Web-Based Boarding House Management Information System with Payment Gateway Integration

Ketut Arya Wisnu Wardana[1*], Tri Widodo[2]

[1,2] Department of Informatics, Faculty of Science and Technology,
Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia
*Corresponding Author:
Email:ayra6793@gmail.com

Abstract.

Manual management at GreenKost in Sleman faces challenges in unstructured recording of room, tenant, and financial transaction data, along with slow and error-prone administrative processes. This study aims to develop a web-based boarding house management information system with Model-View-Controller (MVC) architecture using Laravel framework to optimize business processes and improve tenant services. The research methodology includes direct observation on February 17, 2025, and interviews with the boarding house owner on February 25, 2025, to identify system requirements. The system was developed using Laravel 12.x with MySQL database and Eloquent ORM, integrated with a payment gateway to support QRIS and E-Wallet payments. Main features include user management, room management with real-time status, online reservation and booking, integrated payments, admin dashboard reports, announcement management, and repair request services. Testing using black-box testing method shows all features function as expected under both normal and abnormal conditions. The research results demonstrate that an integrated digital system can automate payment verification, improve administrative efficiency, and provide ease of information access and real-time transactions for tenants.

Keywords: *Blackbox Testing; Boarding House; Laravel; Management Information System; MVC and Payment Gateway.*

I. INTRODUCTION

According to the Jakarta Housing Department, a boarding house (kost) is a house that is used as a source of financial income by its owner, either partially or entirely, by accepting payment from residents for a minimum period of one month [1]. The Association of Indonesian Internet Service Providers (APJII) recorded that the number of people connected to the internet in 2021-2022 reached 210,026,769 people out of Indonesia's total population, indicating that the internet has become a primary necessity for Indonesian society [2]. This data indicates significant potential for digitalization of services, including in the boarding house property management sector. Efficient boarding house property management is one of the main challenges for boarding house owners. This challenge is primarily related to efforts to maintain tenant data accuracy, manage administrative processes regularly, and ensure that every transaction and operational activity runs smoothly. The lack of an integrated management system often leads to data errors and delays in administrative processes [3]. Similar problems are also experienced by GreenKost in Sleman, Special Region of Yogyakarta, where the recording of room data, tenants, and financial transactions is still done manually without a clear system structure. The abundance of available rental options often creates confusion for tenants in determining the right choice. This condition is further exacerbated when the information presented is inaccurate, unverified, or not updated regularly. As a result, tenants have difficulty obtaining a clear picture of availability, facilities, and offered prices, making the booking process less efficient [4].

The problems faced by boarding house service owners include difficulties in marketing boarding house services and providing information to prospective boarding house residents about room availability [5]. The development of information technology using websites is expected to assist the registration process for prospective boarding house residents from outside the area, enabling them to register without having to come to the boarding house location [6][7][8]. Dynamic websites allow both owners and users to update and interact with content directly, so the displayed information can change according to activity or user input, unlike static websites that can only be updated by administrators [9]. Several previous studies have developed web-based boarding house management systems but with limitations. Hasti and Zaelani [1] and Mike et al. [2] developed systems using Laravel framework without payment gateway integration, requiring manual verification. Dhananjaya [3] focused on promotional aspects rather than administrative automation.

Harefa and Harianja [4] and Satria et al. [5] built information systems emphasizing location search and facility browsing but lacked online booking and digital payment features.These studies show that the development of boarding house management information systems has been widely conducted, but there is still a significant gap regarding payment verification automation and robust payment gateway integration. Most developed systems still require manual payment confirmation by admins, which has the potential to cause administrative delays and errors.

This study develops a web-based Boarding House Management Information System integrated with Payment Gateway to automate administrative processes, manage room status in real-time, and provide secure payment services [10]. The system is designed using Waterfall method with Laravel framework implementing MVC (Model-View-Controller) architecture for organized code management. MySQL is used as database system due to its stable performance and seamless Laravel integration [11].Black-box testing focuses on software inputs and outputs to ensure requirement compliance without examining internal mechanisms [12]. MVC design pattern separates application logic, display, and controller as demonstrated in previous system implementations [13].Use Case Diagram visualizes user-system interactions and functional relationships [14], while Activity Diagram describes workflow sequences and process flows [15]. Waterfall method is applied to develop features including room booking, bill management, payment validation, and transaction reporting, improving efficiency and information accessibility [16].Payment gateway selection involves critical factors including transaction costs, security levels, integration ease, customer support, and system reliability [17]. Midtrans was chosen as the payment gateway supporting QRIS, e-wallet, debit cards, and credit cards with features integrated for secure online transactions [18].This study aims to develop a web-based Boarding House Management Information System with MVC architecture using Laravel framework integrated with payment gateway to automate payment verification, improve administrative efficiency, and provide real-time transaction access for tenants. The system addresses manual system limitations and overcomes previous research gaps requiring manual payment confirmation, improving boarding house management service quality at GreenKost Sleman, Yogyakarta.

