WEB-BASED WASTE BANK APPLICATION DEVELOPMENT

Conduct by:

Name : Muhammad Al Fikri Sitorus
Student ID : 21523250

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2026

SUPERVISOR ENDORSEMENT PAGE

WEB-BASED WASTE BANK APPLICATION DEVELOPMENT

THESIS

Conduct By:

Name : Muhammad Al Fikri Sitorus

Student ID : 21523250

Yogyakarta, January 9th, 2026

Advisor,

(Moh. Idris, S.Kom., M.Kom.)

EXAMINER ENDORSEMENT PAGE

WEB-BASED WASTE BANK APPLICATION DEVELOPMENT

THESIS

Has been defended in front of the examiners as one of the requirements to obtain a Bachelor of Informatics degree from the Undergraduate Program in Informatics at the Faculty of Industrial Technology, Universitas Islam Indonesia

Yogyakarta, January 9th, 2026

Chair

Moh. Idris, S.Kom., M.Kom. _____

Examiner 1

Ir. Irving Vitra Paputungan, S.T., M.Sc., Ph.D. 03/02/2026 _____

Examiner 2

Hari Setiaji, S.Kom., M.Eng. _____

Acknowledged by,
Head of Undergraduate Program in Informatics
Faculty of Industrial Technology
Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., IPM., ASEAN Eng.)

AUTHENTICITY STATEMENT

The undersigned:

Name	:	Muhammad Al Fikri Sitorus
Student ID	:	21523250

Final project with title:

WEB-BASED WASTE BANK APPLICATION DEVELOPMENT

Stating that all components and contents in this final project are my own work. If in the future it is proven that some parts of this work are not my own work, the final project submitted as my own work is ready to be withdrawn and ready to bear any risks and consequences.

Thus this statement letter is made, hopefully it can be used properly.

Yogyakarta, February 6th, 2026

(Muhammad Al Fikri Sitorus)

DEDICATION

In the name of Allah SWT, the Most Gracious and the Most Merciful. By His permission, this final project can be completed. With all humility, I dedicate this work to my parents, who have never tired of praying for me, supporting me, and loving me unconditionally. To the teachers and mentors who have been my guiding light on my journey of knowledge. To the friends who have always been faithful companions. And to myself, for the courage, determination, and patience I have endured throughout this process.

MOTTO

Bismillahirrahmanirrahiim

“Semua jatuh bangunmu hal yang biasa, angan dan pertanyaan waktu yang menjawabnya, berikan tenggat waktu bersedihlah secukupnnya, rayakan perasaanmu sebagai manusia.”
(Hindia)

“Allah does not burden a person but according to his ability.”
(Al Baqarah 286)

FOREWORD

Alhamdulillahi rabbil ‘ālamīn, Praise be to Allah, the Most Gracious and the Most Merciful. Thanks to His countless gifts, guidance, and mercy, the author has successfully completed this final project, entitled "Website-Based Waste Bank Application Using the Laravel Framework." May blessings and peace always be upon the Prophet Muhammad, his family, companions, and followers who consistently emulate his teachings.

This final project was prepared as one of the requirements for obtaining a bachelor's degree in Informatics, the Islamic University of Indonesia. During the process of preparing this final project, the author received significant assistance, support, and guidance, both directly and indirectly, from various parties. Therefore, I would like to take this opportunity to express my deepest gratitude to:

1. My beloved family, father, mother and sister, who always give endless prayers and support so that the author can reach this point.
2. Mr. Moh. Idris,S.Kom., M.Kom., as my final project supervisor, for his knowledge, guidance and dedication in guiding the author through the completion of this Final Project.
3. Mr. Irving Vitra Paputungan, S.T., M.Sc., Ph.D., as the author's Academic Advisor, for his guidance from the beginning to the end of the undergraduate study period.
4. Mr. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., as the Head of the Undergraduate Program in Informatics Study Program.
5. My fellow IP students/Class of 21, who have become a second family throughout my studies. Thank you for all your support, togetherness, laughter, discussions, motivation, and unwavering enthusiasm from the beginning of my studies through the process of writing this final project. Your presence has not only helped me academically, but also provided mental and emotional strength in facing every process and challenge. May the friendship and togetherness that have been built continue to be maintained well into the future.
6. Friends whom I cannot mention individually, but who have always been there to provide support, companionship, and encouragement throughout my life and studies. Special thanks to FAFFA, TEAM GACOR and INPO MAIN ZUM for their prayers, motivation, laughter, and camaraderie, which have been a vital part of my journey to the completion of this final assignment.

The author acknowledges that this Final Project is far from perfect, both in terms of structure and writing. Therefore, the author welcomes constructive criticism and suggestions from readers, which will serve as a valuable reference for future improvement. May this Final Project contribute to the readers' knowledge and be beneficial for the advancement of science and knowledge. Wassalamu'alaikum Wr.Wb

Yogyakarta, January 9th, 2026

(Muhammad Al Fikri Sitorus)

ABSTRACT

Waste management remains a complex challenge in many communities, particularly in waste banks that still rely on manual recording systems. Conventional data processing frequently leads to several issues such as inaccurate records, risk of data loss, errors in deposit logging, delays in report generation, and lack of transparency in user transactions. These problems affect operational efficiency, limit access to information, and reduce public trust in waste bank management. Therefore, a technology-based solution is required to facilitate structured, efficient, and accessible management of waste bank data.

This study aims to develop a web-based waste bank application to support administrators in managing user data, recording waste deposits, monitoring balances and transactions, and generating real-time reports. The application is developed using the Laravel framework, selected for its stability, strong security features, and reliable development support. The system development adopts the Waterfall model, consisting of requirement analysis, system design, implementation, testing, and maintenance stages. The system also utilizes Application Programming Interface (API) integrations to support specific digital functionalities and enhance system usability. In addition, the application interface is designed to be user-friendly and accessible for both administrators and waste bank users.

System testing is carried out using two approaches, namely User Acceptance Testing (UAT) to ensure conformity between system functionality and user needs, and System Usability Scale (SUS) to measure the level of usability from the user's perspective. The testing results indicate that the system operates effectively and meets the defined requirements. UAT results show that all key features are accepted and function as expected, while SUS testing produces a score of 65.95, indicating that the system has good usability and is feasible for implementation.

In conclusion, the developed web-based waste bank application provides an effective solution for digital waste bank management. The system is expected to improve operational efficiency, minimize recording errors, enhance transparency, and support more modern and integrated waste management practices. Furthermore, the findings of this study are expected to contribute as a reference for future research and development of digital waste bank systems.

Keywords: waste bank, information system, web-based application, Laravel, Waterfall, User Acceptance Testing, System Usability Scale.

GLOSSARY

Application Programming Interface (API)	An interface that enables systems to communicate and exchange data automatically with other systems or digital services.
Bank Sampah (Waste Bank)	A community-based waste management system that applies banking principles, where users deposit waste and receive economic value in return.
Dashboard	The main page of the system that displays key information summaries visually and in real time.
Database	A structured collection of data that is stored systematically and can be accessed, managed, and updated easily.
Filament	An admin panel framework built on Laravel, used to develop modern, efficient, and interactive data management interfaces.
Framework	A structured environment consisting of tools, libraries, and standards that facilitates application development.
Laravel	A PHP framework based on the Model–View–Controller (MVC) architecture used to build structured, secure, and efficient web applications.
Midtrans	A digital payment gateway service that provides various payment methods such as QRIS, virtual accounts, e-wallets, and other electronic payment services.
Model–View–Controller (MVC)	A software architecture pattern that separates application logic into Model (data), View (interface), and Controller (process flow).
System Testing	A process used to ensure that the system operates correctly and meets user and functional requirements.
System Usability Scale (SUS)	A usability evaluation method using a standardized questionnaire to measure the ease of use of a system, producing a quantitative score.

User Acceptance Testing (UAT)	A testing method involving end users to ensure that the system meets user needs and is ready for implementation.
User Experience (UX)	The overall experience and level of satisfaction felt by users while interacting with the system.
User Interface (UI)	The part of the system that directly interacts with users, consisting of screens, menus, buttons, icons, and other visual elements.
Waste Bank Member	A registered user of the waste bank who deposits waste and receives balance value based on recorded transactions.
Waterfall Model	A sequential software development method consisting of requirement analysis, system design, implementation, testing, and maintenance.
Web-Based Application	A software application that operates through a web browser and can be accessed via the internet.
Web Server	A software or service that receives requests from web browsers and responds by delivering web pages.

TABLE OF CONTENTS

LIST OF TABEL

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Background

The waste problem is a global issue that has a direct impact on environmental quality and public health. Indonesia is one of the countries with the largest waste generation in Southeast Asia. Based on a report by the Ministry of Environment and Forestry (Lingkungan & Kehutanan, 2022) , Indonesia produces more than 67 million tons of waste each year, and around 40% consists of inorganic waste such as plastic, metal, and glass—types of waste that require hundreds of years to decompose and have the potential to contaminate soil, water, and

the food chain. Increasing consumption, urbanization, and the lack of good waste management practices are the main factors that worsen this condition.

As a mitigation measure, one approach developing in various regions of Indonesia is the Waste Bank, a community-based waste management system that adopts banking principles. Under this concept, customers deposit sorted inorganic waste, which is then weighed and converted into savings (Putra & Bhakti, 2024) .

Along with the development of information technology, web-based digital systems have been proven capable of improving the quality of waste bank management. Studies by (Rahmah & Theresiawati, 2022) show that web-based information systems can accelerate transaction recording processes, improve data accuracy, and create transparency for customers. Frameworks such as Laravel are widely used because they provide a Model-View-Controller (MVC) architecture, built-in security, and easy API integration that supports financial transaction management (Putra & Bhakti, 2024; Santoso et al., 2021) .

However, most previous studies are still limited to simple CRUD functions and do not pay attention to secure and low-risk transaction verification mechanisms. In the context of waste banks, errors or data manipulation related to waste weight and transaction value can lead to balance discrepancies and disrupt accountability. Therefore, a Dual Verification (Maker–Checker) mechanism is needed, which is a validation procedure where transactions inputted by field officers must be re-verified by an admin or supervisor. This mechanism is commonly used in financial systems to prevent fraud and increase data integrity (Saranya & Naresh, 2022).

Besides security aspects, waste bank transactions are generally still conducted using cash, making them prone to cash discrepancies, miscommunication, and the absence of payment records. Integrating the Midtrans payment gateway becomes a solution that enables faster, more secure, and well-documented digital transactions. The implementation of Midtrans has been proven to increase transaction efficiency and provide an accurate audit trail in various web-based applications (A. M. Kurniawan & Ariessanti, 2025).

Beyond simply savings transactions, the waste bank information system also has the potential to serve as a participatory fundraising tool through a digital donation feature. Donations are made voluntarily by customers as a form of social contribution for the common good. Donation funds are not used for individual interests but are managed collectively to support the sustainable operations of the waste bank, such as the procurement of trash bins, the purchase of waste management support facilities, and training in waste processing and recycling. Midtrans' integration with the donation feature allows for a cashless, secure,

transparent donation process with clear transaction evidence, thereby enhancing trust and accountability in fund management.

Based on these problems, this research focuses on the development of a web-based Waste Bank Information System with the implementation of a Dual Verification mechanism and the integration of the Midtrans Payment Gateway for transactions and donations, in order to increase the efficiency, accuracy, security, and transparency of inorganic waste management.

1.2 Problem Formulation

Based on the background, the problem formulations for this research are:

a. How to design and develop a web-based waste bank information system with Laravel and Filament technology?
b. How to implement the Dual Verification (Maker–Checker) mechanism to increase the accuracy and security of waste deposit transaction data?
c. How to integrate the Midtrans payment gateway to support non-cash transactions securely and transparently?

1.3 Research Objectives

The objectives of this research are:

a. To develop a web-based waste bank information system.
b. To implement the Dual Verification mechanism in the waste deposit transaction process.
c. To integrate the Midtrans payment gateway to support non-cash transactions.

1.4 Research Scope

This research is limited by the following scopes:

a. The system only focuses on the management of inorganic waste.
b. System users include: Admin, Collector, Customer Groups, and Customers.
c. The Dual Verification mechanism is applied to waste deposit transactions.
d. The payment gateway used is Midtrans with available payment methods (QRIS/VA).

1.5 Research Benefits

Theoretical Benefits

a. To contribute to research related to waste bank digitalization and the Dual Verification mechanism in community systems.

b. To serve as an academic reference in the development of web-based information systems with payment gateway integration.

Practical Benefits

a. To support waste bank management to be more efficient, structured, and transparent.
b. To make it easier for customers to access balance information and transaction history.
c. To reduce the risk of recording errors and data manipulation.
d. To provide a secure and documented non-cash transaction system.

1.6 Writing Systematics

This thesis is structured into five chapters as follows:

a. Chapter I Introduction

This chapter describes the context of the development of the waste bank information system, including the background of the problem, problem formulation, research questions, research limitations, and research objectives to be achieved. In addition, this chapter also discusses the research benefits, an overview of the research methods, and the writing systematics used to explain the content and flow of the final project report.

b. Chapter II Literature Review

This chapter discusses the theories and concepts that form the basis for the development of the waste bank information system. The discussion includes explanations of the concepts of waste bank management and digitalization of waste management, web-based information systems, the Laravel framework as a development platform, and Filament as the admin panel. Furthermore, this chapter explains the software development method used, namely the Waterfall method, as well as system testing methods including User Acceptance Testing (UAT) and System Usability Scale (SUS). This chapter also explains the use of Application Programming Interface (API), including payment gateway integration such as Midtrans, and presents previous research and similar applications as the basis for establishing the research gap and the innovation of the developed system.

c. Chapter III Research Methodology

This chapter explains the stages of research and development of the waste bank information system carried out in this study. The initial stages begin with identifying user needs through preliminary studies and system requirements analysis. Next, a comparative study of waste bank applications or similar systems is conducted as a reference in determining features. System development is carried out using the Waterfall method,

which consists of requirements analysis, system design, implementation, testing, and maintenance stages. The design stage includes system modeling, database design, and user interface design. Implementation is carried out using the Laravel framework supported by Filament, followed by the testing stage using UAT (consisting of alpha testing and beta testing) and SUS to measure the system's usability level. The maintenance stage is carried out based on user evaluation results.

d. Chapter IV Results and Discussion

This chapter presents the results of the entire development process of the waste bank information system. The initial stages, such as needs identification and analysis of similar applications, are explained briefly to show their role in forming system requirements. The main focus of this chapter is on the results of system development using the Waterfall method, including the results of requirements analysis, design, feature implementation, system testing results, and their systematic discussion to demonstrate the effectiveness and performance of the developed system.

e. Chapter V Conclusion and Suggestion

The final chapter contains the conclusions of the research that has been carried out, summarizing the results of system development and the achievement of research objectives. In addition, this chapter presents suggestions that can be considered for further system development and research.

CHAPTER II
LITERATURE REVIEW

2.1 Waste Management and the Waste Bank Concept

Waste is one of the biggest environmental problems faced by Indonesia. The increase in population, urbanization, and economic activities causes the volume of waste to increase every year. If not managed properly, waste can cause negative impacts such as environmental pollution, health problems, and a decline in people's quality of life. Therefore, effective, integrated, and sustainable waste management becomes an important necessity (Lingkungan & Kehutanan, 2022).

One effective approach in waste management is the waste bank concept. A waste bank is a community-based waste management system that applies principles similar to banking institutions, where the community as customers deposits sorted inorganic waste, which is then weighed, given an economic value, and converted into savings balance. This approach not only functions as an environmental solution but also encourages public awareness of waste management and provides economic benefits (Putra & Bhakti, 2024).

However, the implementation of waste banks in various regions still faces challenges, especially related to data and administrative management. Most waste banks still perform manual recording, which is prone to errors, difficult to trace, slow in reporting, and lacks transparency. This condition affects data accuracy and public trust. Therefore, digitalization of waste banks is an important solution to improve the efficiency, accuracy, and transparency of management (Widiyati & Wijayati, 2025).

2.2 Digital Transformation in Waste Bank Management

The development of information technology provides great opportunities to increase the efficiency of waste banks through the digitalization of business processes. Web-based information systems enable transaction recording, weighing, customer data management, and report generation to be carried out more quickly and accurately.

The study by (Rahmah & Theresiawati, 2022), shows that web-based digital systems can accelerate waste bank management processes, reduce human error, and facilitate data access for managers and customers. The study by (Kadhem et al., 2025) on the E-WASTE application proves that digitalization increases community participation and improves the accuracy of deposit recording.

(Gustin et al., 2025) also found that a responsive web-based waste bank system is able to replace manual recording and support real-time monitoring, thus increasing operational transparency. The use of digital technology in waste banks not only increases efficiency but also strengthens public trust in the management.

2.3 Web-Based Information Systems

A web-based information system is a system that runs through a browser and can be accessed via the internet. This system has several advantages such as high accessibility, ease of maintenance, integration capability, and scalability. With a web-based system, users can access information from various devices without requiring special application installation.

In waste bank management, a web-based system allows customer data management, deposit recording, balance management, transaction reporting, and monitoring to be carried out in real time, structured, and secure. This enables better transparency and accountability in waste bank operations.

2.4 Waterfall Method

The software development model known as the Waterfall method, or often called the classical life cycle, is a structured and sequential system development approach. This model describes the software development process that starts from requirements analysis, design, implementation, testing, to system maintenance. The method was first introduced by Winston Royce in 1970 and is still widely used today because of its systematic flow and well-documented process.

The Waterfall method has a main characteristic in the form of a linear workflow. Each stage in development must be completed first before moving on to the next stage. This concept is analogous to a waterfall, where the process moves downward gradually and in an orderly way. Although this method is often considered rigid and less responsive to changing requirements, Waterfall remains relevant, especially for projects that have clearly defined system requirements from the beginning. The complete stages of the Waterfall method can be seen in Figure 2.1.

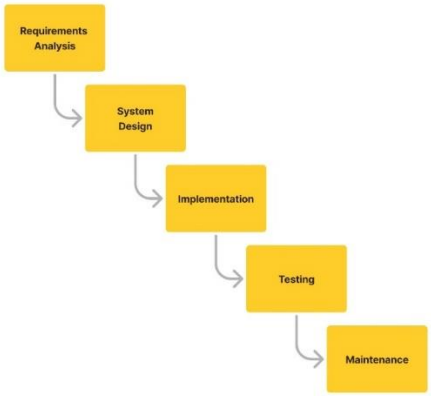


Figure 2. 1 Waterfall Method

The Waterfall method has several advantages, including:

1. Clear focus on each development phase, which can improve system quality and minimize the risk of errors.

2. Phase-by-phase approach facilitates project planning, control, and documentation.
3. Each stage produces structured and complete development documents.
4. The development process is relatively easier to understand and manage, including by less experienced teams.
5. Systematic stages allow regular review and evaluation before moving to the next phase.

However, the Waterfall method also has some weaknesses that need attention, namely:

1. The linear process makes this method less flexible to changes in user requirements.
2. Errors that occur in the early stages of development can have a major impact on later stages and require significant effort to fix.
3. Development time tends to be longer because each stage must be completed before entering the next stage

2.5 Laravel Framework

Laravel is a modern PHP framework that implements the Model–View–Controller (MVC) architecture. Laravel provides various features such as authentication, authorization, Eloquent ORM, structured routing, middleware, and CSRF protection, thus supporting the development of secure, structured, and maintainable systems (Santoso et al., 2021).

In addition, Laravel has a large community and complete documentation that facilitate the development process. In a waste bank information system, Laravel is chosen because it can support data security, development flexibility, and system reliability (Sinlae et al., 2024). The Laravel MVC architecture can be seen in Figure 2.2.

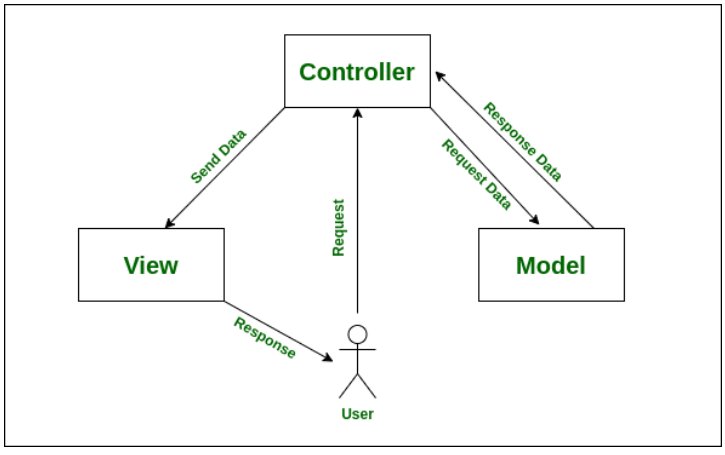


Figure 2. 2 MVC Laravel Artchitecture

2.6 Filament PHP as Admin Panel

Filament is a Laravel-based administration panel that uses the TALL Stack (Tailwind, Alpine.js, Laravel, Livewire). Filament provides a modern administration interface equipped with features such as CRUD generator, interactive tables, statistical charts, user management, roles and permissions, and system notifications (Sudarsono & Vebriandi, 2025).

In the context of waste banks, Filament is highly relevant because it supports transaction data management, weighing verification, report generation, and monitoring of waste bank activities in real time (Wijanarko, 2025).

2.7 Unified Modeling Language (UML)

Unified Modeling Language (UML) is a standard modeling language used to visualize, design, and document software systems. UML helps identify system requirements, describe business processes, and facilitate communication between developers and stakeholders before entering the implementation stage (Siska Narulita et al., 2024).

The roles of UML in system development include:

1. Helping to understand system requirements.
2. Serving as a communication tool between the development team and users.
3. Simplifying system complexity through visualization.
4. Serving as a guideline for system implementation.

2.8 Dual Verification (Maker–Checker) Mechanism

Dual Verification or Maker–Checker is an internal control mechanism where a transaction must go through a validation process by two different parties. The Maker is responsible for creating or inputting the transaction data, while the Checker is responsible for verifying and approving the transaction.

This concept is widely applied in financial and banking systems to prevent fraud, prevent data manipulation, and ensure transaction integrity (Saranya & Naresh, 2022) Systems with a maker–checker mechanism have a clearer audit trail and can significantly reduce administrative errors.

In waste bank management, this mechanism is very important because:

1. waste weight can be incorrectly entered by field officers,
2. transaction value potentially does not match,
3. cross-verification is needed to maintain customer balance accuracy.

With dual verification, a transaction is only considered valid after being verified by the admin, thereby increasing system security, trust, and accountability.

2.9 Payment Gateway and Midtrans

A payment gateway is a service that facilitates digital payment transactions such as QRIS, debit cards, credit cards, and virtual accounts. Payment gateway integration is used to simplify the payment process, reduce the risk of errors, and provide an automatic audit trail.

Midtrans is one of the largest payment gateways in Indonesia that provides structured APIs, real-time callback notification systems, and various payment methods such as QRIS, Virtual Accounts, and e-wallets. (A. M. Kurniawan & Ariessanti, 2025) show that Midtrans integration increases transaction efficiency and provides accurate payment data in accounting information systems. The study by (Djuwitaningrum & Jati, 2025) strengthens this finding by proving that Midtrans improves user convenience and security in digital payment transactions. Midtrans integration into the waste bank system will:

1. reduce cash handling risk,
2. increase transaction transparency,
3. provide a digital payment trail,
4. accelerate the process of disbursing or paying customer balances.

2.10 System Testing

2.10.1 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is a testing stage that aims to ensure that the developed system can be accepted by users because it has met their needs, expectations, and usage objectives. UAT does not only assess whether the system runs well technically, but emphasizes the suitability of the system to user needs in real usage conditions (Rahmah & Theresiawati, 2022) Thus, UAT becomes an important stage to ensure that the system is truly useful, easy to use, and relevant for end users.

In this research, testing is carried out using two approaches, namely alpha testing and beta testing. Alpha testing is conducted internally by developers to ensure that all main system functions run according to the design. This stage aims to identify bugs, functional mismatches, and ensure that the system has met the predetermined functional requirements. At this stage,

the black box testing method is used, where testing focuses on the suitability of system functions without directly looking at the program code (Menora et al., 2023).

After the system is declared stable in alpha testing, the process continues to beta testing involving end users directly. At this stage, users will try the system in usage scenarios that approach real conditions. The goal of this stage is to obtain feedback from users regarding comfort of use, ease of understanding features, and suitability of the system with operational needs. The results of the beta testing process are used as a basis for evaluating and refining the system so that it can be used optimally (Menora et al., 2023).

2.10.2 System Usability Scale (SUS)

System Usability Scale (SUS) is a widely used usability evaluation method designed to measure the perceived usability of a system based on user responses. SUS provides a general overview of how users evaluate their experience when using a system, particularly in terms of ease of use, efficiency, and interaction comfort (Brooke, 1996).

The SUS method uses a standardized questionnaire consisting of ten statements, rated by respondents using a five-point Likert scale. The responses are then converted into a final score ranging from 0 to 100, where a higher score indicates better usability. In general, SUS scoring is calculated by converting responses from odd and even items and multiplying the total score by 2.5 to obtain the final score (Brooke, 1996).

SUS is commonly applied because it is practical, quick to administer, and does not require complex testing procedures. In addition, SUS produces quantitative results that can be used to compare usability levels across different systems. Therefore, SUS is suitable for evaluating web-based applications, including the waste bank information system developed in this research.

2.11 Previous Research

Previous research is used to review various approaches, methods, and findings that have been conducted related to waste bank management and web-based information system development. This review aims to identify the extent to which previous studies have discussed waste bank digitalization, the technologies applied, and the limitations that remain, in order to formulate the research gap that becomes the basis of this study.

Several studies have developed waste bank applications and information systems to replace manual processes and improve operational efficiency. For example, (Rahmah &

Theresiawati, 2022) and (Atin et al., 2022) implemented web-based systems using the Waterfall method to support transaction recording and report generation. (Umar et al., 2025) also developed a web-based application that accelerates data management, although it still lacks non-cash transaction support. In addition, (Assaid et al., 2022) highlighted common problems in manual waste bank operations, such as human error and low transparency. Other studies, such as (A. M. Kurniawan & Ariessanti, 2025) and (Kadhem et al., 2025), emphasized that digitalization can improve transparency and data accuracy, yet important control mechanisms are still limited. A summary of relevant previous research is presented in Table 2.1.

Tabel 2. 1 Previous Research

No.	**Reference**	**Objective**	**Method**	**Result**
1	(Rahmah & Theresiawati, 2022)	To develop a waste bank information system to replace manual recording	Waterfall, web-based information system	The system increases the efficiency of transaction recording and facilitates report generation
2	(Assaid et al., 2022)	To analyze problems in manual waste bank management	Field study and system analysis	Found human error and low transparency due to the manual system
3	(Atin et al., 2022)	To design a web-based waste bank information system	Waterfall and UML	The system is able to manage waste bank data, but does not yet have a transaction security mechanism
4	(Umar et al., 2025)	To develop a web-based waste bank application	Prototype model	The application speeds up data management, but does not yet support non-cash transactions
5	(Sinlae et al., 2024)	To evaluate the use of the Laravel framework in web application development	Literature riview	Laravel increases system development efficiency and security
6	(A. M. Kurniawan & Ariessanti, 2025)	To analyze the impact of waste bank digitalization on data transparency	Case study and observation	Digitalization increases the transparency and accuracy of transaction recording
7	(Kadhem et al., 2025)	To analyze the impact of waste bank digitalization on data transparency	System development and user evaluation	The system increases participation and recording accuracy, but has not

No.	Reference	Objective	Method	Result
				integrated dual verification and payment gateway

Based on Table 2.1, it can be concluded that previous studies have primarily focused on basic transaction recording, data management, and reporting features as the main outcomes of waste bank digitalization (Atin et al., 2022; Rahmah & Theresiawati, 2022; Umar et al., 2025). However, most of these studies have not comprehensively addressed stronger internal control aspects, such as structured transaction verification mechanisms and secure digital payment integration (Assaid et al., 2022; Kadhem et al., 2025). This limitation becomes an important foundation for developing a more reliable waste bank information system in this research.

2.12 Research Gap

Although various previous studies have discussed waste bank digitalization, several research gaps remain unresolved.

First, most developed waste bank systems still emphasize basic CRUD (Create, Read, Update, Delete) functions and transaction recording, without sufficient internal control mechanisms to validate the accuracy and legitimacy of transaction data (Atin et al., 2022; Rahmah & Theresiawati, 2022; Umar et al., 2025). This limitation is critical because waste transactions involve weighing and valuation processes that may be prone to input errors.

Second, the Dual Verification (Maker–Checker) mechanism, which is commonly applied in financial and banking systems, has not been widely implemented in waste bank contexts. This mechanism is important to reduce the risk of data manipulation and to strengthen accountability in transaction validation (Kadhem et al., 2025).

Third, many waste bank systems still rely on cash-based transactions, which can lead to limited audit trails and lower financial transparency. Therefore, integrating a payment gateway as a secure and well-documented non-cash transaction solution is still rarely found in waste bank research (Umar et al., 2025).

Based on these gaps, this research develops a web-based Waste Bank Information System that integrates a Dual Verification mechanism and a payment gateway. This approach is expected not only to improve operational efficiency but also to strengthen transaction security, accuracy, and transparency, thereby providing both scientific and practical contributions compared to previous studies.

