

A Conceptual Framework for a Progressive Web App-Based Student Data Management System

Haldi Alfaisal¹, Nia Rahmadani², Ahmad Ashar³, Elsa⁴, Riksal Rivaldi⁵, Sri Rejeki⁶,
Dimas Rezkianto⁷, Iin Sugiarti⁸, Zulfikar Isnansah⁹

Computer Engineering Study Program, Faculty of Engineering, Muhammadiyah University of North Kolaka

haldialfaisal096@gmail.com¹, nia355481@gmail.com², ahmadashar2525@gmail.com³,
ee4373056@gmail.com⁴, rikسالrifaldi669@gmail.com⁵, srjksrrejek@gmail⁶,
dimasrezkiantto36@gmail.com⁷, iin11sugiarti12@gmail.com⁸, zulfikarisnanansah157@gmail.com⁹

Article Info

Article history:

Received April 25, 2026

Revised May 18, 2026

Accepted May 26, 2026

Keywords:

Progressive Web App, student data management, web-based application, PHP, MySQL, Black Box Testing.

ABSTRACT

The rapid advancement of information technology has accelerated digital transformation across various sectors, including student data management in educational institutions. Conventional manual data management often encounters challenges such as recording errors, difficulties in data retrieval, and the risk of information loss. This study aims to design and develop a Progressive Web App (PWA)-based student data management application using PHP and MySQL to improve data management efficiency while providing a better user experience. The study employed a structured web-based system development approach consisting of four stages: requirements analysis, system design, implementation, and testing. The system was designed using Use Case Diagrams, Flowcharts, and Entity Relationship Diagrams (ERD) as the foundation for application development. The implemented system includes several key features, such as user authentication, an administrative dashboard, student data management, data entry and editing, data search functionality, and responsive web design support. System testing was conducted using the Black Box Testing method through fourteen test scenarios covering login, student data management, dashboard functionality, data search, system responsiveness, and Progressive Web App features. The testing results demonstrated that all system functions operated successfully according to the specified functional requirements, achieving a 100% success rate. Furthermore, the implementation of Progressive Web App technology enables the application to be installed on user devices, supports service workers, and improves application performance through caching mechanisms. Therefore, the proposed application provides an effective, efficient, and responsive solution for computerized student data management in educational environments.

This is an open access article under the CC BY-SA license.



Article Info

Article history:

Received April 25, 2026

Revised May 18, 2026

Accepted May 26, 2026

ABSTRAK

Perkembangan teknologi informasi yang pesat menuntut adanya perkembangan teknologi informasi telah mendorong digitalisasi dalam berbagai bidang, termasuk pengelolaan data siswa di lingkungan pendidikan. Pengelolaan data secara manual masih

Kata kunci:

Progressive Web App,
pengelolaan data
siswa, aplikasi
berbasis web, PHP,
MySQL, Black Box
Testing.

menghadapi berbagai kendala, seperti kesalahan pencatatan, kesulitan pencarian data, serta risiko kehilangan informasi. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi pengelolaan data siswa berbasis Progressive Web App (PWA) menggunakan PHP dan MySQL sehingga dapat meningkatkan efisiensi pengelolaan data sekaligus memberikan pengalaman pengguna yang lebih baik. Metode penelitian yang digunakan adalah metode pengembangan sistem berbasis web dengan pendekatan terstruktur yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Perancangan sistem dilakukan menggunakan Use Case Diagram, Flowchart, dan Entity Relationship Diagram (ERD) sebagai dasar pembangunan aplikasi. Implementasi antarmuka menghasilkan beberapa fitur utama, yaitu autentikasi pengguna, dashboard, pengelolaan data siswa, penambahan dan pengeditan data, pencarian data, serta dukungan responsive web design. Pengujian sistem dilakukan menggunakan metode Black Box Testing terhadap empat belas skenario pengujian yang mencakup fungsi login, pengelolaan data siswa, dashboard, pencarian data, responsivitas sistem, hingga fitur Progressive Web App. Hasil pengujian menunjukkan bahwa seluruh fitur berjalan sesuai dengan kebutuhan fungsional yang telah dirancang dengan tingkat keberhasilan sebesar 100%. Penerapan teknologi Progressive Web App juga memungkinkan aplikasi diinstal pada perangkat pengguna, mendukung pemanfaatan service worker, serta meningkatkan kecepatan akses melalui mekanisme caching. Dengan demikian, aplikasi yang dikembangkan mampu menjadi solusi yang efektif, efisien, dan responsif dalam mendukung pengelolaan data siswa secara terkomputerisasi di lingkungan pendidikan.

This is an open access article under the CC BY-SA license.



