Jurnal Esensi Infokom Vol 10 No. 2 Oktober 2026
e-ISSN: 2828-6707
https://esensijournal.com/index.php/infokom
DOI: 10.55886/infokom.v10i2.542

# Development of a Web-Based Boarding House Management Information System With a Payment Gateway Using the Scrum Framework

M. Dimas Anwar[1], M. Rudi Sanjaya[2], Fidia Adella Wulandari[3], Agnes Carolina[4], M Zefania Sharon P.S.[5], A.A. Aswin Aldira[6], Zira Pertiwi[7]

[1,2,3,4,5,6,7]*Department of Information Systems, Faculty of Computer Science, Universitas Sriwijaya*
*Jl. Raya Palembang-Prabumulih Km. 32, Ogan Ilir 30662, South Sumatra, Indonesia*

[1]dimasanwar315@gmail.com, [2]m.rudi.sjy@ilkom.unsri.ac.id, [3]adellafidia17@gmail.com, [4] agnescrolina@gmail.com, [5]zefaniapriscilla23@gmail.com, [6]aswinaldira01@gmail.com, [7]zirapertiwi0@gmail.com

***Abstract*— Boarding house management involves various activities that generate operational, transaction, resident service, and financial data. At Jessa Kost, some data and information management processes are still carried out manually and through separate media, resulting in information that is not centrally managed. This study aims to develop a web-based Boarding House Management Information System that integrates management information to support monitoring and operational management. The system was developed using the Scrum framework through product backlog, sprint planning, development, and sprint review. The system covers room and resident management, rental payments, electricity and WiFi, guest records, complaints, announcements, expenses, and financial reports with payment gateway support. The testing results show that all tested functions operated according to the expected results. The system provides centralized information through dashboards and reports to support operational management at Jessa Kost.**

*Keywords*— Management Information System, Boarding House Management, Payment Gateway, Scrum, Operational Management, Information Integration.

*Intisari*— **Pengelolaan rumah kost melibatkan berbagai aktivitas yang menghasilkan data operasional, transaksi, layanan penghuni, dan keuangan. Pada Jessa Kost, sebagian proses pengelolaan data dan informasi masih dilakukan secara manual dan melalui media yang terpisah, sehingga informasi belum terkelola secara terpusat. Penelitian ini bertujuan mengembangkan Sistem Informasi Manajemen Kost berbasis web yang mengintegrasikan berbagai informasi pengelolaan untuk mendukung monitoring dan pengelolaan operasional. Pengembangan sistem menggunakan kerangka kerja Scrum melalui tahapan product backlog, sprint planning, development, dan sprint review. Sistem mencakup pengelolaan kamar dan penghuni, pembayaran sewa, listrik dan WiFi, buku tamu, keluhan, pengumuman, pengeluaran, serta laporan keuangan dengan dukungan payment gateway. Hasil pengujian menunjukkan seluruh fungsi yang diuji berjalan sesuai hasil yang diharapkan. Sistem menyediakan informasi terpusat melalui dashboard dan laporan untuk mendukung pengelolaan operasional Jessa Kost.**

*Kata kunci*— Sistem Informasi Manajemen, Manajemen Rumah Kost, Payment Gateway, Scrum, Manajemen Operasional, Integrasi Informasi.

## I. INTRODUCTION

A Management Information System (MIS) transforms data from organizational activities into information that supports operations, monitoring, and managerial decision-making [1]. Its application extends beyond data recording to the management and presentation of information according to managerial needs. In boarding house management, activities such as room and resident administration, payments, utilities, and resident services generate data that must be managed in an integrated manner to support operations.

Boarding house management that is still carried out conventionally can cause data recording and information delivery to be conducted through different media. Research on Boarding House Management Information Systems shows that manual boarding house management still creates a need for the integration of resident and transaction data [2]. Other studies on the development of boarding house information systems also indicate a need for data recording, room availability monitoring, and resident information management [3]. Based on observations and interviews conducted at Jessa Kost, information delivery to residents is still carried out directly in person, while certain urgent information is conveyed through communication media such as WhatsApp. Financial recording by the owner is still performed manually, whereas rental payments are made via direct bank transfer to the owner's account. For specific needs related to electricity, residents still have to communicate with the administrator or boarding house guard. These conditions indicate that operational and transaction information management is still carried out through several separate processes and media.

Several studies have developed systems to support boarding house management. Nugraha et al. [2] developed a Boarding House Management Information System (SIMKO) to integrate resident and transaction data, whereas Puspasari et al. [3] developed a boarding house information system to support the management of resident data, room status, and billing. Tumpao et al. [4] developed a boarding house management application

using Scrum that covers boarding house data, residents, rooms, and payments. Saputro et al. [5] developed a boarding house payment and complaint management system using Scrum, whereas Bintang et al. [6] developed a boarding house management system with cashless payment integration using Scrum. However, a comparison between these existing systems and the system developed in this study reveals several functional gaps. Systems [2], [3], and [4] focus primarily on basic room administration, tenant data, and manual payment recording without automated payment gateway verification, complaint handling, or utility management. While system [5] incorporates complaint management and system [6] integrates cashless payments, neither system provides features for separate electricity and WiFi utility billing, digital guest book logging, operational expense tracking, or an executive financial report calculating real-time net profit. This comparison indicates that the main distinction of the proposed system lies not in introducing a single new function, but in integrating multiple operational and managerial functions within one information environment. This integration is intended to reduce the fragmentation of transaction, service, utility, and financial information so that boarding house management can be monitored more centrally.