CHAPTER III
RESEARCH METHODOLOGY

3.1 Research Framework

Preliminary research was conducted to obtain an initial overview of the condition of the research object and the problems occurring in waste bank management. This stage aims to understand the current workflow, identify the constraints faced, and formulate the initial requirements that form the basis for application development.

Based on initial observations, the waste bank management process still faces several problems, especially in transaction recording, customer data management, and monitoring of waste collection activities. The recording process, which is still carried out manually, has the potential to cause recording errors, delays in report preparation, and limited access to information for interested parties.

In addition, the absence of an integrated system means that information related to transactions and waste savings balances is difficult to access quickly and accurately. This condition has an impact on the low management efficiency and suboptimal transparency in waste bank operations.

3.2 Comparison of Similar Applications

Comparison of similar applications is carried out to obtain an overview of systems or applications that have existed previously and have functions similar to the system to be developed. This stage aims to identify common features available, their advantages, and the limitations still found in similar applications, so that they can be taken into consideration in the design of the system to be developed.

Based on a review of several existing waste bank management applications and systems, generally, similar applications have provided basic features such as waste transaction recording, customer data management, and presentation of transaction reports. These applications help the waste bank management process compared to manual recording, especially in terms of data storage and processing.

However, several limitations are still found in similar applications. Some systems do not fully support structured user access rights management according to roles, so the division of duties and responsibilities between users is not yet optimal. In addition, transaction data transparency and transaction validation processes in some applications are still limited, which has the potential to cause recording errors or data discrepancies.

Based on the results of this comparison, the system to be developed in this research is designed to complement and perfect the features already present in similar applications. The focus of development is directed at clearer user access rights management, structured transaction recording, and integrated data management mechanism support. Thus, the developed system is expected to be able to provide a solution that is more in line with the operational needs of the waste bank.

3.3 Application Development

This subsection outlines the application development design for the web-based waste bank information system developed in this research. Application development is carried out using the Waterfall method, which was selected because it provides a structured and systematic software development framework with clear sequential stages and well-documented outputs for each phase (Haniva et al., 2023; Pressman & Maxim, 2019).

The Waterfall method is used as a reference in this research because each stage is conducted in a logical order, where the output of one stage becomes the input for the next stage. The development stages include: requirements analysis, system design, implementation, testing, and maintenance (Haniva et al., 2023). Each stage plays an interconnected role in ensuring that the system is developed in an organized, measurable, and traceable manner.

Therefore, the development stages in this research are planned to be carried out sequentially, starting from requirements analysis up to system maintenance. The application development flow using the Waterfall method in this research is shown in Figure 3.1.

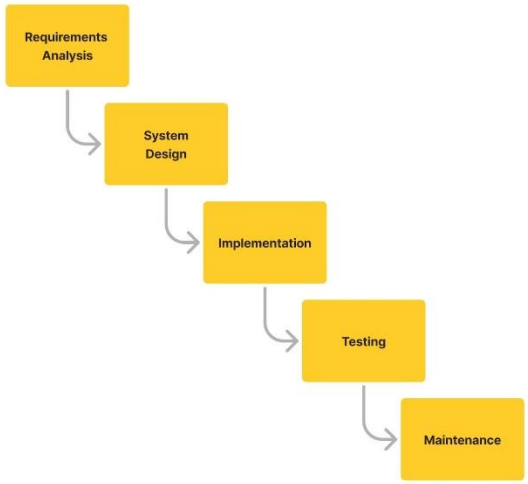


Figure 3. 1 Waterfall Sysstem Development Method

3.3.1 System Requirements Analysis

Requirements analysis is the initial stage in application development which aims to identify and define system requirements comprehensively. This stage is carried out to ensure that the developed waste bank information system application is able to meet user needs and support waste bank operational processes effectively and structured.

Requirements analysis is carried out based on the results of preliminary research and data collection techniques that have been carried out previously. The information obtained is used to determine the functional and non-functional system requirements that form the basis for the application design and development process.

Data Collection

Data collection in this research was carried out through a review process of similar research and similar applications that have been discussed in the previous chapter (Chapter II). This process aims to obtain a deeper understanding of the features, system flow, and approaches that have been applied to the development of waste bank information systems and similar applications.

Through a review of similar research and applications, the researcher identifies the general needs of users and the limitations that are still found in existing systems. This

information is used to obtain a comprehensive overview of the potential user needs, especially in data management, transaction recording, and structured information presentation.

The data obtained from this review process is then used as a basis for analyzing the system requirements to be developed. With this approach, the requirements analysis is carried out based on the findings that have been systematically reviewed in previous research, so that the formulated system requirements remain relevant and appropriate to the context of waste bank management.

3.3.2 Design

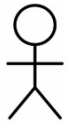
System design is a continuation stage after requirements analysis which aims to describe the design of the system structure and workflow before implementation. This design stage is carried out to ensure that the waste bank information system to be developed has a process flow and data structure that is in accordance with user needs.

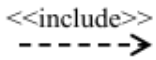
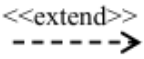
System design in this research includes the design of interaction between users and the system, the design of activity flow, the design of the database structure, and the design of the user interface. All designs compiled at this stage are used as a reference in the application development process at the next stage.

Use Case Diagram

A Use Case Diagram is a visualization of system components which include actors, use cases, and the relationship between these components. This diagram is used to explain the functional design of the system to users and map all the features planned in the waste bank information system. Through this diagram, developers can compile system requirements in a structured manner based on the role of each actor (Siska Narulita et al., 2024). An explanation of the symbols used can be seen in Table 3.1.

Tabel 3. 1 symbols in Use Case Diagram

Sysmbols	Name Component	Deskription
	Actor	Represents individuals, other systems, or devices that use the system.
	Use Case	Representation of the main service or function provided by the system to the actor (user).

Sysmbols	Name Component	Deskription
	Subject	Indicates a system being modeled that includes all use cases.
	Association	Describes the interaction between the actor and the function or service they access.
	Include Relationship	Indicates that a function depends on another function to run.
	Extend Relationship	Indicates that an additional function is performed under certain conditions.
	Generalization Relationship	Indicates that a function is a more general form of another (more specific)

This diagram will be created using the draw.io tool to simplify visual creation and understanding. The diagram will include the actors involved in the system and the main services available to them. For example, Figure 3.2 shows the implementation of a Use Case Diagram in the Development of a Correspondence Application, where there are four main actors, namely superior, admin, work unit, external. The leader actor has access to functions such as sending outgoing mail, editing work unit data, editing user data, etc. Meanwhile, the secretary/admin actor can receive incoming mail, send outgoing mail to the work unit/external, and the work unit/external actor can receive outgoing mail from the secretary and send mail to the secretary.

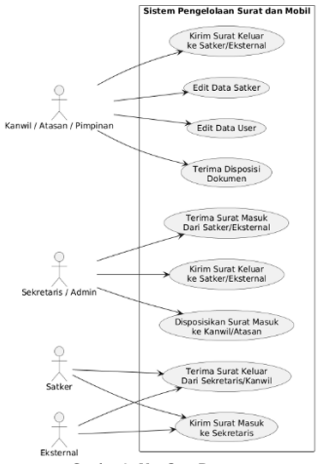


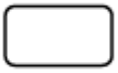
Figure 3. 2 Ecample Use Case Diagram in Correspondence Application Development

Source: (Mustapa et al., 2025)

Activity Diagram

An Activity Diagram is used to represent the flow of processes or activities within the system, starting from the start point, decision process (branching), to the end process. This diagram also functions to visualize the steps or stages that occur sequentially or in parallel when the system is executed. In its design, at least one activity diagram represents each use case to describe the running process in detail (Siska Narulita et al., 2024) Some common symbols used are explained in Table 3.2.

Tabel 3. 2 Activity Diagram Symbols

Symbol	Name	Description
	Start Point	Indicates the start point of a process or activity.
	End Point	Marks the end point of the activity flow.
	Activity	Represents a step or action taken in the system.
	Decision Node	Indicates the branching of the flow based on certain conditions.
	Control Flow	Connects elements in the diagram, showing the direction of the process flow.
	Swimlane	Shows the division of roles or responsibilities in a process.

The activity diagram will also be designed using the draw.io tool to simplify visualization. As an illustration, Figure 3.3 below shows an example of Activity Diagram implementation in the note report process. This diagram illustrates how the activity begins by selecting the note menu, after which the system displays the report filter menu. The admin then selects the date, month, and year they want to view, after which the admin views the note data, then the admin selects details, the admin can directly save the note and the system will display the saved note. If the admin wants to print the note, they can select save and the system will process for printing and the admin can see all the note data. And the admin can return to the note menu. And if the admin does not want to print, they can select cancel, then the system will process back to the note menu.

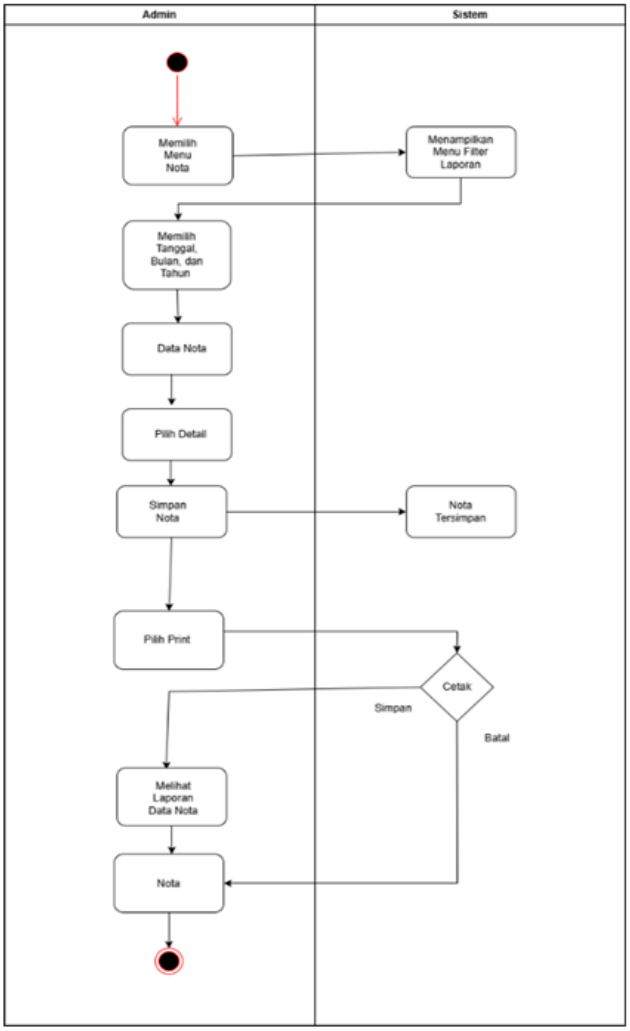


Figure 3. 3 Example Activity Diagram In The Note Report

Source: (Puspita, (2025).

Database Relationship Design

At this stage, the database relationship design is prepared to provide an initial overview of how data in the system will be connected and managed in an integrated and structured manner. Designing table relationships is a part of the system design phase, carried out before the system is fully implemented. The focus of this stage is to define how each table interacts with other tables through primary keys and foreign keys so that data integrity and consistency can be maintained.

In relational database systems, relationships between tables are used to describe logical connections between data stored in different tables. These relationships ensure that data redundancy can be minimized and that data can be accessed efficiently. The use of table relationships also helps developers understand the structure of the database as well as the flow of data that occurs within the system.

Cardinality is an important element in table relationships that defines the numerical association between two related tables. Cardinality explains the minimum and maximum number of data connections that can occur in a relationship. The common types of cardinality used in relational database design are as follows:

1. One-to-One (1:1) The one-to-one relationship is a relationship where one data in one entity is only related to one data in another entity, and vice versa. Example: One user account only has one user profile, and one user profile is only owned by one user account.
2. One-to-Many (1:N) or Many-to-One (N:1) The one-to-many relationship is a relationship where one data in one entity can be related to many data in another entity, but each data in that entity is only related to one data in the first entity. Example: One user can have many activity records, but each activity record is only owned by one user.
3. Many-to-Many (M:N) The many-to-many relationship is a relationship where one data in one entity can be related to many data in another entity, and vice versa. This relationship is generally represented using an associative entity. Example: One user can access many features in the system, and one feature can be accessed by many users.

The following is an example of a database design using Relationship Database Design with Crow's Foot Notation which can be seen in Figure 3.4.

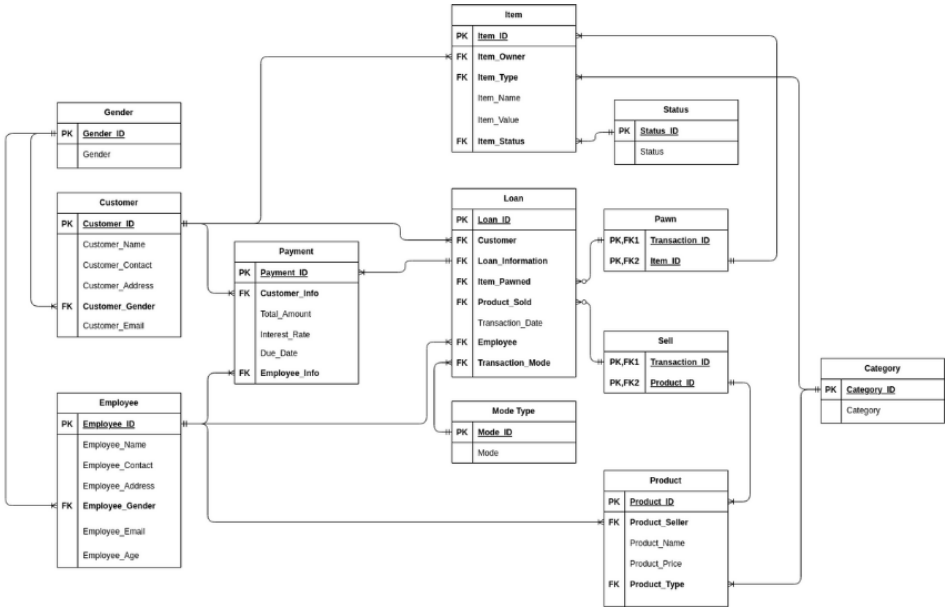


Figure 3. 4 Example Database Relationship Design

Interface Design

Application interface design is carried out using Figma, which is a web-based design tool that supports the creation of visual and interactive high-fidelity prototypes. The use of high-fidelity prototypes is chosen because it is able to represent the interface design more realistically, both in terms of visual appearance and user interaction flow.

Figma is used in the interface design stage because it provides features that support detailed interface design, such as setting the layout of components, placing interface elements in the form of menus, buttons, and input forms, and designing the navigation flow between pages. With these capabilities, the interface design can be compiled systematically and easily understood.

The interface design produced at this stage is used as a reference in the system implementation process at the next stage. With a clear interface design, the process of realizing the application display is expected to be carried out in a more focused manner and in accordance with the planned design.

3.3.3 Implementation

The implementation stage is an important stage in the application development process, where all the system designs and designs that have been compiled previously are realized into an application that can be run. At this stage, the system design is no longer conceptual, but begins to be realized into a system that has functions in accordance with the planning.

The implementation process is carried out using the PHP programming language supported by the Laravel framework. This framework was chosen because it provides an organized development structure and supports efficiency in application development through its various built-in features.

The first step in the implementation stage is to configure the development environment. The local development environment is prepared using Laragon, which provides a local server integrated with PHP, MySQL, and Apache or Nginx web servers. At this stage, the Laravel framework is installed along with the initial settings required so that the application can be run in the local environment.

After the development environment is ready, the implementation process continues with the application of the database design. The database structure that has been designed previously is realized using MySQL as the database management system. Development is carried out by utilizing Laravel's built-in features, such as user authentication management, routing settings, and session management. All features are developed by applying the Model-View-Controller (MVC) architecture which is integrated into the Laravel framework.

On the user interface side, the interface design that has been created using Figma is implemented into the application by utilizing the Blade Templating Engine. Blade is used to build dynamic interface pages based on data processed by the controller. During the interface implementation process, elements such as layout, navigation, and visual components are adjusted to the design that has been planned so that the application display remains consistent and easy to use.

3.3.4 Testing

The testing stage is one of the important stages in the application development process which aims to ensure that the developed system can run according to the specified specifications and meet user needs. Through this stage, the quality of the application's functions and usability can be evaluated before the system is used more widely.

The testing process in this research is planned using two main approaches. The first approach is User Acceptance Testing (UAT) which consists of alpha testing and beta testing. UAT is carried out to assess the suitability of the system with user needs and ensure that every feature developed can be used properly according to its role.

The second approach is the evaluation of the system usability level using the System Usability Scale (SUS) method. This method is used to measure user perceptions of the ease of

use and convenience of the system as a whole. The results of this testing process will be used as a basis for evaluating system quality and will be discussed further in Chapter IV as part of the research results and discussion.

User Acceptance Testing (UAT)

As explained in Chapter II, User Acceptance Testing (UAT) is carried out to ensure that the developed application can be used properly and is in accordance with user needs. This testing aims to validate the suitability of the system functions from the user's perspective before the application is used more widely.

UAT in this research is planned through two stages, namely alpha testing and beta testing. The alpha testing stage is carried out internally by the researcher as the system developer. At this stage, testing is focused on examining the main functions of the application to ensure that the system runs according to the designed specifications.

The testing method used in the alpha testing stage is black box testing, which is a testing method that focuses on testing the functionality of the system without paying attention to the structure of the program code. Testing is carried out by checking whether each feature can produce output that matches the given input. The main focus at this stage is to identify errors (bugs), ensure that features run well, and verify that the application workflow is in accordance with the design.

After the alpha testing stage is complete, testing continues with beta testing which involves end users. This stage aims to assess the experience of using the application from the user's perspective and identify problems that may not be detected in internal testing. Beta testing is carried out based on testing scenarios compiled in the form of test cases.

Tabel 3. 3 Example Test Case Table

ID Test	**Testing Scenario**	**Testing Steps**	**Expected Results**	**Status (Pass/Fail)**
UAT-01	User Authentication	1. Enter NIP and Password. 2. Click Login.	The system recognizes user access rights (Admin/Staff/Leader) and enters the dashboard.	Pass

ID Test	Testing Scenario	Testing Steps	Expected Results	Status (Pass/Fail)
UAT-02	Outgoing Mail Submission	1. Click the "Surat Keluar" menu. 2. Fill in the form and upload the mail draft. 3. Click Send.	Mail successfully saved with "Draft/Awaiting Approval" status.	Pass

System Usability Scale (SUS)

In accordance with the discussion in Chapter II, the System Usability Scale (SUS) method is used to measure the level of application usability based on user experience. This evaluation is carried out by involving users in providing an assessment of the application through a set of instruments consisting of ten statements.

SUS assessment uses a five-point Likert scale, where each statement represents an aspect of system usability, such as ease of use, consistency of appearance and function, and the user's level of confidence in operating the application. This assessment scale allows users to respond based on their level of agreement with each statement given.

The SUS instrument is used to obtain a general overview of the level of application usability quantitatively. The five-point Likert scale used in the SUS method is presented in Table 3.4 (E. Kurniawan et al., 2022).

Tabel 3. 4 SUS Method

Answer	Point
Strongly Disagree (STS)	1
Disagree (TS)	2
Neutral (N)	3
Agree (S)	4
Strongly Agree (SS)	5

The following is a list of ten statements in SUS (Brooke, 1996) :

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need assistance from a technician to use this system.
5. I thought that the functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system

The SUS score calculation process can be done with the following steps (Brooke, 1996):

1. For odd-numbered statements (1, 3, 5, 7, 9), the score is calculated by subtracting 1 from the given value.
2. For even-numbered statements (2, 4, 6, 8, 10), the score is calculated by subtracting the value on the scale from the number 5.
3. The total score is summed and multiplied by 2.5 to get the final score in the range 0 to 100.

$$Total\ skor \quad SUS = (\sum(\text{Skor}pernyataan \quad \text{ganjil}) + \sum(\text{Skor } pernyataan \quad \text{genap})) \times 2.5 \quad (3.1)$$

The result of this calculation provides an overview of the level of application comfort and ease felt by users. Interpretation of the SUS results is done by referring to the user acceptance category, grade value, and adjective rating as shown in Figure 3.5.

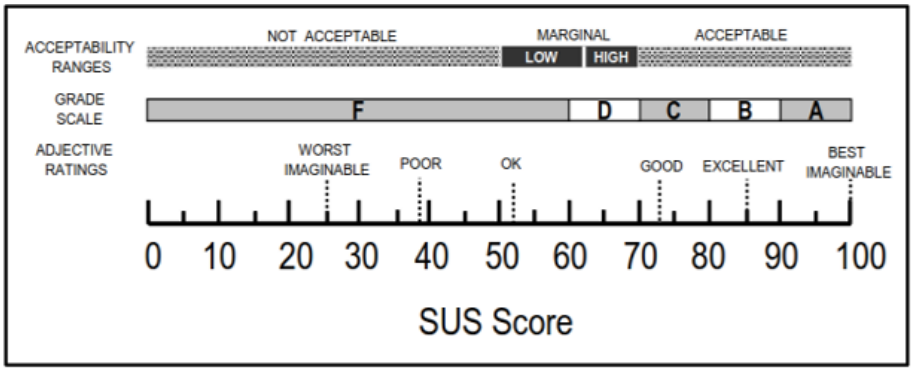


Figure 3. 5 Visualization of SUS Score Result

This SUS evaluation provides an important indicator of the quality of the application from the user's point of view. A high score indicates that the application has a good level of comfort and ease of use.

3.3.5 Maintenance

The maintenance stage is the final stage in the Waterfall system development method which aims to ensure the system can run properly after going through the testing process. At this stage, the system is refined based on the findings and input obtained from the previous testing stages.

In this research, maintenance activities are carried out by making improvements to the application based on user feedback obtained during the beta testing stage. The improvements made include adjustments to the functional aspects of the system, refinement of the user interface, and an increase in code quality.

CHAPTER IV
RESULTS AND DISCUSSION

4.1 Application Development

4.1.1 Requirements Analysis

Based on the results of the requirements analysis discussed in Chapter III, a number of main requirements were obtained that form the basis for application development in this research. These requirements are formulated to ensure that the developed system is able to support the process of data management and user activities optimally. The main system requirements identified are as follows:

a. The system must be able to manage user data according to different roles and access rights, so that each user can access features according to their authority.
b. The system must be able to support the process of recording data and activities in a structured manner to minimize errors and increase the accuracy of the managed data.
c. The system provides data validation mechanisms to ensure the correctness and consistency of recorded information.
d. The system is able to present information and reports that are easy to access, understand, and use for monitoring and evaluating user activities.
e. The system has a simple and user-friendly interface so that it can be operated without requiring a complex learning process.

The system is able to present information and reports that are easy to access, understand, and use for monitoring and evaluating user activities. The system has a simple and user-friendly interface so that it can be operated without requiring a complex learning process.

a. User Data Management, which allows the management of user data and access rights grouping according to respective roles.
b. Activity and Transaction Recording, which is used to record and manage activity data in a systematic and well-documented manner.
c. Data Validation and Verification, which ensures that recorded data has undergone a verification process before further use.

d. Report Presentation, which displays recorded data and activity results in a structured and easy-to-understand format.
e. Information Content Management, which allows management of supporting information provided within the system.
f. System Dashboard, which presents a summary of key information to help users monitor system activities and conditions.
g. User Management, which enables system administrators to supervise and manage user accounts as needed.

4.1.2 System Design

This subsection presents the results of the system design that has been realized based on the requirements identified during the analysis stage. The design results are presented in the form of diagrams and visual designs that illustrate how the system is built and how users interact with it.

The system design in this study includes the design of user-system interactions, activity flows, database structures, and user interface designs. All of these design results serve as the basis for the application implementation process.

Use Case Diagram

The results of the Use Case Diagram design illustrate the interaction between users and the system based on the main functions available. This diagram describes the roles of each actor and the features that can be accessed according to their access rights. As shown in Figure 4.1, the diagram provides an overall view of the interactions that occur within the system.

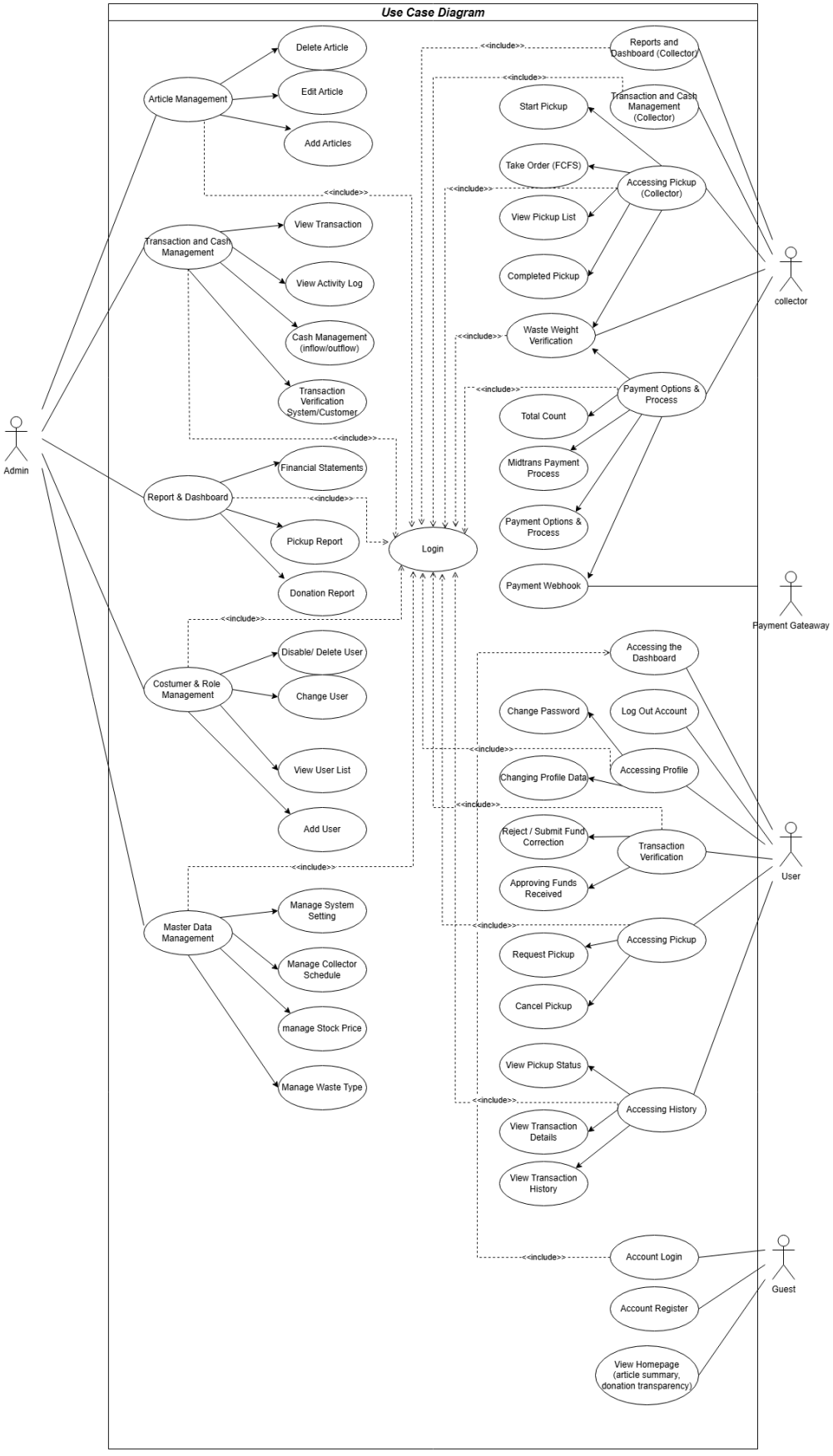


Figure 4. 1 Use Case Diagram of the Waste Bank Application

To provide a more detailed explanation, Table 4.1 presents a description of each actor in the system.

Tabel 4. 1 Actor Description

Actor	Description
Admin	The main system administrator responsible for managing master data, monitoring all transactions, managing article content, as well as managing users and financial reports
Collector	Field staff responsible for collecting waste from users, verifying weight, and processing payments through the system.
User	Customers or application users who submit waste pickup requests, manage profiles, and verify transactions.
Guest	General visitors who have not logged into the system; can only view public information and register.
payment Gateway	An external third-party system that handles automated payment integration through webhooks.

Next, Table 4.2 presents a complete list of system functionalities based on the Use Case Diagram.

Tabel 4. 2 List of System Functionalities

Action	Description
Article Management	Managing educational information (add, edit, delete) displayed on the homepage/application.
Transaction & Cash Management	Monitoring all cash flows, viewing activity logs, and verifying transactions between the system and customers.
Reports & Dashboard	Accessing data visualizations related to finance, pickup performance, and overall system conditions.

Action	Description
User & Role Management	Managing user accounts (add/delete) and setting access roles within the system.
Master Data Management	Managing fundamental data such as waste prices by type, pickup schedules, and system settings.
Access Pickup Tasks	The core field processes start from taking orders (FCFS), viewing the task list, to completing the pick-up.
Waste Weight Verification	Inputting and verifying collected waste weight for price calculation.
Payment Options & Processing	Calculating payable amounts and processing payments (including via Midtrans).
Payment Webhook	Receiving automatic confirmation from the Payment Gateway after successful transactions.
Profile Access	Manage customer personal data such as changing profile photos or changing passwords.
Transaction Verification	Allowing users to approve received funds or request corrections if discrepancies occur.
Pickup Request Access	Submitting waste pickup requests or canceling existing requests.
History Access	Viewing past transaction records, amount details, and current pickup status.
Registration & Login	Registering new accounts or logging into the system.
Homepage Access	Viewing public information such as recent articles and transparency reports.

