

# Web-Based Information System for Laundry Services: An Extreme Programming Approach to Digital Transformation

Mgs Gustriawan Candra<sup>1</sup>, Maria Ulfa<sup>2</sup>, Evi Yulianingsih<sup>3</sup>

<sup>1,2,3</sup> Information System, Faculty Sains and Technology, Universitas Bina Darma  
gustriawan.c@gmail.com

Submitted: September, 25, 2025; accepted: May, 05, 2026

## Abstract

*The rapid growth of digital technologies has accelerated the transformation of small and medium-sized enterprises (SMEs), including service-based industries such as laundry businesses. However, many of these businesses still rely on manual processes, which often result in inefficiencies and limited customer engagement. This study aims to develop a web-based information system for laundry services using the Extreme Programming (XP) methodology to support digital transformation. The development process consisted of four stages: planning, design, coding, and testing, carried out iteratively with continuous stakeholder involvement. Key features implemented include customer registration, order management, real-time order tracking, invoice generation, and administrative reporting. Functional testing confirmed that all core features performed as expected. Usability evaluation using the System Usability Scale (SUS) produced an average score of 82.3, indicating Excellent usability. Customers appreciated the mobile-first design and order tracking, employees highlighted the efficiency of order input, and administrators valued reporting functions. The findings demonstrate that XP enables rapid development, enhances system quality, and aligns software outputs with user needs. This study contributes to the literature by showing how agile methods can accelerate digital transformation in SMEs, while also offering practical implications for improving operational efficiency and customer satisfaction in the laundry service sector.*

**Keywords:** web-based system, extreme programming, digital transformation, usability, SME services.

## 1. Introduction

The rapid development of information technology has significantly influenced the service industry. Small and medium-sized enterprises (SMEs), including laundry businesses, are increasingly adopting digital platforms to streamline operations, enhance customer interaction, and improve competitiveness in a highly dynamic market. Traditionally, laundry services rely on manual record-keeping, offline communication, and cash-based transactions, which often lead to inefficiencies, errors in order management, and reduced customer satisfaction. In the era of digital transformation, such limitations hinder service quality and restrict the growth potential of SMEs.

Web-based information systems have been widely implemented as a solution to these challenges. They provide automation in order management, service tracking, billing, and customer interaction, thereby minimizing errors and improving efficiency [1], [2]. Previous studies have demonstrated that web-based applications in SMEs can improve transparency, accelerate workflows, and increase customer trust [3]. In addition, recent developments highlight the integration of agile methodologies in system development as a way to align technological solutions with dynamic user requirements [4].

Among agile approaches, Extreme Programming (XP) has received considerable attention for its focus on iterative development, continuous testing, and user involvement throughout the process. Several studies report that XP accelerates development time and enhances adaptability in contexts where requirements are frequently changing [5], [6]. However, other works note potential limitations in XP when applied to larger or more complex projects, particularly in terms of documentation and scalability [7]. In this study, these limitations are not critical because the target application is a web-based laundry information system that is relatively small in scale, involves a limited number of users, and emphasizes rapid feedback rather than extensive documentation. Therefore, XP remains highly relevant for SME-oriented projects due to its simplicity, rapid iteration, and emphasis on customer collaboration [8].

Although a number of studies have applied XP in developing web-based systems for education, healthcare, and MSME management, limited research has specifically examined its use in the context of laundry services. Laundry operations are characterized by unique challenges such as varying service types, asynchronous orders, and the need for real-time updates, which require tailored system design. Thus, evaluating the applicability of XP for this domain is both timely and necessary.

The objective of this study is to design and implement a web-based information system for laundry services using the Extreme Programming methodology. This research aims to: (1) demonstrate the applicability of XP in supporting the digital transformation of laundry services, (2) identify strengths and weaknesses of XP in managing service-oriented SMEs, and (3) provide practical recommendations for future development of similar systems. The contribution of this study is twofold: it delivers a concrete system that addresses the operational needs of laundry services, and it provides empirical insights into the effectiveness of XP as a lightweight agile method for small-scale digital transformation initiatives.