II. METHODS

Framework Design

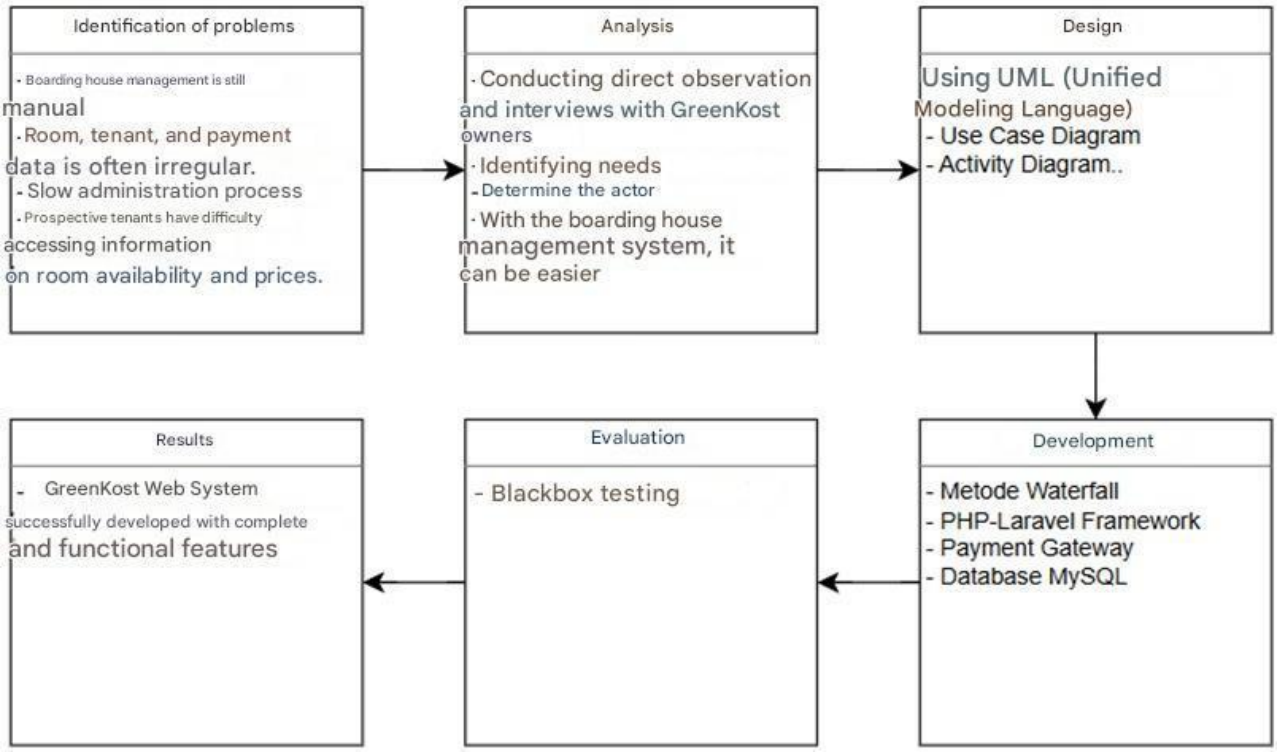

Fig 1. Framework Design

This research uses a systematic framework with six stages to develop a web-based boarding house management system at GreenKost Sleman. Problem Identification identifies issues through direct observation including unstructured data recording and payment delays. Analysis gathers requirements via observation, interviews, and questionnaires. Design creates system architecture using UML diagrams [14][15]. Development implements code using Laravel and MySQL [11]. Implementation and Evaluation tests the system using black-box testing [12]. Maintenance ensures regular upkeep and feature development.