Activity Diagrams

a. User Login Activity Diagram

The User Login Activity Diagram illustrates a systematic sequence of steps performed by the user to obtain access rights to enter the system environment. This process involves a security authentication mechanism to verify the user's identity before granting feature access. This diagram can be seen in Figure 4.2

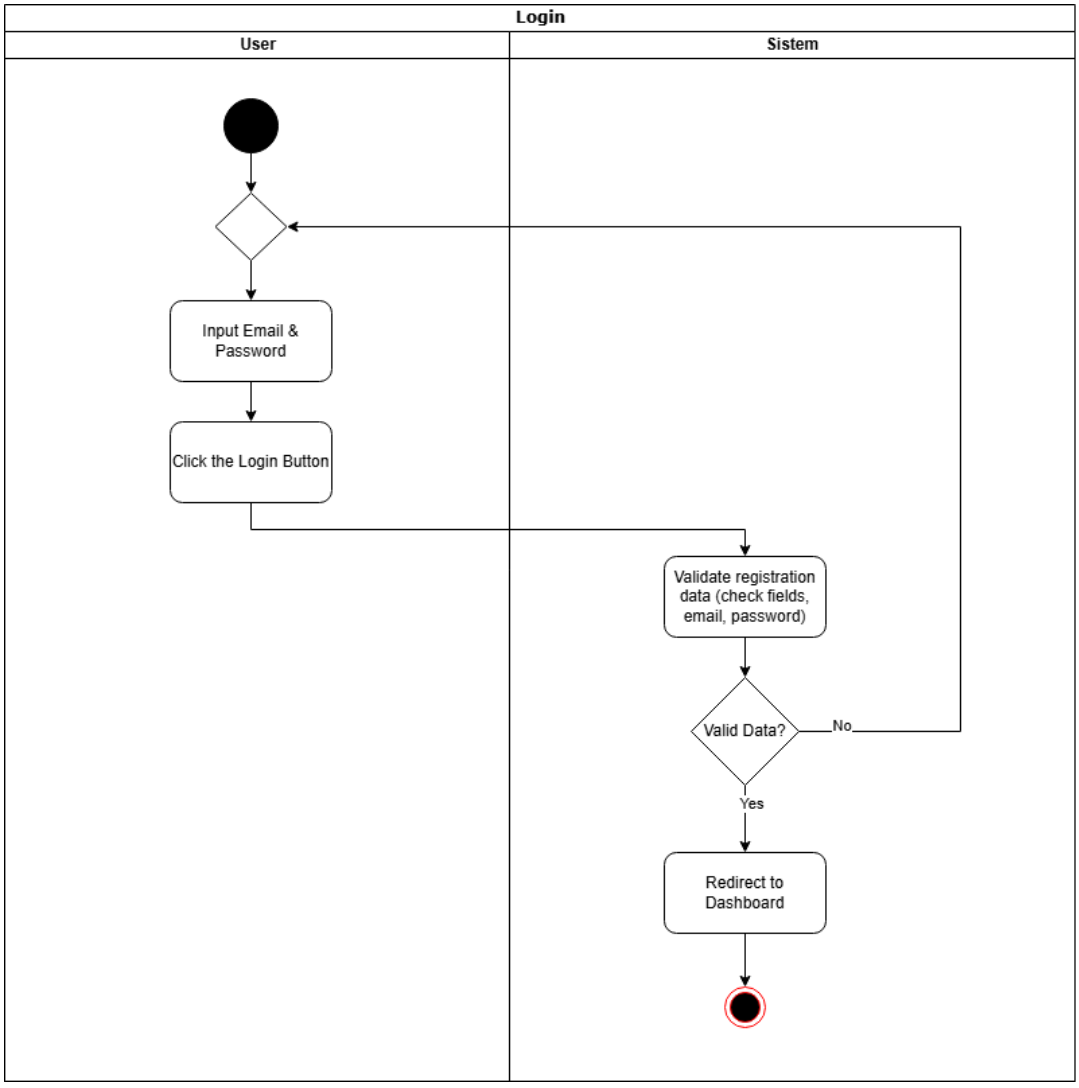


Figure 4. 2 User Login Activity Diagram

The authentication process begins when the user accesses the application's homepage and selects the login menu. The system responds by displaying a form interface that prompts the user to enter access credentials, namely the registered email address and the security password. After the user fills in the data and presses the "Login" button, the data is sent to the server via a secure protocol.

On the server side, the system receives the input and immediately validates data integrity by comparing it against user data records in the database. If a mismatch is found, the system will deny access and display an error warning message to the user. However, if the data is verified as valid, the system proceeds to the role checking stage. Based on the identified role—whether Admin, Collector (*Pengepul*), or Customer (*Nasabah*) the system automatically directs the user to the specific dashboard page corresponding to their authority, marking that the application session has successfully started.

b. Collector Registration Activity Diagram

The Collector Registration Activity Diagram visualizes the self-registration flow for prospective collector partners. This process includes the collection of administrative data and multi-level validation to ensure the credibility of the working partner. This diagram can be seen in Figure 4.3.

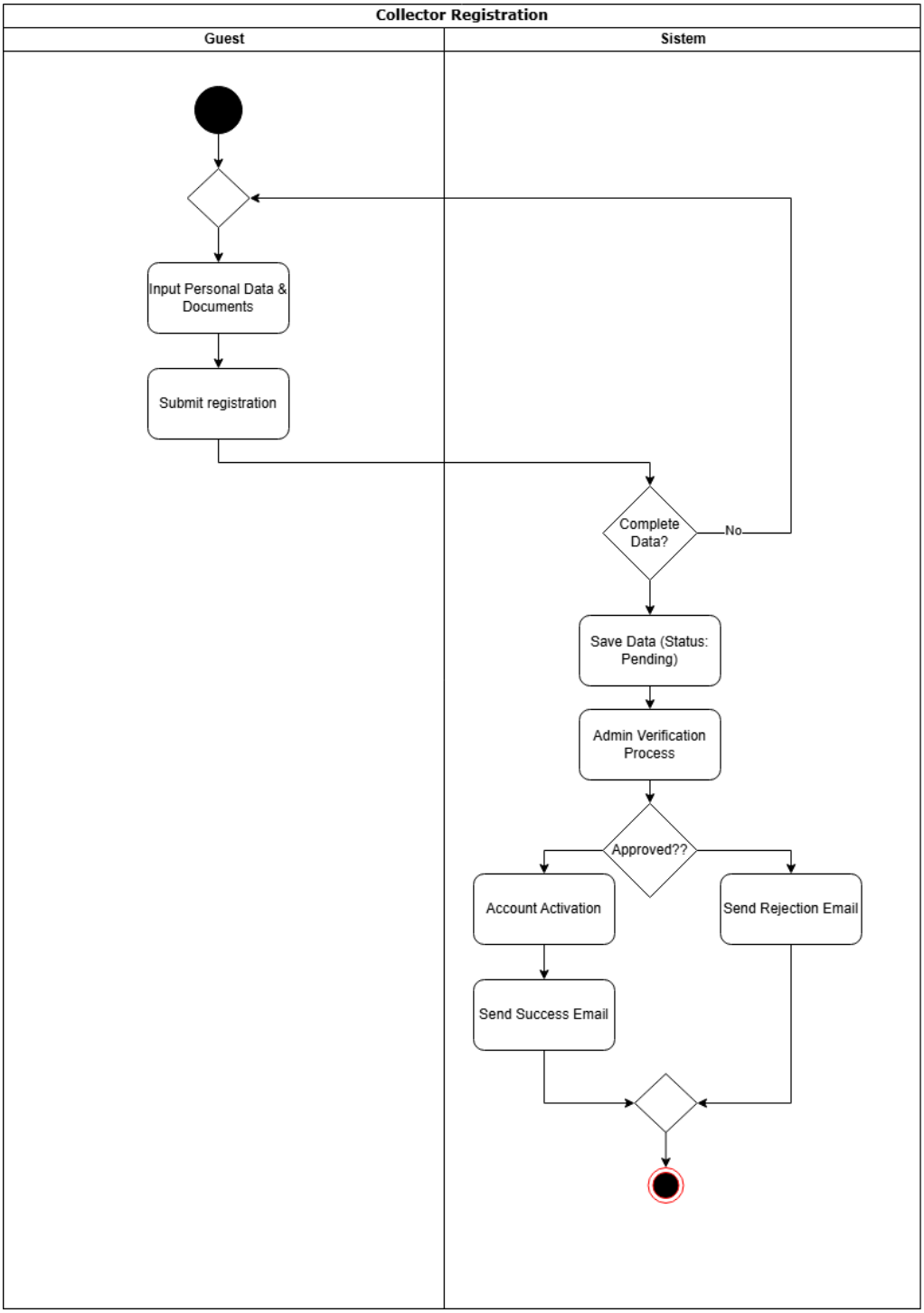


Figure 4. 3 Collector Registration Activity Diagram

The initial step begins when the prospective partner opens the special collector registration page on the application. The prospective partner is required to complete the digital registration form with valid self-identity data (such as Full Name, NIK, Domicile Address) and upload supporting legality documents. After all data is submitted, the system automatically checks the completeness of the input format.

If the data is declared complete, the system saves it to the database with a temporary "Pending" status and sends a review notification to the Central Admin. The Admin then performs a manual verification of the applicant's data validity to decide on partnership eligibility. In the scenario where the registration is approved, the Admin will change the account status to active and the system sends a confirmation email. Conversely, if the data is deemed unqualified, the Admin rejects the application and the system sends a rejection notification, ending the registration process.

c. Customer Group Registration Activity Diagram

This Activity Diagram explains the procedure for forming new community units or customer groups within the system. This flow emphasizes geographical location data collection and the creation of a unique group identity. This diagram can be seen in Figure 4.4.

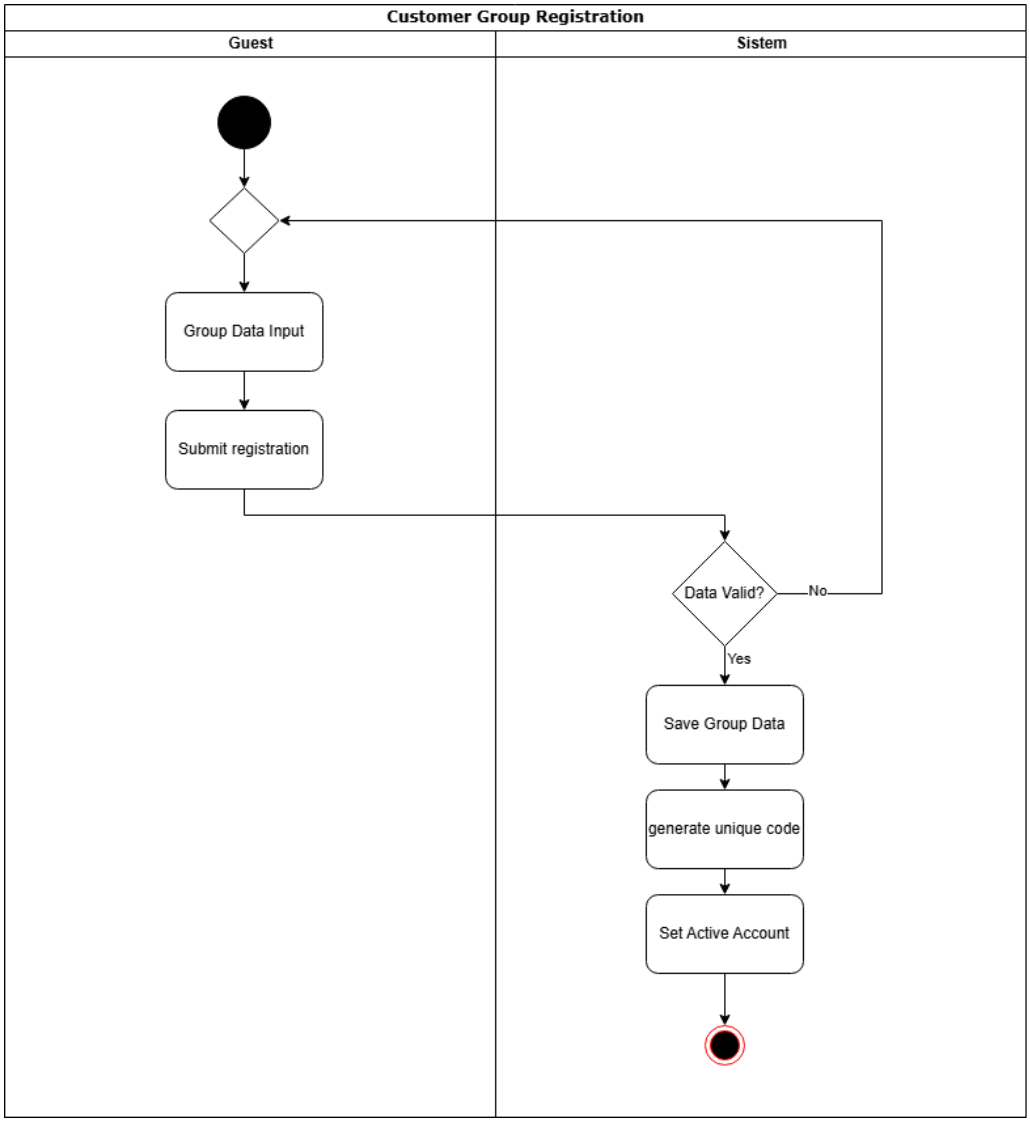


Figure 4. 4 Customer Group Registration Activity Diagram

The process is initiated by a resident representative or community leader who accesses the "Create New Group" feature. The user fills out a form containing the group identity name, operational area description, main pickup point location, and responsible contact person. This data is crucial for future pickup logistics management.

After the data is submitted, the system validates and ensures there is no group name duplication. If the data is valid, the system saves the group entity to the master database. As a key feature, the system runs an algorithm to automatically generate a "Unique Code" (Referral Code). This code is then displayed to the user and the group account is immediately activated (Status: Active), allowing the group leader to immediately distribute the code to residents to start recruiting customer members.

d. Pickup Request Submission Process Activity Diagram

This Activity Diagram describes the upstream operational stage where theCustomer Group submits a request for waste collection. The diagram details the interaction between the user, system validation, and the dissemination of information to collector partners. This diagram can be seen in Figure 4.5

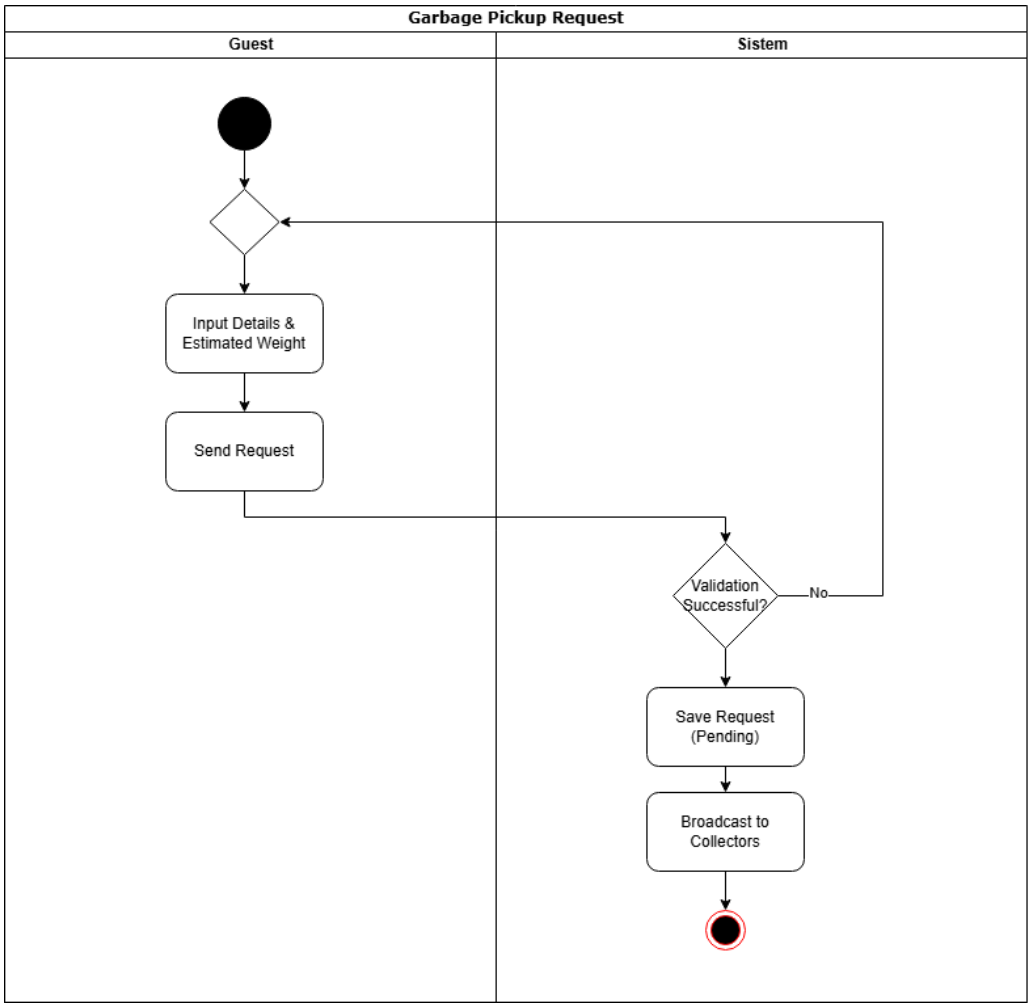


Figure 4. 5 Pickup Request Submission Process Activity Diagram

The Customer logs into the system and selects the "Request Pickup" menu. The user then fills out a detail form which includes the estimated total waste weight, the category of collected waste types, and attached photos of the real waste condition.

After the send button is pressed, the system validates the completeness of the request data. If valid, the system records the request with a "Pending" status into the task queue. Simultaneously, the system triggers a real-time notification broadcasting mechanism to all active Collector accounts in that area, ensuring the waste request information is quickly disseminated for immediate follow-up.

e. Pickup Taking Activity Diagram (FCFS)

This Activity Diagram shows the competitive First Come First Served (FCFS) mechanism used by the system in distributing pickup tasks to collectors. This ensures the aspects of fairness and service response speed. This diagram can be seen in Figure 4.6.

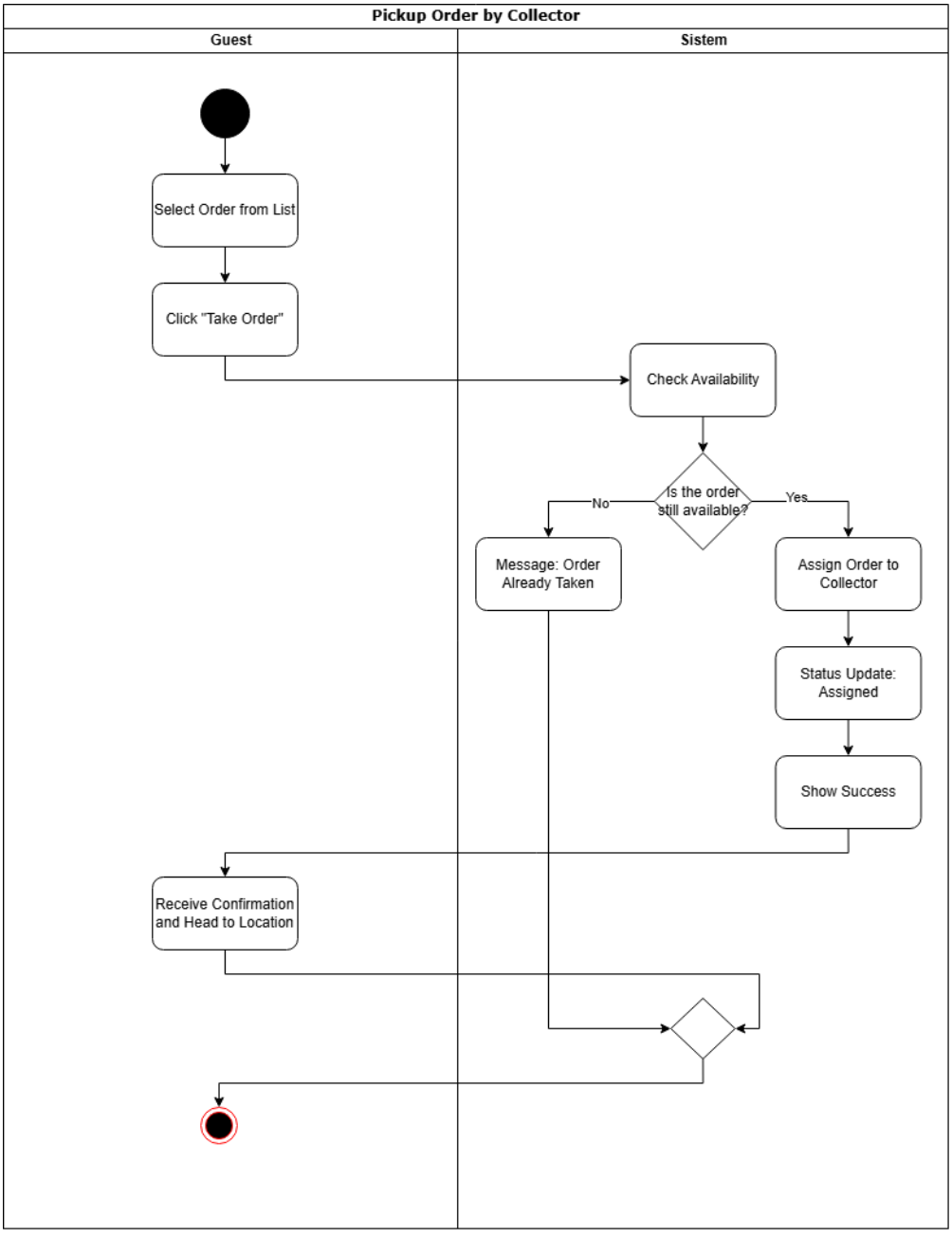


Figure 4. 6 Pickup Taking Activity Diagram (FCFS)

The Collector monitors the list of available pickup requests (Open Orders) through their application dashboard. After analyzing the location and potential load, the Collector selects one order and presses the "Take Order" button. At this critical point, the system performs database concurrency checking (Database Locking) to ensure the order's availability.

If the order is still available (not yet taken by another collector), the system will lock the data, update its status to "Assigned", and record the Collector ID as the official person in charge. The system then provides access to detailed location information and customer contact to the Collector, allowing the mobilization process to the pickup location to be carried out immediately.

f. Waste Weight Verification Activity Diagram

This Activity Diagram illustrates the physical validation process in the field to ensure the accuracy of the weighing data before financial transactions are carried out. This is a crucial stage for transparency between both parties. This diagram can be seen in Figure 4.7.

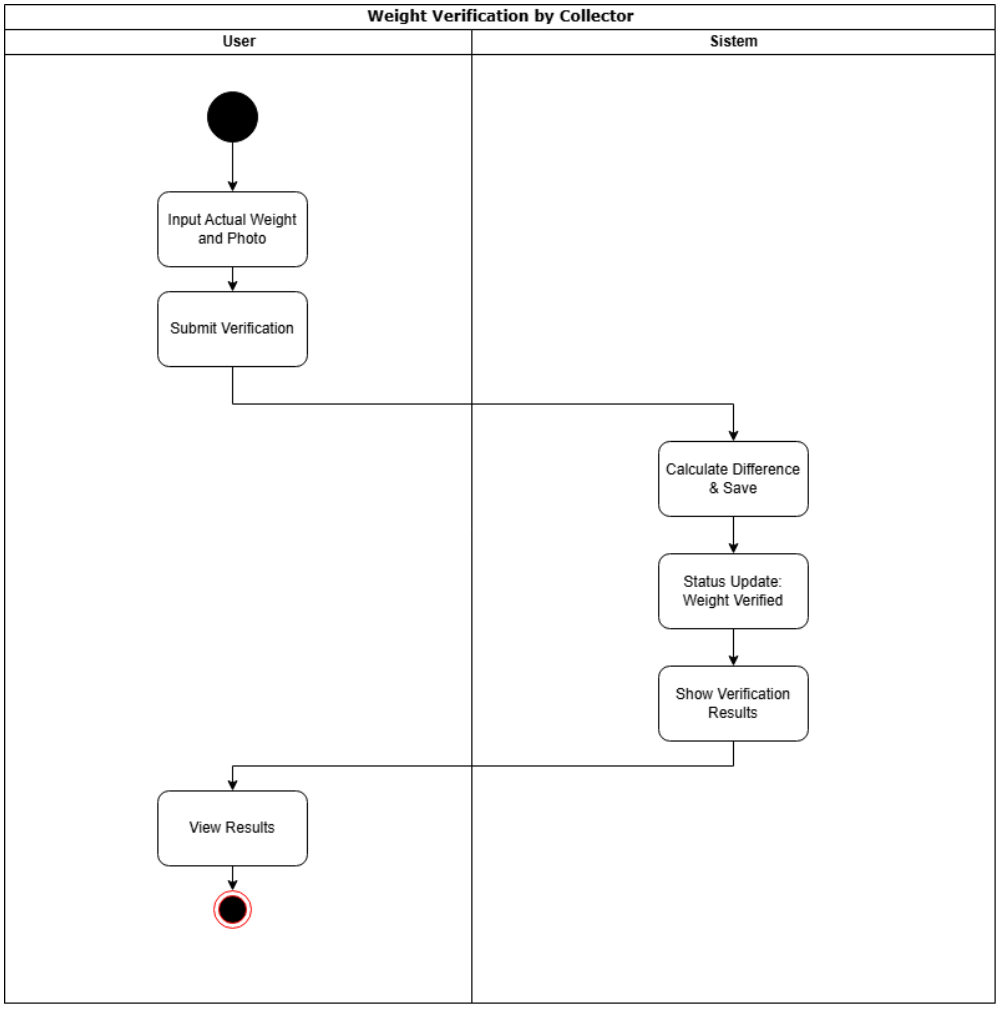


Figure 4. 7 Waste weight Virification Activity Diagram

After the Collector arrives at the location and performs physical weighing, the Collector accesses the verification menu on the application. They input the actual weight (Real Weight) per waste type and upload a photo of the scale as digital evidence. This data is then sent to the system for processing.

The system automatically calculates the variance between the customer's initial estimate and the actual weight, and calculates the total Rupiah value based on the prevailing unit price of the waste. The result of this calculation is saved with the status "Weight Verified" and displayed transparently on both the Customer's and Collector's application screens. Both parties review the final figures to agree on the basis of the transaction to be processed.

g. Payment Process Activity Diagram (Midtrans Gateway)

This Activity Diagram explains the digital financial transaction flow involving integration with an external payment authority (Midtrans). The diagram shows the handling of different payment scenarios according to the donation choice. This diagram can be seen in Figure 4.8.

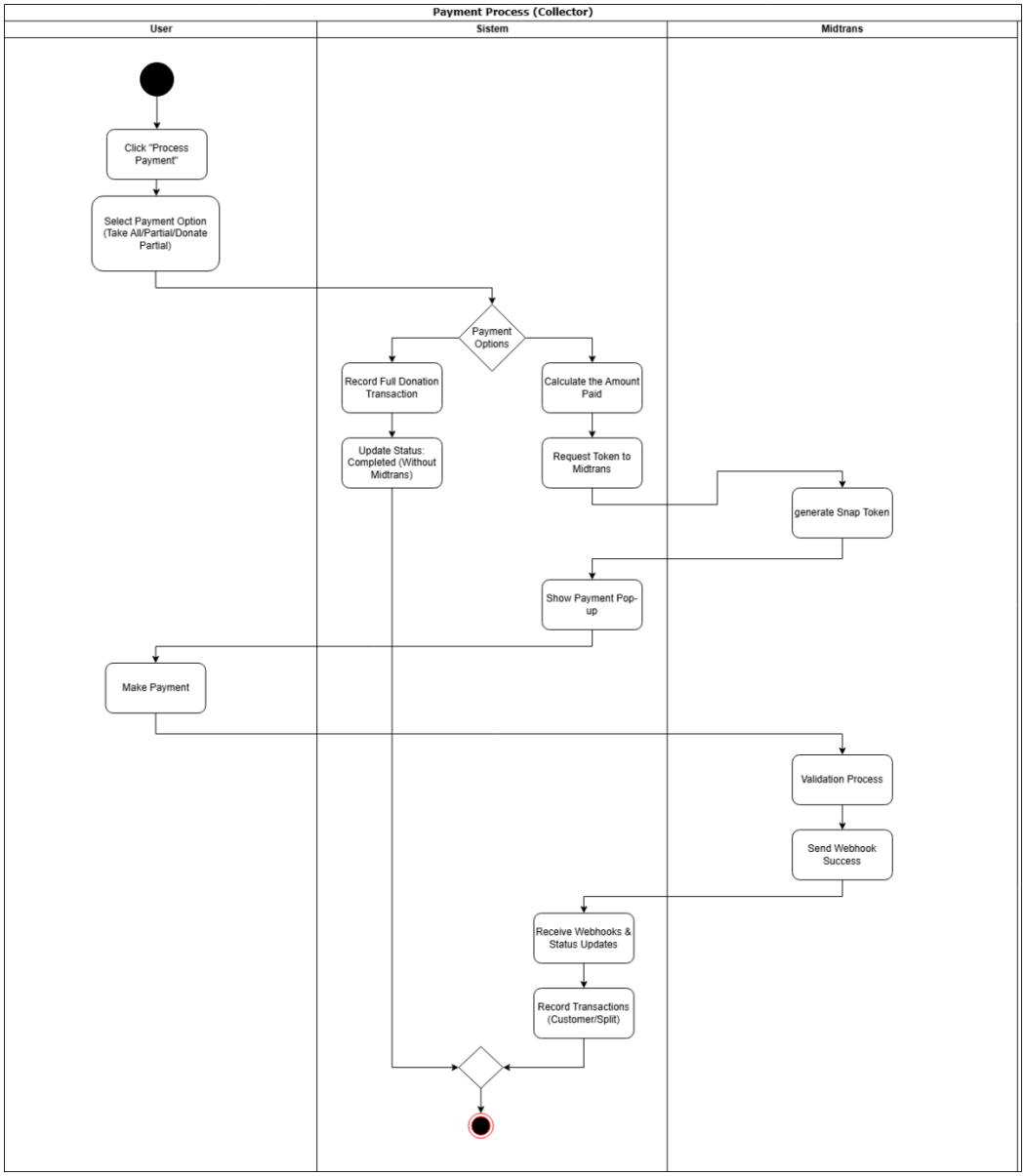


Figure 4. 8 Payment Process Activity Diagram (Midtrans Gateawy)

After the weight is agreed upon, the Collector initiates the payment. The system detects the fund distribution option chosen by the Customer. If the option is "Donate All", the system bypasses the payment gateway and immediately records the internal transaction as a successful donation, ending the process without external fund flow.