Corresponding Author:

In Sugiarti
Universitas Muhammadiyah Kolaka Utara
Email: iin11sugiarti12@gmail.com

INTRODUCTION

The rapid advancement of information and communication technology (ICT) in the digital era has significantly transformed various aspects of human life. This transformation is characterized by the widespread adoption of digital devices, such as computers and smartphones, across numerous sectors, including education. Information technology has evolved beyond being merely a supporting tool; it has become an essential component in improving efficiency, effectiveness, and accuracy in data processing, information management, and service delivery. The integration of digital technologies into educational environments has enabled institutions to optimize administrative processes, improve accessibility to information, and enhance the quality of educational services provided to students and stakeholders.

One of the most significant developments in information technology is the rapid growth of web-based applications. Web applications offer numerous advantages because they can be accessed through web browsers without requiring users to install dedicated software. Furthermore, web applications are generally easier to develop, maintain, and distribute than

native mobile applications, making them an attractive solution for many organizations. However, conventional web applications still suffer from several limitations, including dependence on continuous internet connectivity, relatively slower performance under unstable network conditions, and a user experience that is often less interactive than that provided by native mobile applications.

To address these limitations, Progressive Web App (PWA) technology has emerged as an innovative approach that combines the advantages of web applications and native mobile applications. A Progressive Web App is a modern web application designed to deliver a native-like user experience by utilizing technologies such as Service Workers, Web App Manifest, and intelligent caching mechanisms. These technologies enable web applications to be installed on users' home screens, operate in fullscreen mode, send notifications, and continue functioning even when network connectivity is limited or temporarily unavailable [1].

In addition to providing offline capabilities, PWAs are designed to be fast, responsive, and reliable across various network conditions. According to Russell, Progressive Web Apps are characterized by three fundamental principles: reliable, fast, and engaging. These characteristics ensure that applications load quickly, remain accessible under unstable network conditions, and provide interactive experiences comparable to native mobile applications [2]. Consequently, Progressive Web Apps have become one of the most promising technologies for developing modern web applications that deliver high-quality user experiences while maintaining the flexibility and accessibility of web-based systems.

Within the educational sector, student data management represents one of the most critical administrative activities. Student information—including personal identity, academic records, class assignments, and contact details—must be managed accurately and systematically to support academic administration and decision-making processes. Unfortunately, many educational institutions still rely on manual record-keeping or conventional spreadsheet applications to manage student data. Such practices often lead to problems such as data redundancy, recording errors, inefficient data retrieval, inconsistencies, and an increased risk of data loss due to human error or hardware failure. These limitations reduce administrative efficiency and negatively affect the quality of educational services.

Developing a student information system based on Progressive Web App technology offers an effective solution to these challenges. Through PWA implementation, users can access the application using a standard web browser while also installing it on their devices, enabling an experience similar to native mobile applications. Furthermore, Service Worker technology enables intelligent caching, allowing frequently accessed data and application resources to remain available even during temporary internet disruptions [3]. These capabilities improve system accessibility, usability, and overall user satisfaction while reducing dependency on stable internet connectivity.

The development of software systems also requires an appropriate software engineering methodology to ensure that the resulting application satisfies user requirements and maintains high quality. According to Pressman, software development should follow systematic stages beginning with requirements analysis, followed by system design, implementation, testing,

deployment, and maintenance to produce reliable and maintainable software [4]. Applying a structured development methodology minimizes development risks, improves software quality, and facilitates future maintenance and system enhancement.

Based on these considerations, this study aims to design and develop a Progressive Web App-based Student Data Management Application using PHP and MySQL. The proposed application is expected to facilitate efficient student data management, improve administrative productivity, minimize data processing errors, and provide an enhanced user experience through modern Progressive Web App technologies. Therefore, the implementation of Progressive Web App technology is expected to provide an effective and practical solution for developing modern educational information systems while supporting digital transformation in educational institutions.

METHOD

This study employed a structured web-based system development approach as the research methodology. The structured approach was selected because it provides a systematic and organized framework throughout the software development lifecycle, beginning with requirement identification and continuing through system design, implementation, and testing. By following sequential development stages, each phase can be completed in an orderly manner while minimizing potential development errors. Furthermore, this methodology facilitates system evaluation, debugging, and future improvements whenever deficiencies are identified during the development process.

The primary objective of adopting a structured development methodology is to produce software that fulfills user requirements, maintains high software quality, and remains maintainable and scalable for future development. According to Pressman, software engineering should follow well-defined and carefully planned development stages to ensure that the resulting software operates effectively and satisfies user expectations [5]. Therefore, this research follows four principal stages: requirements analysis, system design, implementation, and system testing.

Requirements Analysis

During the requirements analysis phase, a Use Case Diagram was employed to represent the functional requirements of the proposed system. A Use Case Diagram illustrates interactions between system users (actors) and the system while identifying the primary functionalities that the system must provide. This modeling technique enables system requirements to be visualized clearly, making communication among developers and stakeholders more effective while ensuring that functional requirements are comprehensively documented [6].