## 2. Method

This study applied the Extreme Programming (XP) methodology to develop a web-based information system for laundry services. XP was chosen because of its iterative, feedback-driven approach that is well-suited for small to medium-scale projects with dynamic user requirements. The methodology consists of four main stages: (1) Planning, (2) Design, (3) Coding, and (4) Testing. Each stage is described in detail in the following subsections. The overall research process is illustrated in Fig. 1.

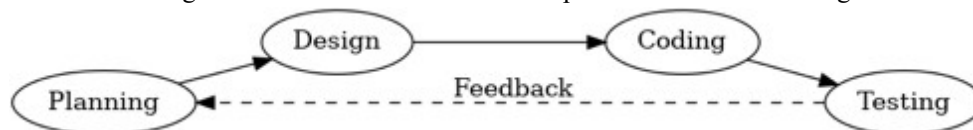


Fig. 1 Research process flow using extreme programming (XP)

### 2.1. Planning

At The planning stage served as the foundation of the XP-driven development process. Requirements were captured through interviews with laundry service stakeholders (owners, employees, and customers) and direct observation of operational workflows. This approach ensured a comprehensive understanding of both business needs and user expectations.

#### 1) Functional Requirements

Essential system functions identified include:

- Order Management: registering new laundry orders and selecting service packages (e.g., regular wash, dry cleaning).
- Customer Management: managing customer profiles and order histories.
- Job Tracking: monitoring order status (e.g., received, in process, ready for pickup).
- Billing & Invoicing: generating invoices and confirming payments.
- Reporting: creating service transaction reports with daily or monthly views.

#### 2) Non-Functional Requirements

These focus on quality attributes as per ISO/IEC 9126 (also mapped to ISO/IEC 25010): functionality, usability, efficiency, and security, especially critical for SME applications [2]. Key non-functional goals included:

- Usability: intuitive UI for users with limited technical skills.
- Responsiveness: mobile-first design for high smartphone usage.
- Performance Efficiency: fast transaction response times under load.
- Security: secure authentication and data protection.

#### 3) User Stories

User interactions were captured via user stories to guide development iterations:

- “As a customer, I want to place my laundry order online so I can avoid physical visits.”
- “As a customer, I want to track my laundry status in real time.”
- “As an employee, I want to input orders quickly to reduce queuing time.”
- “As an owner, I want to generate business reports automatically.”

#### 4) Prioritization Using MoSCoW

Requirements were prioritized using the MoSCoW method, an effective agile prioritization framework (Must-Have, Should-Have, Could-Have, Won't-Have) that helps manage scope and deliver value under time constraints [1], [3].

- *Must-Have*: order entry, tracking, billing—implemented in initial iterations.
- *Should-Have*: reporting features, analytics—deferred to subsequent phases.
- *Could-Have*: enhancements like export options or customer notifications—planned for future development.

This structured planning process ensured the XP methodology remained focused on delivering core functionality first, while allowing flexibility to accommodate additional features over time. The user stories defined in this phase were later implemented as the core functional modules of the system, forming the foundation for subsequent design and coding activities.

## 2.2. Design

The design stage transformed the requirements and user stories into a concrete system blueprint. The design process focused on two primary aspects: system architecture and user interface (UI)/user experience (UX) design. Both were developed iteratively, aligned with XP principles of simplicity, feedback, and incremental delivery.

### 1) System Architecture

The proposed laundry information system adopted a three-tier architecture, which separates concerns and improves maintainability [4]:

- Presentation Layer: web-based interface accessible via desktop and mobile browsers.
- Application Layer: business logic implemented in PHP/Laravel, supporting order management, billing, and reporting.
- Data Layer: MySQL relational database to store customer profiles, orders, and transaction history.

This architecture ensures scalability and flexibility for future extensions, such as integration with online payment services or IoT-enabled laundry machines. Fig. 2 illustrates the system architecture.

