Design Front-End Information System of LKP Aneka Prima Kayuagung using The Waterfall Method

Rizqy Gunafi Hidayatullah[1], Alex Wijaya[2]

[1,2]Universitas Bina Darma
Corresponding Email : Rizqygunafihidayatullah@gmail.com

Received: 2025-11-12 /Accepted: 2025-11-12 / Doi:

ABSTRACT

The rapid development of digital technology has changed how organizations manage and deliver information. Educational institutions, particularly non-formal training centers, require efficient information systems to support administrative and operational processes. This study employs a Research and Development (R&D) method using the Waterfall software development model. The research followed five sequential stages: analysis, design, implementation, testing, and maintenance. Data were collected through interviews and direct observation of LKP's administrative activities. The front-end interface was developed using HTML, CSS, and Bootstrap, while PHP and MySQL were used for the back-end integration. The result is a web-based information system capable of managing student data, class schedules, and training information in a user-friendly, responsive, and structured format. Testing shows that the system functions effectively and enhances the institution's ability to store, update, and access information in real-time. This research demonstrates that a structured development approach such as the Waterfall model can significantly improve digital transformation for non-formal educational institutions.

Keywords: *Digital Transformation, Front-End Design, Information System, LKP, Waterfall Model*

1. Introduction

As today's era becomes increasingly sophisticated and modern, digital advancements have significantly simplified the way people carry out their daily activities. In response to diverse and evolving human needs, numerous digital innovations have been created to address a wide range of societal and organizational challenges. Digital innovation itself is a continuous and dynamic process aimed at identifying, developing, and implementing effective technological solutions to existing problems (Starke & Ludviga, 2024). The steady pace of this development demonstrates that technology has grown into a fundamental component of organizational progress and transformation.

Consequently, digital innovation is not executed solely at the individual level but is primarily driven by organizations such as private companies, government institutions, and educational agencies. These entities leverage digital technologies to enhance business operations, optimize service delivery, streamline administrative processes, and strengthen their competitive advantage in an increasingly digital-centric environment (Awad & Rojas, 2024). As organizations face more complex operational demands and greater expectations from stakeholders, the adoption of systematic, technology-based solutions has become not only beneficial but essential for ensuring long-term effectiveness, adaptability, and sustainability (Mayasari, 2023).

In today's digital era, information systems have an important role in increasing the efficiency and effectiveness of an organization's operations in carrying out its activities. Information systems enable organizations to automate processes, reduce human error, support decision-making, and create a structured flow of information. Numerous forms of information systems have been implemented across various sectors, such as inventory information systems (Sobri et al., 2017), library information system (Solikhin et al., 2018), hotel information system (Estrada & Sobri, 2022) and many more. Their successful adoption demonstrates how digital systems can transform conventional workflows into more integrated and measurable processes. Educational institutions including course and training centers also require well-designed information systems to manage data, streamline communication, support academic and administrative activities, and provide higher-quality services for both students and staff (Mazadu et al., 2022).

LKP (Lembaga Kursus dan Pelatihan) is an institution in the field of non-formal education other than formal which provides education in the form of specific training according to the needs required by the community (Sholeh et al., 2023). LKP Aneka Prima Kayu Agung as a non-formal educational institution offering various training programs, it currently still carries out most of its operational and administrative activities manually. This process often results in data duplication, inefficiency, difficulty in obtaining accurate and up-to-date information, difficulty in accessing data in real time, potential data loss, and a lack of engagement in conveying information to prospective students (Bing & Farhana, 2024).

These challenges not only hinder the smooth execution of daily operational activities but also undermine the institution’s capacity to sustain service quality, maintain stakeholder satisfaction, and remain competitive in an increasingly digital-oriented environment. Consequently, the development of a website-based information system emerges as a highly relevant and strategic solution to overcome these limitations. By integrating core administrative functions into a centralized and accessible digital platform, such a system enables LKP Aneka Prima Kayu Agung to streamline its workflows, improve data accuracy, strengthen communication between staff and learners, and support more informed decision-making. Ultimately, this technological initiative contributes to greater operational efficiency, enhanced service delivery, and overall improvement in institutional performance.