Research Stages

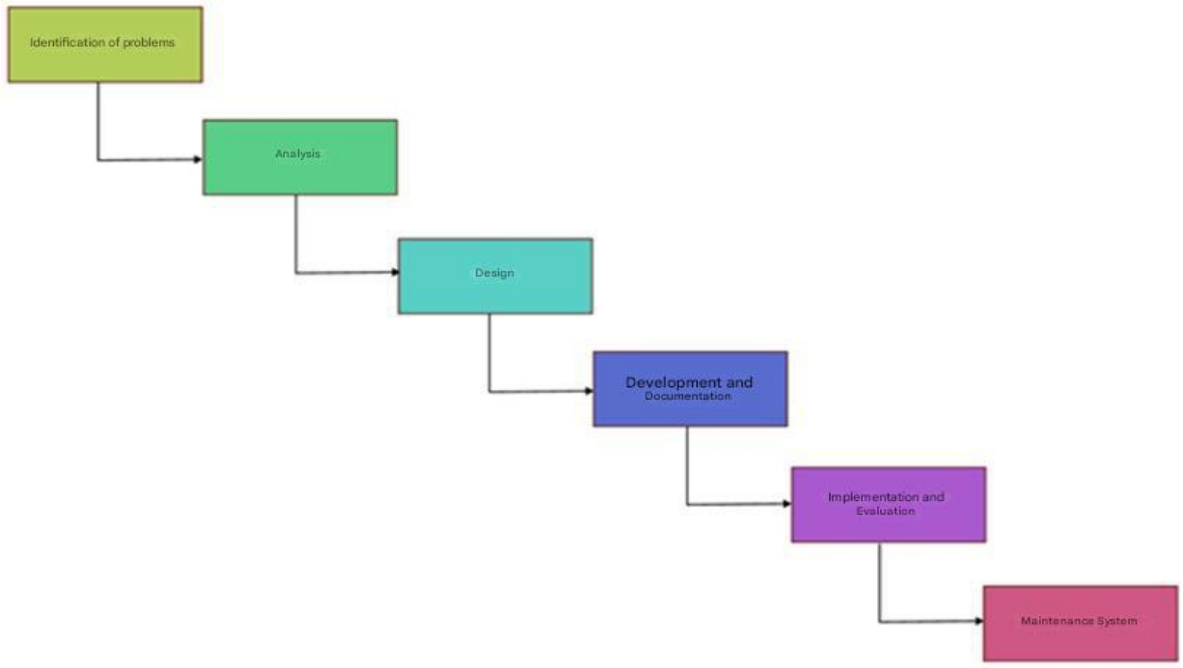

Fig 2. Research Stages

The research stages follow a systematic approach from planning to maintenance, encompassing seven phases: Problem Identification, Information Requirements, System Requirements Analysis, Design, Development and Documentation, Implementation and Evaluation, and System Maintenance as illustrated in Figure 2 [19].

1. Requirement (Requirements Analysis)

Requirements analysis involves collecting and analyzing user needs to understand system functionality and specifications [20]. Data collection uses observation method for natural data acquisition without intervention [21]. and interview method through face-to-face question-and-answer sessions with boarding house owners [22].

2. Design (System Design)

System design uses Unified Modeling Language (UML) to visualize system structure and interactions [14][15]. Use Case Diagram defines three actors: Admin manages user data, rooms, reservations, payment verification, and reports; Tenants register, browse rooms, book, pay online, and submit repair requests; Owner monitors occupancy, transactions, and announcements. Activity Diagram describes critical workflows including room booking (login, search, selection, availability check, form filling, payment, confirmation) and payment gateway flow (QRIS/E-Wallet selection, request generation, QR display, validation, status update, notification).

3. Implementation

The system is built using Laravel 12.x with PHP implementing MVC architecture separating business logic, display, and controller. MySQL database is used under GNU General Public License [23]. Frontend uses Blade templating with Bootstrap styling and JavaScript interactivity. Midtrans payment gateway integration supports QRIS and E-Wallet with API credentials, webhook handlers for automatic status updates, and error handling with retry mechanism [24]. Security implements Laravel Sanctum authentication, bcrypt password hashing, CSRF protection, and HTTPS enforcement.

4. Verification (Testing)

Testing uses Black-Box Testing method focusing on system functionality without examining internal code. Testing includes positive scenarios with valid input and negative scenarios with invalid input and boundary values. Critical features tested include login authentication, room management, concurrent booking, payment gateway with timeout handling, dashboard reporting, and repair requests. Results are documented in Test Case Matrix with test scenarios, expected output, actual output, and Pass/Fail status.

5. Maintenance

System maintenance includes corrective maintenance (bug fixing), adaptive maintenance (framework and API updates), perfective maintenance (new features from user feedback), and preventive maintenance (database backup, security audit, performance monitoring). Early risk identification prevents maintenance burden disruption [25]. Monitoring implementation uses Laravel Telescope for debugging and Sentry for error tracking ensuring long-term stability.

III. RESULT AND DISCUSSION

A. System Design Results

1. Use Case Diagram

Figure 3 illustrates the Use Case Diagram with two main actors: Tenant and Admin. Tenants access eight functions including View Room, Room Booking with payment dependency, Repair Service, Transaction History, Announcements, Registration, Notifications, and Room Management. Admins control eleven functions including room CRUD operations, tenant administration, system analytics, maintenance management, and announcements with clear role separation [14].