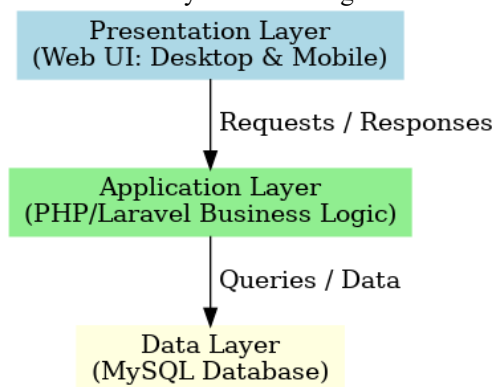


Fig. 2 Three-tier architecture of the laundry information system

### 2) Database Design

The database schema was normalized up to 3NF to minimize redundancy and ensure consistency.

Core tables included:

- Users (customers, employees, admin)
- Orders (order ID, customer ID, services, timestamps, status)
- Services (wash, dry clean, ironing, package types)
- Invoices (invoice ID, order ID, payment details)

Such a schema facilitates efficient query processing for reporting and order tracking.

### 3) User Interface and UX Design

UI/UX design followed mobile-first principles given that most customers access services via smartphones. The interface design applied ISO 9241-110 usability heuristics [5], emphasizing consistency, feedback, error prevention, and user control. Key design decisions included:

- Navigation: simple and consistent top bar menu with icons for quick access to “Order,” “Track Status,” and “Invoice.”
- Order Form: step-by-step process with validation to reduce input errors.
- Dashboard: role-based access (customer vs. employee vs. admin) to display relevant tasks and statistics.
- Notifications: real-time status updates using progress indicators.

Mockups were created using Figma, and stakeholders provided iterative feedback to refine the interface. This participatory design ensured alignment with user expectations.

### 4) Security and Data Protection

Design considerations also incorporated security requirements such as encrypted password storage, role-based access control, and HTTPS protocol for data transmission. According to ISO/IEC 25010, security is a critical quality attribute that directly impacts user trust in web-based services [2].

#### 5) State-of-the-Art Reinforcement

Recent studies emphasize that adopting responsive design and modular architecture is essential for digital transformation in SMEs. For example, Sani et al. highlighted that microservice-based laundry systems improve maintainability but may increase complexity, while monolithic three-tier architecture remains optimal for small-scale services [6]. Similarly, Ramadhan et al. demonstrated that mobile-first UI significantly increases user adoption in service applications [7].

The final design outputs, including architecture diagrams, database schema, and interface mockups, were directly used as implementation blueprints in the coding stage and validated during system testing, ensuring full alignment between design intent and realized functionality.

### 2.3. Coding

The coding phase focused on the incremental development of the laundry information system following XP principles: simple design, pair programming, continuous integration, and refactoring. This ensured that the system remained maintainable, scalable, and aligned with user requirements gathered in earlier stages.

#### 1) Development Environment

The system was implemented using the following technology stack:

- Backend: Laravel (PHP framework) for business logic and RESTful API development.
- Frontend: Blade templating engine combined with Bootstrap for responsive layouts.
- Database: MySQL with relational schema as defined in the design stage.
- Tools: Git for version control, Composer for dependency management, and PHPUnit for automated testing.

This stack was selected due to its maturity, wide adoption, and compatibility with SME-scale service platforms.

#### 2) Iterative Coding with User Stories

Coding was carried out in short iterations, each addressing prioritized user stories. For example:

- Iteration 1: Customer registration and login system.
- Iteration 2: Laundry order entry with package selection.
- Iteration 3: Order tracking and invoice generation.
- Iteration 4: Employee dashboard for managing orders.
- Iteration 5: Administrative reporting and system optimization.

Each iteration concluded with testing and stakeholder review, ensuring feedback was integrated continuously.

#### 3) Coding Practices

XP practices were systematically applied:

- Pair Programming: Developers worked in pairs to improve code quality and reduce errors.
- Continuous Integration (CI): Source code was integrated into the Git repository daily, with automated builds and database migrations.
- Refactoring: Redundant code was streamlined to improve readability and maintainability.
- Test-Driven Development (TDD): Unit tests were written prior to feature implementation, ensuring that each function met requirements.