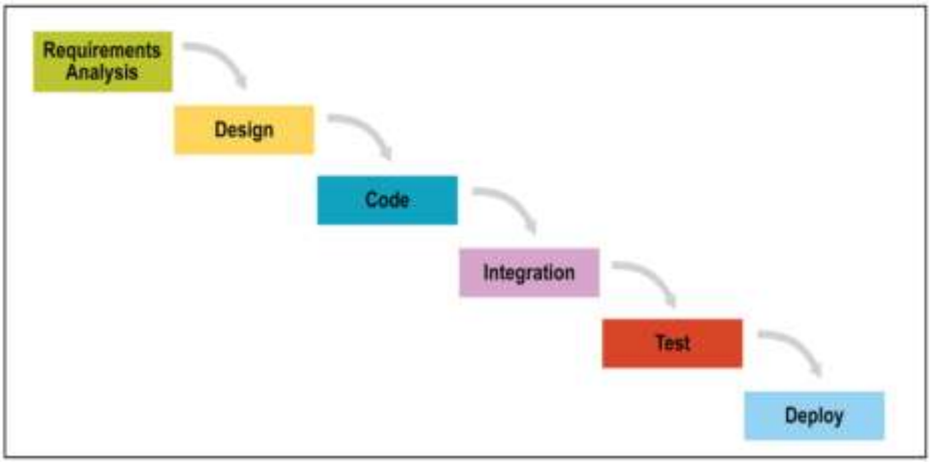
2. Research Methods

This study employs a Research and Development (R&D) method using the

Waterfall software development model. This model was chosen because of its structured and systematic nature, allowing software development to be carried out in clear stages, each phase from requirement analysis, system design, implementation, testing, deployment, to maintenance has a specific objective and produces well-defined outputs that serve as the foundation for the next stage (Bassil, 2012). Such a sequential process helps maintain clarity, consistency, and traceability throughout development. Moreover, the Waterfall model is highly suitable for projects where system requirements can be identified early and are unlikely to undergo major changes. By emphasizing thorough documentation and step-by-step progression, this method minimizes the risk of errors, reduces ambiguity in development activities, and ensures that the resulting system aligns with user needs and organizational requirements. Its predictability and ease of management make it a reliable choice for developing information systems in educational institutions or organizations with structured operational workflows (Tjahjanto et al., 2022).

According to (Kayande & Phadnis, 2024), The Waterfall model consists of several sequential phases, namely requirements analysis, system design, coding or implementation, integration, testing, and deployment, as illustrated in Figure 1. Each phase must be completed before progressing to the next, ensuring a structured and systematic approach to software development.

Figure 1. Waterfall Model according to (Kayande & Phadnis, 2024)



2.1. Requirements Analysis Phase

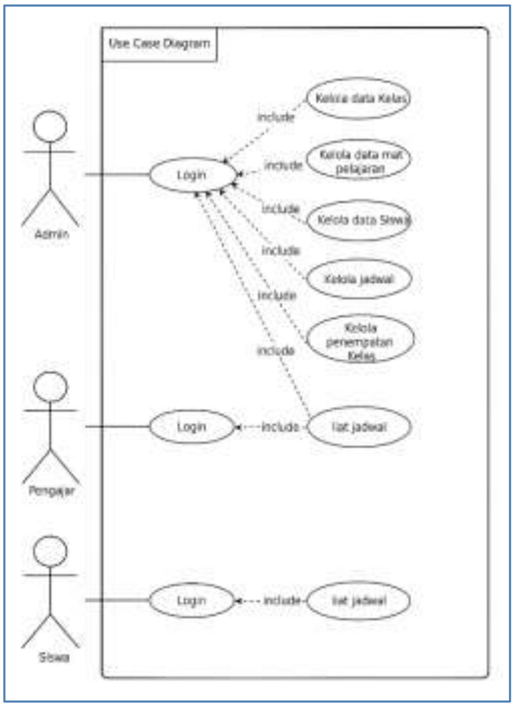
This is the first phase where all system requirements are gathered, clarified, and documented. Activities include: identifying functional requirements (what the system should do), identifying non-functional requirements (performance, security, reliability, etc.), and determining constraints and project goals. In this phase, information was gathered through interviews with the LKP Aneka Prima Kayu administrators to identify system requirements.