In this study, the proposed system is a Progressive Web App-based Student Data Management Application operated by an administrator. Consequently, the administrator serves as the primary actor in the Use Case Diagram and possesses full authorization to access all available system functions. The administrator can perform several primary activities, including user authentication (login), student data management, and system logout. Once authentication

has been successfully completed, the administrator gains access to the system dashboard, where various data management functions are available. The proposed Use Case Diagram is presented in Figure 1.

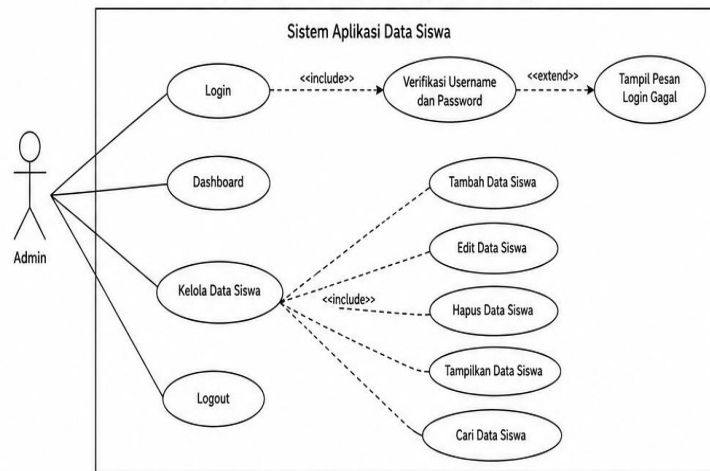


Figure 1. Use Case Diagram of the Proposed Progressive Web App-Based Student Data Management System

The Use Case Diagram illustrates that all student data management activities within the proposed system are performed under the authority of the administrator. Every action executed by the administrator is processed by the system and subsequently stored in the centralized database. Through this modeling approach, the functional requirements of the system can be clearly identified, demonstrating that the proposed application provides comprehensive features for managing student data efficiently and effectively. Furthermore, the Use Case Diagram serves as a fundamental reference during the system design phase, ensuring that the application development process is carried out systematically and remains aligned with user requirements and the objectives of the proposed system.

System Design

System design represents the subsequent phase following the requirements analysis and aims to describe in detail how the proposed application will be structured and implemented. During this stage, all identified functional and non-functional requirements are translated into a comprehensive technical design that serves as the foundation for software development. A well-defined system design is essential because it provides a clear blueprint for developers, minimizes implementation errors, facilitates future maintenance, and ensures that the final application fulfills user expectations while achieving the objectives of the study.

The system workflow was designed to describe the overall operational process of the proposed application. The process begins when a user accesses the application and navigates to the login page to perform user authentication. The credentials entered by the user are validated against the information stored in the database. If the authentication process is successful, the user is granted access to the dashboard, which serves as the main interface for interacting with the system. Conversely, if the authentication fails due to incorrect credentials, the system displays an error notification and requests the user to re-enter valid login information.

After successful authentication, the administrator can access the available system functionalities, including managing student records, searching student information, updating existing data, and performing other administrative operations. Every operation executed by the administrator is processed by the application and synchronized with the centralized database to ensure data consistency and integrity. The workflow concludes when the administrator chooses to log out of the application, terminating the active session and returning the user to the login page. This systematic workflow ensures secure access, efficient data management, and reliable operation of the proposed Progressive Web App-based Student Data Management System.

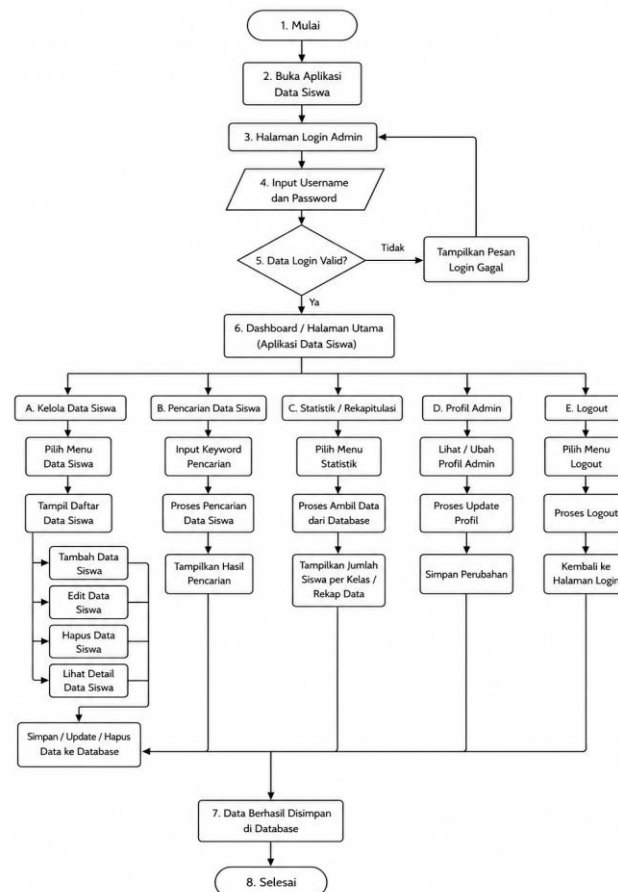


Figure 2. Flowchart of the Proposed Progressive Web App-Based Student Data Management System