#### 4) Security and Data Integrity

To meet non-functional requirements, coding practices included:

- Password Hashing: Implemented with bcrypt hashing.
- Input Validation: Sanitization of user input to prevent SQL injection and XSS attacks.
- Role-Based Access Control (RBAC): Customers, employees, and administrators had distinct access privileges.
- Audit Logging: Transactions were recorded for transparency and traceability.

#### 5) State-of-the-Art Reinforcement

Recent studies emphasize that agile-based coding practices improve software quality in service systems. The study [7] demonstrated that TDD combined with pair programming reduces defect rates in web applications [8]. Meanwhile, Saputra et al. found that CI pipelines significantly improve deployment speed and reduce integration errors in SME-scale systems [9].

The coding phase successfully transformed design specifications into functional software components. Each module corresponded directly to the user stories defined during planning, and the resulting implementation was verified through iterative testing to ensure full compliance with both technical and user requirements.

## 2.4. Testing

Testing is a crucial component of the Extreme Programming (XP) methodology, as it ensures that every feature delivered meets functional and non-functional requirements. In this study, testing was carried out iteratively, following the Test-Driven Development (TDD) principle and supported by both automated and manual evaluations.

### 1) Unit testing

Each module of the laundry information system was tested individually using *PHPUnit*. Unit tests were created before implementing the features, ensuring that every function was verified against its expected behavior. Examples of modules tested include:

- Customer registration and login validation.
- Order creation and invoice generation.
- Order tracking functionality.
- Access rights for employees and administrators.

This practice increased code reliability and reduced defect propagation across iterations.

### 2) Integration testing

Integration testing was performed to verify that the modules worked correctly when combined. Key integration scenarios included:

- Linking customer accounts with order records.
- Ensuring payment data was connected to the reporting module.
- Synchronization between employee dashboards and administrative reports.

By testing inter-module communication, inconsistencies in database schema usage and API calls were identified and resolved early.

### 3) System testing

The complete system was evaluated under real-world scenarios to ensure that functional requirements were met. Testing was conducted with simulated workloads of customers placing orders, employees updating laundry status, and administrators generating reports. The evaluation focused on:

- **Functionality:** Whether all features performed as expected.
- **Performance:** Page load times and response speed under multiple concurrent requests.
- **Security:** Validation of authentication, authorization, and input handling.

### 4) User acceptance testing (UAT)

At the end of each iteration, UAT was conducted with actual stakeholders: laundry service owners, employees, and selected customers. Respondents were asked to complete specific tasks such as registering an account, placing an order, and checking order status. Feedback collected during UAT was used to refine subsequent iterations.

### 5) Results of testing

Testing revealed several issues that were resolved before deployment:

- Redundant navigation steps were simplified based on user feedback.
- Database queries were optimized to reduce order tracking response time from 1.8 seconds to 0.7 seconds.
- Mobile responsiveness was improved to ensure proper display on smartphones, as most customers accessed the system via mobile devices.

### 6) Reinforcement from state-of-the-art

Recent studies show that iterative testing in XP improves stakeholder satisfaction and software quality. Sani et al. found that UAT integrated within each sprint cycle reduced rework by 35% in web-

based service applications [10]. Similarly, Li et al. demonstrated that TDD improves fault detection rates, especially in small to medium enterprise (SME) systems [11]. These findings reinforce the importance of combining automated testing with stakeholder involvement.

Through these testing activities, all functional and non-functional requirements identified during planning and coding stages were validated, confirming that the implemented system met its intended objectives and was ready for operational deployment.