Direct observations of the classroom management processes and learning schedules were also carried out to gain a more comprehensive and contextual understanding of how academic and administrative activities operate on a daily basis. These observations complemented the interviews and document reviews, ensuring that the collected requirements accurately reflected real conditions within the institution. The outcome of this phase is a detailed requirements specification document, which outlines functional and non-functional system needs and serves as a foundational reference for guiding the subsequent design and development phases.

2.2. Design

In this phase, the requirements are transformed into a detailed system design. The team creates: system architecture, database design, user interface (UI/UX) design, module and workflow designs. This phase serves as the blueprint for the developers before coding begins. Begins with the creation of UML diagrams, such as use-case diagrams, activity diagrams, and sequence diagrams, to document the system's process flow in a structured manner. Figure 2 shows the use case of the information system that will be built.

Figure 2. Use Case



Furthermore, a database structure is designed using MySQL to ensure efficient storage of class, schedule, and user data. As a first step in visualizing the user interface, wireframes or page prototypes are created to illustrate key features such as the class management, schedule, and information pages. The following is a database table design based on the use case diagram in Figure 2 above. The diagram shows three actors (Admin, Teacher, Student) and several functions: Login, Manage Class Data, Manage Subject Data, Manage Student Data, Manage Schedule, Manage Class Placement, and View Schedule. Based on these functions, the following is a detailed database structure.

2.2.1. The Users Table

The user table serves as the central repository for storing all types of user accounts within the system, including administrators, teachers, and students. This table ensures that every individual who interacts with the application has a unique and identifiable record. The fields contained in the user table include *user_id* as the primary key, *username* and *password* for authentication, *role* to differentiate the access level of each user, and personal information such as *name* and *email*. These fields work together to support secure login, role-based access control, and user management features. The complete structure of the user table is presented in Table 1 below.

Table 1. The Users Table

No	Field	Type	Description
1	user_id (PK)	INT	Primary key
2	username	VARCHAR(50)	Username to login
3	password	VARCHAR(255)	Password (hash)
4	role	ENUM('admin','teacher','student')	User type
5	name	VARCHAR(100)	Full name
6	email	VARCHAR(100)	Optional

2.2.2. The Class Table

The class table is designed to store comprehensive information about all classes managed within the system. This table helps organize academic data by providing a structured record for each class. The fields included in the class table consist of *class_id* as a unique identifier, *class_name* to specify the name or label of the class, and *level* to indicate the class grade or academic level. Together, these fields support efficient class management, data retrieval, and linkage with other related tables such as students or schedules. The complete structure of the class table is presented in Table 2 below.

Table 2. The Class Table

No	Field	Type	Description
1	class id (PK)	INT	Primary key
2	class name	VARCHAR(50)	class name
3	Level	VARCHAR(20)	Example: X, XI, XII

2.2.3. The Subject Table

The subject table serves as a dedicated repository for storing all data related to the subjects offered within the system. This table includes several key fields that ensure each subject is uniquely identifiable and properly documented. These fields consist of the subject ID, subject name, and subject description. Together, these attributes facilitate efficient data management and retrieval, enabling the system to accurately represent each subject's details. The complete structure of the subject table is presented in Table 3 below.

Table 3. The Subject Table

No	Field	Type	Description
1	subject id (PK)	INT	Primary key
2	subject name	VARCHAR(50)	subject name
3	Subject description	Text	Optional

2.2.4. The Student Table

The student table is designed to store comprehensive data related to all students registered in the system. This table contains several essential fields that support accurate identification and classification of each student. These fields include the student ID, the National Student Identification Number (NIS), and the class ID, which links each student to their respective class. The complete structure and details of the student table are presented in Table 4 below.

Table 4. The Student Table

No	Field	Type	Description
1	student id (PK)	INT	Primary key
2	NIS	VARCHAR(50)	Number Identity Students
3	class id	Text	Class id

2.2.5. The Teacher Table

The teacher table functions as a central repository for storing all data related to the teachers within the system. This table comprises several key fields that enable precise identification and classification of each teacher. These fields include the teacher ID, the National Teacher Identification Number (NIT), and the teacher's area of specialization. A complete overview of the structure and attributes contained in the teacher table is presented in Table 5 below.