Therefore, this study develops a web-based Boarding House Management Information System for Jessa Kost using the Scrum framework. Compared to the existing boarding house systems [2]–[6] that only address partial operational aspects, the proposed system offers a more comprehensive feature set by combining nine core modules: (1) multi-role authentication, (2) room and tenant management, (3) automated rent payment via a payment gateway, (4) electricity and WiFi billing, (5) facility complaint and repair cost tracking, (6) a digital guest book, (7) announcements, (8) operational expense and net profit reporting, and (9) a real-time owner monitoring dashboard. This study aims to develop an integrated information system that supports the operational management of Jessa Kost.

## II. LITERATURE REVIEW

### A. *Previous Studies*

Several previous studies have addressed the development of boarding house information systems and the integration of payment gateways into web-based systems, with varying focuses, methodologies, and feature scopes. Study [7] developed a boarding house information system using the *Rapid Application Development* (RAD) method, with the Next.js framework and PostgreSQL database. The system includes room booking and an administrative dashboard; however, payments are still processed by uploading proof of bank transfer. Therefore, the integration of a payment gateway was recommended as a potential area for further development.

Study [8] developed a web-based boarding house information system using PHP and MySQL with the waterfall method and WhatsApp notifications. The system focuses on searching for and promoting boarding houses and achieved a feasibility test result of 100%. However, it does not yet focus on the digital management of payment transactions. Meanwhile, study [9] implemented a payment gateway in an event ticketing management information system using Node.js and Next.js with the waterfall method. The study demonstrated that payment gateway integration can facilitate transaction processes; however, its implementation was conducted in the event ticketing domain rather than in boarding house management.

In the boarding house domain, study [4] developed a web-based boarding house management application using the Scrum framework, Laravel, and Unified Modeling Language (UML). The system covers the management of boarding house data, residents, rooms, and payments; however, the payment functionality remains focused on recording internal transactions. Another Agile approach was applied in study [10] through the development of the E-PANJO nursing home management information system using Extreme Programming (XP). The study demonstrated that the application of Agile methodologies in management information systems can support information management and address dynamic user requirements, although the research object and system scope differ from those of boarding house management.

Based on previous studies [4], [7], [8], [9], [10], the development of boarding house information systems has covered room management, resident management, booking, payment, and information dissemination using various development approaches. However, this study focuses on developing a system from a Management Information System (MIS) perspective by integrating operational information, transactions, resident services, utilities, and financial information into a single system to support boarding house monitoring and management. The system is developed for Jessa Kost using Scrum and is equipped with a *payment gateway* as part of the payment transaction integration.

### B. *Management Information System*

A Management Information System (MIS) is a set of interrelated components involving people, technology, and procedures that collect, process, store, and distribute information to support decision-making, coordination, and control within an organization [11]. A computer-based MIS enables operational data from various activities to be integrated and presented as information required by management for monitoring and decision-making [12]. In this study, the MIS concept is implemented by integrating operational, transaction, resident service, utility, and financial information at Jessa Kost. The integrated information is subsequently presented through dashboards and reports to support centralized monitoring and boarding house management.

### C. *Information System*

An information system consists of interrelated components designed to collect, process, store, and present information to support organizational activities [13]. Structurally, an information system includes input, processing, output, technology, database, and control components that work together to transform data into useful information. In a web-based environment, these components are implemented

through user interfaces, application processes, and integrated databases that enable users to access and manage information digitally [14]. In this study, room and resident data, transactions, utilities, complaints, and other operational records serve as system inputs, which are processed and subsequently presented through system functions, dashboards, and reports.

### D. *Payment Gateway*

A payment gateway is a technological service that functions as an intermediary for processing electronic payment transactions by connecting customers, service providers (billers), and financial institutions (collecting agents), allowing financial transactions to be processed and verified automatically without requiring manual confirmation. In practice, when a customer initiates a transaction through a payment channel, the transaction data is sent to the payment gateway and forwarded to the biller, which then provides a response indicating the success or failure of the transaction back to the payment gateway and subsequently to the customer [15]. Payment gateways generally support various payment methods, such as credit cards, debit cards, bank transfers, digital wallets, and virtual accounts. Therefore, their selection needs to be adjusted to the type and scale of the business, as each payment gateway provider has different characteristics in terms of transaction fees, fund settlement speed, and payment method coverage [16]. With the implementation of a payment gateway, rental payment transactions in the system developed in this study can be conducted online, recorded automatically, and verified in real time, thereby reducing the risk of recording errors commonly associated with manual payment mechanisms.

### E. *Scrum*

Scrum is a flexible and lightweight framework in which people involved in software product development collaborate to address complex problems and develop products that deliver the highest possible value [17]. Within the Scrum framework, work is organized into a product backlog, which contains a prioritized list of product requirements established by the Product Owner, and is then carried out within a fixed period known as a sprint, which generally lasts two to four weeks through a series of activities including sprint planning, daily scrum, and sprint review. The implementation of Scrum has been shown to accelerate the development of information systems by emphasizing team collaboration and continuous iteration in each sprint, enabling the team to continuously evaluate development progress and adapt to changing requirements throughout the development process [18]. In this study, the Scrum framework is used as the development method for the Boarding House Management Information System because the stages of product backlog, sprint planning, sprint backlog, daily stand-up, and sprint review enable complex system requirements, ranging from room and resident management to payment gateway integration, to be developed in a structured manner and monitored daily through daily stand-ups, before being comprehensively evaluated during the sprint review at the end of the sprint.

## III. RESEARCH METHODOLOGY

This research falls under the category of applied research in the field of software engineering. The research method encompasses qualitative data collection and software testing, while the applied research model for system development is the Agile Scrum framework. The methodology is structured to ensure the systematic development of the system and to achieve the expected output efficiently.

### A. *Data Collection*

Data collection in this study was conducted through interviews and observations of the Jessa Kost management process. Interviews were held with stakeholders involved in the boarding house administration to acquire information regarding the ongoing business processes and system requirements. The gathered information encompasses room and tenant data management, rent payments, information dissemination, utilities, as well as tenant reports and needs. Observations were carried out to comprehend the management workflow and information flow during operational activities. The results from both interviews and observations were subsequently analyzed using a fishbone diagram to identify the root causes of the problems, which served as the foundation for formulating the product backlog.