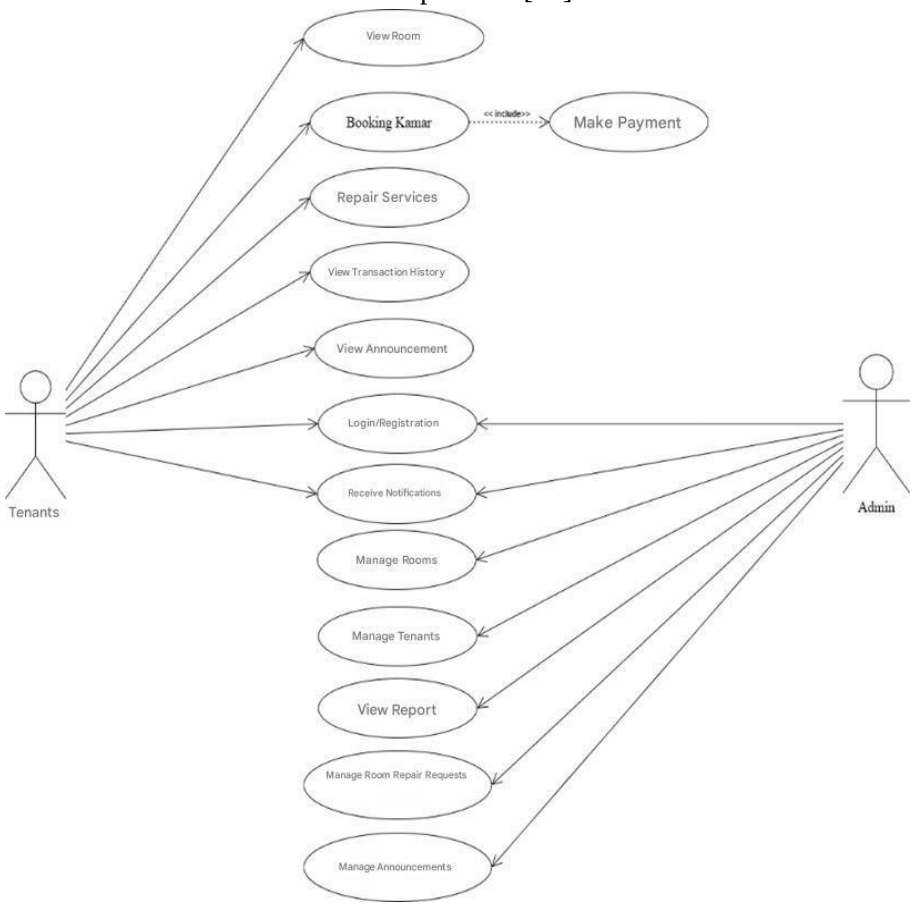

Fig 3. Use Case Diagram Showing Interaction Between Penyewa And Admin Actors

2. Activity Diagram

Figure 4 presents the Tenant Activity Diagram beginning with system access and login validation. The dashboard offers two pathways: viewing available rooms or proceeding to room booking. After room selection, the workflow progresses to payment via payment gateway integration, followed by admin validation.

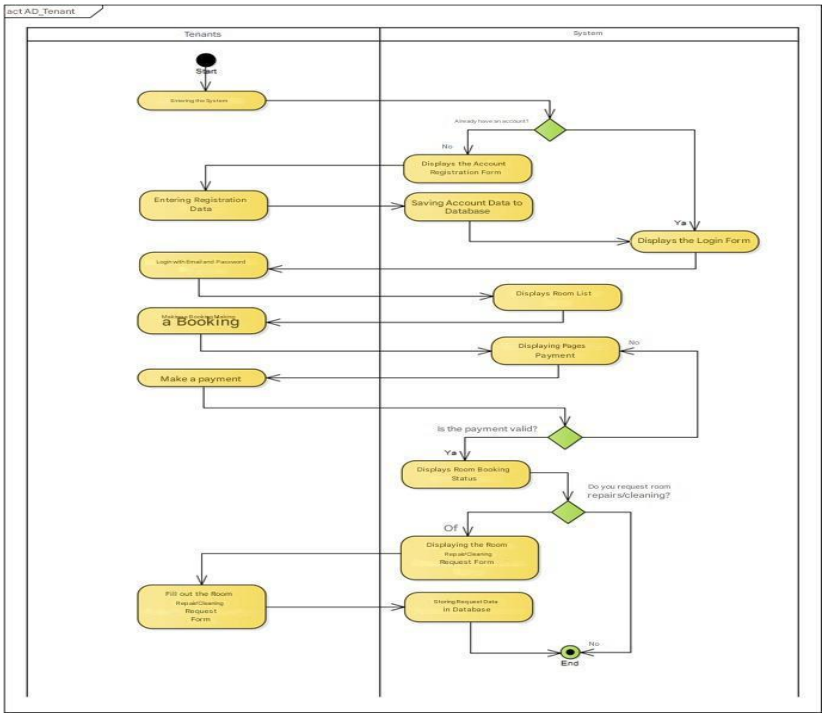

Fig 4. Activity Diagram Penyewa

Figure 4 depicts the Admin Activity Diagram demonstrating comprehensive management workflow. After login, the dashboard presents four operational paths: tenant data management with CRUD operations, room data management including pricing and availability updates, transaction verification for payment confirmation review, and announcement management. Transaction verification triggers automatic notification upon approval or loops back for retry. The workflow terminates upon logout [15].