Table 5. The Teacher Table

No	Field	Type	Description
1	teacher id (PK)	INT	Primary key
2	NIT	VARCHAR(50)	Number Identity Teacher
3	Specialist	Text	optional

2.2.6. The Class Placement Table

The class placement table serves as a dedicated repository for managing all data related to class assignments within the system. This table includes several essential fields that facilitate the organization and tracking of teaching activities. These fields consist of the placement ID, teacher ID, class ID, and subject ID, each of which plays a specific role in linking teachers to the classes and subjects they are responsible for. A detailed structure of the class placement table is presented in Table 6 below.

Table 6. Class Placement Table

No	Field	Type	Description
1	placement id (PK)	INT	Primary key
2	Teacher id	INT	Teacher id
3	Class id	INT	Class id
4	Subject id	INT	Subject id

2.2.7. The Schedule Table

The schedule table is designed to store all data related to class schedules within the system. This table includes a series of key fields that support the detailed arrangement and management of teaching activities. These fields consist of the schedule ID, class ID, subject ID, teacher ID, day, start time, and end time. Together, these attributes enable the system to accurately represent when and by whom each subject is taught. The complete structure of the schedule table is presented in Table 7 below.

Table 7. Schedule Table

No	Field	Type	Description
1	schedule id (PK)	INT	Primary key
2	class id	INT	class id
3	Subject id	INT	subject id
4	teacher id	INT	teacher id
5	Day	Enum	Monday to Saturday except Sunday (holiday)
6	Start time	Time	Start time study
7	End time	Time	End time study

The resulting database supports all the functions in the use case diagram:

1. Admins can manage classes, subjects, students, schedules, and class placements.
2. Teachers and students can log in and view their respective schedules.
3. Schedules are linked to classes, subjects, and teachers.

2.3. Code Phase

During this phase, developers write the actual program code according to the design specifications. Each module is built individually, following coding standards to ensure readability and maintainability. Once the design is complete, the system is implemented by starting the back-end development using PHP to manage processes such as data storage, input validation, and business logic processing.

The front-end of the system is built using HTML and CSS to ensure an attractive and user-friendly user interface, and Bootstrap supports responsive design. All 20 components are then integrated with a MySQL database so that data can be stored and accessed as needed.

2.4. Integration Phase

After each individual module has been successfully developed and tested, the next step is to combine them into a unified and fully functioning system. In this phase, the primary goal is to ensure that all modules interact seamlessly and support each other's functionality. Integration activities verify that data can flow smoothly between components, interfaces are compatible, and no conflicts arise when the modules operate together.

At this stage, system-level behaviors begin to emerge, allowing developers to observe how the overall application performs as a whole and identify any issues that only appear when components are integrated. This phase is crucial for ensuring the system's stability, coherence, and readiness for subsequent testing.

2.5. Test Phase

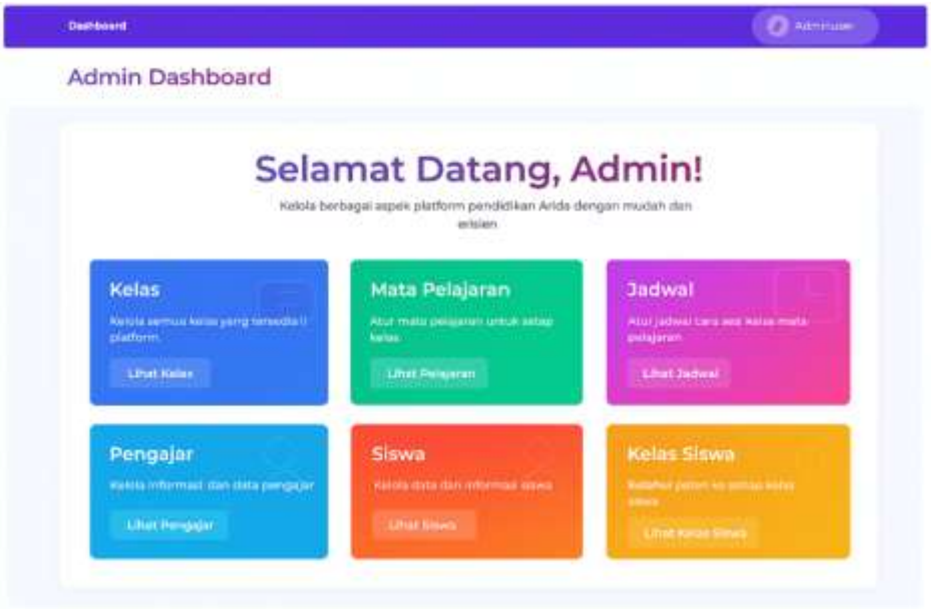
In the testing phase, the entire system is thoroughly examined to ensure that it is free from errors and fully meets the requirements and specifications defined earlier. This phase includes several types of testing, such as unit testing to verify the logic of individual modules, integration testing to ensure that different modules work together correctly, system testing to evaluate the performance and behavior of the system as a whole, and user acceptance testing (UAT), where end-users assess whether the system meets their expectations and is ready for deployment.