### B. *System Development*

The development of the Boarding House Management Information System was executed using the Scrum framework. Scrum was chosen because it implements an iterative and incremental development process, allowing system requirements to be developed gradually and evaluated continuously during the development lifecycle [17] , [18] . The stages applied in this study include the product backlog, sprint planning, sprint backlog, development and daily stand-up, and sprint review. The Scrum workflow is illustrated in Figure 1.




Figure 1. Scrum Method Workflow

The applied development stages are described as follows:

*1) Product Backlog:* Contains a list of system requirements and features derived from the results of interviews and observations. These requirements are arranged based on priority to determine the features that will be developed in the sprint.

*2) Sprint Planning Meeting:* During this stage, requirements from the product backlog are selected based on

priority to determine the tasks that will be executed during the sprint. Planning and task estimation are also conducted at this stage [4].

*3) Sprint Backlog:* A prioritized list of tasks selected from the product backlog that have been broken down into smaller tasks to be completed in the upcoming sprint. The sprint backlog outlines the tasks that the development team must complete during the sprint. It is managed and regularly updated by the development team [18].

*4) Development & Daily Stand Up:* The development stage is the process of system implementation based on the established Sprint Backlog. Each team member executes tasks according to the work distribution during the sprint. Throughout the development process, a short Daily Stand-up is routinely held to synchronize work, discuss progress and obstacles, and adjust the development plan accordingly.

*5) Sprint Review:* The final evaluation stage of the sprint after all tasks have been completed. This stage also serves as an opportunity to obtain feedback and determine future product developments based on the review results.

In this study, the development was conducted in a single sprint with a duration of two weeks. The breakdown of the development activities is shown in Table I.

TABLE I
SYSTEM DEVELOPMENT TIMELINE USING SCRUM

| Stage | Activity | Timeframe |
|---|---|---|
| Product Backlog | Identification and formulation of system requirements | Day 1 |
| Sprint Planning | Determining features and tasks to be executed | Day 1 |
| Sprint Backlog | Breaking down features into development tasks | Day 1 |
| Development & Daily Stand-up | System development and progress monitoring | Days 2-11 |
| Sprint Review | Evaluation of development results | Day 12 |
| Finalization | System refinement and completion | Days 13-14 |

### C. System Testing

System testing was conducted using the black-box testing method to evaluate system functionalities based on the inputs and generated outputs without examining the internal program code structure. Testing was performed by executing test scenarios on each developed function and comparing the actual results with the expected outcomes. This approach is utilized to ensure that every system function operates properly according to user requirements. The application of black-box testing in information system research has also been implemented in previous studies [19] . The test results are deemed successful if the system's output aligns perfectly with the expected results.

### D. Expected Output

The expected output of this research is a functional web-based Boarding House Management Information System for Jessa Kost. The system is designed to integrate room and tenant management, payments, utilities, complaints, guest records, announcements, and financial information within a single platform. The system is also expected to provide centralized operational and financial information through dashboards and reports for management monitoring.

## IV. RESULTS AND DISCUSSION

### A. Problem Analysis

Problem analysis in this study was conducted using a fishbone diagram to systematically identify the root causes of problems in the management of Jessa Kost, as shown in Figure 2. The primary problem identified is the absence of an integrated management information system, which stems from four root causes: (1) unintegrated room, resident, and transaction data, (2) uncentralized operational monitoring, (3) manual payment and financial recording, and (4) uncentralized information delivery, complaint handling, and electricity/WiFi utility services. The data category indicates that the system must manage four main types of data, namely payment data, room and tenant data, operational data, and digital files, while the platform category confirms that the system is built as a web-based application. In addition to root causes, this fishbone diagram maps the system implementation scope through three additional categories: users consisting of the owner, staff, and residents; development tools comprising PHP, the Laravel framework, Tailwind CSS, MSSQL, GitHub, as well as Google SMTP and Google Maps integration; and the study object, namely Jessa Kost as the research site. The results of this fishbone analysis serve as the foundation for compiling the product backlog during the system development stage using the Scrum framework, where each root cause in the problem category is translated into prioritized feature requirements to be executed during the sprint.

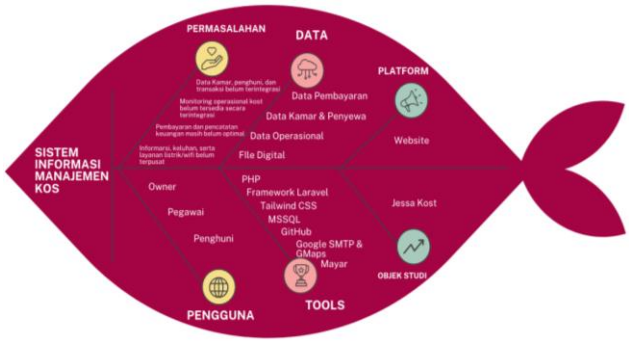



Figure 2. Fishbone diagram for the Boarding House Management Information System.

### B. Product Backlog

The product backlog was compiled based on the results of interviews, observations, and problem analysis at Jessa Kost. Each backlog item represents a system requirement associated with the operational needs of the owner, staff, and tenants. The

Jurnal Esensi Infokom Vol 10 No. 2 Oktober 2026
e-ISSN: 2828-6707
https://esensijournal.com/index.php/infokom
DOI: 10.55886/infokom.v10i2.542

identified features were prioritized according to their relevance to core system processes and implementation dependencies, while the estimated development time for each feature is presented in Table II.