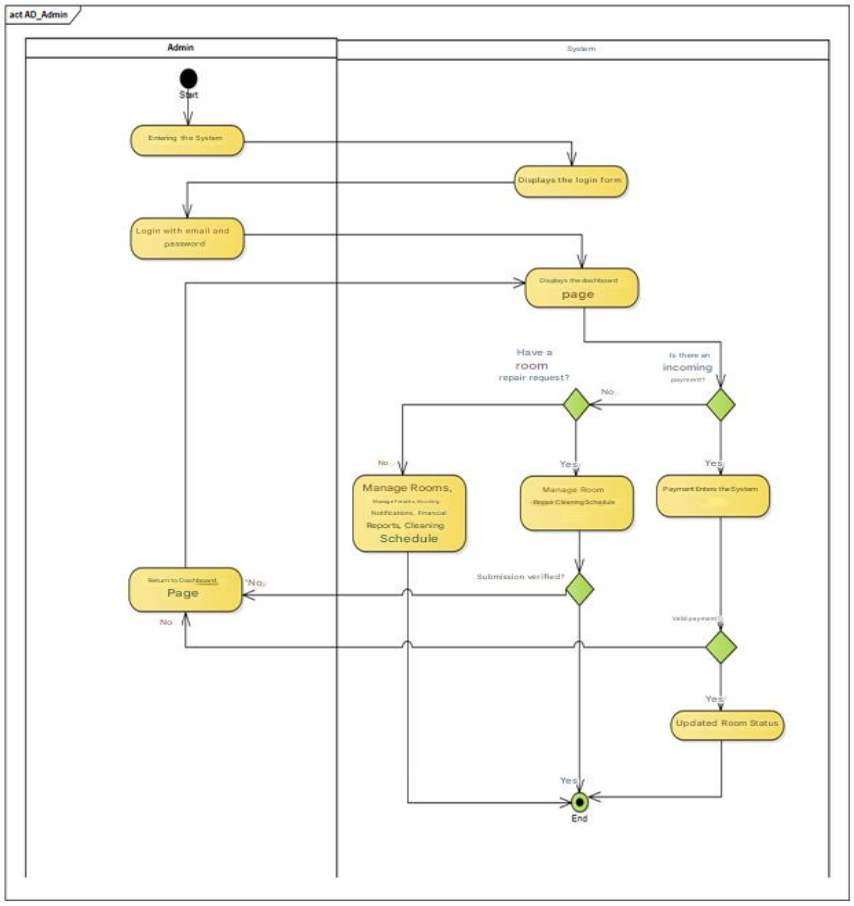

Fig 5. Activity Diagram Pengelola

B. System Implementation and User Interface Display

1. Database Implementation

The system database was implemented using MySQL with normalized table structure following Third Normal Form (3NF) principles to ensure data integrity and optimize query performance. Figure 6 presents the Entity Relationship Diagram (ERD) showing fourteen interconnected tables supporting comprehensive boarding house operations.

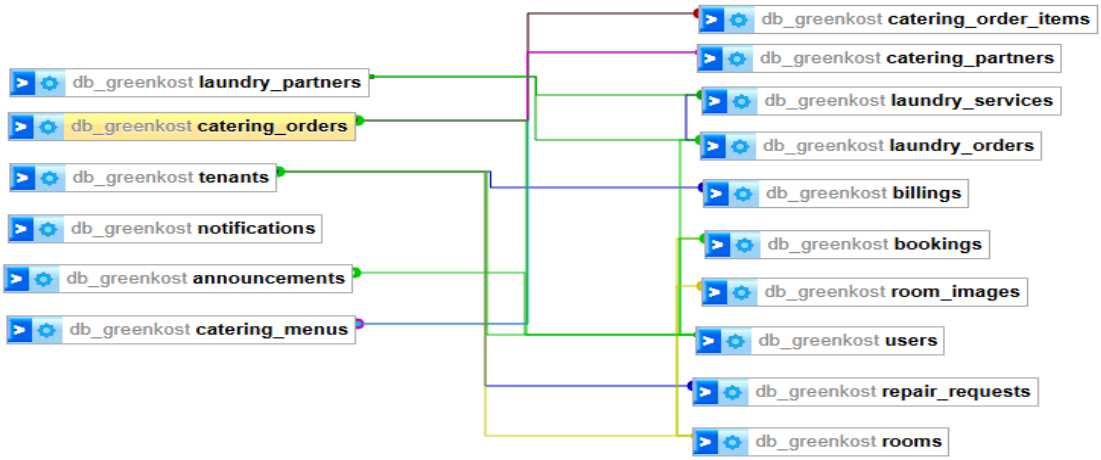

Fig 6. Database Implementation With Fourteen Tables

Core entity tables include users (role-based access control), rooms (accommodation details with room_number, type, capacity, facilities, price, status), and bookings (reservations with payment and booking status tracking). Supporting tables implement catering services (catering_orders, catering_menus), laundry services (laundry_orders, laundry_services), notifications, announcements, and repair_requests for maintenance workflows. Foreign key constraints ensure referential integrity while indexes on frequently queried columns (user_id, room_id, status fields) optimize performance. The modular structure supports scalability and future extensions using InnoDB storage engine with ACID compliance [11][21].