This phase plays a crucial role in ensuring that the system meets the expected standards of quality, stability, security, and functionality prior to deployment. Through a structured and comprehensive testing process, various components of the system are thoroughly examined to detect defects, inconsistencies, or performance issues that may affect usability. Identifying and resolving these potential problems at an early stage helps prevent failures during actual operation, thereby ensuring that the system is reliable, robust, and fully prepared for real-world implementation.

3. Result

The implementation stage produced a fully functional prototype of the LKP Aneka Prima Kayuagung information system. This prototype integrates several essential features designed to support academic and administrative activities, including modules for kelas (class), mata pelajaran (subject), jadwal (schedule), pengajar (teacher), siswa (student), and kelas siswa (student class). Each module is connected to ensure seamless data flow and consistent information management across the system. These features are illustrated in Figure 3.

Figure 3. Admin Dashboard


Dashboard
Admin User
Admin Dashboard
Selamat Datang, Admin!
Kelola berbagai aspek platform pendidikan Anda dengan mudah dan efisien
Kelas
Kelola semua kelas yang tersedia di platform.
Lihat Kelas
Mata Pelajaran
Atur mata pelajaran untuk setiap kelas.
Lihat Pelajaran
Jadwal
Lihat Jadwal
Pengajar
Kelola informasi dan data pengajar
Lihat Pengajar
Siswa
Kelola data dan informasi siswa
Lihat Siswa
Kelas Siswa
Lihat Kelas Siswa

1. Kelas (class): This feature is designed to manage all available classes, allowing administrators to add, update, and organize class information efficiently. Class page interface consist of:
   - Class list have title to class management, button to class add, and class search to find no, class name, level, student sum, and action button (edit, delete, class detail).
   - Add and edit class form have fields: class name, level (X, XI, XII), homeroom teacher (optional), and button (save and cancel).
   - Class detail have fields: class name, level (X, X1, XII), student list, subject list, and teacher list.

2. Mata Pelajaran (subject): This feature is used to organize and manage the subjects assigned to each class, enabling administrators to add, modify, and structure subject information in an efficient and systematic manner. Subject page interface consist of:
   - Subject list have element: title to subject management, button to add subject, search, and table (no, subject name, description and action).
   - Subject add and edit form have fields: subject name, description, button add and cancel.

3. Jadwal (schedule): This feature is used to manage and organize the schedule for each subject, allowing administrators to define class times, adjust session allocations, and ensure that learning activities are arranged in a structured and efficient manner. Schedule page interface consist of:
   - Schedule list have element: title (schedule management), dropdown select class, button add schedule, and table (no, day, start time, end time, subject, teacher, and action).

   - Schedule add form have fields: select class, select day (Monday until Saturday), start time, end time, subject, teacher, and button save.

4. Pengajar (teacher): This feature is used to manage and maintain teacher information, enabling administrators to store, update, and organize data related to teaching personnel in an accurate and structured manner. Teacher page interface consist of:
   - Teacher list have elements: title (teacher management), button add teacher, search, table (no, name, NIT, specialist, and action).
   - Teacher add and edit form have fields: name, NIT, email, specialist or subject and button save.
   - Teacher detail have information name, NIT, teaching schedule, and class taught.

5. Siswa (student): This feature is used to manage and maintain student information, allowing administrators to record, update, and organize student data systematically to support academic and administrative processes. Student page interface consist of:
   - Student list have elements: title (student management), button add student, search, and table (no, name, NIS, class, and action).
   - Student add and edit form have fields: name, NIS, email, select class and button save.
   - Student detail have information student biodata, currect class, class history (optional), and student schedule.