Database Design (Entity Relationship Diagram)

Database design is a fundamental stage in the software development process because it defines how data are structured, stored, and managed within the system. A well-designed database ensures efficient data organization, minimizes redundancy, maintains data integrity, and supports reliable information retrieval. In the proposed student data management system, MySQL is utilized as the Database Management System (DBMS) due to its reliability, scalability, and ability to efficiently manage relational data. The database serves as the central repository for storing all information related to the Progressive Web App-based Student Data Management System.

The proposed database is designed using the Entity Relationship Diagram (ERD) approach to model the relationships among the entities within the system. The ERD provides a

conceptual representation of the database structure, illustrating how data entities are connected through primary keys and foreign keys. This modeling approach facilitates the development process by providing a clear understanding of data relationships while ensuring that the database design supports the functional requirements identified during the requirements analysis phase.

The proposed system consists of three primary database tables: the Admin table, the Student table, and the Class table. The Admin table stores authentication and administrator profile information, including attributes such as `admin_id`, `username`, `password`, and `admin_name`. This table plays a crucial role in the authentication and authorization process by verifying user credentials before granting access to the system. Through this mechanism, only authorized administrators are permitted to access and manage the application's functionalities.

The Student table is designed to store comprehensive student information, including student identification numbers, personal information, class assignments, gender, contact details, and other relevant academic data. This table serves as the primary repository for student records and supports various operations such as data insertion, modification, deletion, searching, and reporting. Meanwhile, the Class table contains information related to class management, including class identifiers, class names, and other attributes required to organize students according to their respective classes. The relationship between the Student table and the Class table enables the system to associate each student with the appropriate class efficiently while maintaining data consistency.

The relationships among these entities are established according to relational database principles, ensuring that each data item is stored only once while maintaining referential integrity through the use of primary and foreign keys. This relational database design minimizes data redundancy, improves storage efficiency, and guarantees data consistency throughout all system operations. Furthermore, the centralized database architecture enables every module of the application to access synchronized information, thereby supporting reliable data management and facilitating future system expansion.

The proposed Entity Relationship Diagram (ERD), which illustrates the relationships among all database entities within the system, is presented in Figure 3. This database model serves as the foundation for implementing the database structure during the software development phase and ensures that the proposed application can efficiently support all functional requirements of the Progressive Web App-based Student Data Management System.

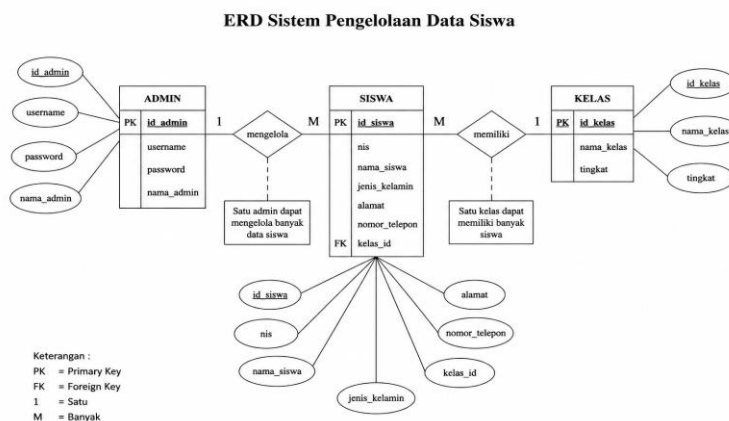


Figure 3. Entity Relationship Diagram (ERD) of the Proposed Student Data Management System

The proposed database design is represented using an Entity Relationship Diagram (ERD) to provide a clear visualization of the database structure and the relationships among entities within the system. The ERD serves as a conceptual model that illustrates how data are organized, connected, and managed throughout the application. By providing a comprehensive representation of the database architecture, the ERD facilitates developers' understanding of data flow and entity relationships, thereby supporting a more systematic and efficient implementation process. Furthermore, the ERD serves as an essential reference during database implementation, ensuring that the developed database structure remains consistent with the functional requirements identified during the system analysis phase. The proposed Entity Relationship Diagram is presented in Figure 3.

User Interface Implementation

User interface implementation represents the stage in which the proposed interface design is translated into a functional web application. During this phase, all interface components designed in the previous stage are implemented using web technologies, including HTML, CSS, and JavaScript. The primary objective of this implementation is to provide an intuitive, efficient, and user-friendly interaction between users and the system while ensuring that all functional requirements are properly represented within the application.