However, for the "Take All" or "Donate Some" options, the system connects to the Midtrans API to request a Payment Token. Midtrans issues a token that allows the Collector to make payments via various methods (QRIS, E-Wallet, Transfer). Following a successful payment, the system receives a confirmation signal (Webhook) from Midtrans, verifies the paid status, and automatically splits and distributes the balance to the Customer's digital wallet according to the predetermined percentage.

h. Transaction Verification by User Activity Diagram

This Activity Diagram illustrates the transaction quality control mechanism performed by the Customer. This stage gives authority to the user to audit fund receipt. This diagram can be seen in Figure 4.9.

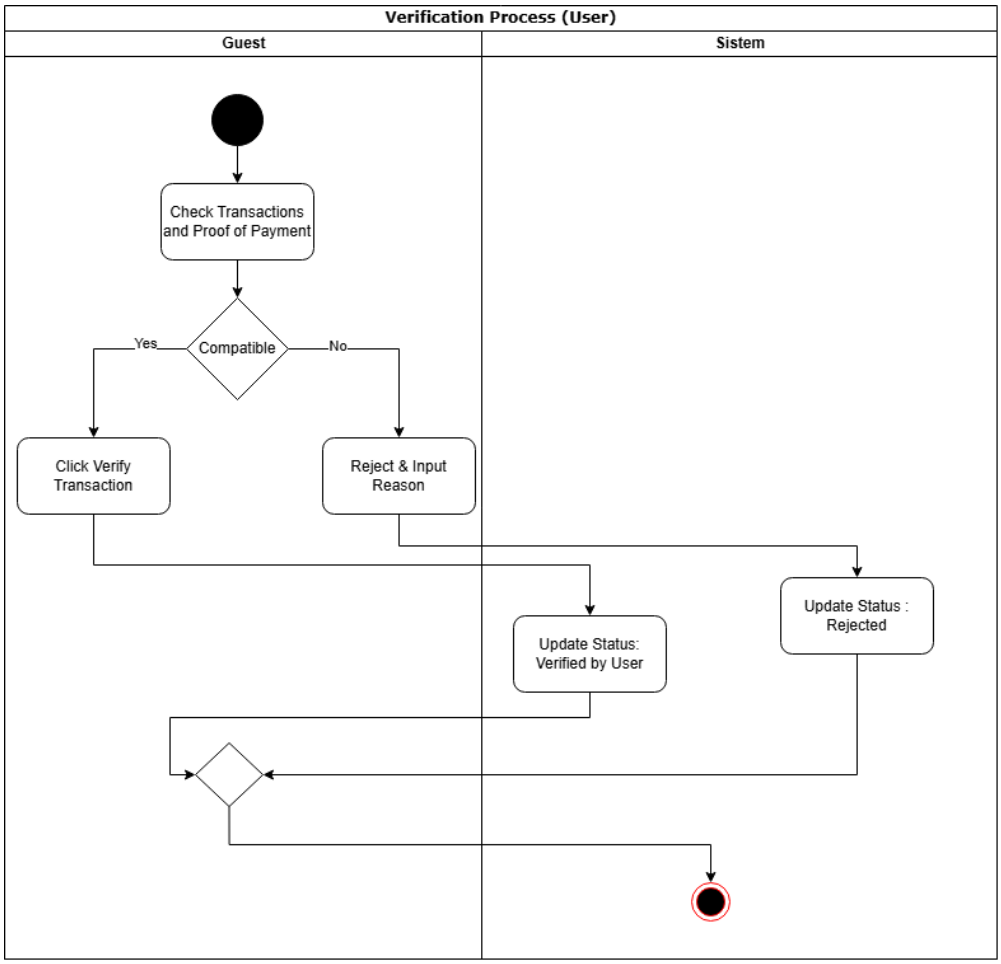


Figure 4. 9 Transaction Verification by User Activity Diagram

The Customer accesses the system's transaction history to check incoming funds. The Customer compares the nominal amount listed in the application with the physical receipt or agreement in the field. If the data matches, the Customer confirms by pressing the "Verify" button, which the system will record as transaction completion ("Completed").

If there is a discrepancy, the Customer has the right to file a complaint by pressing the "Reject" button and filling out the complaint reason form. The system responds by changing the transaction status to "Dispute" and suspending the transaction completion until the Admin performs manual mediation intervention.

i. Transaction Verification by Admin Activity Diagram

This Activity Diagram visualizes the Admin's role as an auditor in validating transactions involving public funds (donations). This process is important for the financial accountability of the system. This diagram can be seen in Figure 4.10.

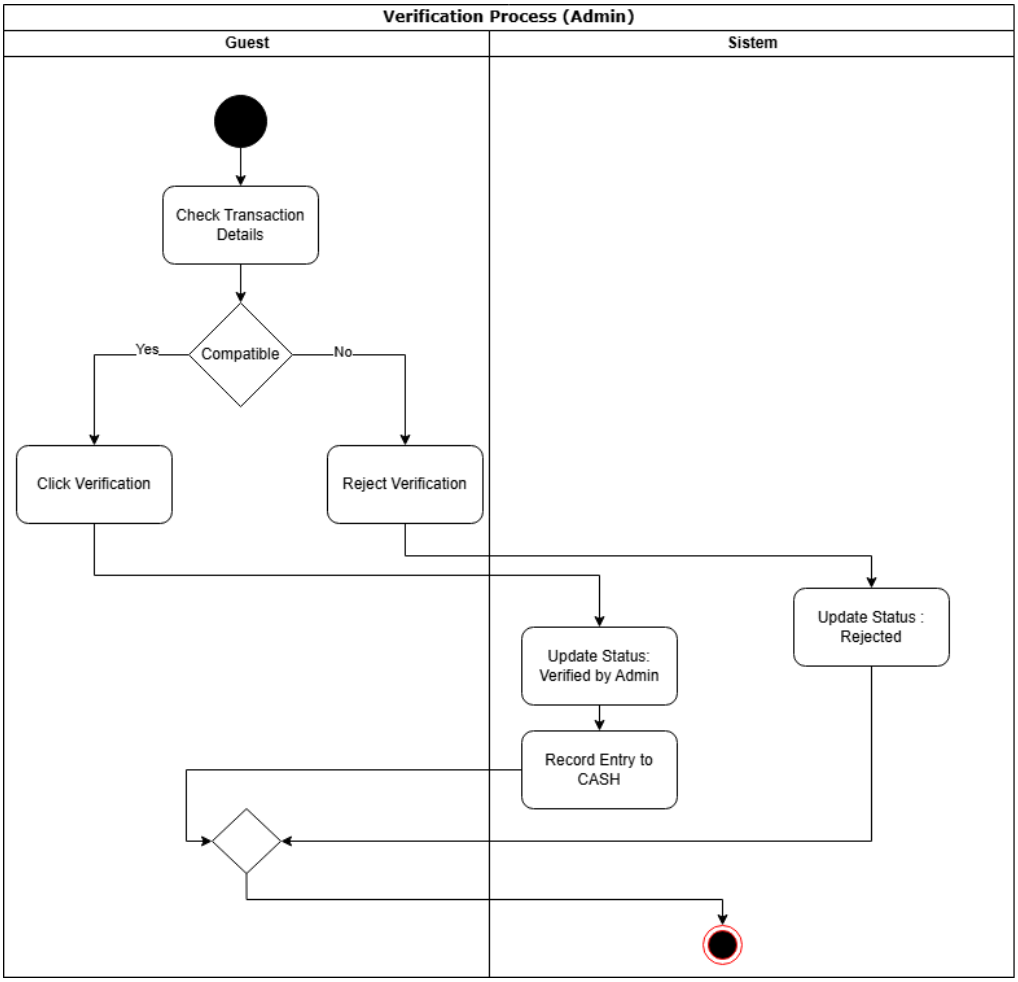


Figure 4. 10 Transaction Verification By Admin Activity Diagram

The Admin periodically monitors the list of donation transactions entering the system. The Admin performs data reconciliation by checking the system transfer evidence and the foundation's bank account statement. If the funds are confirmed to have legally entered, the Admin provides digital approval ("Approval"). The system then permanently records the nominal amount into the system's Cash In ledger. Conversely, if suspicious or fictitious transactions are found, the Admin rejects the transaction to maintain the cleanliness of the financial data.

j. Cash Management Activity Diagram

This Activity Diagram explains the management of the system's cash outflow operated by the Admin. The diagram emphasizes fund availability validation before expenditure is recorded. This diagram can be seen in Figure 4.11.

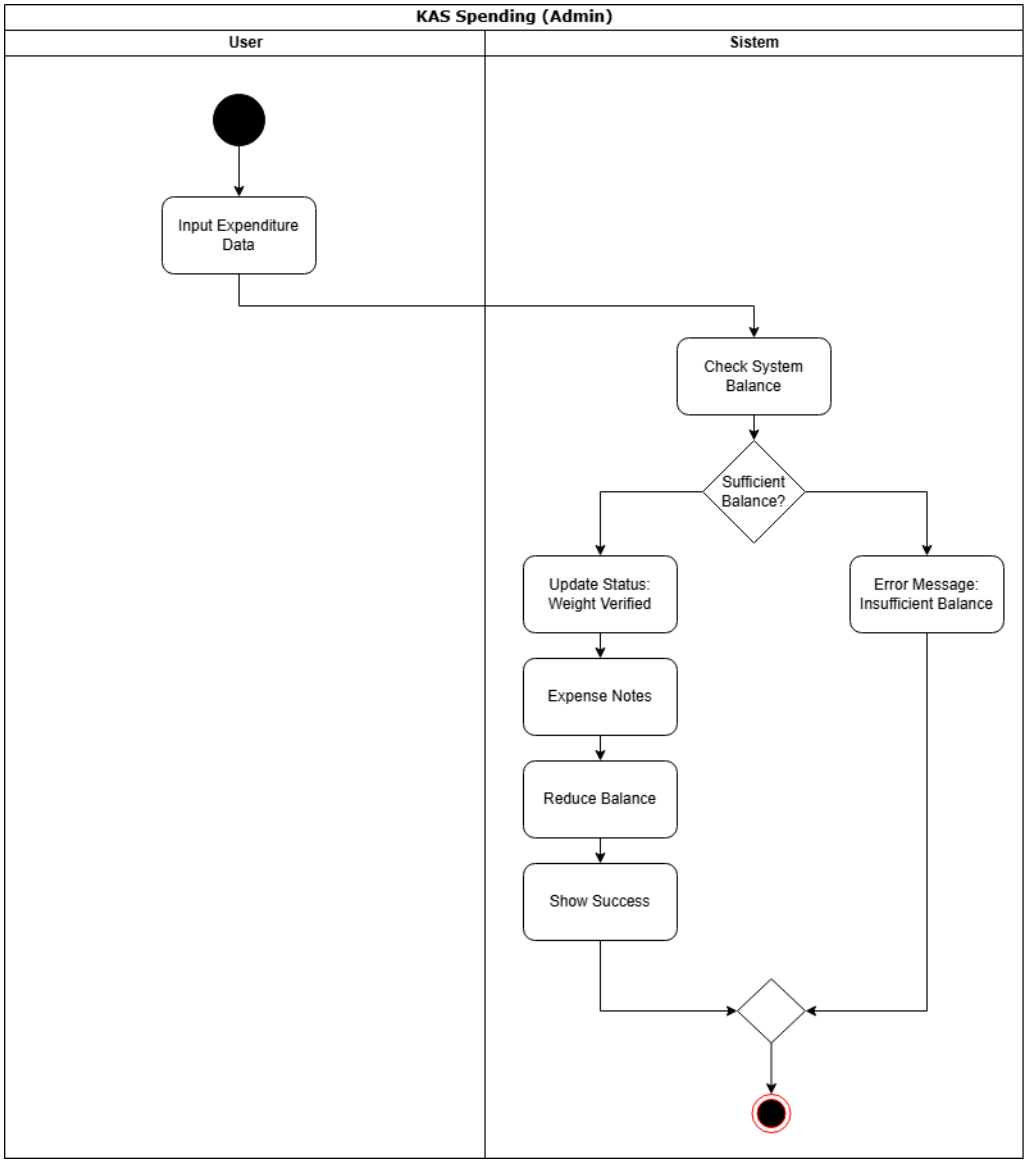


Figure 4. 11 Cash Mangement Activity Diagram

The Admin inputs planned operational cost expenditures or social disbursements into the Cash Management module, complete with the nominal amount and supporting evidence. Before the data is saved, the system algorithm performs a Balance Check. The system calculates the current total liquid cash balance. If the balance is sufficient to cover the expenditure, the system allows the transaction, saves it to the expense database, and automatically debits the cash balance. If the balance is insufficient, the system blocks the transaction to prevent budget deficit.

k. User Data Management Activity Diagram

The User Data Management Activity Diagram describes the user account management cycle performed by the Admin to ensure access security. This diagram can be seen in Figure 4.12.

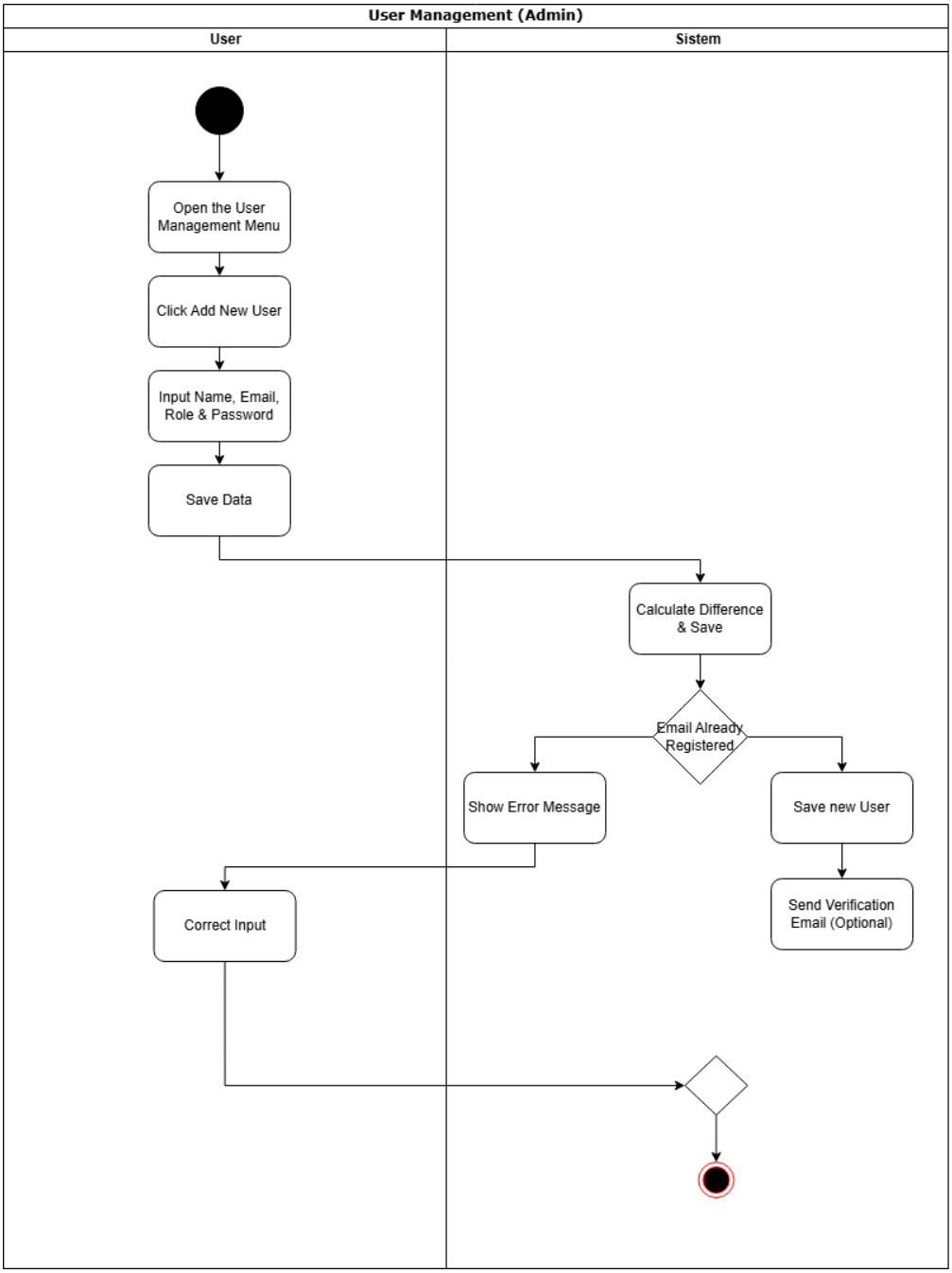


Figure 4. 12 User Data Management Activity DIagarm

The Admin inputs new user data through the management dashboard. The system validates the uniqueness of the email and the strength of the password. Once valid, the system encrypts the sensitive data and saves it to the database, granting access rights to the new user.

l. Data Management Activity Diagram (type of waste and price)

This Activity Diagram explains the process of updating critical reference data, namely the types and prices of waste. This diagram can be seen in Figure 4.13.

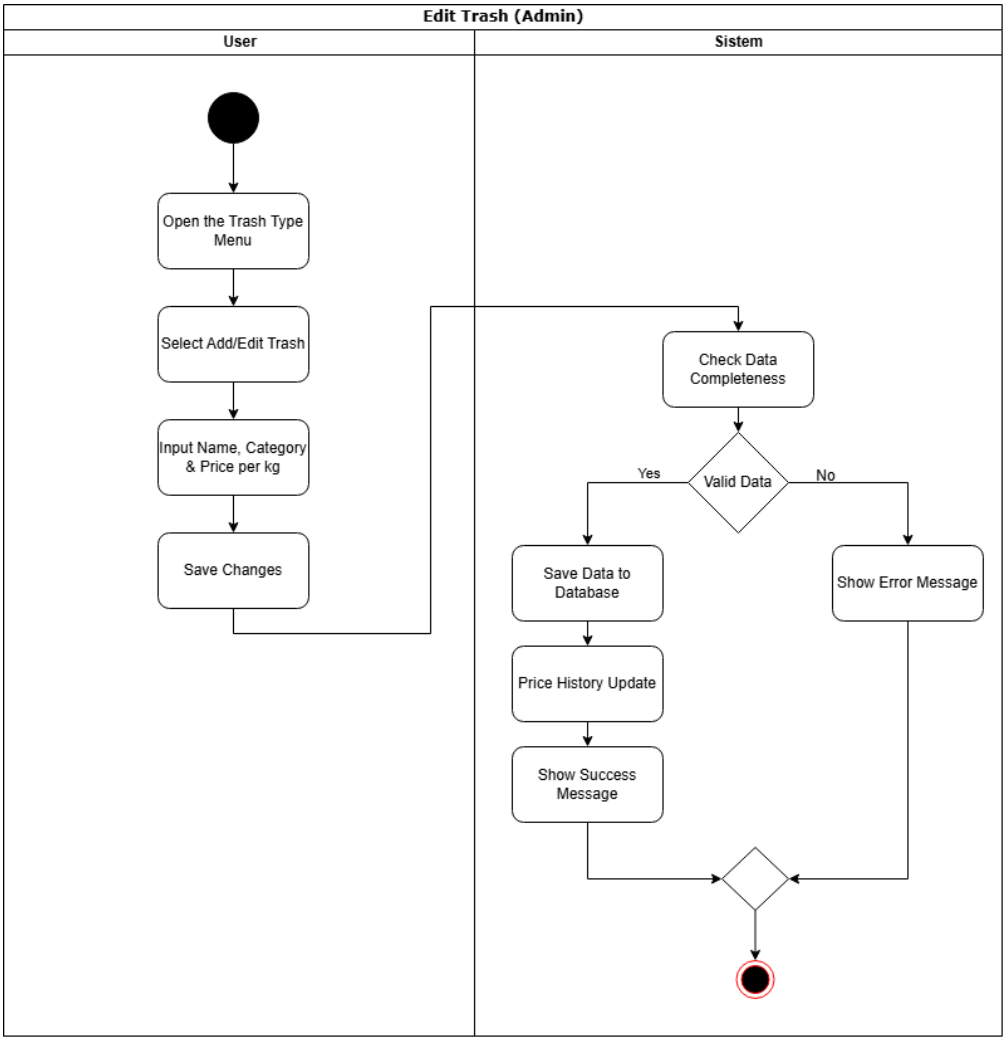


Figure 4. 13 Data Type of Waste Management Activity Diagram

The Admin adjusts the unit price of waste following market trends. The system saves this price change and applies it in real-time as a calculation reference for all pickup transactions that occur after the update time.

m. Article Management Activity Diagram

This Activity Diagram describes the management of environmental education content on the platform. This diagram can be seen in Figure 4.14.

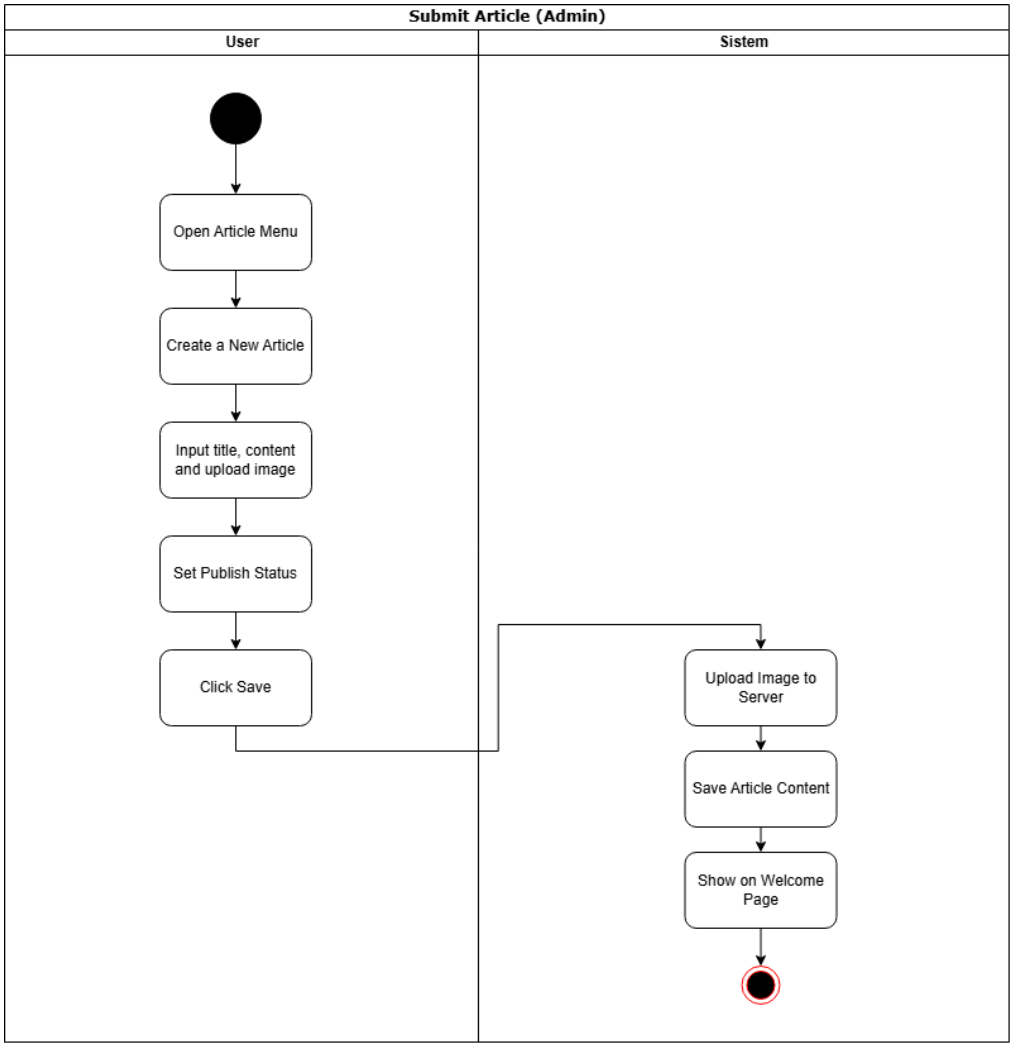


Figure 4. 14 Article Management Activity Diagram

The Admin compiles educational articles and uploads visual media assets. The system processes the content storage to the application server and then publishes it to the homepage so that it can be accessed by the entire user base as a means of literacy.

Database Design

f. Table Relationships

The database design of the developed system is intended to support integrated management of user, groups, costumer, pick up and transaction. The database structure is designed so that each system process is interconnected and well documented. The database design results are represented in the form of an Entity illustrating the relationships between the main tables in the system. The Table Relationship is shown in Figure 4.15.

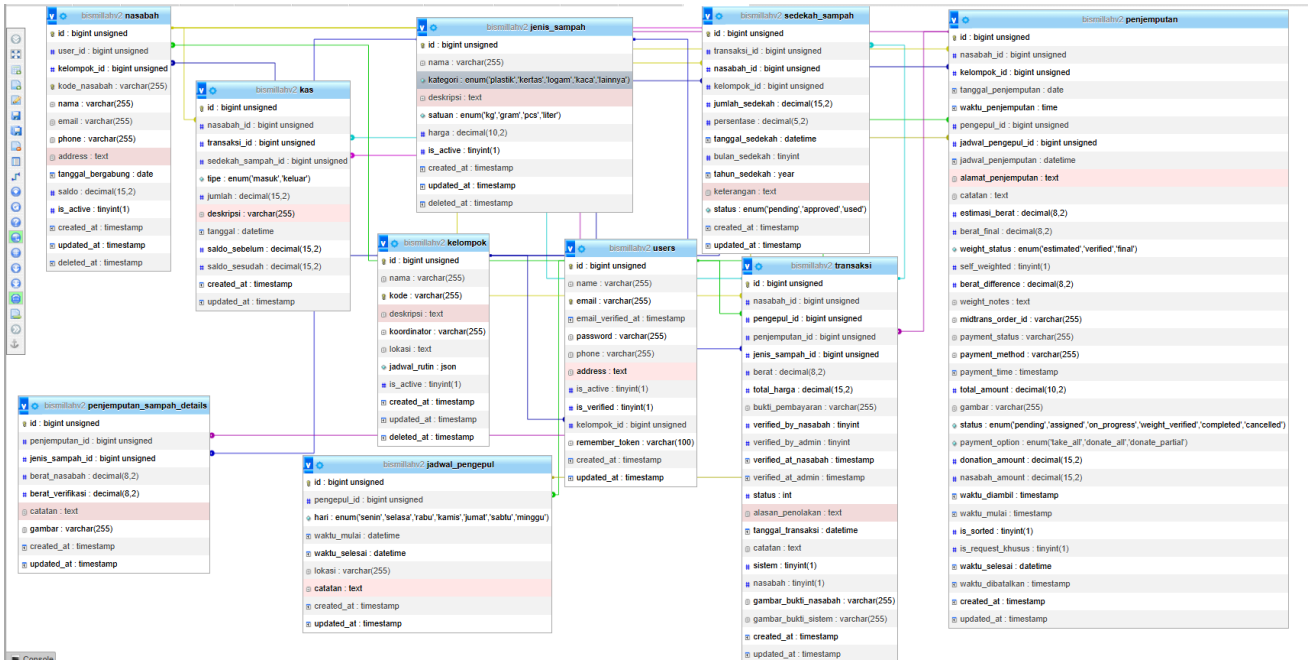


Figure 4. 15 Database Realitionships of the Waste Bank Application

Core Table Explanation

1. Users: Parent table for authentication (Login). All roles (Admin, Collector, Customer Group) are stored here. Each user account can have multiple roles managed through Spatie Laravel Permission package. The kelompok_id field links users with the kelompok_nasabah role to their respective group.
2. Group: Entity to group customers by organization/region (similar to RT/RW concept). This is important because the system is "Customer Group" based, where waste collection is organized by groups rather than purely individual customers. Each group has a coordinator and can define regular pickup schedules.
3. Costumer: Customer profile details. Connected to Users (for login account) and Kelompok (for organizational grouping). Has a saldo (balance) column to save accumulated money from verified waste transactions. Customers can exist as data-only records or have linked user accounts for system access.
4. Pick Up: Busiest operational table with 46 columns covering the entire workflow. Recording cycle: Request (by customer) → Assigned (to collector) → Picked up → Weighed (Weight Verification) → Completed (Payment processed). This central transaction table implements denormalization strategy to optimize query performance by storing all pickup lifecycle data in one place.