6. Kelas Siswa (student class): This feature is used to manage the assignment of students to their respective classes, ensuring that teaching sessions are organized effectively and that each student is placed in the appropriate learning group. Student class consist of:
   - Student class list have elements: title (student class management), dropdown select class, and table (no, student name, NIS, class and action).
   - Class placement and move form have fields: student name, current class, moto to class (dropdown), and button to student move.

Testing was conducted through Black Box Testing, confirming that all major functions (data input, retrieval, and display) worked as intended without critical errors. Figure 3 the final front-end interface. The evaluation phase also involved usability testing with five administrative staff members. Feedback indicated that the new system improved task completion time by approximately 40% compared to manual recording, reduced data redundancy, and minimized documentation errors.

4. Discussion

The results of the study indicate that the implementation of a structured Waterfall methodology provides a clear, sequential, and predictable workflow that is highly suitable for small-to-medium-scale information system development projects. Each phase in the Waterfall model from requirements analysis to design, implementation, testing, and deployment produces well-defined outputs that guide the subsequent stage, thereby reducing ambiguity and ensuring that development progresses in an orderly manner. Compared to more flexible agile approaches, the Waterfall model offers the advantage of comprehensive documentation at every phase.

This characteristic proved particularly valuable in the context of LKP Aneka Prima Kayuagung, where the development team had limited technical expertise and required a methodical framework to maintain clarity and consistency throughout the project. On the front-end side, the system's interface was designed with an emphasis on simplicity, usability, and responsive interaction. These principles align with Nielsen's usability heuristics, as highlighted by (Ginting et al., 2021), ensuring that the interface remains intuitive and accessible, especially for non-technical users who form the majority of the institution's stakeholders. Furthermore, the integration of Bootstrap components contributed significantly to enhancing design consistency across pages and streamlining the front-end development process. By leveraging Bootstrap's predefined styles and responsive utilities, the development team was able to reduce both design complexity and implementation time while maintaining a professional and cohesive user experience.

The results of this study align with previous research, which highlights that a well-designed and user-centered web interface plays a crucial role in enhancing both user satisfaction and operational efficiency (Wijayanto et al., 2020). A clear, intuitive interface not only facilitates smoother interaction but also minimizes the cognitive load on users, enabling them to complete tasks more quickly and with fewer errors. This finding conducted research by (Yonata et al., 2024) that reinforces the importance of thoughtful interface design as a foundational component of effective information system development.

Moreover, the outcomes of this project demonstrate that non-formal educational institutions, such as LKP Aneka Prima Kayuagung, can significantly benefit from the adoption of structured digital systems. Implementing such systems contributes to improved transparency, more efficient administrative workflows, and better data management processes, even in environments where technological resources and digital literacy may be limited, this is the same as research conducted by (Jimoh et al., 2024). Research in education institutions shows that the use of digital technology for administrative functions significantly improves the speed and quality of administrative services. This demonstrates that information technology supports administrative efficiency and effectiveness in higher education institutions.

The project showcases how digital transformation can support decision-making, enhance accountability, and streamline routine operations within smaller-scale educational organizations, this is the same as research conducted by (Purwani et al., 2025). This study explores the impact of implementing digital technologies in educational administration: the results show that digital administrative transformation helps improve operational efficiency, reduce bureaucracy, and speed up services supporting the idea of more efficient workflows and leaner administration.

Despite these positive results, several limitations were identified during the development and evaluation stages. The system's scalability is currently constrained, particularly in scenarios requiring the handling of larger datasets or increased user activity. Additionally, the absence of real-time data analytics capabilities limits the institution's ability to conduct immediate monitoring and generate dynamic insights from operational data. Addressing these issues in future development phases would require adopting more modular architectural approaches and integrating API-based components to support flexible expansion, interoperability, and advanced data processing functionalities, this is the same as research conducted by (Zhu et al., 2019). (Zhu et al., 2019), offers a layered architecture for big data processing (ingest → store → process → analyze → present) useful when you add advanced data processing capabilities to a modular education system.

