gmail.com, Password: admin111	Admin successfully logs in and is redirected to the admin dashboard	Success

2.	User Login	Admin enters an invalid username or password	Email: admin@gmail.com, Password: pustakawan	Admin fails to log in and is redirected back to the login page	Success
3.	Book Data Management	Admin adds, edits, or deletes book data	Title: Matematika SMP, Category: Textbook, Author: Yadi Mulyadi, Publisher: Yrama Widya, Year: 2025, Stock: 120, ISBN: 9786232056947, Description: Math textbook, Rack: 2-b, Cover: mtk.png	Data is successfully added, edited, or deleted	Success
4.	Student Data Management	Admin views, edits, or deletes student data	Name: Siswa, Email: siswa@gmail.com, NIS: 200022, Class: IX, WhatsApp: 08912345678, Address: Makamhaji	Data is successfully viewed, edited, or deleted	Success
5.	Librarian Data Management	Admin adds or deletes librarian data	- Full Name: Admin Perpus 1, Email: admin2@gmail.com, Password: admin222, Confirm Password: admin222	Data is successfully added or deleted	Success
6.	Loan Data Management	Admin receives requests, approves loans, or processes book returns	Student Data: siswa (Class IX, WhatsApp: 08912345678); Book Data: Matematika SMP (Stock: 99)	Data is received, approved, WhatsApp message is sent, and book is returned	Success
7.	Send Overdue Notification	System automatically sends an overdue return message	Student Data: Valid, Book Data: Valid, Loan Status: Overdue	Message is successfully sent	Success
8.	Send Overdue Notification	System attempts to send overdue message for non-overdue status	Student Data: Valid, Book Data: Valid, Loan Status: Borrowed (Ongoing)	Message is not sent; status remains "borrowed"	Success
9.	Fines and Sanctions Management	Admin views overdue data and approves fine payments	Name: siswa, Class: IX, Title: Matematika SMP, Due Date: Jan 21, 2026, Total Fine: Rp. 1,000, Status: Unpaid	Admin successfully views overdue records and approves fine payments	Success
10.	Print Report	Admin prints monthly borrowing reports	Month: December, Year: 2025	Admin successfully prints the monthly borrowing report	Success
11.	Admin Logout	Admin logs out of the system	- -	Successfully exits the system and returns to the login page	Success
12.	Student Login	Student enters a valid username and password	Email: siswa@gmail.com, Password: siswa111	Student successfully logs in and is redirected to the student dashboard	Success
13.	Student Login	Student enters an unregistered username or password	Email: Mochraihan@gmail.com, Password: raihan123	Student fails to log in and is redirected back to the login page	Success
14.	Book Loan Request	Student submits a loan request	Student Data: siswa (Class IX, WhatsApp: 08912345678); Book Data: Matematika SMP (Stock: 99)	Request is successfully sent	Success
15.	Book Loan Request	Student submits a request with incomplete personal profile	Student Data: Name/Class/WhatsApp (Empty); Book Data: Matematika SMP	Request fails to send; student is redirected to the personal profile completion page	Success

16.	Fill Personal Profile	Student fills in personal profile data	Email: siswa@gmail.com, NIS: 200022, Class: IX, WhatsApp: 08912345678, Address: Makamhaji	Student successfully adds/updates personal profile data	Success
17.	Student Logout	Student logs out of the system	-	Successfully exits the system	Success

3.2.2 System Usability Scale (SUS)

The usability evaluation was conducted using the System Usability Scale (SUS) method involving 36 respondents (librarians and students). Respondents were asked to provide assessments for 10 statements based on a 1-5 Likert scale. The final SUS score is not calculated by directly summing the raw values, but through the following standardization procedure:

- Odd Statements (Positive): For each statement numbered 1, 3, 5, 7, and 9, the calculated score is $(Score - 1).
- Even Statements (Negative): For each statement numbered 2, 4, 6, 8, and 10, the calculated score is $(5 - Score).
- Score Accumulation: The total score from the 10 statements is then multiplied by 2.5 to obtain a final value within the range of 0-100.