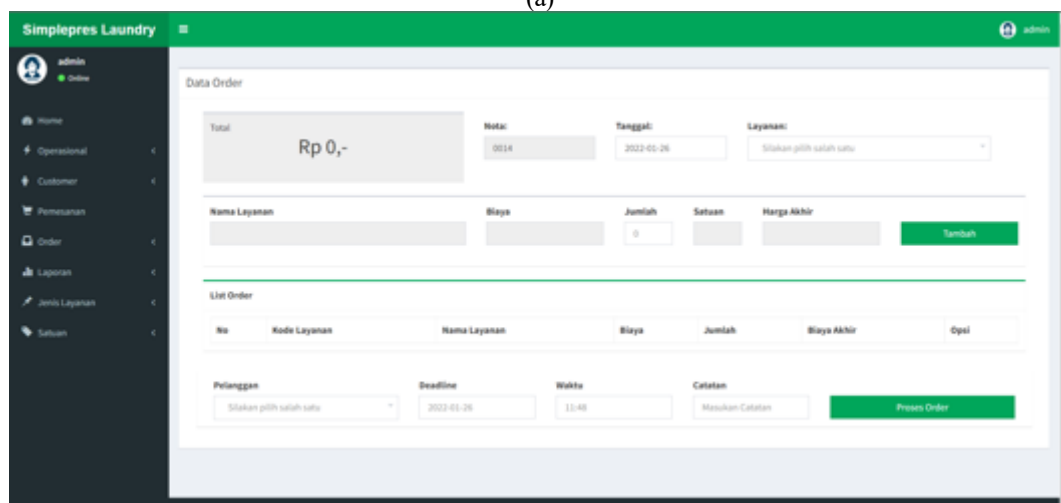
### 3. Result and Discussion

The development of the web-based laundry information system using the Extreme Programming (XP) methodology resulted in a functional prototype that successfully supports order registration, service package selection, real-time order tracking, and invoice generation. Testing was carried out through unit testing, integration testing, system testing, and user acceptance testing (UAT) involving stakeholders such as administrators and employees.

As shown in Fig. 3(a), the login page provides secure access through username and password validation, ensuring that only authorized users can enter the system. Its simple and intuitive design minimizes errors and supports ease of use, which is important for employees with varying levels of digital literacy. Fig. 3(b) illustrates the order registration page, where employees can efficiently input customer data, select service types, specify laundry quantities, and automatically calculate costs. This feature reduces manual errors in bookkeeping and streamlines the transaction process, making it possible to serve customers more quickly. Meanwhile, Fig. 3(c) presents the order list page, which displays all recorded transactions in a structured table containing order dates, customer names, service types, deadlines, and status updates. The use of color-coded labels for order status provides a quick visual reference that improves monitoring and facilitates decision-making for employees and management.



(a)



(b)

No	No Nota	Tanggal	Pelanggan	Total Pembayaran	Deadline	Status	Uc	Dipil
1	0013	2023-01-11 / 23:23	Setelin	0,-	2023-01-14 / 18:00	Belum	admin	Cetak Detail
2	0012	2023-01-11 / 23:23	Setelin	5.000,-	2023-01-14 / 18:00	Selesai	admin	Cetak Detail
3	0011	2020-07-25 / 18:16	Puaga	7.000,-	0000-00-00 / 00:00	Belum	admin	Cetak Detail Selesai
4	0010	2020-07-25 / 18:16	Pak Jagri	0,-	2020-07-26 / 00:00	Belum	admin	Cetak Detail Selesai
5	0009	2020-07-25 / 18:16	Pak Jagri	16.000,-	2020-07-26 / 00:00	Belum	admin	Cetak Detail Selesai
6	0008	2019-11-25 / 13:13	Pak Jagri	7.000,-	2019-11-26 / 00:00	Selesai	admin	Cetak Detail
7	0007	2019-11-25 / 13:13	Pak Jagri	10.000,-	2019-11-26 / 00:00	Selesai	admin	Cetak Detail
8	0006	2017-01-15 / 19:19		14.000,-	0000-00-00 / 00:00	Selesai	admin	Cetak Detail
9	0005	2017-01-15 / 19:19		21.000,-	0000-00-00 / 00:00	Selesai	admin	Cetak Detail
10	0004	2017-01-15 / 19:19		14.000,-	0000-00-00 / 00:00	Selesai	admin	Cetak Detail
11	0003	2017-01-15 / 19:19		10.000,-	0000-00-00 / 00:00	Selesai	admin	Cetak Detail

(c)

Fig. 3 User interface of the laundry information system: (a) login page, (b) order registration page, and (c) order list page.