During the implementation phase, several challenges emerged across different stages of the system development process. One of the most significant obstacles involved gaining a comprehensive understanding of the specific operational needs of LKP Aneka Prima Kayu Agung during the initial requirement-gathering stage. Since the institution's manual procedures had never been formally documented, the development team faced difficulties in accurately capturing the workflow and translating it into system requirements. This situation necessitated multiple rounds of in-depth interviews, field observations, and cross-verification with staff members. As a result, the requirements document underwent several iterations to ensure alignment with actual practices and institutional goals, this is the same as research conducted by (Barcelos & Penteado, 2017).

(Barcelos & Penteado, 2017) demonstrates that using requirements patterns as an initial foundation enables analysts to produce more comprehensive, structured, and internally consistent requirements documents. By relying on established patterns, the specification process becomes significantly more efficient, particularly when new projects exhibit similarities to previously developed systems. This approach not only streamlines the identification of functional and non-functional requirements but also enhances clarity, reduces ambiguity, and supports better communication among stakeholders. Furthermore, leveraging reusable patterns makes the requirements development process more systematic and repeatable, allowing iterative refinement and ensuring that the final documentation aligns more closely with project objectives and real-world user needs.

In addition to conceptual challenges, technical issues also arose during the implementation phase. The integration of PHP with MySQL proved particularly demanding, especially in maintaining proper data validation and ensuring smooth communication between the application layer and the database, this is the same as research conducted by (Baah & Kabari, 2015). Several errors were encountered as inconsistencies in data formatting and incomplete input handling triggered unexpected system behavior. Addressing these issues required extensive debugging, refinement of validation rules, and adjustments to both front-end and back-end code. Although time-consuming, these efforts were essential to ensure system reliability, data integrity, and overall functional correctness. Bug detection should incorporate both static and dynamic analysis techniques that evaluate the interaction between the front-end and back-end components to identify inconsistencies across system layers. This is essential because addressing issues on only one side—whether in the interface or the server logic—often fails to resolve the underlying problem and may even introduce new errors. A comprehensive, cross-layer debugging approach ensures that data flow, validation rules, and system responses are aligned, ultimately improving system stability, functionality, and overall reliability.

5. Conclusion

This study successfully developed a front-end information system for LKP Aneka Prima Kayuagung by employing the Waterfall method as the primary software development framework. The structured and sequential nature of the Waterfall model enabled the development team to proceed through each phase requirements analysis, design, implementation, testing, and deployment in a systematic manner. This approach not only facilitated clearer documentation and communication but also helped reduce implementation errors by ensuring that each stage was thoroughly validated before progressing to the next.

The resulting web-based system significantly improves the institution's administrative efficiency by providing an organized and user-friendly platform for managing class data, schedules, and student information. Through its intuitive interface, staff members can perform essential tasks more quickly and with fewer errors, while the system's consistent layout and responsive design contribute to an overall positive user experience. These improvements are particularly valuable for non-formal educational institutions like LKP Aneka Prima Kayuagung, where digital literacy may vary and streamlined workflows are essential for effective daily operations.

Overall, the research underscores the importance of adopting a methodological and well-structured software engineering approach when undertaking digital transformation initiatives in educational settings. By applying a clear development model and emphasizing usability-focused design principles, institutions can achieve more reliable, maintainable, and user-centered systems. Future research may further enhance the system by incorporating mobile-based access, automated notifications, and advanced reporting and analytics features. These enhancements would not only expand the system's functionality but also increase its practicality and accessibility for a broader range of users.

6. References

Awad, J. A. R., & Rojas, R. M. (2024). Digital transformation influence on organisational resilience through organisational learning and innovation. *Journal of Innovation and Entrepreneurship*, 1–24. https://doi.org/10.1186/s13731-024-00405-4

Baah, B., & Kabari, L. (2015). An Enhanced Data Validation System for a Relational DataBase. *An Enhanced Data Validation System for a Relational DataBase*, *2*, 27–35.

Barcelos, L. V., & Penteado, R. D. (2017). Elaboration of software requirements documents by means of patterns instantiation. *Journal of Software Engineering Research and Development*, *5*(3), 1–23. https://doi.org/10.1186/s40411-017-0038-9

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (IJET)*, *2*(5).

Bing, L., & Farhana, N. (2024). Efficient Management of Student's Academic Records: Factors to Consider. *Informative Journal of Management Sciences (IJMS)*, *3*(3).