2. Frontend Implementation

The frontend implementation leverages Bootstrap 5 framework to create responsive user interfaces accessible across desktop, tablet, and mobile devices. Figure 7 shows the landing page with hero section displaying "Temukan Kamar Ideal Anda di GreenKost" headline, call-to-action button, and navigation links. The value proposition area features three benefit cards highlighting strategic location, security, and complete facilities.

Fig 7. Landing Page Interface Displaying Hero Section With Prominent Call-To-Action

Figure 8 presents the admin dashboard with personalized greeting, notification system, and four KPI cards displaying available rooms, active tenants, unpaid bills, and repair requests using color-coded visual distinction. The "Akses Cepat Menu" section provides six icon-based navigation cards for room management, tenant administration, billing operations, maintenance workflow, reporting, and announcements. The persistent left sidebar maintains contextual awareness throughout administrator workflows.

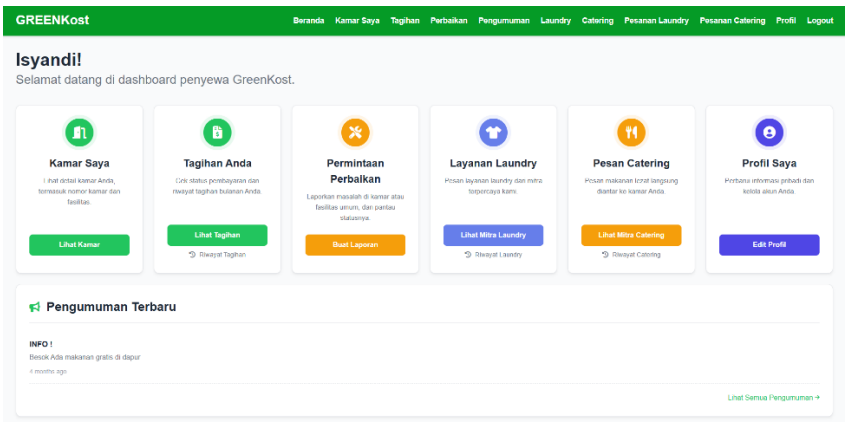

Fig 8. Admin Dashboard Displaying Key Performance Indicators And Modular Menu System

Figure 9 depicts the tenant dashboard with six service cards: room details, billing status, repair requests, laundry orders, catering services, and profile management. Each card combines colored icons, service titles, primary action buttons, and secondary history links. The real-time notification dropdown panel displays recent announcements ensuring tenants remain informed without manual page refresh.

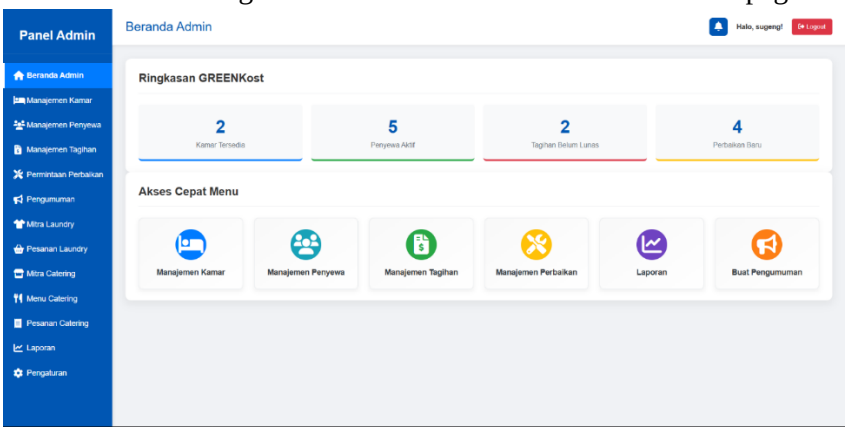

Fig 9. Tenant Dashboard Featuring Service Cards With Notification System And Quick Action Buttons

3. Backend Implementation

The system architecture adheres to MVC pattern: Models handle database interactions via Eloquent ORM with automatic relationship loading; Controllers process business logic with service layer pattern for reusable functionality; Views render through Blade templating engine with template inheritance and XSS protection. Middleware provides authentication verification, role-based authorization, CSRF token validation, and request throttling.Payment gateway integration utilizes Midtrans Snap API with server key authentication. The integration includes payment request generation with transaction details, snap token retrieval for frontend rendering, webhook handler processing notifications asynchronously via queue worker, transaction signature validation, atomic status updates within database transactions, email confirmation via Laravel Mail, and real-time notifications through Laravel Broadcasting with WebSocket connection.Security implementation follows industry best practices: bcrypt password hashing with minimum 8 characters, SQL injection prevention through Eloquent ORM parameter binding, XSS protection via Blade automatic escaping, CSRF token validation on form submissions, HTTPS enforcement through middleware, and secure session management with httpOnly and sameSite cookie attributes [24][25].

C. System Testing Results

1. Black-Box Testing Implementation

Black-box testing covered 87 test cases across all modules: 85 passed (97.7%) and 2 initially failed (2.3%) due to payment timeout handling edge cases. Failed cases were analyzed through debugging sessions identifying webhook retry logic issues. Code refactoring implemented exponential backoff strategy and enhanced error handling. Retesting validated all issues resolved with zero remaining defects, confirming production readiness.