The interface is designed by considering User Experience (UX) and User Interface (UI) principles to create a modern, responsive, and visually appealing application. Consistent navigation, organized layouts, readable typography, and responsive components are incorporated to improve usability across various devices, including desktop computers, tablets, and smartphones. This design approach enables users to manage student information efficiently while minimizing operational complexity and improving overall interaction with the application.

RESULTS AND DISCUSSION

Login Page Implementation

The Login Page is the initial interface displayed when the application is launched. This page functions as the primary authentication mechanism to prevent unauthorized users from accessing the system. The login interface contains two mandatory input fields, namely username and password, which must be completed before users are granted access to the application.

When the Login button is selected, the system validates the submitted credentials against the administrator data stored in the MySQL database. If the authentication process is successful, the user is redirected to the Dashboard page, where all system functionalities become accessible according to the assigned access privileges. Conversely, if the entered username or password is incorrect, the system displays an authentication error message and prompts the user to re-enter valid credentials. This authentication mechanism enhances system security by ensuring that only authorized administrators can access and manage student information.

The implementation of the login interface emphasizes simplicity, usability, and security. A clean and responsive layout enables users to perform authentication quickly while maintaining a consistent visual appearance across different screen sizes and devices. This implementation supports the overall objective of providing an efficient and reliable student data management system based on Progressive Web App (PWA) technology. The implemented login page is presented in Figure 4.

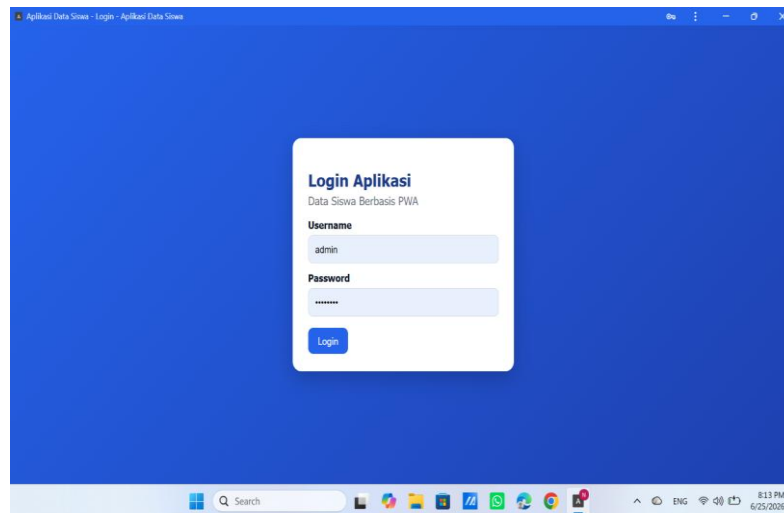


Figure 4. Login Page

Dashboard Page Implementation

The Dashboard serves as the main interface displayed after users successfully authenticate through the login process. It is designed as the application's central control panel, providing quick access to all primary features available in the Progressive Web Application (PWA) for student data management. The dashboard aims to improve system usability by presenting essential information in a concise, organized, and user-friendly manner. According to Pressman and Maxim (2022), a well-designed user interface enhances user interaction efficiency and facilitates effective information retrieval.

The dashboard displays several statistical summaries automatically retrieved from the database, including the total number of students, the number of male students, the number of female students, and the total number of registered classes. These statistics are presented in real time, ensuring that any updates made by the administrator are immediately reflected on the dashboard. Consequently, users can monitor the latest data without manually accessing individual modules.

In addition to statistical information, the dashboard provides navigation menus that allow users to access the application's primary modules, including student management, class management, data searching, and other administrative functions. The navigation layout is designed with simplicity and consistency to ensure smooth transitions between pages and to improve overall usability, as recommended by Nielsen (2020).

The dashboard presents information using statistical cards and graphical elements with a modern and responsive design. This visual representation enables users to quickly understand

the overall status of the stored data while improving the application's visual appeal. Furthermore, data visualization supports monitoring activities and facilitates more effective decision-making by presenting key information in an intuitive format.

The dashboard implementation also adopts the Responsive Web Design approach, allowing the interface to adapt seamlessly to various screen sizes, including desktop computers, laptops, tablets, and smartphones. By leveraging Progressive Web App technology, the dashboard can be accessed through modern web browsers without requiring traditional application installation, thereby providing users with greater accessibility and flexibility.

Overall, the dashboard implementation provides an informative, user-friendly, and efficient interface that supports student data management activities. It functions not only as the main landing page after authentication but also as the central hub for navigation and information within the proposed system. The implemented dashboard interface is presented in Figure 5.