TABLE II
PRODUCT BACKLOG

| No | Name (Feature) | Est (Hari) | Acceptance Criteria |
|---|---|---|---|
| 1 | User Authentication & Role Management | 1 | Owners, staff, and tenants can log in according to their respective roles |
| 2 | Room & Tenant Management | 1 | Admin can add, edit, and delete room data and tenants |
| 3 | Rent Payment with Payment Gateway | 2 | Tenants can pay rent and the payment status is updated automatically |
| 4 | Electricity & WiFi Billing | 1 | Tenants can view and pay electricity/WiFi bills |
| 5 | Tenant Complaints & Reports | 1 | Tenants can submit complaints, and admin can respond to them |
| 6 | Digital Guest Book | 1 | Admin can record data of guests visiting the boarding house |
| 7 | Announcements | 1 | Admin can create announcements, and tenants can read them |
| 8 | Financial & Expense Reporting | 1 | Owner can view reports of income and expenses at the boarding house |
| 9 | Owner Monitoring Dashboard | 1 | Owner can monitor a summary of operational costs in real time |

User authentication, room and tenant management, and rental payment were prioritized during early development because these functions serve as the foundation for other system features. All identified backlog items were subsequently included in the sprint and decomposed into smaller development tasks during the sprint backlog stage.

### C. *Use Case*

The use case diagram illustrates the interactions between users and the Boarding House Management Information System, as shown in Figure 3. The system involves four actors, namely Visitor, Resident, Admin, and Owner, each with different access rights according to their roles. Most system functions require authentication through the Login use case, while the Visitor can access public information without an account. The Visitor can view boarding house information, facilities, and room availability as initial information for prospective residents. The Resident can access announcements, submit complaints or damage reports, manage guest records, view billing information, and make rental payments through the system.

The Admin is responsible for daily operational management, including room data, utilities, complaints, announcements, guest records, and operational expenses. Meanwhile, the Owner has broader access to user management, payment monitoring, financial reports, net profit information, and overall operational conditions. This role-based access structure separates public, operational, resident service, and managerial functions while ensuring that each actor accesses information according to their respective responsibilities.

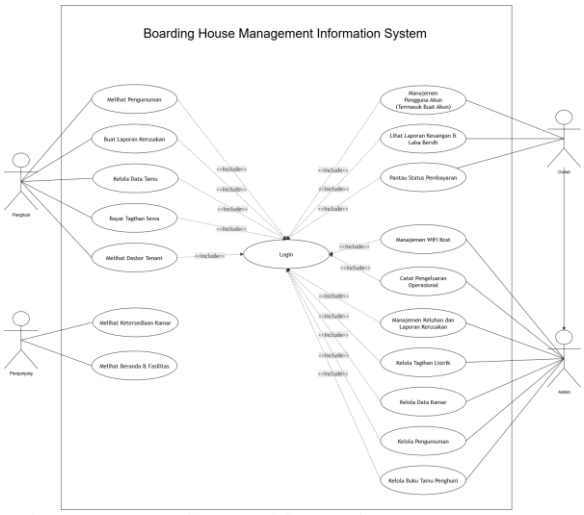


Figure 3. Use case diagram of the Boarding House Management Information System.

### D. *Flowchart*

Process design in this study is illustrated through four flowcharts representing the interaction of Visitor, Admin, Owner, and Resident with the system. Each flowchart describes a different process according to the actor's responsibilities and access rights. The process design was developed based on the requirements identified during the problem analysis, product backlog, and use case design stages. Therefore, the flowcharts do not only represent navigation within the application, but also illustrate how operational, transaction, service, and managerial activities are carried out through the developed system.

The Visitor process is shown in Figure 4 and represents public interaction with the system before a user becomes a registered resident. Visitors can access general information about Jessa Kost, including its profile, facilities, room availability, and contact information without authentication. Room information is presented to support prospective residents in identifying available accommodation, while integration with Google Maps and WhatsApp provides access to location information and further communication with the administrator. This process separates publicly accessible information from functions containing operational and resident data, allowing the system to support promotional and information needs without providing access to restricted management functions.

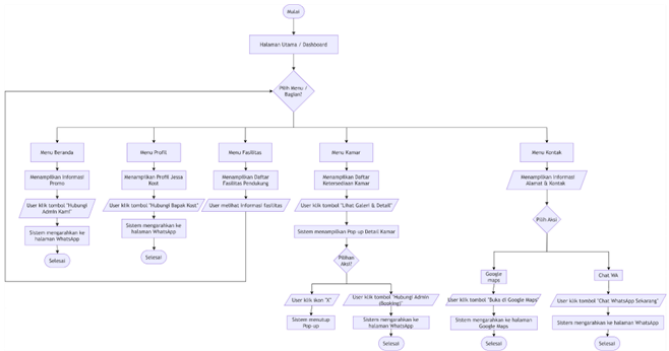


Figure 4. Flowchart for the Visitor (public) actor.

The Admin process is shown in Figure 5 and represents the daily operational management of Jessa Kost. After authentication, the Admin gains access to functions related to room management, guest records, electricity and WiFi services, complaints, announcements, and operational expenses. Rather than operating as separate recording processes, these activities are managed within the same system and follow a consistent data management process in which information can be recorded, updated, and maintained according to operational needs. Through this process, the Admin acts as the primary operational user responsible for maintaining the accuracy and continuity of information generated from daily boarding house activities.

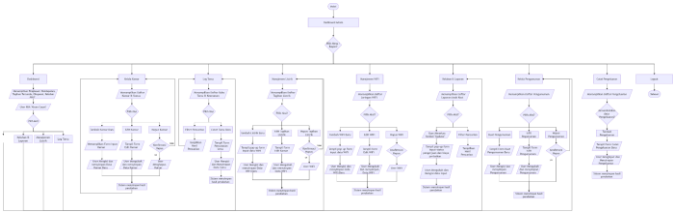


Figure 5. Flowchart for the Admin actor.