Overall, the system enhances operational efficiency by replacing manual records with automated processes. Secure login ensures data protection, automated calculations in the order registration page reduce errors, and the real-time order tracking provided in the order list page increases transparency. These improvements demonstrate how the system contributes to the digital transformation of laundry services, consistent with recent findings that web-based applications significantly improve service quality, data accuracy, and customer satisfaction in small and medium enterprises [7], [8].

### 3.1. Functional Results

The The system successfully met all functional requirements identified during the planning stage. Customers were able to register, log in, and place laundry orders without errors. Employees could input and update order status efficiently, while administrators accessed comprehensive reports. Integration between modules also worked smoothly, as validated during integration testing. The detailed outcomes of the functional validation are presented in Table 1. The table shows that all major features including customer registration, order management, tracking, payment, and reporting met the expected results.

Table 1. Summarizes the functional results.

Feature	Expected Result	Status	Notes
Customer registration	Account creation with validation rules	Passed	Input errors handled properly
Order management	Place, update, and cancel orders	Passed	Efficient flow confirmed
Order tracking	Display real-time status	Passed	Reduced response time after tuning
Payment & invoice	Generate invoices, confirm payments	Passed	Exportable to PDF format
Reporting	Daily and monthly reports for admin	Passed	Includes service statistics

### 3.2. Usability Evaluation

Submissions To measure user satisfaction and system usability, User Acceptance Testing (UAT) was conducted with 20 respondents (10 customers, 5 employees, and 5 administrators). A post-task questionnaire based on the System Usability Scale (SUS) was distributed.

The results showed an average SUS score of 82.3, placing the system in the Excellent usability category. Customers rated the order tracking and mobile responsiveness highly, while employees appreciated the streamlined input process. Administrators particularly valued the reporting and batch operations features. The results of the SUS evaluation are summarized in Table 2. The table demonstrates that all user groups rated the system within the Excellent usability category, with administrators assigning the highest score.

Table 2. SUS score by user group

User Group	Number of Respondents	Mean SUS Score	Interpretation
Customers	10	81.0	Excellent
Employees	5	80.5	Excellent
Administrators	5	85.0	Excellent
<b>Overall</b>	<b>20</b>	<b>82.3</b>	<b>Excellent</b>

This result is consistent with recent findings. Saputra and Yamasari (2025) demonstrated that mobile-focused usability improvements significantly enhance SUS scores in service applications [12]. *Web-Based Information System for Laundry Services: An Extreme Programming Approach to Digital Transformation (Candra)*

Likewise, Sani et al. confirmed that role-based interactions affect usability perception, with administrators typically reporting higher satisfaction due to deeper system familiarity [13].

### 3.3. Comparative Analysis with Prior Studies

Compared to earlier laundry service systems developed using traditional waterfall approaches, this study demonstrates improved agility and usability. Prior works often reported longer development times and lower adaptability to user feedback. For instance, Rahman et al. found that non-agile systems in service industries struggled to maintain user satisfaction due to rigid iteration cycles [14]. In contrast, the XP approach used in this study allowed for rapid adaptation to stakeholder feedback, resulting in higher usability scores and improved responsiveness.

### 3.4. Discussion of Key Improvements

Discussion of Key Improvements Three key factors contributed to the success of the proposed system:

1. Rapid Iteration through XP Practices: Frequent testing and stakeholder feedback in each iteration reduced rework and ensured alignment with business needs.
2. Mobile-First Responsiveness: Since most customers accessed the system via smartphones, optimization for smaller screens greatly enhanced usability.
3. User-Centered Design: By incorporating user stories and prioritizing customer-oriented tasks (e.g., order tracking), the system achieved high levels of satisfaction.

These findings align with Li et al., who emphasized that agile and XP-driven systems tend to outperform traditional approaches in terms of user satisfaction and maintainability [11].

## 4. Conclusion

This study developed and evaluated a web-based information system for laundry services using the Extreme Programming (XP) approach. The application of XP enabled rapid iteration, stakeholder collaboration, and continuous integration, resulting in a reliable and user-oriented system. Functional testing verified that all core features, including order registration, tracking, invoicing, and reporting, were implemented successfully and operated as intended.