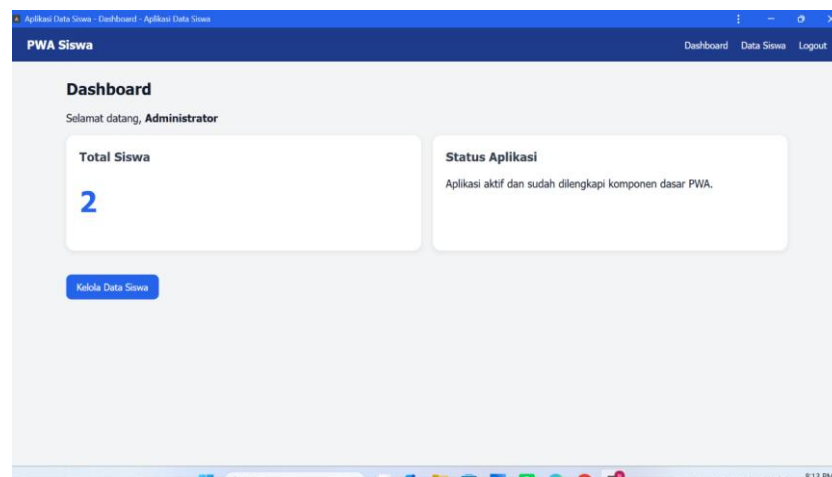


Figure 5. Dashboard page

Student Data Page Implementation

The Student Data page is one of the main modules of the application designed to manage all student information stored in the database. This page serves as the primary administrative interface, enabling administrators to perform student data management efficiently, systematically, and effectively. The implementation of this page aims to simplify data entry, retrieval, updating, and deletion processes, thereby ensuring that student information is managed in an integrated manner. According to Pressman and Maxim (2022), a well-designed information system enhances data management quality and facilitates efficient information processing.

Student records are presented in a data table containing essential information such as the Student Identification Number (Student ID), student name, gender, class, address, telephone number, and other relevant information required by the system. Presenting data in tabular form provides a structured and organized view, allowing administrators to easily identify and manage student records. Furthermore, all displayed information is retrieved directly from the database, ensuring that any updates made by the administrator are immediately reflected within the system.

To support administrative activities, the Student Data page provides complete Create, Read, Update, and Delete (CRUD) functionalities, including Add, Edit, Delete, and View Details features. The Add feature enables administrators to register new student records, while the Edit feature allows existing information to be updated whenever changes occur. The Delete feature removes obsolete records after a confirmation process to prevent accidental data loss. These integrated functions enable administrators to perform all data management tasks efficiently within a single interface.

In addition, the page includes search and sorting functionalities, allowing administrators to quickly locate student records based on keywords such as student name, Student ID, or class. These features significantly improve data retrieval efficiency, particularly when the database contains a large number of student records. According to Nielsen (2020), search functionality is a critical usability component that enhances user efficiency by reducing the time required to locate specific information.

The Student Data page also adopts the Responsive Web Design approach, enabling the interface, data tables, and action buttons to adapt seamlessly to various screen sizes, including desktop computers, laptops, tablets, and smartphones. By leveraging Progressive Web App (PWA) technology, the application can be accessed through modern web browsers without requiring conventional software installation. This approach provides greater flexibility, allowing administrators to manage student data anytime and from any compatible device.

Overall, the implementation of the Student Data page provides a user-friendly, informative, and efficient interface for student data administration. The combination of structured data presentation, comprehensive management features, fast search capabilities, and responsive interface design enables the system to support student data management effectively and efficiently. The implementation of the Student Data page is presented in Figure 6.