5. Pick Up Waste Details: Breakdown table for pickup requests. One pickup can contain multiple waste types, each with separate weight tracking (berat_nasabah for customer estimate, berat_verifikasi for collector verification). Connected to Jenis Sampah for pricing calculation: total = berat × harga.
6. Waste Type: Master data table storing waste type catalog and pricing. The harga field stores current price per kilogram directly in this table (simplified approach without price history). Implements soft delete to preserve historical transaction references while preventing selection of inactive waste types.
7. Transaction: Financial table recording incoming/outgoing money from waste sales. This transaction is verified by Customer (user via verified_by_nasabah field) and Admin (if donating via verified_by_admin field). Supports dual verification system for split payment scenarios where part goes to customer and part goes to system donation fund.

Database Table Structure

The database structure consists of several main tables with attributes adjusted to system requirements. The explanation of the main table structures is as follows.

a. User Table

The users table serves as the parent entity for authentication. This table stores login credentials for all system actors (Admin, Collector, Customer, Group). U can see at Tabel 4. 3.

Tabel 4. 3 User Table Structure

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
name	VARCHAR(255)	Full name of User
email	VARCHAR(255)	Email address (unique) for login
password	VARCHAR(255)	Encrypted password (bcrypt hashing)
phone	VARCHAR(255)	User's phone/WhatsApp number
address	TEXT	User's domicile address
kelompok_id	BIGINT(20) UNSIGNED	Foreign Key to kelompok (for kelompok_nasabah role)
is_active	BOOLEAN	Account active status (1=active, 0=inactive)
is_verified	BOOLEAN	Account verification status

Column Name	Data Type	Information
remember_token	VARCHAR(100)	Session persistence token
created_at	TIMESTAMP	Account creation time
updated_at	TIMESTAMP	Time of last account update

The users table is used to store credential data for all actors involved in the system, from Administrators, Collectors, to Customer Groups. The email and password fields are used as authentication keys during the login process. The system implements Spatie Laravel Permission package for role management, allowing users to have multiple roles (admin, pengepul, kelompok_nasabah, nasabah) stored in separate permission tables. The kelompok_id field links users with the kelompok_nasabah role to their respective group. Contact information such as phone and address is stored for operational communications purposes. The is_active flag allows administrators to enable or disable user accounts without permanently deleting them.

b. Group Table

Table 4.4 displays table stores individual customer/member data within waste collector groups. This table represents end-users who generate recyclable waste.

Tabel 4. 4 Group Table Structure

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
nama	VARCHAR(255)	Group name
kode	VARCHAR(50)	Unique group code
deskripsi	TEXT	Group description
koordinator	VARCHAR(255)	Coordinator/leader name
lokasi	TEXT	Group location/address
jadwal_rutin	JSON	Regular pickup schedule (flexible array format)
is_active	BOOLEAN	Group active status (1=active, 0=inactive)
created_at	TIMESTAMP	Group registration time
updated_at	TIMESTAMP	Time of last group data update
deleted_at	TIMESTAMP	Soft delete timestamp (NULL if not deleted)

The group table manages waste collector group organizations. Each group has a unique kode identifier used for administrative purposes. The jadwal_rutin field stores flexible schedule information in JSON format, allowing groups to define their regular pickup schedules. The table implements soft delete functionality through the deleted_at column, ensuring that historical data is preserved even when a group is deactivated. The koordinator field identifies the group leader responsible for coordination with the system administrators. This table serves as an aggregation point for multiple individual nasabah (customers) who belong to the same collection group.

c. Costumer Table

Table 4.5 table stores individual customer/member data within waste collector groups. This table represents end-users who generate recyclable waste.

Tabel 4. 5 Costumer Table Structure

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
user_id	BIGINT(20) UNSIGNED	Foreign Key to users (nullable, for account linkage)
kelompok_id	BIGINT(20) UNSIGNED	Foreign Key to kelompok (required)
kode_nasabah	VARCHAR(50)	Unique customer code
nama	VARCHAR(255)	Customer full name
email	VARCHAR(255)	Customer email (optional)
phone	VARCHAR(255)	Customer phone number
address	TEXT	Customer address
tanggal_bergabung	DATE	Date joined the group
saldo	DECIMAL(15,2)	Customer account balance (from waste sales)
is_active	BOOLEAN	Customer active status (1=active, 0=inactive)
created_at	TIMESTAMP	Customer registration time
updated_at	TIMESTAMP	Time of last customer data update
deleted_at	TIMESTAMP	Soft delete timestamp (NULL if not deleted)

The customer table manages individual customers who are members of waste collector groups. Each customer is uniquely identified by kode_nasabah and must belong to one Kelompok (group). The optional user_id field allows customers to have a linked user account for accessing the system's customer portal, enabling self-service features such as viewing transaction history and requesting waste pickups. The saldo field tracks the customer's account balance, which accumulates from verified waste transactions. This balance represents the financial value of recyclable waste collected from the customer. The table implements soft delete to maintain historical transaction records even when a customer is no longer active. The relationship between kelompok and nasabah follows an aggregation pattern where customers can logically exist independently but are organizationally grouped for operational efficiency.

d. Pick Up Table (Operational)

The Table 4.6 is the central transaction table that manages the entire lifecycle of waste pickup requests from creation to payment completion. This table contains comprehensive tracking of pickup workflow stages.

Tabel 4. 6 Pickup Table Structure

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
nasabah_id	BIGINT(20) UNSIGNED	Foreign Key to nasabah (optional, for individual requests)
kelompok_id	BIGINT(20) UNSIGNED	Foreign Key to kelompok (required)
pengepul_id	BIGINT(20) UNSIGNED	Foreign Key to users (collector assigned to pickup)
jadwal_pengepul_id	BIGINT(20) UNSIGNED	Foreign Key to jadwal_pengepul (optional schedule link)
tanggal_penjemputan	DATE	Requested pickup date
waktu_penjemputan	DATETIME	Requested pickup datetime
jadwal_penjemputan	DATETIME	Scheduled pickup datetime
alamat_penjemputan	TEXT	Pickup location address
catatan	TEXT	Additional notes from customer
gambar	TEXT	Photo evidence of waste (uploaded by customer)

Column Name	Data Type	Information
status	ENUM	Workflow status: pending, assigned, on_progress, weight_verified, completed, cancelled
is_sorted	BOOLEAN	Whether waste is pre-sorted by customer
is_request_khusus	BOOLEAN	Special request flag (outside regular schedule)
waktu_diambil	DATETIME	Timestamp when collector accepts the pickup
waktu_mulai	DATETIME	Timestamp when collector starts pickup
waktu_selesai	DATETIME	Timestamp when pickup is completed
waktu_dibatalkan	DATETIME	Timestamp if pickup is cancelled
payment_option	ENUM	Payment allocation: self, donate_all, donate_partial
donation_amount	DECIMAL(15,2)	Donation amount (if donate_partial)
nasabah_amount	DECIMAL(15,2)	Customer amount (if donate_partial)
estimasi_berat	DECIMAL(10,2)	Estimated weight provided by customer (kg)
berat_final	DECIMAL(10,2)	Final verified weight by collector (kg)
weight_status	VARCHAR(50)	Weight verification status
self_weighted	BOOLEAN	Whether customer weighed themselves
berat_difference	DECIMAL(10,2)	Difference between estimated and verified weight
weight_notes	TEXT	Notes regarding weight verification
midtrans_order_id	VARCHAR(255)	Midtrans payment gateway transaction ID
payment_status	VARCHAR(50)	Payment gateway status
payment_method	VARCHAR(50)	Payment method used (if applicable)
payment_time	DATETIME	Payment completion timestamp
total_amount	DECIMAL(15,2)	Total payment amount

Column Name	Data Type	Information
created_at	TIMESTAMP	Pickup request creation time
updated_at	TIMESTAMP	Time of last pickup data update

Pick Up table is designed as a central transaction table containing the entire lifecycle of waste pickup requests. This table has 46 columns grouped into five concerns: (1) identification and relationships, (2) basic pickup information, (3) workflow status tracking, (4) weight verification, and (5) payment processing. The denormalization approach was chosen to optimize query performance, as most system operations require access to complete pickup data in a single query. The status field tracks pickup workflow progression through six stages: pending (awaiting collector assignment), assigned (collector has accepted), on_progress (actively picking up), weight_verified (weight has been verified), completed (fully processed), and cancelled. The payment_option field enables flexible fund allocation: 'self' (100% to customer), 'donate_all' (100% to system), or 'donate_partial' (split between customer and system based on donation_amount and nasabah_amount). Workflow timestamps (waktu_diambil, waktu_mulai, waktu_selesai) provide detailed audit trail of pickup operations. Integration with Midtrans payment gateway is supported through midtrans_order_id and related payment fields for cashless transaction options. The extensive column count reflects the system's comprehensive tracking requirements, capturing every aspect from initial request through final payment.

e. Waste Collection Details Table

Table 4.7 displays the structure of the transaction table which records the mutation of the waste exchange rate.

Tabel 4. 7 Waste Collection Details Table

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
penjemputan_id	BIGINT(20) UNSIGNED	Foreign Key to penjemputan (parent pickup request)
jenis_sampah_id	BIGINT(20) UNSIGNED	Foreign Key to jenis_sampah (waste type)
berat_nasabah	DECIMAL(10,2)	Weight estimated by customer (kg)
berat_verifikasi	DECIMAL(10,2)	Weight verified by collector (kg)
catatan	TEXT	Notes regarding this waste type

Column Name	Data Type	Information
gambar	TEXT	Photo evidence of this specific waste type
created_at	TIMESTAMP	Detail record creation time
updated_at	TIMESTAMP	Time of last detail update

The Waste Collection Details table implements a composition relationship with Pick Up, where detail records cannot exist without a parent pickup request (CASCADE delete). Each pickup can contain multiple waste types, with each type having separate weight tracking. The dual weight fields (berat_nasabah and berat_verifikasi) support the system's two-stage weight verification process: customers provide estimated weights when requesting pickup, and collectors verify actual weights during collection. The system's business logic uses berat_verifikasi for final price calculations when available, falling back to berat_nasabah if verification hasn't occurred yet. This approach provides transparency by allowing customers to see both their estimates and verified weights, building trust in the verification process. The gambar field allows documentation of specific waste conditions, useful for quality control and dispute resolution. Total transaction value for each detail row is calculated as: berat_verifikasi × jenis_sampah.harga.

f. Table Waste Types

Table 4.8 display The Waste Types table is a master data table that defines waste types and their pricing. This table serves as a reference for categorizing and pricing recyclable waste.

Tabel 4. 8 Table Waste Types

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
nama	VARCHAR(255)	Waste type name (e.g., Plastic Bottles, Aluminum Cans)
kategori	VARCHAR(100)	Waste category grouping
deskripsi	TEXT	Description of waste type
satuan	VARCHAR(50)	Unit of measurement (kg, pcs, etc.)
harga	DECIMAL(10,2)	Price per unit (direct storage, no price history)
created_at	TIMESTAMP	Waste type registration time
updated_at	TIMESTAMP	Time of last waste type update

Column Name	Data Type	Information
deleted_at	TIMESTAMP	Soft delete timestamp (NULL if not deleted)

The table manages the master data for recyclable waste types accepted by the system. The harga field stores the current price per kilogram for each waste type. An important architectural decision was made to store price directly in this table rather than using a separate price history table (harga_sampah). This simplification prioritizes ease of use and query performance over historical price tracking, which is acceptable for the thesis project scope. The kategori field allows logical grouping of waste types (e.g., plastic, metal, paper) for reporting and filtering purposes. The table implements soft delete to prevent accidental removal of waste types that have historical transaction records. When a waste type is deactivated, existing transactions remain valid while new pickups cannot select the deactivated type. The pricing stored in this table is used throughout the system for automatic transaction value calculation, ensuring pricing consistency across all modules.

g. Transaction Table

Transaction table records financial transactions generated from waste pickups. This table implements dual verification system for split payment scenarios (customer and system/donation parts).

Tabel 4. 9 Transaction Table

Column Name	Data Type	Information
id	BIGINT(20) UNSIGNED	Primary Key, Auto Increment
nasabah_id	BIGINT(20) UNSIGNED	Foreign Key to nasabah (transaction beneficiary)
pengepul_id	BIGINT(20) UNSIGNED	Foreign Key to users (collector who processed)
penjemputan_id	BIGINT(20) UNSIGNED	Foreign Key to penjemputan (source pickup)
jenis_sampah_id	BIGINT(20) UNSIGNED	Foreign Key to jenis_sampah (waste type)
berat	DECIMAL(10,2)	Weight in kilograms

Column Name	Data Type	Information
total_harga	DECIMAL(15,2)	Total price (berat × harga per kg)
tanggal_transaksi	DATETIME	Transaction timestamp
catatan	TEXT	Transaction notes
sistem	BOOLEAN	Whether transaction is for system (donation)
nasabah	BOOLEAN	Whether transaction is for customer
gambar_bukti_nasabah	TEXT	Payment proof for customer part
gambar_bukti_sistem	TEXT	Payment proof for system/donation part
verified_by_nasabah	INT	Customer user ID who verified (for nasabah=true)
verified_by_admin	INT	Admin user ID who verified (for sistem=true)
verified_at_nasabah	TIMESTAMP	Customer verification timestamp
verified_at_admin	TIMESTAMP	Admin verification timestamp
status	INT	Transaction status: 0=pending, 1=verified, 99=rejected
alasan_penolakan	TEXT	Rejection reason (if status=99)
created_at	TIMESTAMP	Transaction creation time
updated_at	TIMESTAMP	Time of last transaction update

This table implements a sophisticated dual verification system to support flexible payment allocation options. When a pickup has payment_option='donate_partial', the system creates separate transaction records with different flag combinations: some records have nasabah=true (require customer verification), others have system=true (require admin

verification for donation tracking). This design ensures that customers can verify their payment portion independently of administrators verifying donation portions. The verified_by_nasabah and verified_by_admin fields record who performed verification, providing clear audit trail. Transaction records are generated automatically from penjemputan_sampah_details data when a pickup reaches 'weight_verified' status. The total_harga field may be proportionally calculated based on donation percentages for split transactions. The gambar_bukti_nasabah and gambar_bukti_sistem fields store separate payment proof images when collectors upload evidence of fund transfers. The status field uses integer encoding where 0 indicates pending verification, 1 indicates verified and processed, and 99 indicates rejected with reason stored in alasan_penolakan. This table serves as the foundation for financial reporting and reconciliation processes.

User Interface Design

The user interface design at this results stage illustrates the implementation of the application interface developed based on the interface design in Chapter III. The interface is designed to support usability and ensure that each user can access system features according to their respective roles.

The interface implementation refers to the previously designed high-fidelity prototype, which was then realized into functional web pages. Each interface is designed consistently so that the system workflow can be easily understood by users. The following are some of the results of the interface design. Here some interface design :

a. Home Page Interface Design (Guest)

The home page is designed as the main gateway for general users who have not yet entered the system. This display carries an informative concept by displaying an environmental illustration banner ("Hero Section") at the top, followed by points of the application's main features.

The middle of the page displays a prominent call to action to encourage users to register as customers or collection partners. The main navigation is placed at the top (Header) for easy access to the Login and Registration menu. The Home Page interface design can be seen in Figure 4.16.

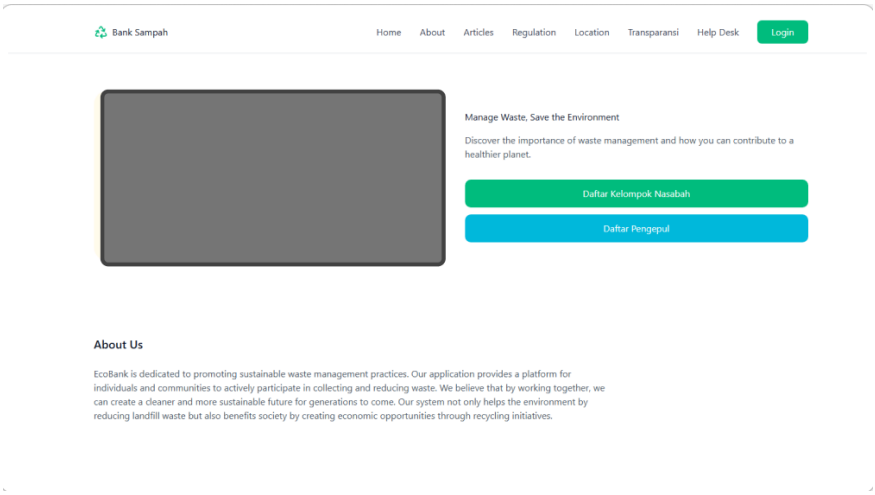


Figure 4. 16 Home Page Interface

b. Collector Registration Page Interface Design

The interface design for the collector registration page is focused on collecting partner identity data efficiently. This page displays a form to fill in personal data which includes Full Name, NIK, and Telephone Number.

There is a special area for uploading verification documents (KTP) to guarantee partner validity. At the bottom of the form, there is a terms and conditions agreement box and a "Register Partner" button in green primary color. The interface design for the collector registration page can be seen at Figure 4.17.

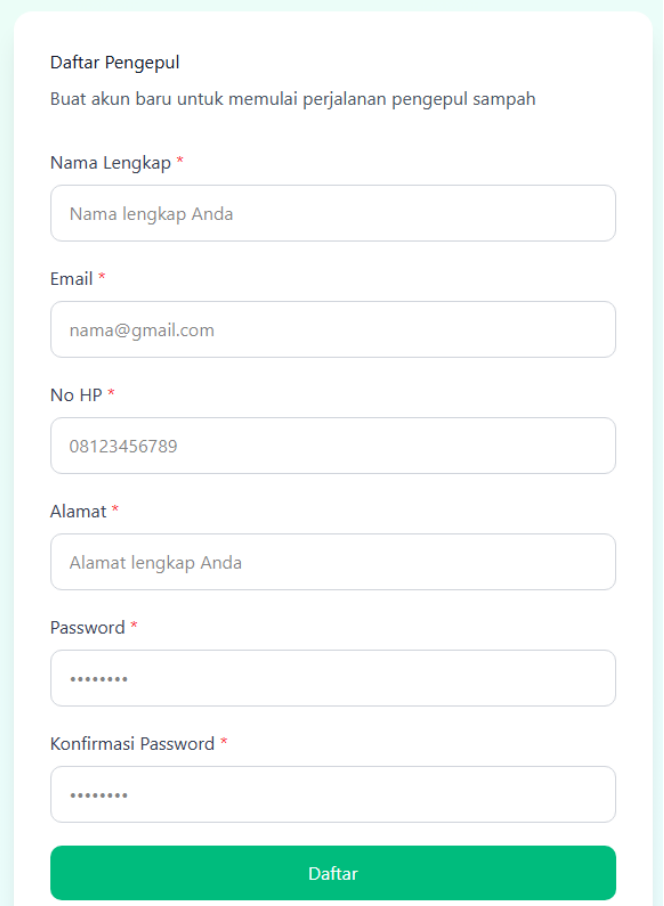


Figure 4. 17 Collector Registration Page

c. Main Dashboard Interface Design

The main dashboard is designed as an information control center for users after successful login (Login). The top of the interface displays statistics cards (Stat Cards) that provide a quick summary of the total balance, weight of waste collected, and account active status.

On the left there is a navigation panel (Sidebar) which contains quick access menus to operational features. The center area of the page provides a monthly activity visualization graph to help users monitor their trash deposit trends. The main dashboard interface design can be seen in Figure 4.18.

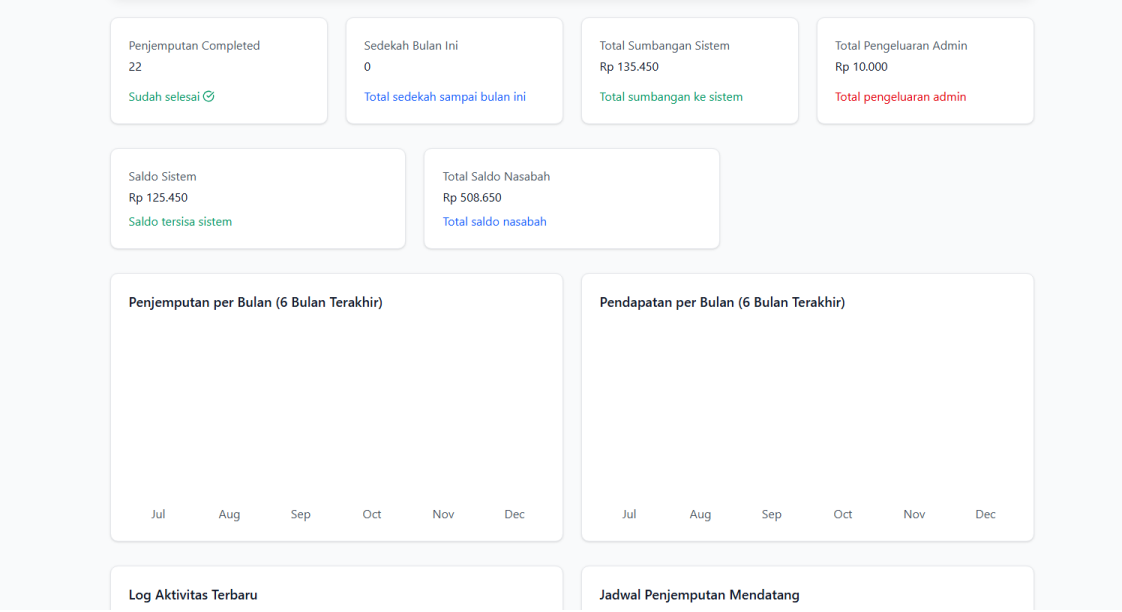


Figure 4. 18 Main Dashboard Interface

d. Create Pickup Page Interface Design

This page is designed to facilitate customers in submitting waste pickup requests. The interface is structured in a structured manner starting from selecting a waste category, inputting estimated weight, to uploading photos of waste evidence.

A key feature of this plan is the choice of donation options, where users can choose to take the proceeds themselves or donate them. The "Send Request" button is placed in a strategic position to complete the process. The interface design of the create pickup page can be seen in Figure 4.19.

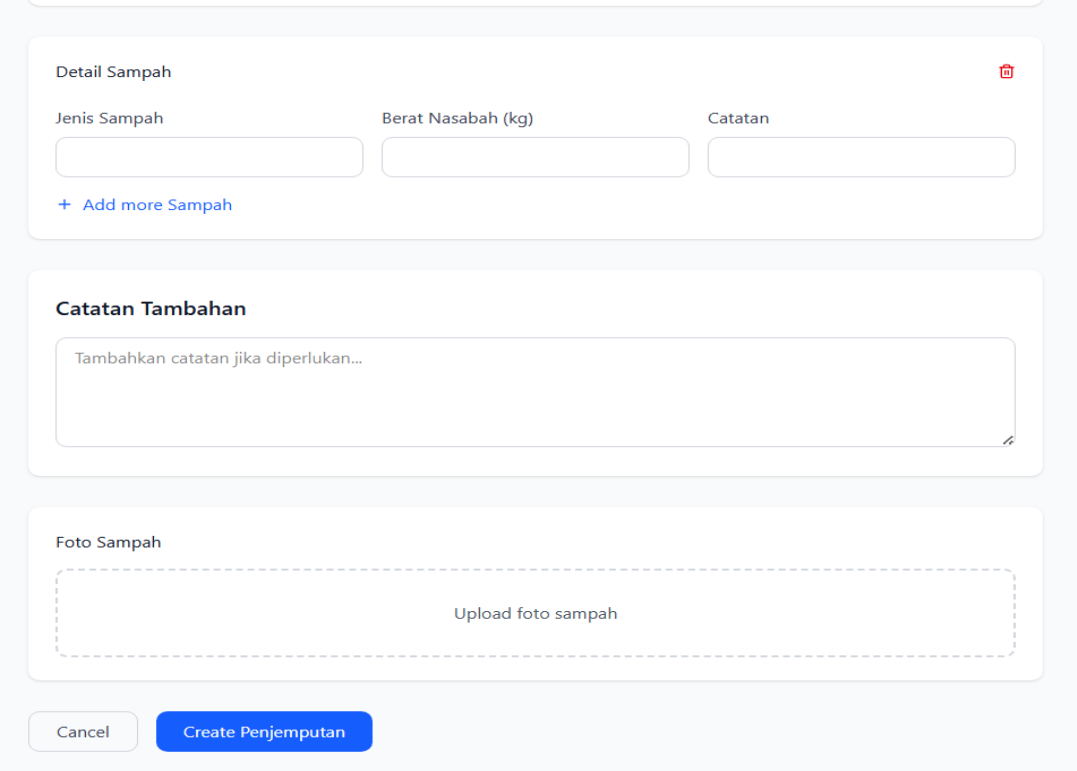


Figure 4. 19 Create Pickup Page Interface

e. Waste Weight Verification Page Interface Design

The heavy verification page is designed for transaction transparency between collectors and customers. This interface displays a comparison table between the customer's "Initial Estimate" and the "Actual Weight" from the collector's weighing results.

The system automatically recalculates the total price based on actual weight input. There is also a column for uploading photos of digital scale evidence as an audit trail. The plan ends with a verification button to approve the transaction. The design of the weight verification page interface can be seen in Figure 4.20.

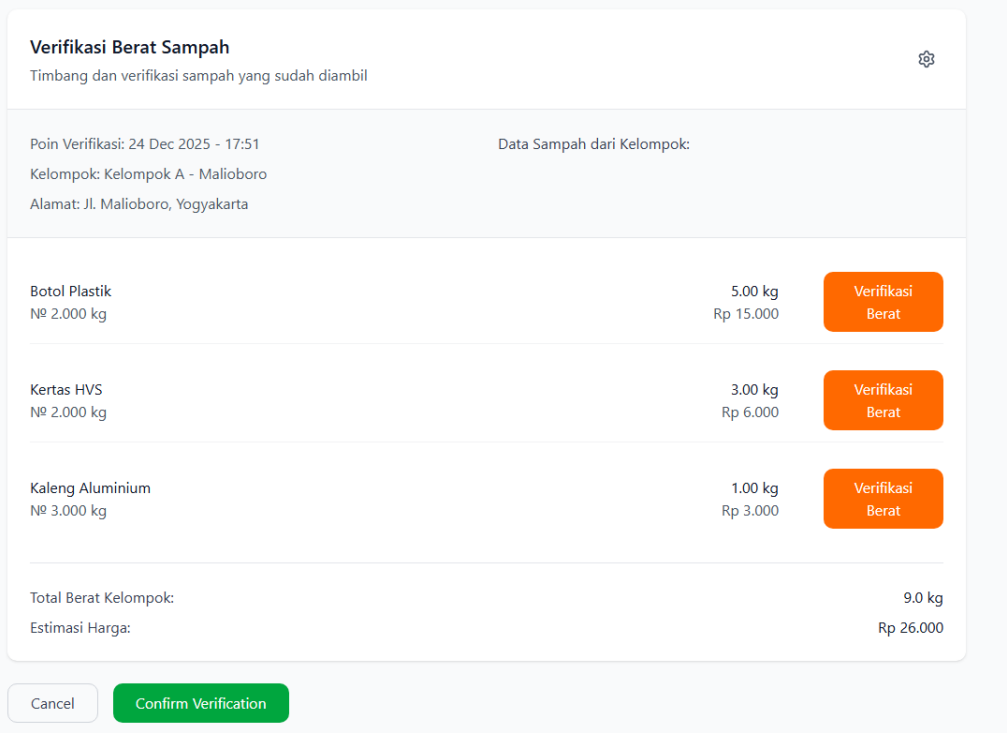


Figure 4. 20 Waste Weight Verification Page

f. Payment System Interface Design

The payment confirmation interface is designed to enhance transaction transparency and accountability in the waste bank system before non-cash payment processing is executed. This interface presents a clear summary of the transaction, including the total payment amount, detailed waste information (waste type, weight, unit price, and subtotal), and the distribution of transaction value. By displaying this information explicitly, users can verify the accuracy of transaction data prior to payment confirmation.

In addition, the interface provides payment allocation options, allowing users to determine whether the transaction value is fully credited to the customer's savings balance, fully allocated as a donation, or partially distributed between balance and donation. This design supports transparency in fund allocation and reduces ambiguity regarding transaction outcomes.