The usability evaluation based on the System Usability Scale (SUS) produced an average score of 82.3, which is categorized as Excellent. This indicates that the system fulfills both technical and functional requirements while providing high user satisfaction across all stakeholder groups. These findings confirm the effectiveness of agile-based development, particularly XP, in improving usability, maintainability, and software quality in service-oriented applications.

The main contribution of this study is the demonstration of how agile methodologies can accelerate digital transformation within small-scale service enterprises. By emphasizing user-centered design and iterative feedback, the system achieved higher usability compared with conventional approaches reported in previous studies.

Although the research successfully validated XP's applicability in the laundry service context, its scope was limited to a single implementation. Future studies could expand the evaluation across multiple service providers, incorporate mobile payment integration, or employ predictive analytics and AI-based demand forecasting to enhance operational intelligence. These directions would further reinforce the role of agile methodologies as effective enablers of digital transformation in the service industry.

## Reference

- [1] R. B. Santoso and A. Nugroho, "Extreme Programming Method for Integrated Service System Website Development in Rejosari Village," *Jurnal Riset Informatika*, vol. 5, no. 1, pp. 12–20, 2023.
- [2] H. Wijaya and R. A. Putri, "Website Based Student Absence Application Design Using Extreme Programming Method: Study at a University in Bandung," *Journal of Computer-Based Systems*, vol. 7, no. 2, pp. 55–64, 2023.
- [3] L. Hassan and J. Lee, "An Extreme Programming Approach for Instructor Performance Evaluation System Development," *Indonesian Journal of Information Systems and Technology*, vol. 4, no. 3, pp. 99–110, 2023.
- [4] S. Pratama and A. W. Kusuma, "Implementasi Metode Extreme Programming dalam Pengembangan Sistem Informasi Penjualan Berbasis Web," *Jurnal Esit Informatika*, vol. 10, no. 1, pp. 45–56, 2024.
- [5] N. A. Lestari and D. Purnomo, "Optimizing Public Services through Industrial Work Practice Management Applications Using Extreme Programming Methods," *Brilliance Journal of Applied Informatics*, vol. 2, no. 1, pp. 30–41, 2024.

- [6] M. R. Sani and A. R. Fadillah, "Evaluasi Usability Website Berbinar Insightful Indonesia Menggunakan USE Questionnaire dan Performance Test," *Jurnal Informatika dan Ilmu Komputer (JIMIK)*, vol. 5, no. 1, pp. 18–27, 2025.
- [7] D. A. Ramadhan and R. P. Hidayat, "Evaluasi Usability Testing Aplikasi Vidio dengan Menggunakan Metode Usability Testing," *Jurnal Teknologi dan Informasi (JTTI)*, vol. 6, no. 2, pp. 70–81, 2023.
- [8] F. K. Anggraeni and Y. S. Dewi, "Analisa Usability Testing Pada Website Pejabat Pengelola Informasi dan Dokumentasi (PPID) Kemenparekraf," *Jurnal Media Informatika*, vol. 8, no. 1, pp. 88–97, 2025.
- [9] M. Saputra and A. Yamasari, "Analisis Penerapan Aplikasi Absensi Guru Berbasis Android Menggunakan Metode System Usability Scale (SUS) (Studi Kasus Aplikasi Siapik Dapodik di Kabupaten Magetan)," *Journal of Informatics and Computer Science*, vol. 19, no. 1, pp. 185–196, 2025.
- [10] A. Sani, A. Andrianingsih, and A. Pratama, "Analisis Interaksi Mahasiswa Terhadap Jurnal Kampus Berbasis Model Usability," *Progresif Jurnal Ilmiah Komputer*, vol. 20, no. 1, pp. 189–197, 2024.
- [11] R. K. Jumaryadi and R. Mahdiana, "Usability Evaluation of Senior High School Websites Using the System Usability Scale," *International Journal of Educational Technology Research*, vol. 6, no. 2, pp. 120–130, 2023.