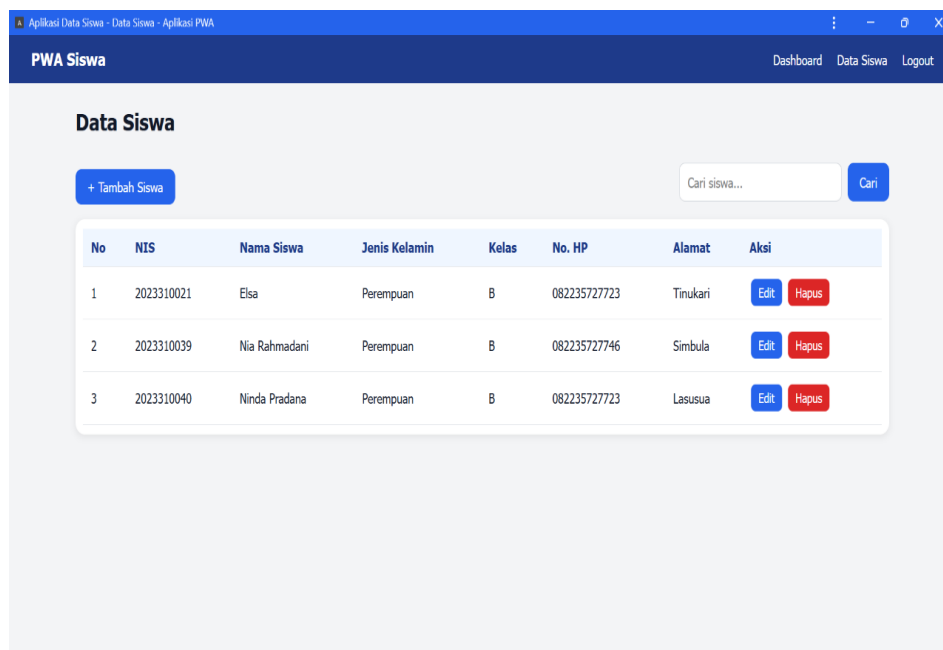


Figure 6. Student Data Management Page

Student Data Add and Edit Form Implementation

The Student Data Add and Edit Form is designed to facilitate the creation and updating of student records within the proposed application. This interface enables administrators to efficiently manage student information through a structured and user-friendly data entry form. The implementation of this form aims to simplify the process of maintaining student records while ensuring data consistency and accuracy throughout the system.

The form allows users to enter essential student information, including the Student Identification Number (Student ID), student name, gender, class, address, and telephone number. Each input field is designed according to its corresponding data type to improve usability and minimize input errors. Before the submitted information is stored in the database, the system performs an input validation process to verify that all required fields have been completed correctly and comply with the predefined validation rules.

During the data creation process, newly entered student information is validated and then stored as a new record in the database. Conversely, during the data editing process, the system automatically retrieves the existing student information and displays it within the form, allowing administrators to modify only the required fields. Once the update process is completed, the modified data are automatically synchronized with the database to ensure that the stored information remains accurate and up to date.

The implementation of the form adopts a simple, clean, and intuitive user interface to improve the overall user experience. The arrangement of labels, input fields, and action buttons is designed to facilitate efficient data entry while reducing the likelihood of user errors. In addition, the interface follows the Responsive Web Design approach, enabling the form to function properly across various devices, including desktop computers, laptops, tablets, and smartphones.

Overall, the implementation of the Student Data Add and Edit Form provides an effective mechanism for managing student records by combining structured data entry, systematic validation, and a responsive interface. These features improve administrative efficiency while maintaining the accuracy, consistency, and integrity of student information stored in the application. The implementation of the Student Data Add and Edit Form is presented in Figure 7.

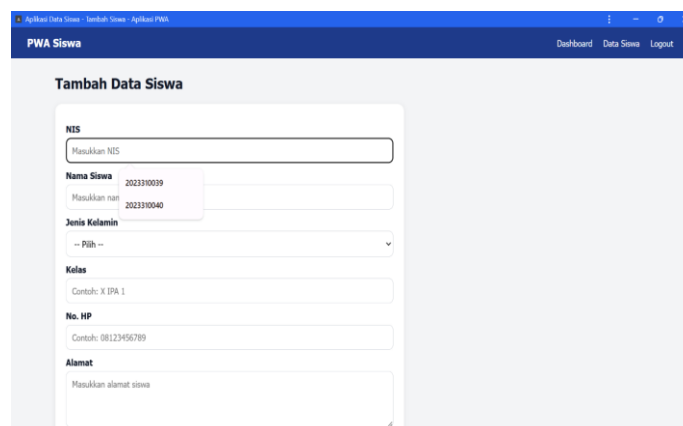


Figure 7. Student Data Add and Edit Form

Search Page Implementation

The Search Page is implemented to assist users in locating student records quickly and efficiently within the application. This feature becomes particularly important when the database contains a large number of student records, making manual searches time-consuming and inefficient. By providing an integrated search mechanism, the system enables users to retrieve the required information with minimal effort, thereby improving overall system usability and operational efficiency.

Users can perform a search by entering keywords such as the student name or Student Identification Number (Student ID) into the search field provided. Once the search button is selected, the system processes the entered keyword by querying the database and displays all records that match the search criteria. The search results are presented immediately in a structured table, allowing users to access the desired information quickly and accurately.

The search functionality is designed to improve information retrieval efficiency by reducing the time required to locate specific student records. In addition, the feature enhances the user experience by providing a simple and intuitive interface that supports fast navigation through large datasets. This functionality is particularly beneficial for administrators who frequently manage and update student information as part of their daily administrative activities.

Overall, the implementation of the Search Page significantly improves the effectiveness of student data management by providing a fast, accurate, and user-friendly search mechanism. The feature contributes to increased productivity and supports efficient access to information within the proposed Progressive Web Application. The implementation of the Search Page is presented in Figure 8.