Once the user confirms the transaction by selecting the "Pay Now" option, the system activates the Midtrans Snap payment popup as a secure payment gateway interface. Through this popup, users can choose various payment methods such as QRIS, e-wallets, and Virtual Accounts, ensuring flexibility and secure payment processing. The integration with Midtrans enables automatic payment status updates, supports transaction traceability, and provides a reliable audit trail for financial records. The design of the payment confirmation page interface

is presented in Figure 4.21, while the Midtrans Snap payment popup interface is shown in Figure 4.22.

Figure 4. 21 Payment Confirmation Page Interface

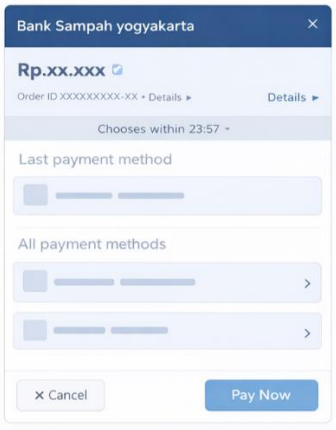


Figure 4. 22 Midtrans Snap payment popup interface

4.1.3 Implementation

After going through the stages of needs analysis and system design, the next stage is implementation. At this stage, the interface design that has been created is realized into a

functional web-based application. The following are the results of the implementation stages that have been carried out in the development of the Waste Bank application.

a. Landing Page (Guest)

The landing page shown in Figure 4.23 is the initial application display, accessible to all users without requiring authentication. This page serves as a general information portal regarding the Waste Bank system and serves as the entry point for users wishing to register or log in. At the top of the page is a sticky navigation bar that displays the Waste Bank logo, the main navigation menu, and a login button for users wishing to access the system.

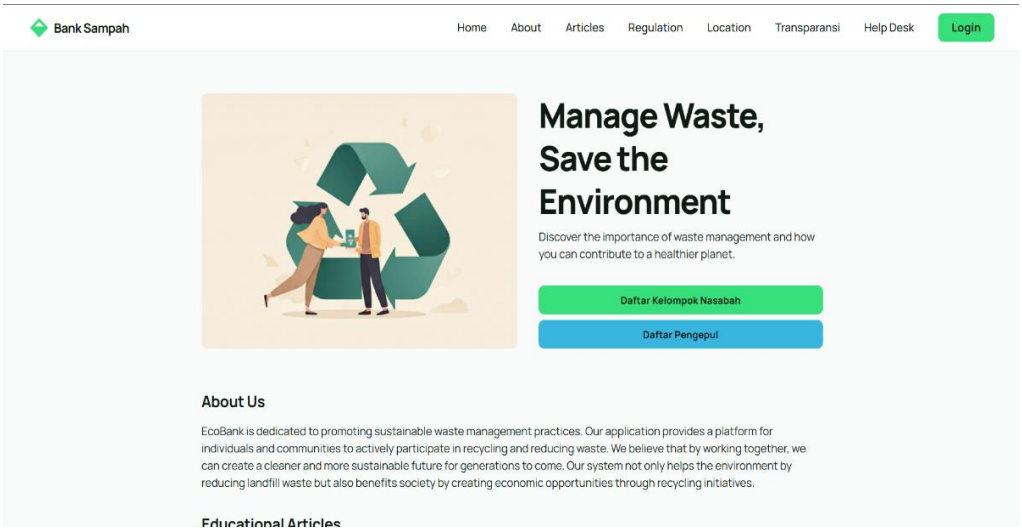


Figure 4. 23 Landing Page Implementation

The Hero Section on this page displays the main headline accompanied by a brief description of the importance of waste management and a visual illustration representing the waste management process. This section also provides two Call-to-Action buttons, namely the Customer Group List and the Collector List, which direct users to the registration page according to their selected role. In addition, the landing page is also equipped with an About Us section that explains the system profile, an Educational Articles section that displays the latest educational articles in card form, and a Donation Transparency section that provides a summary of donations, expenditures, donation balances, and fund distribution history in real-time from the system database. Thus, this landing page functions not only as a system introduction page, but also as a transparent and informative public information tool.

b. Account Login Page (Login)

The login page is the primary authentication gateway for all system users, including Administrators, Collectors, and Customers. As shown in Figure 4.24, this page displays an

authentication form that prompts users to enter their registered email address and password. This page is equipped with a validation mechanism, so if a user enters incorrect credentials, the system displays an error message. After successful authentication, the system redirects the user to the dashboard page appropriate to their access rights or role.

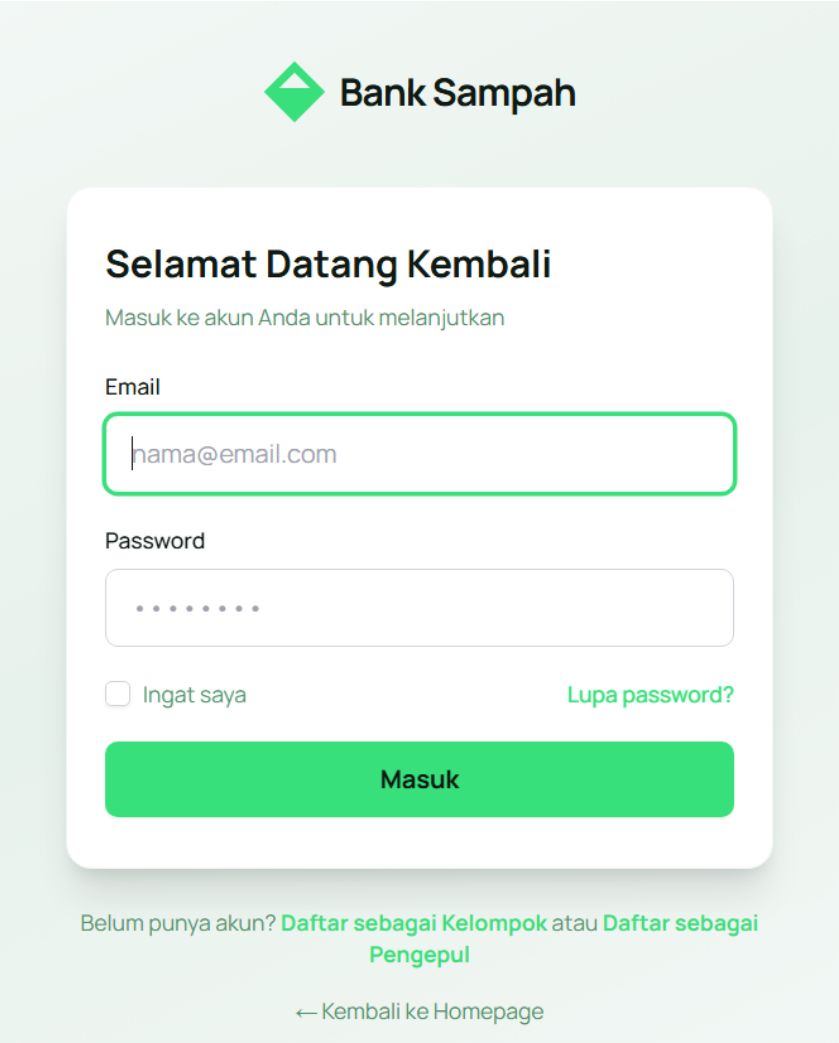


Figure 4. 24 Login Page Implemantation Result

The login page is designed with a simple layout and focuses on the authentication form, placed in the center of the page with a light green background. The login form consists of two main input fields: Email and Password. The email field is equipped with a "name@email.com" placeholder and email format validation. The password field displays hidden characters to ensure the security of user input. If the input format is incorrect or the email/password is invalid, the system displays a red error message below the field.

Below the password field are two additional elements: a "Remember me" option that allows the system to store user logins for a longer period of time, and a "Forgot password?"

link that directs users to the password recovery page. The "Login" button sends credentials to the server for validation using Laravel's authentication mechanism. If the data is valid, the system checks the user's role using Spatial Permissions and redirects them to the appropriate dashboard: the Filament Admin Panel for Admins, the Collector Dashboard for Collectors, and the Customer Dashboard for Customers or Customer Groups.

c. Customer Dashboard Page

After successfully logging in as a customer group, users will be directed to the Customer Group Dashboard page, as shown in Figure 4.25. This page serves as the main information center for the customer group, displaying a summary of account status, waste deposit statistics, collection activity, and transaction history. This dashboard page is designed to allow customer groups to comprehensively monitor activity developments, including balances, order quantities, and waste management achievements.

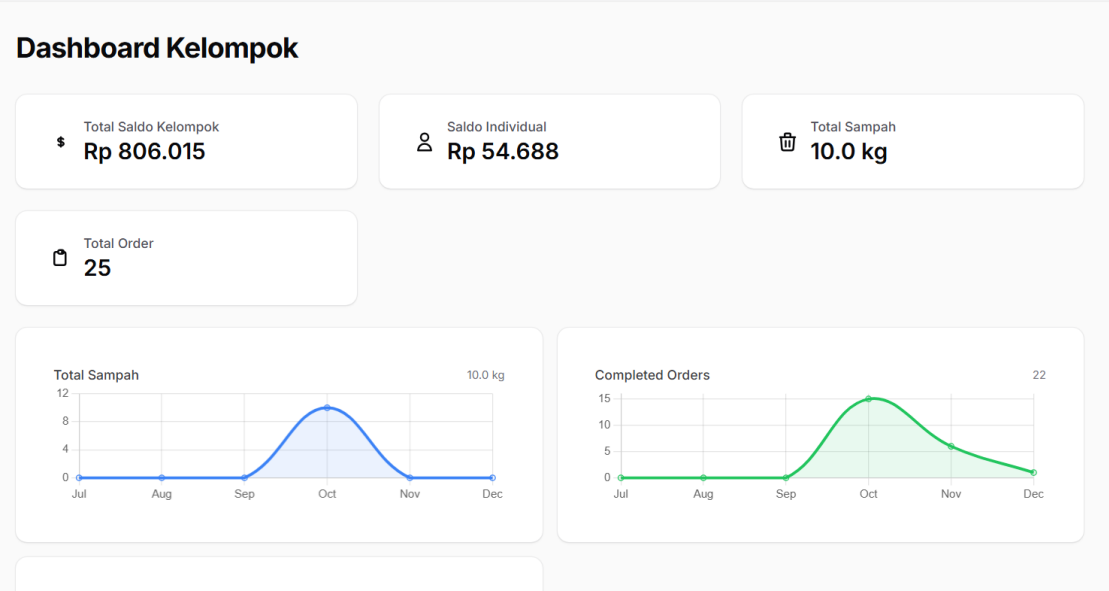


Figure 4. 25 Customer Dashboard Implementation Result

At the top of the page, a statistics widget section is displayed, consisting of four information cards arranged in a responsive grid layout. The first card displays the Total Group Balance, which represents the accumulated balance of all group members. The second card displays the Individual Balance, which is the balance of the logged-in user. The third card displays the Total Waste, which indicates the total accumulated weight of waste collected by the group in kilograms. The fourth card displays the Total Orders, which is the total number of pickup requests made. All cards are displayed with representative icons and an easy-to-read numerical format, making it easy for users to quickly understand the information.

The next section is the Statistics Charts section, which displays three graphs representing the development of group activity over the past six months. The first graph shows the trend in the total weight of waste collected per month. The second graph displays the number of pickup requests with a completed status, while the third graph displays the number of pickups that are still pending. These graphs are visualized using Chart.js, providing a responsive and easy-to-understand interface, helping users analyze the development of group activity over time.

The Active Pickup section displays a list of pickup requests that are still in progress, including those with pending, assigned, or in progress status. Each pickup request is displayed as a card containing the status, date, pickup address, collector name (if assigned), and details of the waste type and estimated value. For items with pending status, users are also provided with a Cancel button to cancel the pickup request after first going through a confirmation dialog. If there are no active pickups, the system displays an empty screen, indicating that no pickups are currently in progress.

Furthermore, the Pickup History section displays the last five pickup activities that have been completed or canceled. Information displayed includes the pickup status, date, collector name, and the total compensation for successfully completed pickups. This section also includes a Details button for viewing more complete pickup history information, as well as a Repeat Order button for successful pickups, which allows users to create a new request with the same data.

At the bottom of the page, there's a Quick Actions section, which contains two main buttons: the Create New Pickup button to create a waste pickup request, and the View All Pickups button to access a complete list of all pickups. Navigation to this page is done through a sidebar configured according to role, so only users with access rights as a customer group can view this dashboard menu.

d. Pickup Request Form Page (Customer)

The core feature of the Waste Bank application is the waste pickup request service, implemented through the Pickup Request Form page, as shown in Figure 4.26. On this page, users with the customer group role can submit a waste pickup request by filling in the required information. This page is designed to ensure that the pickup request process is organized, systematic, and in accordance with the operational capacity of the collector.

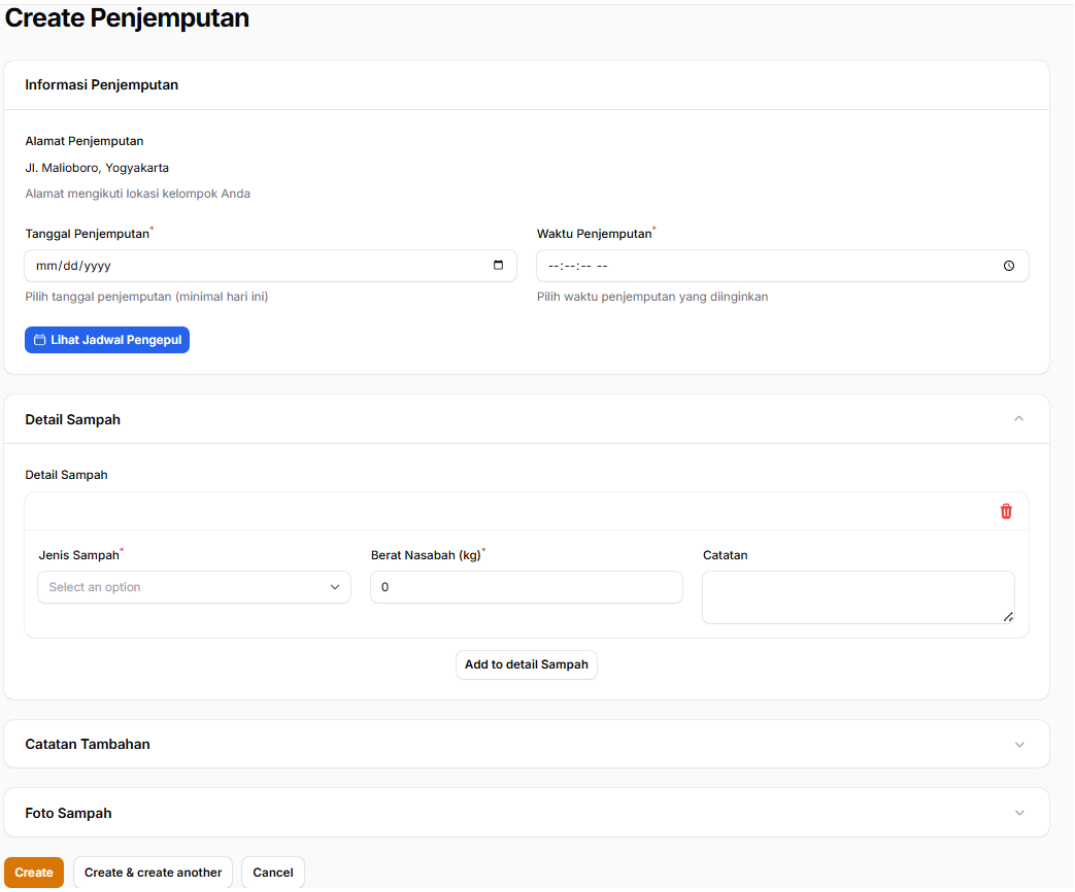


Figure 4. 26 Result of Implementation of Pick Request Form

The pickup form page is divided into several sections to facilitate user input. The first section is the pickup information section, which consists of the Pickup Date and Pickup Time input. Date selection is performed using the date picker component, limited to dates within the permitted operating hours. Meanwhile, time selection is performed using the time picker, which only provides options for pickup operating hours, preventing users from selecting a time outside of service hours.

The next section is the Full Address input, which is automatically filled in based on the address stored in the customer group profile. Users are still given the flexibility to change this address if the pickup location differs from the address stored in the system. A minimum character limit is applied to ensure the entered address is clear and informative.

The Waste Details section allows users to add more than one type of waste to a single pickup request. Here, users can select the waste types available in the system and enter the estimated weight in kilograms. The system is also equipped with validation to ensure the weight input is logical and to prevent duplicate selections of the same waste type in a single request.

In addition, an optional Additional Notes section is provided. This section is used to provide additional information to collectors, such as location access, environmental conditions, or special needs during the pickup process. There's also a section for uploading waste photos to provide a visual overview of the condition of the waste to be picked up. Although optional, this feature helps collectors estimate and prepare before the pickup.

At the end of the page, there's a Save button to submit the pickup request data. Before saving the data, the system will validate all mandatory fields. If the validation process is successful, the pickup request will be saved to the database with an initial pending status, and the user will receive a notification that the request was successfully created. The user will then be directed to the pickup list page to monitor the status of their submitted request.

e. Collector Dashboard Page (Open Orders)

For users with the collector role, the main page displayed is the Open Orders list, as shown in Figure 4.27. This page displays all waste pickup requests that are still pending and ready to be picked up by collectors. The order pickup mechanism uses a First Come First Served (FCFS) system, so each order can only be claimed by the collector who made the first pick-up. Information displayed on each order includes the pickup area, waste type, estimated weight, and time the request was made. To maintain user privacy, full address details are only displayed after the order has been successfully claimed. Each order card provides a Pick Up Order button, which serves to lock the order and assign it to the relevant collector.

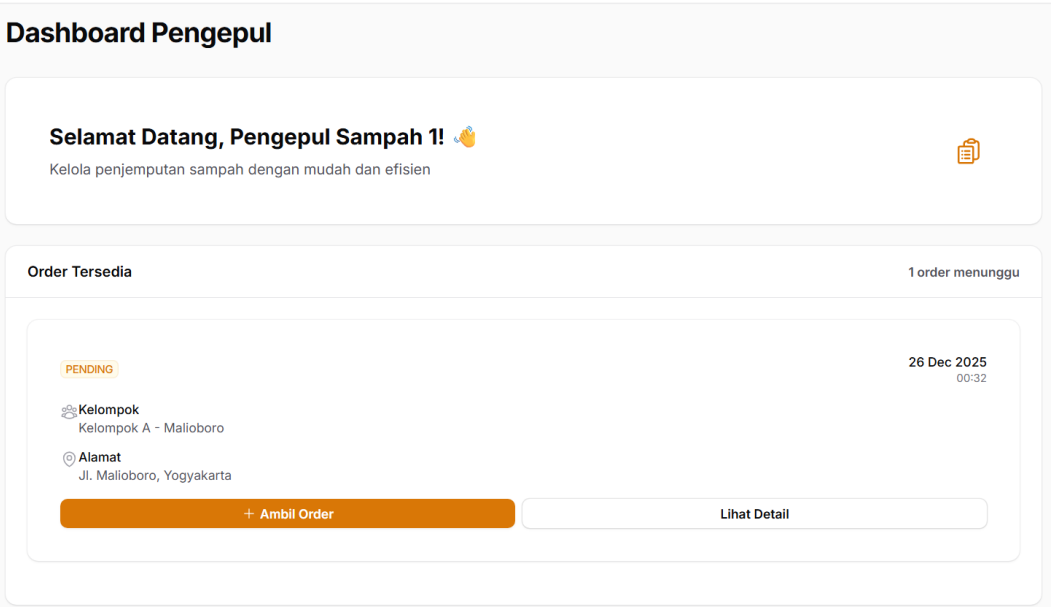


Figure 4. 27 Collector Dashboard Implementation Result

The collector dashboard is implemented with a focus on operational efficiency. The top of the page displays the Available Orders section, which shows the number of pending orders ready for pickup, along with an order quantity indicator and a data refresh feature. The list of pending orders is displayed as a card containing the order status, pickup schedule, customer group identity, location address, and additional notes, if available. Orders are displayed sequentially based on pickup schedule, with orders with closer priority appearing first, in accordance with the FCFS principle. When the Pick Up Order button is pressed, the system assigns the order to that collector and changes the order status to assigned, preventing it from being picked up by another collector. The system also provides an order detail view for viewing waste types, estimated weight, and compensation amounts.

Below the list of pending orders is the Collector Statistics section, which displays four key indicators: total pickups, the number of pickups for the current day, total payments generated from completed pickups, and a reminder indicator for uploading outstanding proof of transactions. This section helps collectors periodically monitor their activity performance.

Furthermore, a Quick Actions section provides quick access buttons to important operational features such as waste weight verification and payment processing. This feature aims to streamline collector workflows, allowing for more efficient operational processes.

At the bottom of the dashboard, an Analytics Dashboard presents data visualizations in graphical form. This graph displays trends in the number of orders over the past six months, the number of successfully completed orders, and the number of pending orders. This

visualization helps collectors and system managers analyze operational activity trends over time.

f. Details & Weight Verification Page

After the collector successfully picks up the order and arrives at the pickup location, the system provides a Waste Weight Verification page, which serves as a field validation process for the actual waste weight. This page displays detailed pickup information such as the customer group name, pickup date and time, location address, and any additional notes the user may provide. This page also provides a "Weight Verification" button to proceed to the actual weight input process, as shown in Figure 4.28.

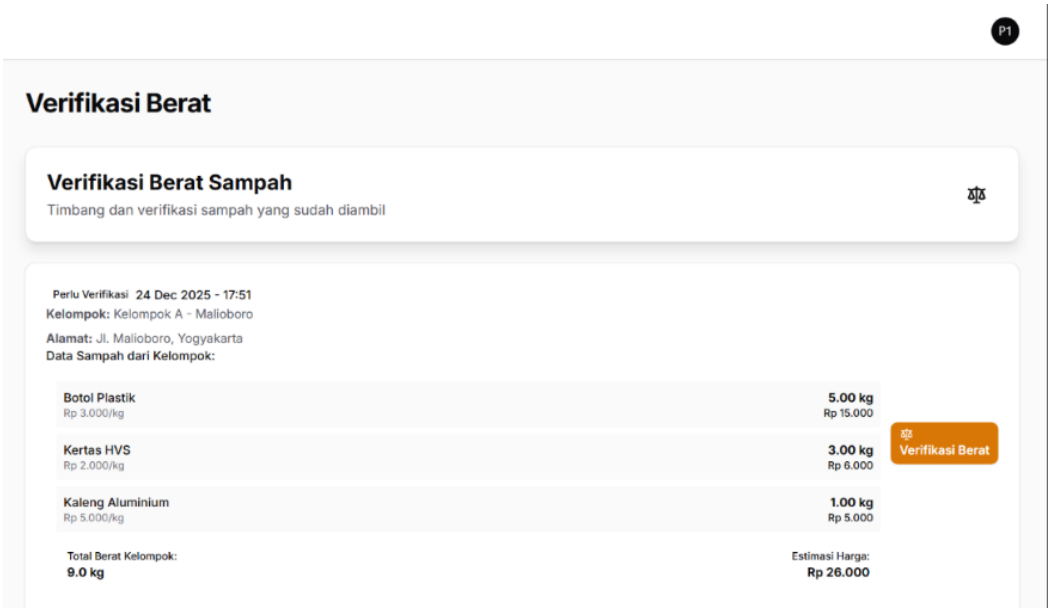


Figure 4. 28 Implementation Result Verification Page Waste Weight

This page ensures that collectors receive complete information before carrying out the weighing process. If a customer group has previously entered an estimated waste weight, the system will display it as a reference before the actual weighing. Additionally, there is a facility to upload photos of the weighing receipts, which can be used as documentation to minimize the potential for data disputes later.

After the collector clicks the "Verify Weight" button, the system will direct the user to the full weight verification page. This page allows collectors to enter the actual weight of each type of waste from the field weighing and forms the basis for calculating the transaction value, as shown in Figure 4.29.

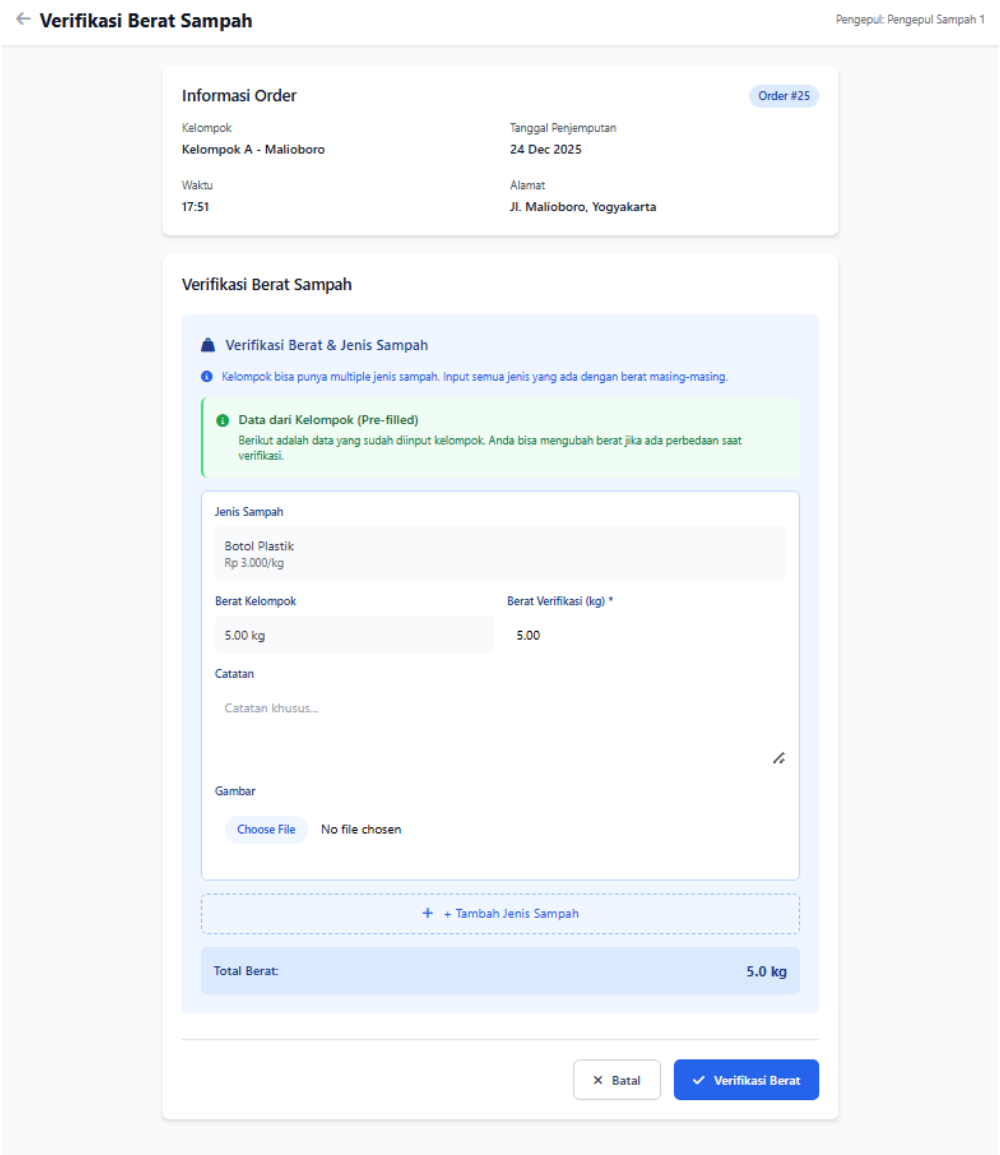


Figure 4. 29 Weight Verification Input Page Implementation Results

The weight verification page is implemented in full-page format, making it convenient to use on both desktop and mobile devices. The page title and back navigation button are displayed at the top of the page. Below it, an information card displays the pickup details for reference during the verification process.

The verification form allows collectors to enter the actual weight for each type of waste being picked up. If the user has previously entered estimated data, the system will display it again, allowing collectors to confirm and adjust. The system also provides a feature for adding new waste types if additional waste is added. All input is validated to ensure the values are logical and to avoid data duplication.

At the bottom of the page, the total weight is displayed, automatically calculated based on the collector's input. Two main buttons are available: Cancel, which returns to the previous

page without saving changes, and Verify Weight, which saves the verification results. If all data is valid, the system will update the pickup status to Weight Verified and direct the process to the next stage, payment.

g. Payment Page (Payment Gateway & Donation Options)

After the verification process is complete, the system will direct the user to the Payment Option Selection page. This page allows users to determine the distribution scheme for transaction funds based on their preferences. This page displays a summary of the payment amount along with three main options: Take All, Donate Partially, and Donate All. The Take All option will allocate all funds to the customer's balance, Donate Partially will divide the funds between the customer's balance and donations according to a specified percentage, while Donate All will direct all funds to the cash donation system. A screenshot of this page can be seen in Figure 4.30.