2. Performance Testing

Performance testing using Apache JMeter with 50 concurrent users showed average response times: 1.2 seconds for page loads, 2.5 seconds for payment gateway integration, and 0.8 seconds for database queries. The system maintained stability without crashes or data corruption during peak load.

Discussion

The developed system successfully addresses the manual management challenges at GreenKost through comprehensive digitalization of boarding house operations. Compared to previous research by Hasti and Zaelani [1] which lacked payment gateway integration, this system provides automated payment verification significantly reducing administrative workload. The implementation of real-time room status updates resolves information accuracy issues identified in earlier studies [4][5].Midtrans payment gateway integration supporting QRIS and e-wallet provides tenant flexibility and secure transactions. Webhook notification enables automatic payment confirmation without manual intervention, addressing gaps in Mike et al. [2] research requiring manual verification.Laravel 12.x MVC architecture provides maintainable codebase with clear separation of concerns, facilitating scalability and future enhancements.

Normalized database design ensures data integrity while Eloquent ORM reduces SQL injection vulnerabilities. Responsive UI design with intuitive navigation improves user experience over earlier implementations [3][6][7][8]. The admin dashboard provides business insights for informed decision-making on occupancy trends and revenue optimization.Black-box testing results with 97.7% pass rate demonstrate system reliability and proper implementation of business logic validation. The identified issues in payment timeout handling were edge cases successfully resolved through enhanced error handling mechanisms.System limitations include internet dependency for payment gateway and potential scalability challenges with database optimization as user base grows. Future enhancements could incorporate caching mechanisms and indexing strategies for performance at scale.

IV. CONCLUSION

This research successfully developed a web-based boarding house management system with MVC architecture using Laravel framework to address manual management problems at GreenKost Sleman, Yogyakarta. The system provides integrated features including user management, real-time room management, online reservation, QRIS and E-Wallet payment integration, dashboard reporting, announcements, and digital repair requests. Black-box testing demonstrates proper functionality under normal and abnormal conditions with 97.7% pass rate. The main innovation is payment verification automation through payment gateway integration, addressing previous research limitations requiring manual confirmation.

This integrated system improves administrative efficiency, minimizes manual errors, and enhances user experience through real-time information access.Future development recommendations include UI/UX evaluation using System Usability Scale (SUS) method to measure user satisfaction, and responsive design optimization for mobile devices as the primary access platform. Payment gateway enhancements should include automatic confirmation via email/SMS, detailed transaction history, and refund system for cancellations. Implemented notification, laundry, and catering features require service status tracking for transparency. Analytics and reporting with interactive data visualization will support informed business decision-making by boarding house owners.

REFERENCES

[1] N. Hasti and M. I. Zaelani, “Sistem Informasi Pengelolaan Rumah Kost Budi Sari Berbasis Website,” ***Jurnal Pengabdian Masyarakat dan Riset Pendidikan,*** vol. 2, no. 2, pp. 26–32, 2023.

[2] A. Mike, B. P. Zen, and A. Utami, “Penerapan metode Agile pada website Indekost Sruntul menggunakan framework Laravel,” ***Jurnal Ilmiah Media Sisfo***, vol. 17, no. 2, pp. 266–279, 2023.

[3] I. G. Dhananjaya, “PENGEMBANGAN SISTEM MANAJEMEN KOST BERBASIS WEBSITE UNTUK MENINGKATKAN EFISIENSI PENGELOLAAN DAN PROMOSI PROPERTI KOST,” ***Jurnal Widya Laksmi: Jurnal Pengabdian Kepada Masyarakat***, vol. 5, no. 2, pp. 356–362, 2025.

[4] J. L. Harefa and A. P. Harianja, "Rancang Bangun Ruteakost Sistem Informasi Rumah Kost Berbasis Web di Kota Medan," ***SATESI: Jurnal Sains Teknologi dan Sistem Informasi***, vol. 4, no. 2, pp. 164–171, 2024.

[5] S. Satria, D. Gusman, and E. Azrialdi, "Rancang Bangun Sistem Informasi Kost Berbasis Web di Kecamatan Tampan: Design and Build of Web-Based Boarding Information System In Tampan District," ***MALCOM: Indonesian Journal of Machine Learning and Computer Science***, vol. 2, no. 1, pp. 28–36, 2022.

[6] G. Gunawan, M. Juansen, S. A. Saputra, and D. T. Putra, "PERANCANGAN APLIKASI RUMAH KOST BERBASIS WEBSITE," ***Variable Research Journal***, vol. 2, no. 03, pp. 953–956, 2025.

[7] S. Suminnten, I. D. Sintawati, and W. Indrarti, "Perancangan Sistem Informasi Sewa Rumah Kost Melalui Aplikasi Berbasis Web," *TEKNIKA*, vol. 17, no. 2, pp. 581-â, 2023.