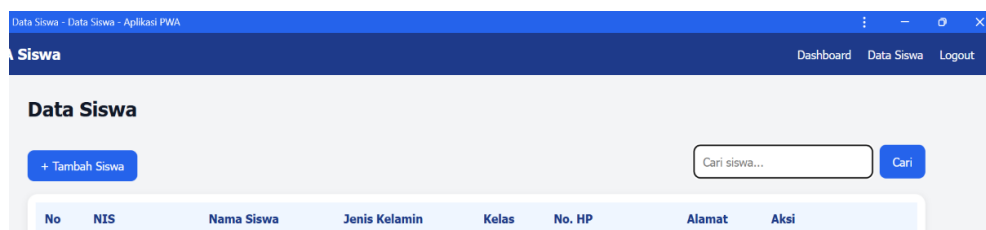


Figure 8. Search Page

Responsive System Implementation

In addition to implementing the core functionalities, the application was developed using the Responsive Web Design (RWD) approach to ensure that the user interface adapts seamlessly to different screen sizes and device resolutions. Responsiveness testing was conducted on various devices, including desktop computers, laptops, tablets, and smartphones, to verify that all interface components were displayed correctly and functioned as intended across multiple platforms.

The responsive implementation enables users to access the application anytime and anywhere without experiencing display or usability issues. This approach not only improves accessibility but also enhances the flexibility of the application by providing a consistent user

experience regardless of the device being used. Consequently, the proposed Progressive Web Application supports efficient student data management while offering greater convenience and mobility for administrators.

Analysis of the User Interface Implementation

Based on the implementation results, all user interface components functioned according to the system design developed during the design phase. Every feature was successfully accessed and operated without significant technical issues. The clean interface design, intuitive navigation, and responsive layout contributed to an improved user experience, allowing users to interact with the application more efficiently.

The implementation of a user-friendly interface also enables administrators to understand and utilize the available features without requiring extensive training. As a result, the developed interface successfully supports the primary objective of the application, namely facilitating effective and efficient student data management while improving the overall usability of the system.

System Testing Results

System testing was conducted to ensure that all application functionalities operated according to the requirements identified during the system analysis and design phases. The testing process employed the Black Box Testing method, which evaluates the functional behavior of the application without considering the internal structure or source code. The testing procedure involved providing various input scenarios to the application and observing the corresponding outputs to verify that each function produced the expected results.

The testing focused on validating the primary functionalities of the application, including user authentication, student data management, data searching, data updating, and other administrative features. Each test scenario was designed to verify whether the system responded correctly to valid and invalid inputs while maintaining data integrity and operational consistency.

The testing results demonstrated that all major application features performed successfully according to their functional specifications. The system was able to process user requests correctly, display accurate information, and perform data management operations without significant errors. These findings indicate that the proposed Progressive Web Application meets the specified functional requirements and is capable of supporting efficient student data management in an educational environment.

Research Limitations

This study has several limitations that should be acknowledged. First, the developed application focuses primarily on student data management and does not yet include more advanced features such as automatic notifications, report generation and printing, or integration with other academic information systems. Second, the system evaluation was limited to Black-Box Testing, which assessed the functional correctness of the application but did not include

usability testing, user satisfaction evaluation, or comprehensive performance testing. Furthermore, the proposed application was developed and evaluated on a limited scale and has not yet been deployed or validated in a real school environment. Therefore, future studies are expected to implement the system in educational institutions, conduct more comprehensive testing involving end users, and enhance the application by incorporating additional features to improve its functionality, usability, and overall effectiveness.

Conclusion

Based on the results of this study, it can be concluded that the proposed Progressive Web Application (PWA) for student data management was successfully designed and developed in accordance with the identified user requirements. The application provides essential functionalities, including user authentication, student data management, data searching, and logout, enabling a structured and efficient management process. The results of the Black-Box Testing indicate that all implemented features functioned as expected and met the predefined functional requirements. Furthermore, the adoption of Progressive Web App (PWA) technology enhances application accessibility, improves performance, and enables responsive operation across various devices and screen sizes. Overall, the proposed application offers a practical solution for improving the efficiency, accuracy, and digitalization of student data management processes in educational institutions. Future work should focus on integrating additional functionalities, conducting comprehensive usability and performance evaluations, and validating the system through real-world implementation in school environments.

References

- Kotler, P., Keller, K. L., & Chernev, A. (2022). *Marketing management* (16th Global ed.). Pearson.
- Nielsen, J. (2020). *Usability engineering* (Updated ed.). Morgan Kaufmann.
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Russell, A. (2022). *Progressive web apps*. A Book Apart.
- Sommerville, I. (2021). *Software engineering* (11th ed.). Pearson.
- Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T. M. (2017). Progressive web apps for the unified development of mobile applications. In *International Conference on Web Information Systems and Technologies* (pp. 64–86). Springer.
- Fortunato, D., & Bernardino, J. (2018). Progressive web apps: An alternative to native mobile apps. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). IEEE. <https://doi.org/10.23919/CISTI.2018.8399283>
- Gambhir, A., & Raj, G. (2018). Analysis of cache in service worker and performance scoring of progressive web application. In *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (pp. 294–299). IEEE.