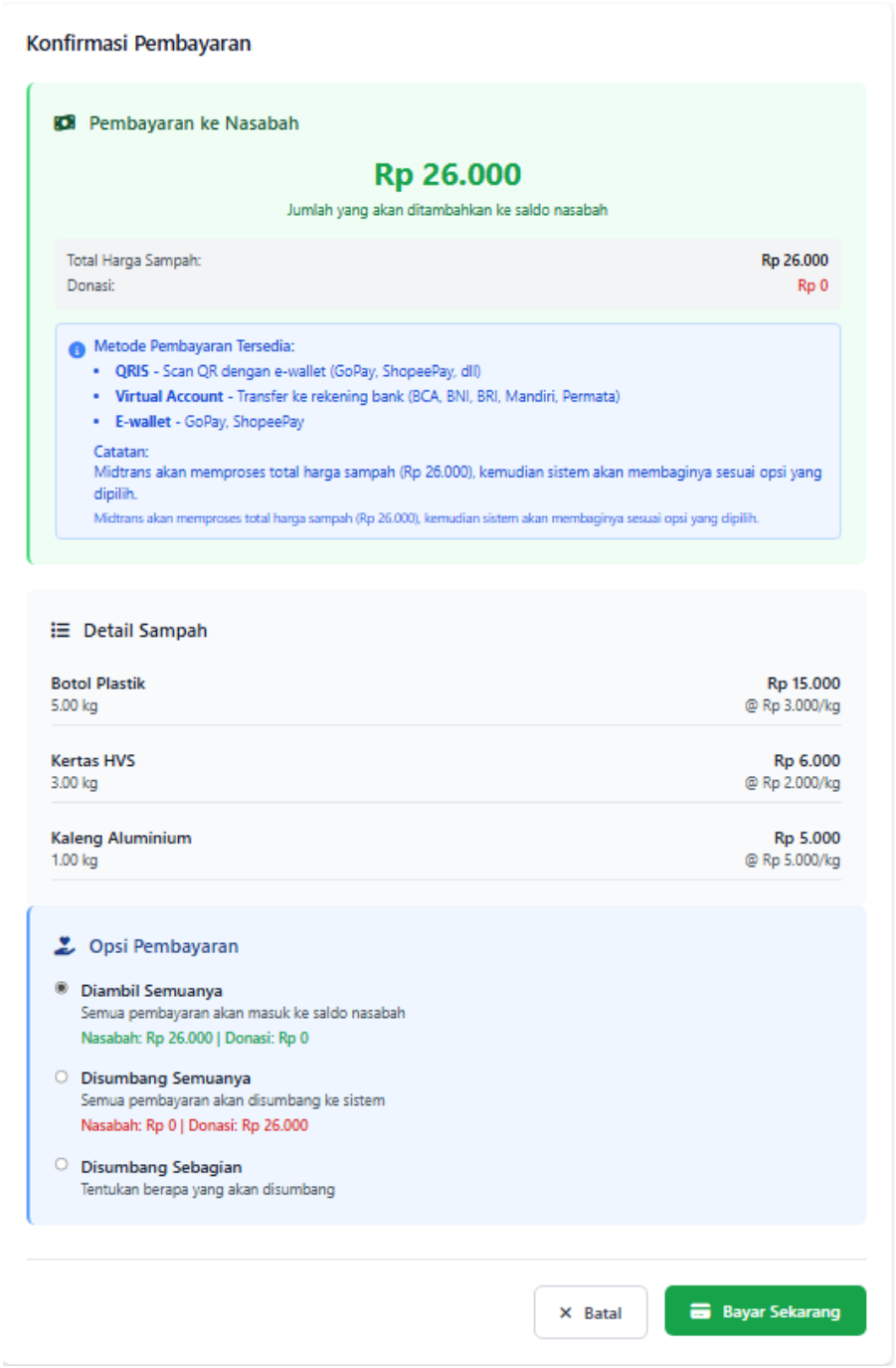


Figure 4. 30 Payment Option Selection Page

In addition to providing a selection of options, this page also displays a summary of the transaction value, including the total payment, estimated funds received by the customer, and estimated funds allocated for donations (if the donation option is selected). This information is presented to ensure transparency and provide users with a clear understanding before proceeding with the payment process.

After the user selects a payment option, the next step is to complete the digital transaction. When the "Process Payment" button is pressed, the system will initiate the payment through the Midtrans Payment Gateway and display a pop-up payment window (Snap Popup) on the same page, eliminating the need for users to switch pages. The payment window is shown in Figure 4.31 below.

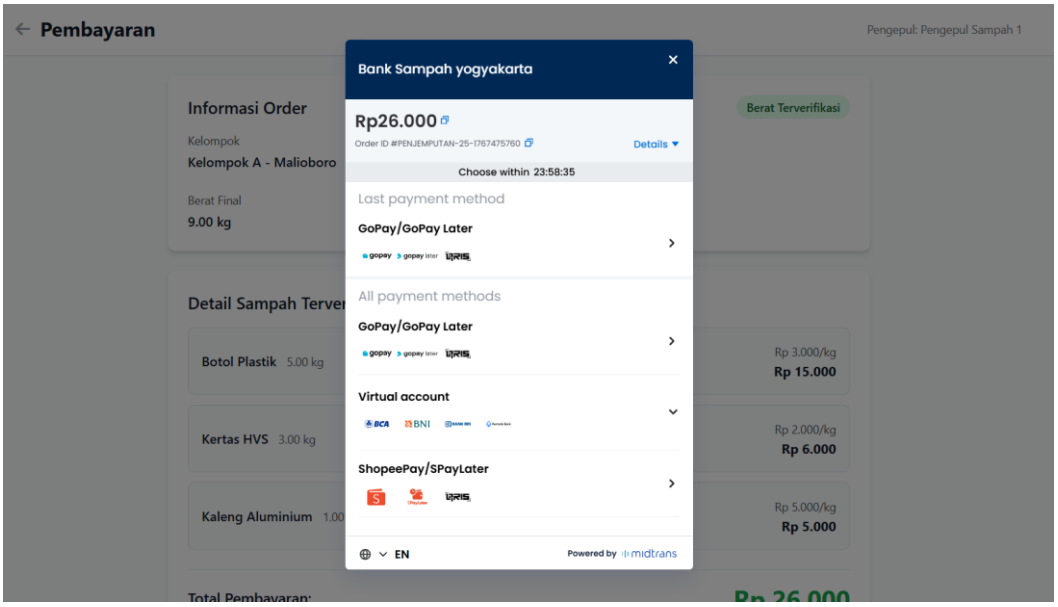


Figure 4. 31 Implementation of Midtrans Payments (Snap Pop-up)

In this payment window, the system displays a transaction summary along with various payment methods to choose from, such as QRIS, Virtual Account, E-Wallet, and credit card. With these various payment methods available, users have the flexibility to choose the most suitable payment method. Once the payment process is complete, Midtrans will automatically send a transaction status notification to the system. If the payment is successful, the system will update the transaction status to successful, record the payment history, update the customer's balance, and record the donation funds if there is a donation allocation. Conversely, if the payment fails, the system will display a failure notification and provide the option to re-pay.

h. Admin Dashboard Page

From a system management perspective, the Admin Dashboard serves as the central control and monitoring point for the overall application performance, as shown in Figure 4.32. This page presents a summary of the system's operational statistics in the form of graphical visualizations and informative numerical indicators, such as user growth, total volume of waste managed, and cash flow from donations coming through the system. With this information, administrators can monitor system performance in real time and make data-driven decisions.

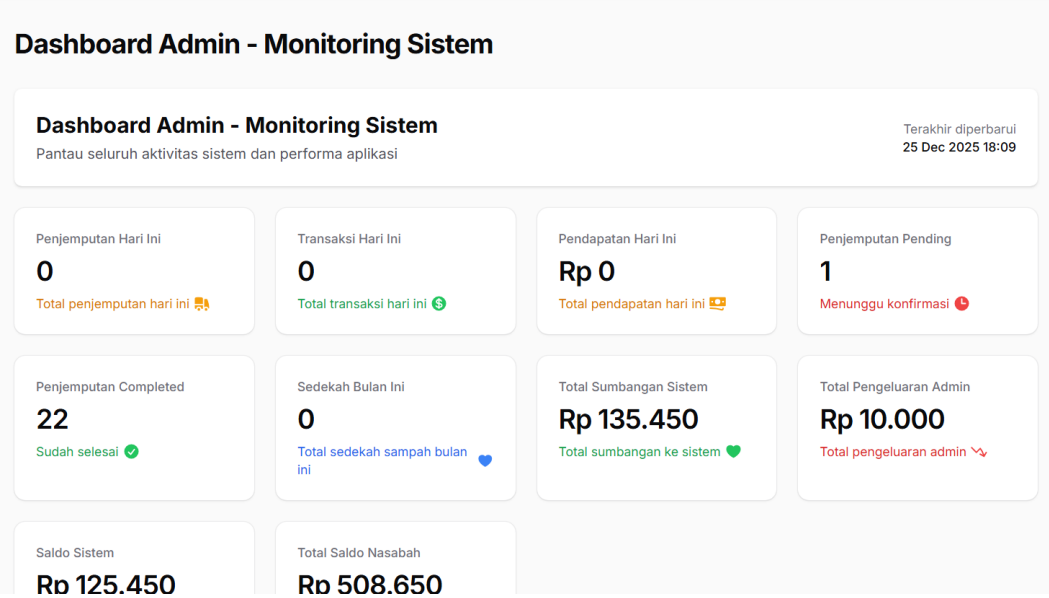


Figure 4. 32 Admin Dashboard Implementation Result

The admin dashboard displays several systematically arranged statistics widgets. These widgets present key indicators such as the number of daily pickups, the number of transactions, and the total revenue generated over a specific period. They also display information regarding the number of pending pickups, completed pickups, the total customer balance, the total donation funds collected, and the remaining system balance after deducting expenses. Presenting data in the form of information cards and monthly trend graphs helps administrators clearly and comprehensively understand the system's operational progress.

In addition to the statistical summary, the admin dashboard also features a monthly pickup trend graph and a monthly revenue trend graph. These graphs display the development of pickup activity and revenue generation over the past six months, allowing administrators to analyze system usage patterns, identify increases or decreases in activity, and plan service management strategies based on these trends.

Furthermore, the system also provides a Recent Activity Log widget that displays a list of recent user activity within the system. This feature serves as an audit trail mechanism that helps administrators monitor system security and trace historical actions in the event of operational errors or system misuse. Additionally, there's an Upcoming Pickup Schedule widget that displays a list of scheduled pickups for the next period, allowing administrators to monitor and coordinate operations more effectively.

All admin dashboard pages are integrated with a panel navigation system that includes the main system management menus, including user management, pickups, transactions, customer groups, master data, reports, and system settings. Thus, the admin dashboard serves

not only as a monitoring center but also as a system management control center, supporting effective, structured, and data-driven monitoring, control, and decision-making.

i. User Management Page (Admin)

The Customer Management page in the Waste Bank system is divided into two main sections: the Customer Management page and the Customer Group Management page, allowing for a more structured and focused administrative process. This separation facilitates administrators' ability to monitor user data, both individually and as a group, and ensures effective, controlled, and accurate customer data management.

On the Customer Management page, as shown in Figure 4.33, the system displays a list of all registered customers in tabular form. Information presented includes email address, telephone number, group (if grouped), date of registration, balance, and account activity status. Administrators can update data, activate or deactivate accounts, and edit customer profiles using the provided action buttons. Furthermore, search, sorting, and pagination features are available, making it easier for administrators to efficiently manage large amounts of customer data.

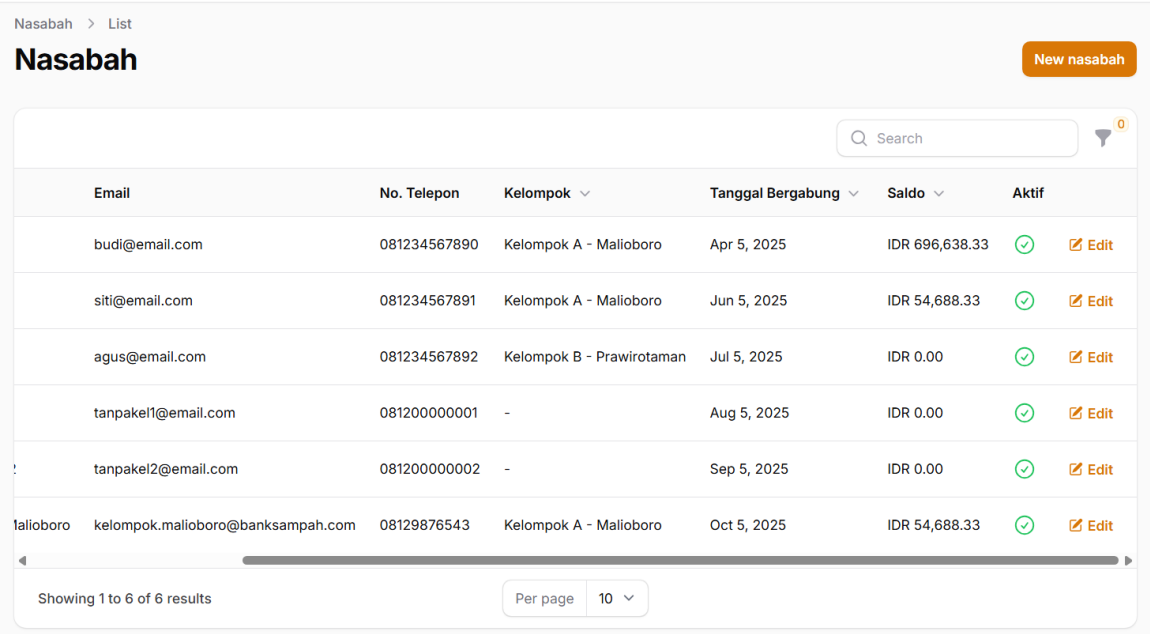


Figure 4. 33 User Management Implementation Result

Furthermore, on the Customer Group Management page shown in Figure 4.34, the system provides a group data management facility that serves as a forum for individual customers. The table on this page displays important information such as the group name, group code, coordinator name, number of members, total group balance, and activity status.

Administrators can add new groups, view group details, and update data using the available action buttons. With this feature, admins can monitor the activity and development of each group more comprehensively, while ensuring that the group's organizational structure remains well-managed.

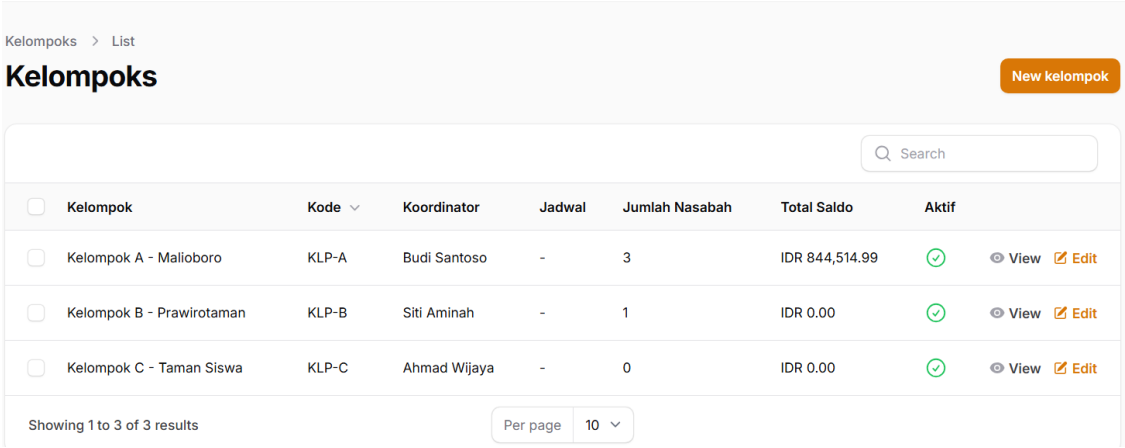


Figure 4. 34 Customer Group Management Page

Both the Customer Management and Group Management pages are equipped with search features, status indicators, and pagination, making it easier for administrators to audit users. Furthermore, the system utilizes a secure data change mechanism through action confirmation and a soft delete approach to maintain data integrity and enable historical retrieval if needed later.

j. Customer Transaction Verification Page

After the collector completes the payment process, the next step is to confirm receipt of funds by the customer, as shown in Figure 4.35. This page is designed to give customers complete control over ensuring the clarity and accuracy of the transaction before the funds are officially disbursed to their account balance. On this page, customers can view complete transaction details, including the payment amount received and the components that make up the transaction value.

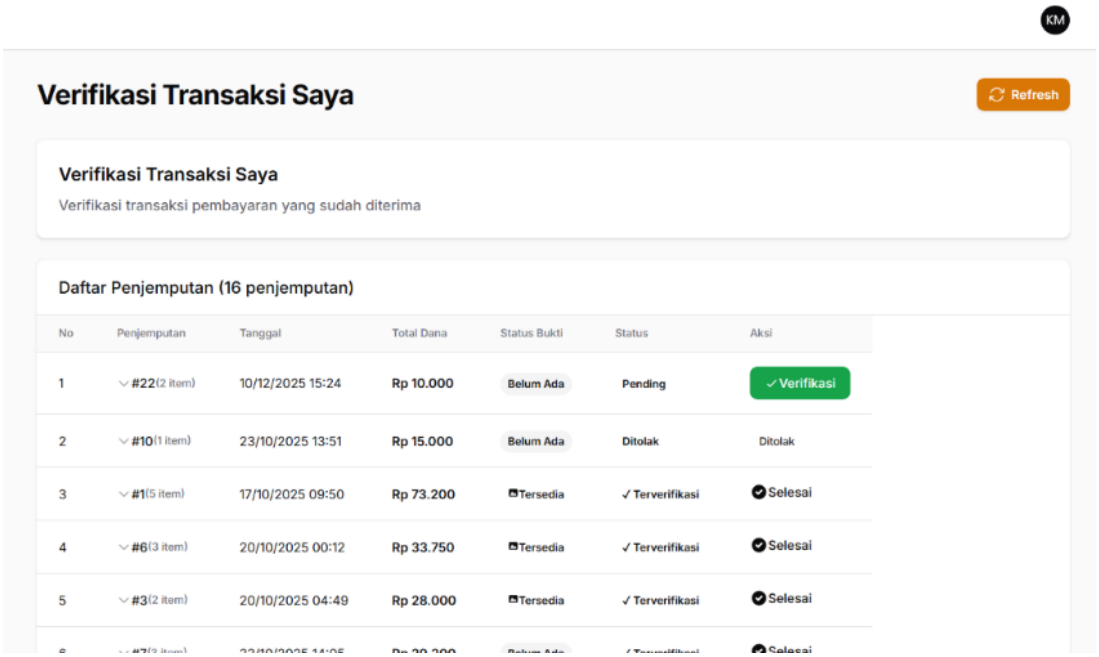


Figure 4. 35 Result of Costumer Verification Implementation

This verification page displays a list of transactions awaiting customer approval. Information provided includes the pickup code, transaction date, number of waste items processed, total payment to be received in rupiah, receipt availability status, and transaction verification status. Users can view transaction details in greater detail using the row expansion feature, which displays the waste type, verified weight, price per kilogram, and total value of each item. This presentation ensures full transparency, allowing customers to review all information before making a decision.

This page provides two main action options for customers. The "Accept" button is used if the payment amount is correct and there are no issues, allowing the transaction to be completed and the balance will automatically be credited to the customer's account. Conversely, the "Submit Complaint" button is provided if there are discrepancies, such as differences in weight, price, or calculation errors. In this case, the system prompts the customer to provide a clear reason for the rejection, which will serve as a basis for further review by the administrator or collector. This mechanism not only ensures transaction accuracy but also protects customer rights and minimizes the potential for disputes.

k. Admin Transaction Verification Page

As a final layer of security in the financial transaction flow, the system provides a Transaction System Verification page accessible only to users with administrator roles, as shown in Figure 4.36. This page serves as an audit mechanism to ensure that all fund

transactions related to donations or the fund system are functioning accurately before being recorded as official transactions. Through this page, administrators have full control to review, validate, and make decisions on transactions involving public funds and are of a sensitive nature.

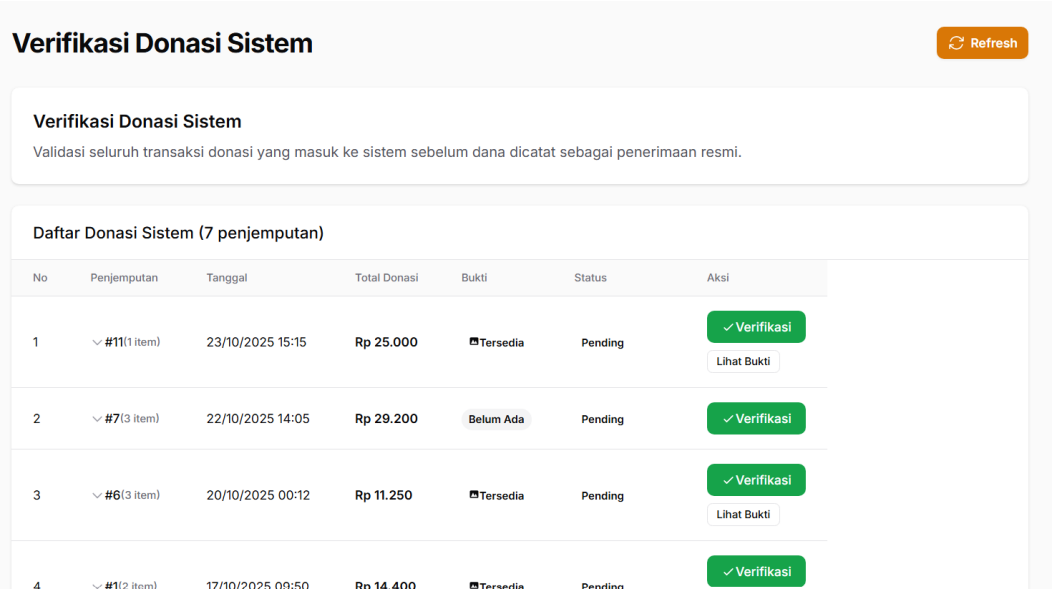


Figure 4. 36 Admin Transaction Verirfication

This page displays a list of system transactions that require verification, particularly those containing donation components, high-risk transactions, or transactions with large amounts of money. Information provided includes the pickup code, transaction date, number of related items, total donation value to be credited to the system cash register, proof of payment availability status, and transaction verification status. Administrators can also view transaction details, including waste type, verified weight, price per kilogram, and the total value allocated to the system. This information helps administrators cross-check transactions before making a final decision.

This page provides two main actions for administrators. The "Verify" button is used if the data is deemed valid and aligns with the available evidence, allowing the transaction to be authorized and officially recorded in the system cash register. Conversely, if data discrepancies, inconclusive evidence, or indications of fraud are found, administrators can press the "Reject" button. In this case, the system requires administrators to provide a clear reason for the rejection as audit documentation and as a basis for follow-up action by relevant parties.

Every verified or rejected transaction is recorded as part of an audit trail, allowing for retrieval of all activity if necessary. Furthermore, this page features pagination for easy

navigation when a large number of transactions need to be verified. Unlike the customer verification page, this page displays transactions from all groups without restrictions, as administrators have full access to all donation data in the system.

l. Financial Report Page

The financial accountability and transparency features in this application are realized through the Financial Reports page, as shown in Figure 4.37. This page is designed as a financial monitoring center, allowing administrators to monitor all incoming and outgoing funds in the system in real time. Every financial transaction is systematically recorded, along with information on the date, transaction type, description, and amount, so that all financial activities can be audited clearly and structured.

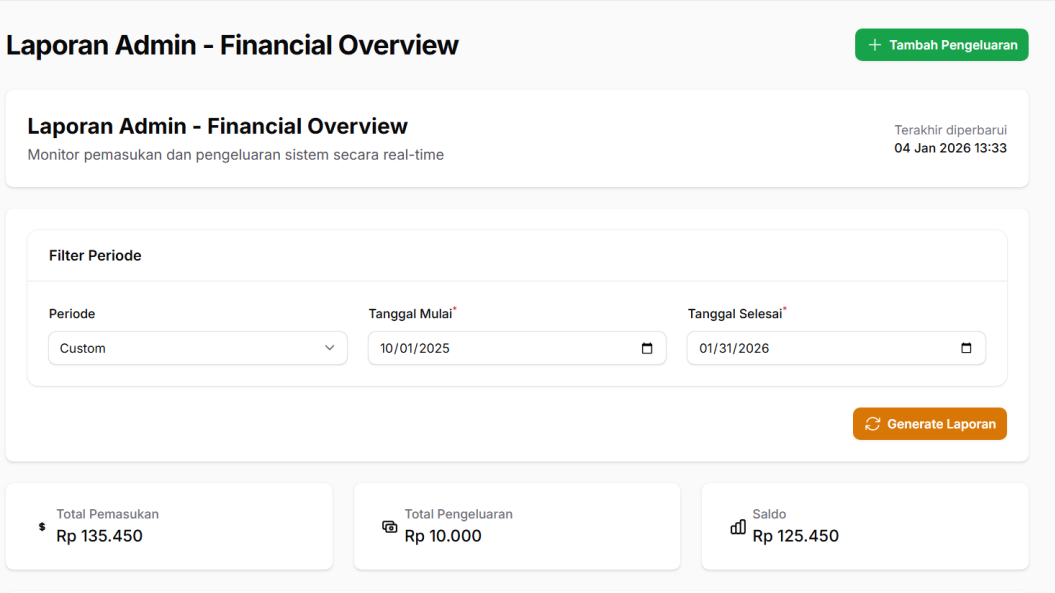


Figure 4. 37 Result of Implementation of Financial Reports

This page provides a financial summary in the form of total income, total expenses, and the system's ending balance for a specific period. Admins can specify the reporting period, whether daily, weekly, monthly, annually, or for a specific date range, as needed. Once the period is selected, the system will automatically display the financial summary based on the filter, allowing administrators to analyze financial conditions quickly and accurately.

The income section displays a list of incoming financial transactions, such as full donations, partial donations, waste donations, and other income sources. Each entry includes the transaction time, income type, description, and nominal amount displayed in rupiah. If there is no income data for a specific period, the system will display a notification indicating that no data is available.

Furthermore, the expenses section displays a list of all outgoing financial transactions made by the system, such as operational expenses, use of donation funds, or other administrative needs. Expense data also includes the transaction date, expense type, description, and nominal amount. To maintain data integrity, some expense types cannot be changed or deleted, while others can still be managed by the admin with the granted access rights.

The system also provides a feature for adding new expenses, complete with balance validation to ensure expenses do not exceed available funds. This ensures the system's financial sustainability, and every transaction is officially recorded. This page also features pagination and a dynamic data update mechanism, ensuring the report display is always up-to-date without requiring a full page refresh.

4.1.4 Testing

Testing is one of the crucial stages in the Waste Bank application development process to ensure that the developed functions operate according to user needs and the designed system specifications. In accordance with the research methodology explained in CHAPTER III, testing in this research was conducted through User Acceptance Testing (UAT), which consists of alpha testing and beta testing, followed by a usability evaluation using the System Usability Scale (SUS).

Respondent Profile and Justification

The respondents involved in the end-user evaluation were selected from a waste bank community group representing potential users who are familiar with waste bank activities. A total of 29 respondents participated in the beta testing and usability evaluation process. Respondents accessed the system through a shared link and were asked to explore the main features before completing the evaluation instruments.

The number of respondents (n = 29) was determined based on participant availability and time constraints during the research period. Nevertheless, this number is considered sufficient to provide an initial evaluation of user acceptance and usability perception, as well as to identify improvement priorities prior to broader implementation.

User Acceptance Testing (UAT)

User Acceptance Testing is divided into two execution stages, namely alpha testing and beta testing. The following is an in-depth discussion of the test results:

a. Alpha Testing

Alpha testing was conducted internally by the researcher as the system developer on December 1, 2025, with the aim of verifying core business logic and the stability of third-party integration, particularly the Midtrans payment gateway, before evaluation by end users. This stage applied the Black-Box Testing method, which focuses on validating system functionality based on input-output behavior without analyzing the internal program code structure. All test scenarios were based on predefined test case documents (details are provided in Appendix A).

Based on the alpha testing results, the main features of the Waste Bank application ran stably. Input validation performed properly to prevent invalid data from being stored, and navigation flow between modules functioned as expected. Several minor issues were identified in responsive display on small mobile screens; however, these issues were corrected prior to the beta testing phase.

Among the tested features, two mechanisms were considered crucial due to their complexity and operational impact:

1. First Come First Served (FCFS) Logic Testing

The first feature that was rigorously tested was the order assignment mechanism in the Collector module to prevent race conditions. In the test scenario, two different Collector accounts attempted to click the “Pick Up Order” button for the same pickup request almost simultaneously. The system successfully handled concurrency using a database locking mechanism. The collector who initiated the request first received the assignment and the order status changed to “Assigned,” while the second collector received a notification stating that the order had already been taken. This result confirms that the FCFS mechanism prevents duplicate assignments and ensures fair order distribution, as shown in Figure 4.38.