The recapitulation of the questionnaire results is presented in Table 4.

Table 4. Result From SUS

No.	Name	Question										Total	Total x 2,5
		1	2	3	4	5	6	7	8	9	10		
1.	Responden 1	4	4	4	4	4	4	4	4	4	4	40	100
2.	Responden 2	3	3	3	2	3	3	3	3	3	2	28	70
3.	Responden 3	3	3	3	3	3	2	3	3	2	1	26	65
4.	Responden 4	4	4	4	4	4	4	3	3	3	4	37	93
5.	Responden 5	4	4	3	4	3	4	4	4	4	4	38	95
6.	Responden 6	4	4	4	4	3	2	4	4	4	4	37	93
7.	Responden 7	4	3	4	3	4	3	4	3	4	3	35	88
8.	Responden 8	1	3	3	4	3	4	4	0	4	4	30	75
9.	Responden 9	3	3	4	4	3	1	4	3	4	4	33	83
10.	Responden 10	3	3	4	1	3	3	3	4	3	1	28	70
11.	Responden 11	4	3	4	3	4	4	3	4	4	3	36	90
12.	Responden 12	4	3	4	4	3	4	4	4	4	3	37	93
13.	Responden 13	4	3	4	2	3	3	3	3	4	3	32	80
14.	Responden 14	4	4	4	4	4	4	4	4	4	4	40	100
15.	Responden 15	3	3	4	3	3	4	2	4	4	3	33	83
16.	Responden 16	3	4	4	4	4	4	4	4	4	4	39	98
17.	Responden 17	3	4	3	3	2	4	4	3	3	2	31	78
18.	Responden 18	4	4	4	3	4	4	4	4	4	2	37	93
19.	Responden 19	4	3	3	3	3	4	3	3	4	3	33	83
20.	Responden 20	3	4	4	3	2	4	3	4	0	3	30	75
21.	Responden 21	4	4	4	3	3	3	4	4	3	3	35	88
22.	Responden 22	4	4	4	4	4	4	4	4	4	2	38	95
23.	Responden 23	3	3	3	3	3	3	3	3	3	3	30	75
24.	Responden 24	4	4	4	4	4	4	4	4	4	4	40	100
25.	Responden 25	4	4	4	4	4	4	4	4	4	4	40	100
26	Responden 26	3	3	4	4	4	3	3	3	3	3	33	83
27.	Responden 27	3	2	2	2	3	3	2	3	2	2	24	60
28.	Responden 28	4	4	4	4	4	4	4	4	4	4	40	100
29.	Responden 29	3	3	4	3	4	3	3	3	3	3	32	80
30.	Responden 30	3	4	4	4	4	4	4	4	4	4	39	98
31.	Responden 31	3	3	4	4	4	4	3	4	3	3	35	88
32.	Responden 32	3	4	4	3	4	3	4	3	3	3	34	85
33.	Responden 33	3	0	4	1	2	2	1	2	3	0	18	45
34.	Responden 34	3	3	3	2	3	1	3	3	2	1	24	60
35.	Responden 35	3	2	3	3	3	3	2	2	3	3	27	68
36.	Responden 36	2	1	2	3	3	2	2	2	2	1	20	50
	Final Result												**83**

The interpretation of the value "0" in the respondent recapitulation table is a result of the raw score transformation process in accordance with the standard System Usability Scale calculation rules. It should be emphasized that respondents provided complete answers on the 1-5 Likert scale; however, these values

were converted to 0 after undergoing the standardization procedure. Based on the calculations presented in Table 4, the "SiMuda" system obtained an average SUS score of 83. The visualization of this final score is presented in Figure 13.