The Owner process is shown in Figure 6 and emphasizes monitoring and managerial access to information generated from operational activities. In addition to accessing several operational functions available to the Admin, the Owner has broader access to user management, rental payment status, operational expenses, financial reports, and net profit information. This process reflects the different information requirements of the Owner, whose role is not limited to recording daily activities but also includes monitoring the overall condition of the boarding house. By consolidating operational and financial information within the same system, the Owner can obtain a centralized overview of room, transaction, and financial conditions for management monitoring.

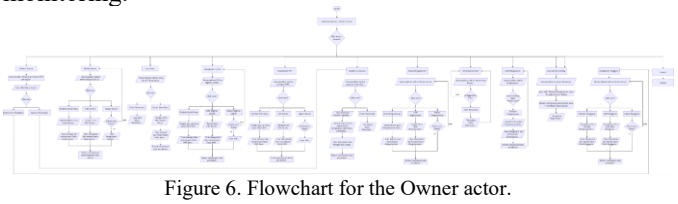


Figure 6. Flowchart for the Owner actor.

The Resident process is shown in Figure 7 and represents interactions related to residential services and financial transactions. After authentication, residents can access information associated with their occupancy, including billing summaries, WiFi information, announcements, guest records, complaints, and personal data. The payment process is connected to the billing information recorded in the system, allowing residents to identify outstanding bills and continue to the integrated payment process when payment is required. Residents can also submit facility-related reports and monitor their status through the system. This process centralizes information and services that were previously handled through separate communication and recording mechanisms, providing residents with a single access point for activities related to their stay at Jessa Kost.

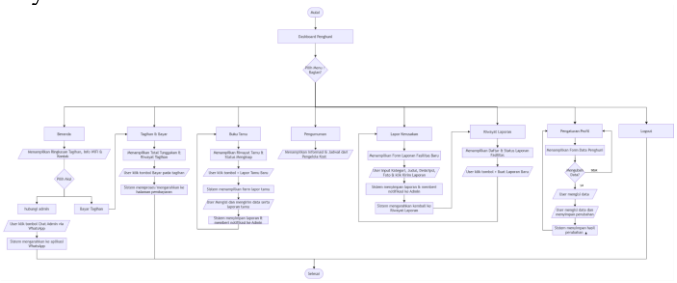


Figure 7. Flowchart for the Resident actor.

### *E. Entity Relationship Diagram (ERD)*

The database design is represented through an Entity Relationship Diagram consisting of ten interrelated entities, namely users, tenant_profiles, announcements, expenses, guest_logs, tickets, leases, bills, rooms, and electricity_readings, as shown in Figure 8. These entities represent the operational, resident, transaction, utility, and financial data requirements identified in the product backlog. The users entity serves as the primary entity for storing Owner, Admin, and Resident account information based on their respective roles and is connected to tenant_profiles, announcements, expenses, guest_logs, tickets, and leases.

The rooms entity is connected to leases, tickets, and electricity_readings to support room occupancy, complaint records, and electricity usage. The leases entity represents the rental relationship between residents and rooms and is connected to bills, which stores billing information such as bill type, amount, due date, and payment status. Through these relationships, the database provides a centralized structure for managing operational, resident, transaction, utility, and financial information within the Boarding House Management Information System.

Jurnal Esensi Infokom Vol 10 No. 2 Oktober 2026
e-ISSN: 2828-6707
https://esensijournal.com/index.php/infokom
DOI: 10.55886/infokom.v10i2.542

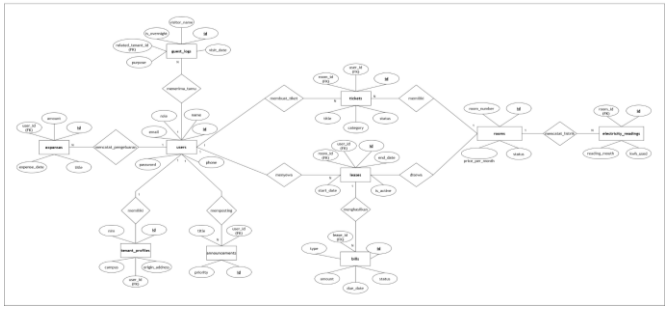


Figure 8. Entity Relationship Diagram of the Boarding House Management Information System.

### *F. Payment Gateway Flow Design*

The payment gateway implementation in this system is designed to automate the verification of rent and other bill payments, which was previously done manually. The transaction flow begins when a tenant selects a bill in the Billing and Payment menu, then presses the payment execution button. The system then sends the transaction parameters (request), consisting of the bill ID, tenant details, and bill amount, to the payment gateway's Application Programming Interface (API). Next, the payment gateway responds by displaying an invoice (checkout) page containing various payment method options, such as QRIS, e-wallet, and bank transfer. Once the tenant completes the payment through the chosen method, the payment gateway sends a real-time callback notification (webhook) to the Jessa Kost server. The server then validates the notification and automatically updates the bill status in the database from "Unpaid" to "Paid." This automated flow significantly reduces the manager's workload of manually checking bank account mutations and minimizes the risk of transaction recording errors.