Estrada, Y., & Sobri, M. (2022). Evaluasi Sistem Informasi Manajemen Hotel Dengan Metode Unified Theory Of Acceptance And Use Of Technology (UTAUT) 2 Di Bina Darma Hotel Palembang. *Smatika Jurnal*, *12*(02), 200–209. https://doi.org/10.32664/smatika.v12i02.694

Ginting, L. M., Sianturi, G., & Panjaitan, C. V. (2021). Perbandingan Metode Evaluasi Usability Antara Heuristic Evaluation dan Cognitive Walkthrough. *Jurnal Manajemen Informatika (JAMIKA)*, *11*(2), 146–157. https://doi.org/10.34010/jamika.v11i2.5480

Jimoh, Adisa, T., & Bamidele, I. (2024). Exploring the Influence of Digital Technology on Administrative Service Delivery in Tertiary Institutions : An Empirical Insight from The Federal Polytechnic , Ilaro. *International Journal of Research and Innovation in Social Science*, *VIII*(XI), 316–326. https://doi.org/10.47772/IJRISS

Kayande, P. S., & Phadnis, S. (2024). *" A Study on the Software Development Life Cycle – Waterfall Model " at a Aviation Management Consultant*. *8339*(1), 1–20. https://doi.org/10.54105/ijsepm.A9019.014124

Mayasari, A. (2023). Peran Transformasi Digital dan Inovasi Terhadap Kinerja Organisasi Pada Sekolah Yayasan Marsudirini Cabang Yogyakarta. *Journal on Education*, *05*(04), 16659–16672.

Mazadu, U. H., Ibrahim, M. M., Ibrahim, A. S., & Mansur, M. S. (2022). Social Sciences & Humanities Open Examining the instructor management benefits of student information system : An empirical investigation. *Social Sciences & Humanities Open*, *6*(1). https://doi.org/10.1016/j.ssaho.2022.100322

Purwani, R., Fathoni, A., Sarilan, & Siswanto, H. (2025). Digital, Transformasi Administrasi Pendidikan untuk Mengoptimalkan Efisiensi dan Kualitas Layanan Pendidikan pada Era Digital. *Jurnal Keilmuan Dan Keislaman*, *4*(1), 53–58. https://doi.org/10.23917/jkk.v4i1.261

Sholeh, B., Soffiatun, S., & Afriliani, F. (2023). *Peran Lembaga Kursus dan Pelatihan (LKP) dalam Mempersiapkan Wirausaha*. *4*, 895–900.

Sobri, M., Indriani, P., & Marlindawati. (2017). Perancangan Sistem Informasi Perawatan dan Inventaris Menggunakan Metode Enterprise Architecture Planning Studi Kasus Universitas Bina Darma. *Seminar Nasional Teknologi Informasi Dan Komputasi*.

Solikhin, I., Sobri, M., & Saputra, R. (2018). Sistem Informasi Pendataan Pengunjung Perpustakaan. *Jurnal Ilmiah Betrik*, *9*(03), 140–151.

Starke, S., & Ludviga, I. (2024). Impeding Digital Transformation by Establishing a Continuous Process of Competence Reconfiguration: Developing a New Construct and Measurements for Sustained Learning. *Sustainability*, *16*(23). https://doi.org/10.3390/su162310218

Tjahjanto, Arista, A., & Ermatita. (2022). Application of the Waterfall Method in Information System for State-owned inventories Management Development. *Jurnal Dan Penelitian Teknik Informatika*, *6*(4), 2182–2192.

Wijayanto, E., Tavares, O. C. da C., & dos Santos, A. (2020). The Web-Based Usability Heuristic Survey Supports User Satisfaction. *Indonesian Journal of Information Systems*, *3*(1), 75–81.

Yonata, Y., Sipayung, E. M., & Theresa, N. (2024). Analisis User Interface Sistem Informasi Akademik Berbasis Mobile pada Aspek Usability (Studi Kasus : Aplikasi XYZ). *Jurnal Telematika*, *15*(1), 55–62.

Zhu, J. Y., Tang, B., & Li, V. O. K. (2019). A five-layer architecture for big data processing and analytics. *International Journal Big Data Intelligence*, *6*(1), 38–49.