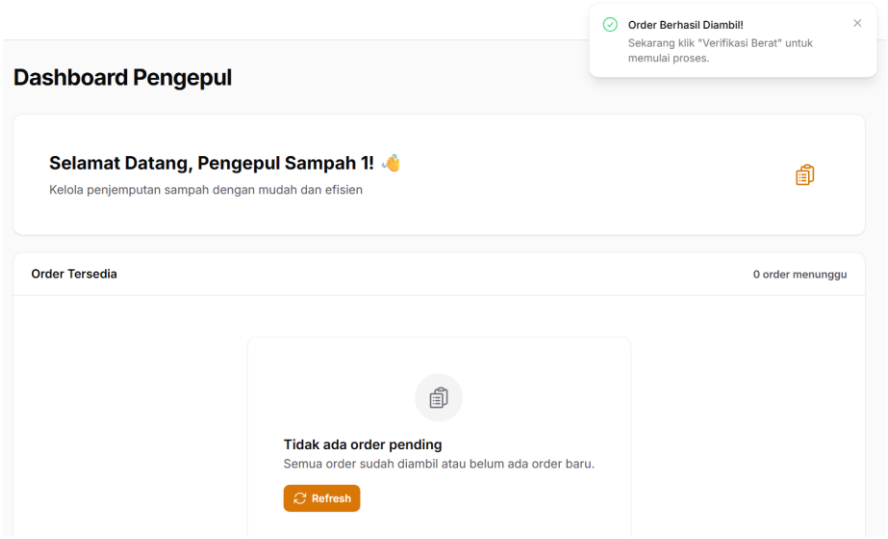


Figure 4. 38 Test Result of Collector FCFS Mechanism

2. Automated Calculation & Gateway Integration Testing

The second tested feature was the accuracy of automated transaction calculation and the reliability of Midtrans integration. Testing was performed by entering different waste weight values (e.g., 10.5 kg) to validate whether the system calculated totals correctly based on predefined prices. After the billing amount was generated, a payment simulation was executed using the Midtrans Mock Simulator. The system successfully received payment status updates via webhook and updated the customer's wallet balance according to the transaction value, including donation deductions when applicable. This confirms that the calculation process and payment integration operated correctly, as shown in Figure 4.39.

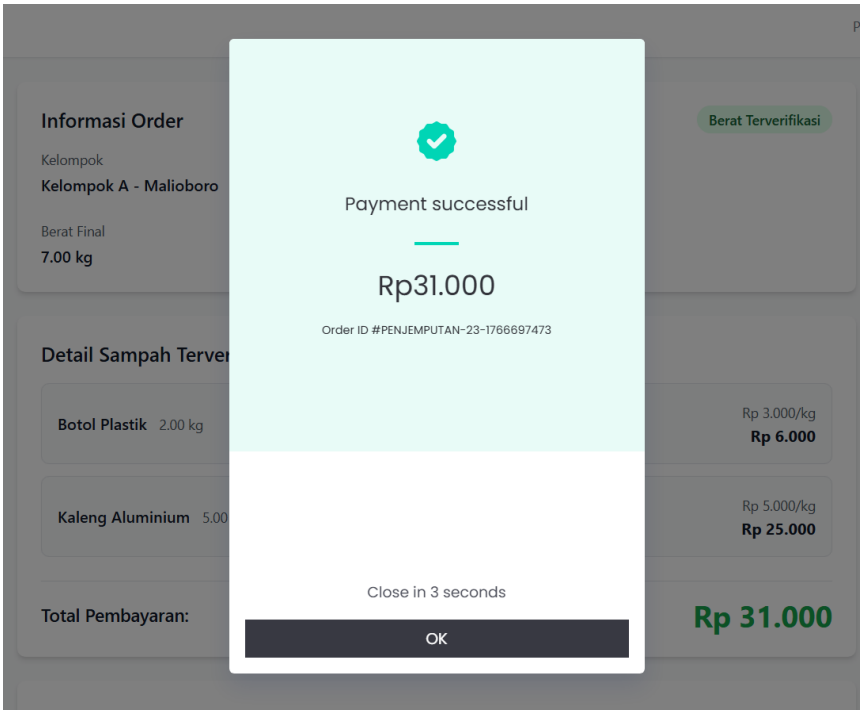


Figure 4. 39 Payment & Test Result

b. Beta Testing

Beta testing was conducted after alpha testing to evaluate the Waste Bank application directly by end users in a usage context that resembles real operational conditions. This stage also applied black-box testing and was carried out using predefined test scenarios. A total of 29 respondents participated in beta testing, representing end users from the waste bank community with backgrounds relevant to the system's operational context.

Overall, beta testing results indicate that the core application functions were usable and operated according to the designed scenarios. Users did not experience major functional obstacles during the testing process. However, several respondents provided feedback related to interface consistency and user comfort. From the total respondents, two respondents submitted constructive written suggestions, which were considered and implemented as system refinement input.

Tabel 4. 10 Summary of User Feedback in the Beta Testing Process

Responden	User Input	Application Section
3	It is recommended to add sound or vibration notifications when a new order comes in so that collectors don't miss opportunities.	Notification FCFS (collector)
11	The transaction verification button for Admin should be made consistent with the Customer verification button to avoid confusion.	Admin Page Verification

System Usability Scale (SUS)

At this stage, usability testing was carried out on the Waste Bank application using the System Usability Scale (SUS) method. This test aims to measure the level of application usability based on user experience after using the system in the beta testing stage. Users are asked to fill out the SUS questionnaire which consists of 10 statements, with a Likert rating scale of 1–5. The results of this assessment are shown in Table 4.11.

Tabel 4. 11 System Usability Scale (SUS) Questionnaire Assessment Results

Respondents	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
R1	4	3	4	3	5	2	5	2	5	1
R2	4	2	5	2	2	4	5	1	4	2
R3	5	4	5	5	5	2	4	2	5	5
R4	4	3	4	3	3	4	3	3	3	2
R5	5	5	5	5	5	5	5	5	5	5
R6	5	1	5	1	5	1	5	1	5	1
R7	4	3	3	3	3	3	3	3	3	3
R8	5	2	4	3	3	3	4	2	5	1
R9	5	4	5	5	4	4	4	5	5	5
R10	4	5	4	5	4	5	4	5	4	5
R11	4	5	4	5	4	5	4	5	4	5
R12	4	5	4	5	4	5	4	5	4	5
R13	5	1	4	2	5	2	5	2	5	1

Respondents	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
R14	4	5	4	5	4	5	4	5	4	5
R15	5	2	5	2	4	1	5	2	4	1
R16	4	5	4	5	4	5	4	5	4	5
R17	5	1	4	1	4	2	5	2	5	2
R18	5	2	4	1	4	2	4	1	5	2
R19	5	1	4	1	4	1	5	2	5	1
R20	2	4	1	4	2	5	2	4	1	4
R21	5	1	4	1	5	2	5	2	5	1
R22	2	4	1	4	2	4	1	4	2	4
R23	5	2	4	2	5	2	4	1	4	2
R24	5	1	4	2	5	2	5	1	5	1
R25	5	4	5	5	4	5	4	4	5	5
R26	4	2	5	2	4	2	5	2	4	2
R27	4	1	5	2	5	2	4	2	5	3
R28	5	2	4	2	5	3	4	2	5	2
R29	5	2	4	2	5	2	5	2	4	1

Next, the questionnaire data were processed according to the SUS scoring procedure described in Chapter III. Each respondent's responses were converted into SUS scores by applying the standard scoring rules for odd and even items, and the final score was calculated on a scale of 0 to 100. The processed results and individual SUS scores are presented in Table 4.12.

Tabel 4. 12 System Usability Scale (SUS) Score Processing Results

Respondents	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	Skor SUS
R1	3	2	3	2	4	3	4	3	4	4	32	80.0
R2	3	3	4	3	1	1	4	4	3	3	29	72.5
R3	4	1	4	0	4	3	3	3	4	0	26	65.0
R4	3	2	3	2	2	1	2	2	2	3	22	55.0
R5	4	0	4	0	4	0	4	0	4	0	20	50.0
R6	4	4	4	4	4	4	4	4	4	4	40	100.0
R7	3	2	2	2	2	2	2	2	2	2	21	52.5
R8	4	3	3	2	2	2	3	3	4	4	30	75.0
R9	4	1	4	0	3	1	3	0	4	0	20	50.0
R10	3	0	3	0	3	0	3	0	3	0	15	37.5
R11	3	0	3	0	3	0	3	0	3	0	15	37.5
R12	3	0	3	0	3	0	3	0	3	0	15	37.5
R13	4	4	3	3	4	3	4	3	4	4	36	90.0
R14	3	0	3	0	3	0	3	0	3	0	15	37.5
R15	4	3	4	3	3	4	4	3	3	4	35	87.5
R16	3	0	3	0	3	0	3	0	3	0	15	37.5
R17	4	4	3	4	3	3	4	3	4	3	35	87.5
R18	4	3	3	4	3	3	3	4	4	3	34	85.0
R19	4	4	3	4	3	4	4	3	4	4	37	92.5

Respondents	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	Skor SUS
R20	1	1	0	1	1	0	1	1	0	1	7	17.5
R21	4	4	3	4	4	3	4	3	4	4	37	92.5
R22	1	1	0	1	1	1	0	1	1	1	8	20.0
R23	4	3	3	3	4	3	3	4	3	3	33	82.5
R24	4	4	3	3	4	3	4	4	4	4	37	92.5
R25	4	1	4	0	3	0	3	1	4	0	20	50.0
R26	3	3	4	3	3	3	4	3	3	3	32	80.0
R27	3	4	4	3	4	3	3	3	4	2	33	82.5
R28	4	3	3	3	4	2	3	3	4	3	32	80.0
R29	4	3	3	3	4	3	4	3	3	4	34	85.0
Rata-Rata Skor SUS												65.95

Based on Table 4.12, the average SUS score obtained from all respondents is 65.95. To support the interpretation of the SUS result, the position of the average score is visualized in Figure 4.40.

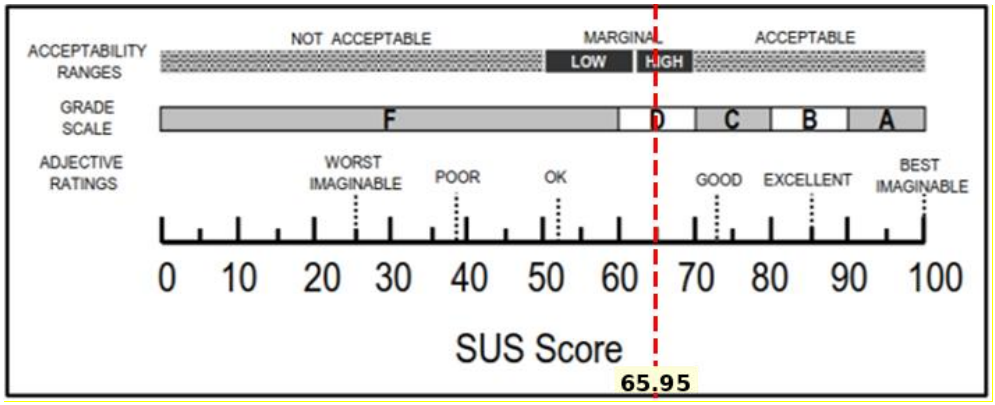


Figure 4. 40 Visualization of the Average SUS Score

Analysis of Testing Results

Based on the visualization of the SUS score results presented in Chapter III, the average SUS score of 65.95 falls into the “Marginal” acceptability category and is described as “OK.” This indicates that the Waste Bank application has a usable and acceptable level of usability according to users, although improvements are still needed to enhance user comfort and overall ease of use.

Discussion of Findings

This research aims to develop a web-based Waste Bank information system that improves operational efficiency while strengthening transaction transparency and accountability through controlled verification and secure digital payments. Based on the UAT results, the developed

system is able to execute the main operational workflow as designed, including pickup request handling, waste weighing input, transaction calculation, verification, and Midtrans payment processing. In addition, alpha testing confirmed that critical mechanisms such as the FCFS order assignment logic and the automated calculation with payment gateway integration functioned properly, supporting internal control by preventing duplicate assignments and ensuring transaction records remain traceable.

From the usability evaluation, the SUS assessment produced an average score of 65.95, which falls into the Marginal acceptability category and is described as “OK.” This indicates that the application is generally usable and acceptable for users to complete the intended tasks, although improvements are still needed to enhance comfort and ease of use. Therefore, the system can be considered feasible for implementation in a real operational environment, with further refinement recommended particularly in interface consistency, navigation clarity, and overall user experience.

4.1.5 Maintenance

The maintenance phase was conducted after the testing process, with the aim of correcting deficiencies and improving user experience in using the Waste Bank application. Improvements were made based on user feedback during the beta testing process, as previously described in Section 4.3 Testing. This feedback was carefully considered and adapted to the application's needs before final implementation.

The improvement included standardizing the design of the transaction verification buttons between the customer verification page and the admin verification page. Previously, there was a visual inconsistency between the "Approve" and "Reject" buttons on both pages, with the Admin page using a different color scheme and size than the Customer page. This discrepancy was considered confusing for users, especially for Admins who also act as Customers in the system.

Therefore, design adjustments were made, with the "Approve" button now being green and the "Reject" button being red on both pages, with uniform size and spacing. This adjustment improves the system's visual consistency and reduces the potential for operational errors.

Figure 4.41 and Figure 4.42 below are a comparison of the appearance of the admin verification page before and after improvements were made based on user input.

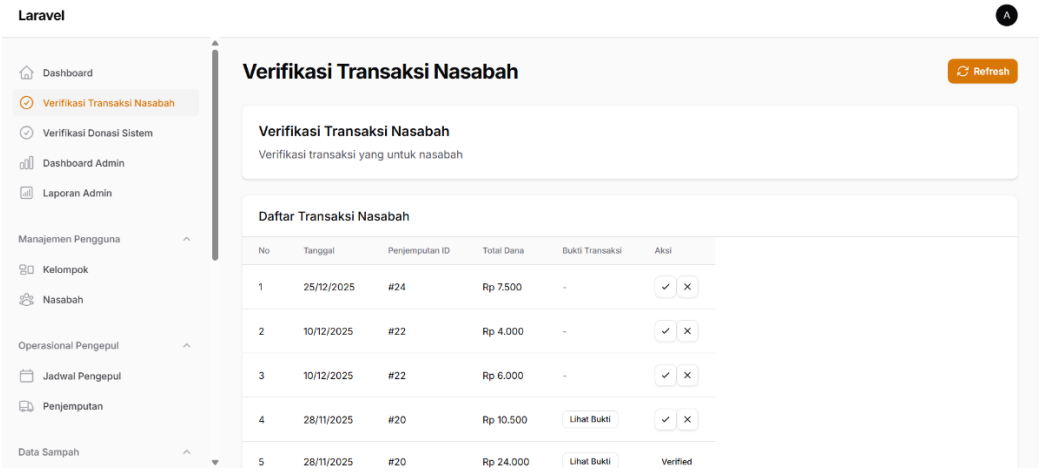


Figure 4. 41 Admin Verification Page Display Before Maintenance

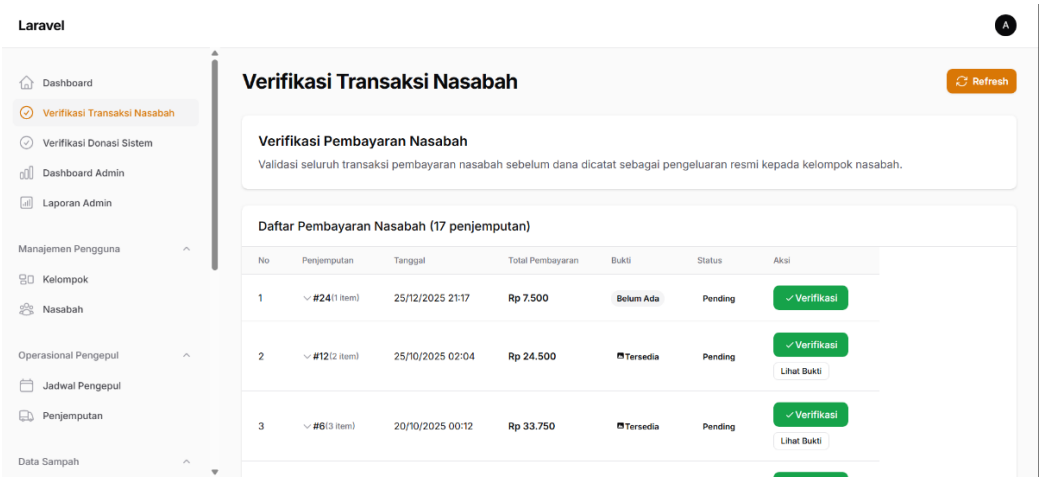


Figure 4. 42 Admin Verification Page Display After Maintenance

CHAPTER V
CONCLUSIONS AND SUGGESTIONS

5.1 Conclusion

Based on the results of system design, implementation, and testing that have been conducted, several conclusions can be drawn as follows:

1. Achievement of Research Objectives

   This research successfully developed a web-based Waste Bank Information System to support waste bank operational activities, including waste pickup requests, weighing input, verification, transaction recording, financial management, and reporting. The system provides a structured workflow that addresses the need for a digital solution to reduce manual recording issues and improve operational efficiency.

2. Contribution to Transaction Transparency and Control (Dual Verification)

   The implementation of the Dual Verification (Maker–Checker) mechanism strengthens transaction control by ensuring that critical transaction data must be validated by more than one party before being finalized. This mechanism supports transparency and accountability because each transaction status is clearly recorded, traceable, and can be reviewed by authorized users, reducing the risk of input errors and potential manipulation.

3. Support for Secure and Traceable Non-Cash Payments (Midtrans Integration)

   The integration of the Midtrans payment gateway enables secure and well-documented non-cash transactions through multiple payment methods such as QRIS, Virtual Account, and e-wallet. The Snap Popup interface provides convenient payment interaction, while webhook-based synchronization ensures that payment status updates are recorded automatically in the system's financial records, improving reliability and auditability.

4. System Feasibility Based on Testing Results (UAT and SUS)

   The UAT results show that the main system functions operate according to the defined scenarios and meet operational requirements. In terms of usability, the SUS evaluation produced an average score of 65.95, which falls into the Marginal acceptability category with an adjective rating of "OK." This indicates that the system is generally usable and feasible for implementation, although improvements are still required to enhance user comfort, interface consistency, and interaction efficiency.

5. Overall Research Outcome

Overall, this research demonstrates that the developed system can serve as a feasible digital solution for waste bank operations by improving transaction control, supporting transparent financial records, and enabling secure digital payments. However, usability enhancement and broader-scale evaluation are recommended to strengthen user adoption and long-term effectiveness.

5.2 Suggestions

Based on the results of the research and system development conducted, several recommendations can be considered for future development, including:

1. Interface Improvements Based on User Feedback Beta testing such as adding real-time notifications for collectors and verification button consistency to improve usability scores in the next iteration.
2. Development of Mobile Application Version Development of a mobile-based application version (Android/iOS) can be considered to increase user accessibility and comfort in accessing the system, especially for collectors who work in the field.
3. System testing on a larger user scale is recommended to obtain a more comprehensive evaluation of system performance and reliability in real operational conditions.
4. Integration with additional external services can be developed to expand system functionality and improve overall process efficiency.
5. Integration with additional services such as mapping systems (Google Maps API) for pickup routes and automatic notification systems (push notifications) can be developed to expand functionality and increase overall process efficiency.

REFERENCE

Assaid, L., Marpaung, W., & ... (2022). Membangun Sinergi Masyarakat pada Program Bank Sampah Sebagai Dampak Lingkungan Hidup oleh Bumdes di Dusun Karang Sari Desa Tanjung Putus …. *Jurnal …*, *6*.

Atin, S., Mutia, S., Widayanti, A., Yatawa, H. S., Rafdhi, A. A., & Afrianto, I. (2022). PERANCANGAN SISTEM INFORMASI BANK SAMPAH BERBASIS WEBSITE. *IJIS - Indonesian Journal On Information System*, *7*(1). https://doi.org/10.36549/ijis.v7i1.194

Brooke, J. (1996). SUS—a quick and dirty usability scale. 1996. In *Usability evaluation in industry* (Vol. 189, Issue 194, pp. 4–7). https://hell.meiert.org/core/pdf/sus.pdf

Djuwitaningrum, E. R., & Jati, I. B. W. (2025). Implementasi Payment Gateway Midtrans pada Website E-commerce Toko Buah dan Sayur. *Jurnal Ilmu Pengetahuan Dan Teknologi*, *9*(1), 19–24. https://doi.org/10.31543/jii.v9i1.390

Gustin, S. S. N., Junaeti, E., & Arianti, A. S. (2025). Picolearn: web-based learning media development on primm approach to enhance cognitive process in programming. *Research and Development Journal Of Education*, *11*(2), 1395–1406. https://doi.org/10.30998/rdje.v11i2.28782

Haniva, D. T., Ramadhan, J. A., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. *Journal of Information Engineering and Educational Technology*, *7*(1). https://doi.org/10.26740/jieet.v7n1.p36-42

Kadhem, A. J., Muslim, M. N., Hamad, A. A., Fadhil, B. H., & Abduljaleel, H. Y. (2025). Assessing university students' awareness and perceptions of electronic waste hazards in Baghdad, Iraq. *Cleaner Waste Systems*, *11*. https://doi.org/10.1016/j.clwas.2025.100309

Kurniawan, A. M., & Ariessanti, H. D. (2025). Perancangan Layanan Wedding Organizer Berbasis Web Terintegrasi Midtrans. *Jurnal Minfo Polgan*, *14*(2), 2109–2115. https://doi.org/10.33395/jmp.v14i2.15308

Kurniawan, E., Nofriadi, N., & Nata, A. (2022). PENERAPAN SYSTEM USABILITY SCALE (SUS) DALAM PENGUKURAN KEBERGUNAAN WEBSITE PROGRAM STUDI DI STMIK ROYAL. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, *5*(1). https://doi.org/10.54314/jssr.v5i1.817

Lingkungan, K., & Kehutanan, H. dan. (2022). Status lingkungan hidup indonesia. In *Laporan Status Lingkungan Hidup Indonesia 2022 olh Kementrian Lingkungan Hidup dan Kehutanan Republik Indonesia*. https://www.menlhk.go.id/cadmin/uploads/SLHI_2022_upload_final_77f9948571.pdf

Menora, T., Primasari, C. H., Wibisono, Y. P., Sidhi, T. A. P., Setyohadi, D. B., & Cininta, M. (2023). Implementasi Pengujian Alpha dan Beta Testing Pada Aplikasi Gamelan Virtual Reality. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, *3*(1), 48–60. https://doi.org/10.24002/konstelasi.v3i1.6625

Mustapa, A. D., Sulastri, A., & Ramadhan, M. A. P. (2025). *Pengembangan aplikasi persuratan berbasis website (studi kasus :kanwill djpb provinsi sulawesi tenggara)*. *9*(6), 1–89. https://doi.org/https://doi.org/10.36040/jati.v9i6.15675

Pressman, R. S., & Maxim, B. R. (2019). SOFTWARE ENGINEERING: A PRACTITIONERS APPROACH, EIGHTH EDITION. In *McGraw-Hill Education* (Vol. 11, Issue 1).

Putra, A. D. E., & Bhakti, H. D. (2024). Implementasi website bank sampah pada kelurahan pekelingan. *Jurnal Sains Dan Teknologi Informasi*, *2*(5), 12–22. https://doi.org/10.62951/switch.v2i5.177

Rahmah, F. H., & Theresiawati, T. (2022). Aplikasi bank sampah berkah melimpah berbasis website pada kelurahan nanggewer. *Informatik : Jurnal Ilmu Komputer*, *18*(2), 131. https://doi.org/10.52958/iftk.v18i2.4641

Santoso, G. B., Sinaga, T. M., & Zuhdi, A. (2021). MVC implementation in laravel framework for development web-based e-commerce applications. *Intelmatics*, *1*(1), 37–42. https://doi.org/10.25105/itm.v1i1.7867

Saranya, A., & Naresh, R. (2022). Dual Authentication for Payment Request Verification Over Cloud using Bilinear Dual Authentication Payments Transaction Protocol. *International Journal of Advanced Computer Science and Applications*, *13*(7), 288–297. https://doi.org/10.14569/IJACSA.2022.0130737

Sinlae, F., Irwanda, E., Maulana, Z., & Syahputra, V. E. (2024). *Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP*. *2*(2), 119–132.

Siska Narulita, Ahmad Nugroho, & M. Zakki Abdillah. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Bridge : Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, *2*(3), 244–256. https://doi.org/10.62951/bridge.v2i3.174

Sudarsono, E., & Vebriandi, M. Y. (2025). Implementasi Framework Laravel Filament Pada Sistem Crm Untuk Optimalisasi Data Pelanggan Dan Program Loyalitas Poin Di Toko Branding Telemarco. *Journal of Innovation And Future Technology (IFTECH)*, *7*(1), 23–34. https://doi.org/10.47080/iftech.v7i1.3839

Umar, N., Asrul, B. E. W., & Wabula, Y. (2025). Evaluating the Performance of an LBS-Based Waste Reporting Application for Digital Waste Management. *Journal of Applied Informatics and Computing*, *9*(4), 1730–1739. https://doi.org/10.30871/jaic.v9i4.9945

Widiyati, D. A., & Wijayati, F. L. (2025). The effect of earnings management and dividend policy on company performance: the moderating role of corporate governance quality. *International Conference on Education Innovation : Social Science*, 420:426. https://proceedings.ums.ac.id/iceiss/article/view/6347

Wijanarko, S. (2025). Implementasi filament form builder dalam sistem informasi berbasis laravel. *Jurnal Teknik Informatika STMIK Antar Bangsa*, *11*(1), 14–17. https://doi.org/10.51998/jti.v11i01.619

APPENDIX

TEST CASE BLACK BOX TESTING APP

Alpha Testing oleh Peneliti/Developer

A. Pengujian Form Registrasi (Daftar)

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Semua input form login kosong, klik tombol "Masuk"	Email: (kosong) Password: (kosong)	Sistem menolak dan menampilkan pesan "Please fill out this field" pada input yang kosong.	Sesuai dengan yang diharapkan	Berhasil
2	Input email salah format, klik tombol "Masuk"	Email: pengepul Password: 123456	Sistem menolak dan menampilkan alert "Email atau password salah".	Sesuai dengan yang diharapkan	Berhasil
3	Email valid dan Password salah	Email: pengepul@test.com Password: SalahPass	Sistem menerima login dan mengarahkan ke Dashboard Pengepul.	Sesuai dengan yang diharapkan	Berhasil
4	Semua input valid (Akun Pengepul)	Email: pengepul1@banksampah.com Password: password	Sistem menerima login dan mengarahkan ke Dashboard Nasabah.	Sesuai dengan yang diharapkan	Berhasil
5	Semua input valid (Akun Nasabah)	Email: kelompok.malioboro@banksampah.com	Sistem menerima login dan mengarahkan ke	Sesuai dengan yang	Berhasil

		Password: password	Dashboard Nasabah.	diharapkan	
6	Registrasi Pengepul (Input Valid)	Isi Nama, NIK, dan Upload Foto KTP. Klik "Daftar".	Data tersimpan dengan status "Pending" dan menunggu verifikasi Admin.	Sesuai dengan yang diharapkan	Berhasil

B. Pengujian Menu Nasabah (Request Penjemputan)

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Klik menu "Request Jemput"	Tidak ada input, klik menu di sidebar.	Sistem menampilkan form request penjemputan sampah.	Sesuai dengan yang diharapkan	Berhasil
2	Submit Request dengan input kosong	Biarkan berat kosong, klik "Kirim".	Sistem menampilkan "Field Berat Wajib Diisi".	Sesuai dengan yang diharapkan	Berhasil
3	Pilih Opsi "Donasi Semua"	Pilih radio button "Donasi Semua", isi berat, upload foto.	Request terkirim, dan sistem mencatat preferensi pembayaran sebagai donasi.	Sesuai dengan yang diharapkan	Berhasil
4	Cek Riwayat Request	Klik menu "Riwayat".	Data request yang baru dibuat muncul dengan status "Pending".	Sesuai dengan yang	Berhasil

				diharapkan	
5	Verifikasi Penerimaan Dana	Klik tombol "Terima" pada transaksi yang sudah dibayar pengepul.	Status transaksi berubah menjadi "Completed" dan saldo dompet bertambah.	Sesuai dengan yang diharapkan	Berhasil

C. Pengujian Menu Pengepul (Operasional FCFS)

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Daftar Open Order	Login sebagai Pengepul, lihat dashboard.	Muncul daftar request nasabah yang berstatus "Pending".	Sesuai dengan yang diharapkan	Berhasil
2	Ambil Order (FCFS)	Klik tombol "Ambil Order" pada salah satu kartu.	Status order berubah menjadi "Assigned" dan hilang dari list pengepul lain.	Sesuai dengan yang diharapkan	Berhasil
3	Verifikasi Berat Aktual	nput berat real (misal: 10 kg) di form verifikasi.	Sistem otomatis menghitung total harga (10kg x Harga Standar).	Sesuai dengan yang diharapkan	Berhasil
4	Pembayaran via Midtrans	Klik "Bayar Sekarang".	Muncul Pop-up (Snap) Midtrans dengan pilihan metode pembayaran.	Sesuai dengan yang diharapkan	Berhasil

D. Pengujian Menu Admin (Manajemen Data)

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Manajemen Master Harga	Ubah harga plastik dari Rp 2000 jadi Rp 2500.	Perubahan tersimpan, dan transaksi baru akan menggunakan harga Rp 2500.	Sesuai dengan yang diharapkan	Berhasil
2	Verifikasi Pengepul Baru	Buka menu Pengepul, klik "Approve" pada pendaftar baru.	Status akun pengepul berubah jadi "Active" dan bisa login.	Sesuai dengan yang diharapkan	Berhasil
3	Monitoring Laporan	Buka menu Laporan Keuangan.	Tabel menampilkan mutasi dana masuk dan keluar secara real-time.	Sesuai dengan yang diharapkan	Berhasil