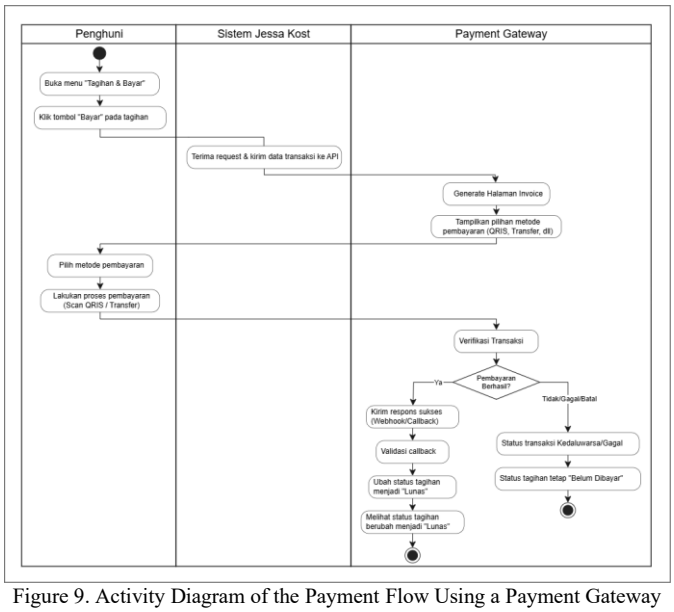



Figure 9. Activity Diagram of the Payment Flow Using a Payment Gateway

### *G. System Implementation*

1) *Public Page and Room Availability (Visitor):* The system provides a public portal accessible without login that displays Jessa Kost's profile, facilities, and real-time room availability, as shown in Figure 10. This interface serves as a digital promotional medium and facilitates direct room booking through an integrated WhatsApp link.

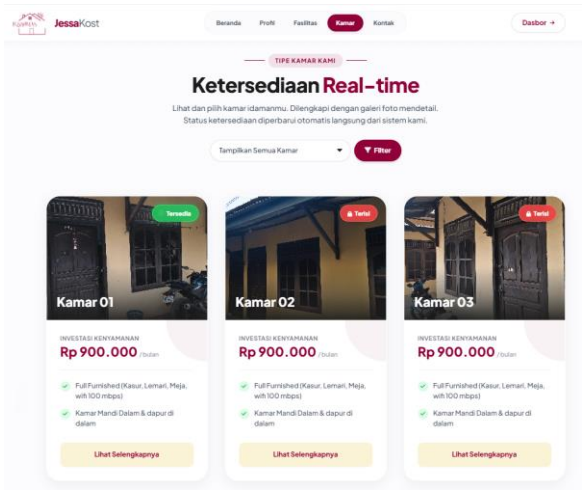



Figure 10. Public interface for boarding house room availability

*1) Resident Dashboard and Payment Gateway Integration (Tenant)*: Registered tenants are provided with a self-service dashboard displaying room details, financial status summaries, WiFi credentials, and quick access to daily operational features, as shown in Figure 11. For cashless transactions, the system integrates a third-party payment gateway that dynamically generates a QRIS code and digital payment options on the invoice page (Figure 12), replacing manual receipt uploads with automated system validation.

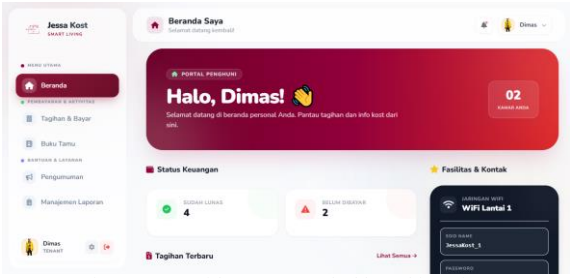



Figure 11. Resident (tenant) dashboard interface

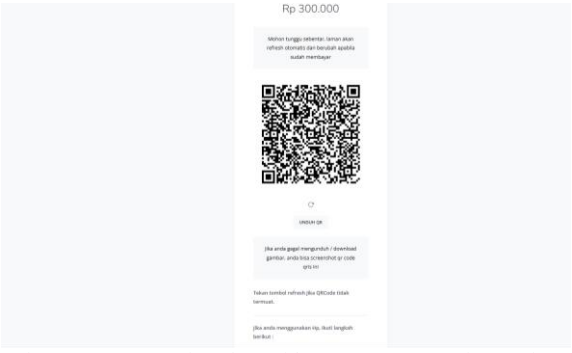



Figure 12. Payment interface with payment gateway integration

*2) Complaint Management and Operational Reports (Admin)*: As shown in Figure 13, the Admin interface

Jurnal Esensi Infokom Vol 10 No. 2 Oktober 2026
e-ISSN: 2828-6707
https://esensijournal.com/index.php/infokom
DOI: 10.55886/infokom.v10i2.542

centralizes facility complaints submitted by residents along with their processing statuses (New, In Progress, Completed), which is directly integrated with operational expense recording when repairs require funding.

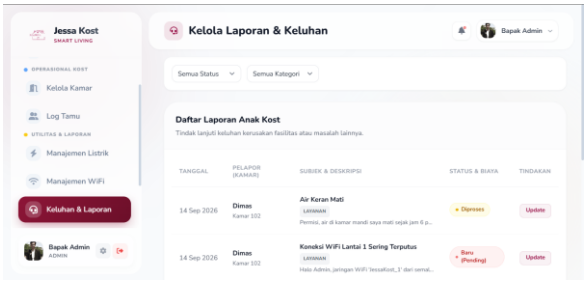



Figure 13. Facility complaint and report management interface

*3) Executive Summary Dashboard and Financial Reports (Owner)*: As the primary Management Information System output for strategic decision-making, the Owner interface (Figure 14) visualizes real-time financial reports, covering total income, arrears, room occupancy, net profit, and monthly cash flow details.