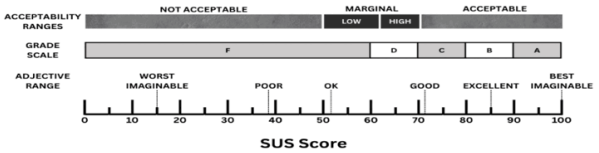


Figure 13. SUS Score Acquisition Chart

The score of 83 was calculated using the standard SUS formula, where the respondent's total score is multiplied by a correction factor of 2.5. This figure places the system in the "Acceptable" category within the acceptability range. On the grade scale, a score of 83 falls into the "B" grade category. Meanwhile, qualitatively based on adjective ratings, this score is categorized as "Excellent." This indicates that the "SiMuda" system is perceived as easy to use, possesses a user-friendly interface, and functions optimally in supporting library activities.

4. Conclusion

Based on the results of the research and development conducted, the web-based library information system "SiMuda" has been successfully designed and implemented at SMP Muhammadiyah 2 Kartasura with optimal functionality. All primary features including user authentication, book data management, borrowing and return circulation, automatic fine calculation, and WhatsApp notifications—have proven to function according to the requirement specifications. The system's reliability is reinforced by Black Box testing results, which show a valid status across all test scenarios with no functional errors detected. Furthermore, the usability evaluation using the System Usability Scale (SUS) yielded an average score of 83, placing the system in the "Acceptable" category with an "Excellent" rating, indicating that the application is easy to use and well-received by both librarians and students. As an effort toward future refinement, it is suggested that system development integrate an ISBN QR Code scanning feature to expedite the book data entry process, as well as add a digital library (e-book) module to make access to literacy collections more flexible, extensive, and relevant to information technology advancements.

Bibliography

[1] A. Yudhistira, "Perancangan dan Implementasi Sistem Informasi Perpustakaan Berbasis Web," *JSK (Jurnal Sist. Inf. dan Komputerisasi Akuntansi)*, vol. 7, no. 1, pp. 14–20, 2023, doi: 10.56291/jsk.v7i1.95.

[2] I. Ilhami, M. Satrio Rofi Fadil, R. Ferina, and S. Aulia Assabilla, "Analisis Tantangan dan Peluang yang dihadapi Perpustakaan di Era Digital," *Blantika Multidiscip. J.*, vol. 3, no. 2, pp. 211–219, 2024, doi: 10.57096/blantika.v3i2.284.

[3] A. Junaedi, D. Drajat, R. I. Syihabuddin, U. M. Damayanti, and M. F. Wahyutama, "Perancangan Perpustakaan Digital Berbasis Website Pada SMAN 18 Kabupaten Tangerang," *ADI Bisnis Digit. Interdisiplin J.*, vol. 2, no. 2, pp. 20–26, 2021, doi: 10.34306/abdi.v2i2.550.

[4] J. R. Permana, M. Muhaqiqin, and A. S. Puspaningrum, "Implementasi Metodologi Web Development Life Cycle Untuk Membangun Sistem Perpustakaan Berbasis Web (Studi Kasus: Man 1 Lampung Tengah)," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 2, no. 4, pp. 435–446, 2021, doi: 10.33365/jatika.v2i4.1407.

[5] J. Nugraha, M. D. Ageng Sudarna, and D. Moeis, "Sistem Informasi Profil Perusahaan Berbasis Website Menggunakan Laravel 8," *J. Rekayasa Sist. Inf. dan Teknol.*, vol. 2, no. 1, pp. 554–567, 2024, doi: 10.70248/jrsit.v2i1.852.

[6] T. N. Putri and A. Miftahuddin, "Pengaruh Media Sosial Terhadap Kesehatan Mental Manusia di Era Digital Pengaruh Media Sosial Terhadap Kesehatan Mental Manusia di Era Digital," no. December, 2024.

[7] M. Amini *et al.*, "Mahamgostar.Com As a Case Study for Adoption of Laravel Framework As the Best Programming Tools for Php Based Web Development for Small and Medium Enterprises," *J. Innov. Knowl.*, no. May, pp. 100–110, 2021, [Online]. Available: https://www.mahamgostar.com/

[8] A. Herdiansah, R. I. Borman, and S. Maylinda, "Sistem Informasi Monitoring dan Reporting Quality Control Proses Laminating Berbasis Web Framework Laravel," *J. Tekno Kompak*, vol. 15, no. 2, p. 13, 2021, doi: 10.33365/jtk.v15i2.1091.