[8] Y. D. Arimbi, D. Kartinah, and A. N. W. Della, "Rancangan sistem informasi kost putri malika berbasis website menggunakan framework laravel dan mysql," ***Jurnal Ilmiah Multidisiplin***, vol. 1, no. 03, pp. 93–103, 2022.

[9] Y. Tumpao, K. Santa, and G. C. Rorimpandey, "Aplikasi Manajemen Rumah Kost Berbasis Web Menggunakan Metode SCRUM," ***Journal Of Informatics, Business, Education And Innovation Technology,*** vol. 2, no. 6, pp. 65–77, 2024.

[10] I. P. S. I. S. Warmadewa, I. M. B. Adnyana, and I. M. Sudarsana, "Optimasi Manajemen Kost Dengan Sistem Informasi Berbasis Web dan Bot Whatsapp Untuk Pembayaran dan Layanan Penghuni," in *Seminar Hasil Penelitian Informatika dan Komputer (SPINTER)| Institut Teknologi dan Bisnis STIKOM Bali*, 2025, pp. 565–570.

[11] A. A. Putri, S. Sepriano, and Y. Ramadhani, "Perancangan Sistem Informasi Pemesanan Kamar Kost Berbasis Website di Gala Residence," ***RIGGS: Journal of Artificial Intelligence and Digital Business***, vol. 4, no. 2, pp. 7108–7121, 2025.

[12] S. R. M. Ali, *Object Oriented Software Engineering*. Leilani Katie Publication, 2024. [Online]. Available: https://books.google.co.id/books?id=9gQSEQAAQBAJ

[13] K. Brown, *The Routledge Companion to Digital Humanities and Art History*. in Routledge Art History and Visual Studies Companions. Taylor & Francis, 2020. [Online]. Available: https://books.google.co.id/books?id=uWHdDwAAQBAJ

[14] A. S. Fitri, R. N. Viddyartha, and R. I. Imroatunnadhiroh, "Rancang Bangun Sistem Informasi Pengelolaan Indekos Berbasis Website dengan Metode Waterfall," ***Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi,*** vol. 13, no. 1, pp. 182–191, 2024.

[15] N. N. Syarif and A. Voutama, "Sistem Informasi Pengelolaan Rumah Kos Berbasis Website Menggunakan Penerapan UML," ***INFORMAL: Informatics Journal***, vol. 9, no. 1, pp. 1–11, 2024.

[16] L. H. Naufal and N. Setiani, "Rancang Bangun Sistem Infomasi Manajemen Kost," *AUTOMATA*, vol. 5, no. 1, 2024.

[17] E. R. Susanto and K. C. Tyas, "Analytical Hierarchy Process Untuk Pemilihan Payment Gateway Pada E-Commerce," ***Jurnal Fasilkom***, vol. 14, no. 2, pp. 471–477, 2024.

[18] E. R. Djuwitaningrum and I. B. W. Jati, "Implementasi Payment Gateway Midtrans pada Website E-commerce Toko Buah dan Sayur," ***Jurnal Ilmu Pengetahuan dan Teknologi***, vol. 9, no. 1, pp. 19–24, 2025.

[19] L. Syahadiyanti, R. P. Putra, and P. Pamudi, "SISTEM INFORMASI PENJUALAN KEMASAN BERBASIS WEBSITE DI PT. ASIA KEMASAN CANTIK," ***Jurnal Informatika Teknologi dan Sains (Jinteks),*** vol. 7, no. 3, pp. 1113–1122, 2025.

[20] Y. S. Rahayu, Y. Saputra, and D. Irawan, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-Disarpus," ***ZONAsi: Jurnal Sistem Informasi***, vol. 6, no. 2, pp. 523–534, 2024.

[21] S. Romdona, S. S. Junista, and A. Gunawan, "Teknik pengumpulan data: Observasi, wawancara dan kuesioner," ***JISOSEPOL: Jurnal Ilmu Sosial Ekonomi dan Politik***, vol. 3, no. 1, pp. 39–47, 2025.

[22] A. Rivaldi, F. U. Feriawan, and M. Nur, "Metode pengumpulan data melalui wawancara," *Sebuah Tinjauan Pustaka*, pp. 1–89, 2023.

[23] D. Alpina and H. Witriyono, "Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web," ***Jurnal Media Infotama***, vol. 18, no. 1, pp. 36–42, 2022.

[24] R. A. Siahaan and R. A. Sianturi, "Analisis Perbandingan Payment Gateway untuk Sistem Pembayaran Berbasis Aplikasi dengan Comparative Study," ***Jurnal Teknologi Informasi dan Ilmu Komputer***, vol. 11, no. 2, pp. 291–296, 2024.

[25] B. Hendrik and B. R. Suteja, "Identifikasi Risiko Program Maintenance dalam Pengelolaan Proyek Berbasis Agile Menggunakan Pohon Klasifikasi," ***Jurnal Teknik Informatika dan Sistem Informasi***, vol. 7, no. 1, 2021.