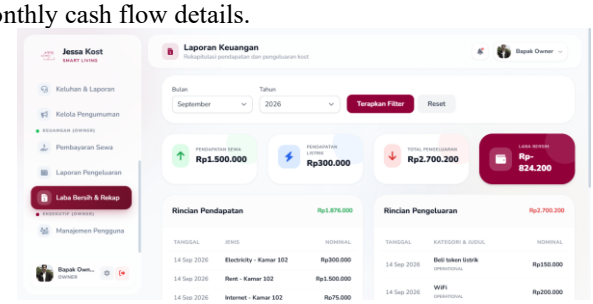



Figure 14. Owner financial report dashboard interface

## *H. System Testing*

Blackbox testing was used to verify that the functionality of this information system matches the requirements set out in the product backlog stage. The testing focuses on the output the system produces for a given input, without regard to the underlying program code structure, so it can confirm that each feature works as intended from the user's point of view. Technically, the testing was carried out by running test scenarios on the nine main features of the system, performed by the researcher together with the users, namely the owner, the admin, and the tenants of Jessa Kost. The results of the blackbox testing are shown in Table III

TABLE III
BLACKBOX TESTING RESULT

| No | Name (Feature) | Scenario | Achieved Result | Status |
|---|---|---|---|---|
| 1 | User Authentication & Role Management | The user enters a valid email and password for their role (owner, admin, tenant) on the login page, then presses the "Enter Dashboard" button | The system successfully verifies the credentials and displays the dashboard according to the user's role (owner/admin/tenant) | Successful |
| 2 | Room & Tenant Management | Add new room data through the "Add Room" form, change the room status, and delete room data | The system successfully saves, updates, and deletes room and tenant data | Successful |
| 3 | Rent Payment with Payment Gateway | The tenant selects an unpaid bill, presses the "Pay" button, and completes the transaction on the payment gateway page | The application successfully redirects to the payment gateway, and the bill status automatically changes to "Paid" | Successful |
| 4 | Electricity & WiFi Billing | The admin enters a new electricity bill for a specific room through the "Electricity Management" menu, then the tenant opens the "Billing & Payment" menu on their dashboard | The system successfully saves the bill and displays it on the relevant tenant's dashboard | Successful |
| 5 | Tenant Complaints & Reports | The tenant fills in and submits a damage and complaint report form with a description. The admin receives the report and updates its status | The application successfully saves the report, sends a notification to the admin, and updates the report status | Successful |
| 6 | Digital Guest Book | The tenant enters guest data in the "Report Guest" form and saves it | The system successfully saves and displays the guest visit history in the guest log list | Successful |
| 7 | Announcements | The admin fills in the "Create New Announcement" form (title, priority, content) and presses the save button | The application successfully saves the announcement and displays it on the dashboards of all tenants | Successful |
| 8 | Financial & Expense Reporting | The admin records operational expenses through the "Record Expense" menu, | The system successfully records the expense and displays total income, total expenses, and | Successful |

| | | | | |
|---|---|---|---|---|
| | | then the owner opens the financial report page | net profit | |
| 9 | Owner Monitoring Dashboard | The owner logs in to the system and opens the "Dashboard" menu to view a summary of room status, bills, and revenue | The application successfully displays a summary of room status, payments, and operational conditions in real time | Successful |
| 10 | User Management – Input Validation (Invalid Phone Number & Required Fields) | The owner fills out the "Add New User" (*Tambah Pengguna Baru*) form by entering non-numeric characters (e.g., "ss") in the phone/WhatsApp number field or leaving the room assignment empty when the selected role is "Tenant", then submits the form | The system rejects the submission, prevents invalid data from being stored in the database, and displays an error validation alert message on the form | Successful |

The black-box testing results show that the nine main system functions and the additional invalid-input validation scenario produced outputs consistent with the defined functional requirements. These functions include user authentication, room and tenant management, rental payments, utility billing, complaint management, guest records, announcements, financial reporting, and managerial monitoring. The results confirm the functional implementation of the requirements previously identified in the product backlog.

From an MIS perspective, the developed system extends the scope of previous boarding house applications by integrating operational, transaction, resident service, utility, and financial information within a single system. Previous studies [2], [4], and [5] primarily focused on room and tenant management, payment recording, or complaint management, while study [6] incorporated cashless payment integration. In the developed system, payment processing is connected with automatic payment-status updates, while operational and financial information is consolidated through the Owner dashboard. This integration allows information generated from different boarding house activities to be centrally presented for management monitoring, demonstrating the role of the developed system not only in supporting individual operational processes but also in integrating information across boarding house management activities.

In addition to testing valid functional flows, negative testing scenarios were also conducted to evaluate system robustness against invalid user inputs, particularly in the User Management feature. When the owner attempts to create a new tenant account with an invalid contact number format (such as alphabetical characters like 'ss' in the phone/WhatsApp field) or omits mandatory role-dependent fields (such as room assignment for a tenant), the system automatically blocks the data storage process and displays an error validation alert to guide the user in correcting the input.

### *I. Sprint Review*

The sprint review identified two functional issues during the initial implementation: payment processing through the payment gateway and overnight guest report submission. Both issues were subsequently corrected before final system testing. This evaluation process demonstrates the iterative nature of the Scrum implementation, in which identified functional problems were reviewed and refined before the system was finalized.

## V. CONCLUSION

This study developed a web-based Boarding House Management Information System for Jessa Kost that integrates operational, tenant service, utility, transaction, and financial information within a single platform. The Scrum framework supported the iterative implementation of the identified system requirements, while black-box testing showed that the tested core functions operated according to the expected outcomes. The integration of a payment gateway enables automatic payment-status updates, while the Owner dashboard centralizes operational and financial information for monitoring purposes. Therefore, the developed system fulfills the research objective of providing integrated information to support boarding house management. This study is limited to functional testing within a single case environment. Future research may extend the evaluation through usability testing, system performance measurement, and quantitative assessment of the system's impact on operational management.

## REFERENCES

[1] A. Syahputra, R. Wiranti, And W. Astita, "Peran Sistem Informasi Manajemen Organisasi Dalam Pengambilan Keputusan," *Jurnal Manajemen Sistem Informasi (Jmasif)*, Vol. 1, No. 1, Pp. 26–31, Apr. 2022, Doi: 10.35870/Jmasif.V1i1.67.

[2] M. K. Nugraha, N. C. Wibowo, And I. R. Mukhlis, "Design And Development Of The Boarding House Management Information System (Simko) Using Laravel With Agile Methodology," *Jurnal Teknologi Dan Open Source*, Vol. 8, No. 1, Pp. 112–122, Jun. 2025, Doi: 10.36378/Jtos.V8i1.4304.