[9] I. S. Nue *et al.*, "Implementasi API WhatsApp Pada Rancang Bangun Sistem Informasi Penjualan

Berbasis Web," *INCREATE*, pp. 21–32, 2025, [Online]. Available: https://jurnal.nusanipa.ac.id/index.php/increate/article/view/418

[10] D. Gunawan *et al.*, "Implementasi MERN Stack pada Pengembangan Sistem Penerimaan Peserta Didik Baru," *Swabumi*, vol. 11, no. 2, pp. 102–110, 2023, doi: 10.31294/swabumi.v11i2.15965.

[11] K. O. Simatupang and A. F. Pakpahan, "Metode Agile Dalam Perancangan Sistem Informasi Reservasi Fasilitas Universitas Advent Indonesia," *J. Inf. Syst. Res.*, vol. 3, no. 4, pp. 608–617, 2022, doi: 10.47065/josh.v3i4.1816.

[12] R. T. Aldisa and A. I. Kushartanto, "The Agile Development Methodology with Data Search to Design a Website-Based Mail Management Information System," *Int. J. Inf. Syst. Technol. Akreditasi*, vol. 6, no. 158, pp. 512–516, 2022.

[13] M. Ikhsan, R. A. Praptiwi, B. M. Jamaluddin, H. Salsabilah, A. M. Akbar, and K. Khaidir, "Sistem Informasi Fotografi Berbasis Website Menggunakan Metode Agile," *J. Secur. Comput. Information, Embed. Network, Intell. Syst.*, vol. 8329, pp. 62–74, 2024, doi: 10.61220/scientist.v2i2.20242.

[14] S. Rokhmah, C. Utomo, and I. Muslihah, "Web Based Distribution of Zakat , Infaq , and shodaqoh (Case Study Of Surakarta City Region)," *Int. J. Comput. Inf. Syst.*, vol. 01, no. 01, 2020, [Online]. Available: https://www.neliti.com/publications/341468/web-based-distribution-of-zakat-infaq-and-shodaqoh-case-study-of-surakarta-city

[15] D. Gunawan and V. N. Andika, "Implementasi Teorema Bayes Pada Sistem Informasi Posyandu Dalam Mendeteksi Stunting Pada Balita," *J. Sist. Komput. dan Inform.*, vol. 4, pp. 692–698, 2023, doi: 10.30865/json.v4i4.6146.

[16] R. R. A. F. Mudhoffar and W. Widayat, "Perancangan Sistem Informasi Rekam Medis Berbasis Web," *Indones. J. Comput. Sci.*, vol. 13, no. 2, pp. 3169–3181, 2024, doi: 10.33022/ijcs.v13i2.3822.

[17] A. R. Manik, M. B. Akbar, and J. Purba, "Academic Portal with MFA (WhatsApp OTP via Fonnte), Role-Based Access Control , and Logging System for Network Monitoring," vol. 4, no. 3, pp. 674–687, 2025.

[18] I. M. ary Subawa, A. A. I. I. Paramitha, and I. P. B. Suyasa, "Pengujian Blackbox Sistem Informasi Penerimaan Mahasiswa Baru (PMB) Primakara University Berbasis Equivalence Partitions dan SUS Kuesioner 1,2," vol. 12, no. 2, pp. 86–96, 2025.

[19] L. B. Herdianto, "Evaluasi Sistem E-Learning Pt. Otak Kanan Melalui Pengujian Blackbox Testing Dan System Usability Scale (Sus)," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 290–296, 2023.

[20] C. A. Putri, R. K. Anwar, S. Chaerani, D. Amar, and E. N. Rukmana, "Keamanan Informasi dan Privasi Pengguna dalam Layanan Perpustakaan Digital," vol. 31, no. 3, 2024, doi: 10.37014/medpus.v31i3.5317.