[3] B. D. Puspasari, J. D. Irawan, R. A. Wicaksono, D. R. Pratama, And A. B. Ilmi, "Development Of Information System For Boarding Houses," *International Journal Of Computer Science And Information Technology*, Vol. 2, No. 1, Pp. 8–12, Apr. 2025, Doi: 10.36040/Ijcomit.V2i1.13659.

[4] Y. Tumpao, K. Santa, And G. C. Rorimpandey, "Aplikasi Manajemen Rumah Kost Berbasis Web Menggunakan Metode Scrum," *Journal Of Informatics, Bussines, Education, And Innovation Technology*, Vol. 2, No. 6, Pp. 65–77, Nov. 2024.

[5] R. E. Saputro, G. G. Setiaji, And A. Rifa'i, "Implementasi Metode Scrum Pada Aplikasi Web Manajemen Pembayaran Kos Berbasis Laravel Dan Filament," *Information System Journal*, Vol. 8, No. 02, Pp. 71–84, Nov. 2025, Doi: 10.24076/Infosjournal.2025v8i02.2283.

[6] A. Bintang, I. Purnamasari, And Carudin, "Transformasi Digital Pengelolaan Kos: Sistem Berbasis Laravel Dengan Integrasi Pembayaran Nontunai," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 14, No. 1, Jan. 2026, Doi: 10.23960/Jitet.V14i1.8261.

[7] U. P. A. Saputra, L. Sugesti, And K. N. M. Ngafidin, "Perancangan Sistem Informasi Rumah Kos Menggunakan Metode Rapid Application Development," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 12, No. 2, Apr. 2024, Doi: 10.23960/Jitet.V12i2.4269.

[8] A. Prasetyo And M. A. Rosid, "Perancangan Sistem Informasi Rumah Kos Berbasis Web Dan Short Message Service (Whatsapp) Menggunakan Php Dan Mysql," *Indonesian Journal Of Applied Technology*, Vol. 1, No. 2, P. 18, Jul. 2024, Doi: 10.47134/Ijat.V1i2.3040.

[9] A. Y. Prasetyo And G. Swalaganata, "Sistem Informasi Manajemen Ticketing Event Dengan Payment Gateway Berbasis Node.Js Dan Next.Js," *Journal Of Information System And Application Development*, Vol. 2, No. 2, Pp. 103–120, Aug. 2024, Doi: 10.26905/Jisad.V2i2.12982.

[10] S. S. Wibagso And I. Celesta, "Extreme Programming Approach In E-Panjo Design To Support Information Management At Nursing Home," *Jurnal Buana Informatika*, Vol. 13, No. 02, Pp. 93–104, Oct. 2022, Doi: 10.24002/Jbi.V13i02.5707.

[11] S. Armah And S. Firdaus, "Konsep Dan Penerapan Sistem Informasi Manajemen," *Jurnal Inovasi Manajemen, Kewirausahaan, Bisnis Dan Digital*, Vol. 1, No. 3, Pp. 50–56, Jun. 2024, Doi: 10.61132/Jimakebidi.V1i3.192.

[12] S. Albadri, R. Septima, And H. Syahputra, "Sistem Informasi Manajemen Kepegawaian Berbasis Web," *Jurnal Teknik Informatika Dan Elektro*, Vol. 6, No. 1, Pp. 111–120, Mar. 2024, Doi: 10.55542/Jurtie.V6i1.993.

[13] E. Effendi, S. Harahap, And H. M. Rambe, "Komponen Sistem Informasi," *Jurnal Pendidikan Dan Konseling*, Vol. 5, No. 2, Pp. 5076–5080, 2023, Doi: 10.31004/Jpdk.V5i2.14317.

[14] I. Hasridayyana, L. Ahmad, And R. Junaidi, "Sistem Informasi Manajemen Kos-Kosan Menggunakan Metode Content Management System Di Kota Banda Aceh," *Jurnal Sistem Komputer (Siskom)*, Vol. 3, No. 2, Pp. 84–94, Aug. 2023, Doi: 10.35870/Siskom.V3i2.797.

[15] H. Mantik, "Penerapan Payment Gateway Sebagai Jembatan Transaksi Yang Aman Di Era Digital. Studi Kasus Pt Abc," *Jurnal Sistem Informasi Universitas Suryadarma*, Vol. 11, No. 1, Jun. 2024, Doi: 10.35968/Jsi.V11i1.1127.

[16] 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, Apr. 2024, Doi: 10.25126/Jtiik.20241127680.

[17] M. D. Anggraeny, A. Kurniawati, And D. Anggraini, "Tingkat Kematangan Implementasi Scrum Menggunakan Scrum Maturity Model Pada Direktorat Tsi," *G-Tech: Jurnal Teknologi Terapan*, Vol. 8, No. 1, Pp. 247–254, Dec. 2023, Doi: 10.33379/Gtech.V8i1.3690.

[18] D. Pungki, A. Aziz, And H. Santoso, "Optimasi Delivery Produk Menggunakan Agile Scrum Pada Pengembangan Aplikasi Monitoring Mbkm Unikama," *Jati (Jurnal Mahasiswa Teknik Informatika)*, Vol. 8, No. 6, Pp. 11650–11656, Nov. 2024, Doi: 10.36040/Jati.V8i6.11436.

[19] F. Perdana, D. S. Y. Kartika, And R. Permatasari, "Pengujian Sistem Informasi Manajemen Aset Pada Perusahaan Perkebunan Menggunakan Metode Black Box Testing," *Jati (Jurnal Mahasiswa Teknik Informatika)*, Vol. 8, No. 3, Pp. 4011–4016, Jun. 2024, Doi: 10.36040/Jati.V8i3.9809.