

Volume: 5
Number: 1
Page: 1397 - 1414

Article History:

Received: 2026-06-07
Revised: 2026-03-08
Accepted: 2026-17-09

DESIGN AND DEVELOPMENT OF A WEBSITE-BASED ACCOUNTING INFORMATION SYSTEM USING THE LARAVEL FRAMEWORK FOR THE NASI KUNING ANGEL MSME

¹Theofani Imanuel Sondakh Masengi, ²Fikky Bawoel, ³Chievo Vicanso Jeferli Simboh

^{1,2,3}Accounting, Manado State Polytechnic, Indonesia

Corresponding author: Theofani Imanuel Sondakh Masengi

E-mail: imanueltheo26@gmail.com

Abstract:

This study aims to design and develop a web-based sales accounting information system using the Laravel framework to address the lack of structured financial record-keeping at the MSME "Nasi Kuning Angel." Previously, the absence of such a system hindered the owner from evaluating business performance and preparing the financial reports often required for external financing applications. The study employs the Waterfall development methodology, encompassing the stages of requirements analysis, system design, implementation, testing, and maintenance. The system design utilizes Use Case Diagrams and Activity Diagrams, which are technically implemented via the Model-View-Controller (MVC) architecture in Laravel. The results demonstrate the successful development of a system featuring a public landing page, transaction journal recording, a general ledger, and automated financial dashboards and reports. Testing conducted using Use Case Testing techniques confirms that all operational functions and authorization validations operate correctly in accordance with user requirement specifications. This web application modernizes the MSME's administrative management, facilitates transaction recording, and generates standard-compliant profit and loss statements and trial balances.

Keywords: Web-Based Accounting; Financial Reporting; Laravel

INTRODUCTION

Rudjito stated that Micro, Small, and Medium Enterprises (MSMEs) are a crucial component of economic activity, as they support income generation, job creation, and the operational continuity of local businesses (Nasution, 2023). The sustainability of MSMEs depends not only on the ability to sell products but also on the capacity to systematically manage operational and financial information (Wahyuningrum & Wibowo, 2025). In an increasingly digital business environment, the use of information systems can assist MSMEs in managing business data, improving access to information, and supporting more structured decision-making processes (Farina, 2022). Therefore, MSME digitalization efforts can be directed not only toward online promotion but also toward strengthening internal business management, particularly regarding sales and financial administration.

Financial management is a crucial aspect of MSME operations, as transaction records serve as the foundation for assessing business performance and compiling financial information (Wahyuningrum & Wibowo, 2025). Structured accounting records enable business owners to identify revenues, expenses, assets, liabilities, and the profit or loss generated within a specific period (Nento et al., 2026). Consequently, the availability of financial reports supports business evaluation, planning, and external financing needs, given that financial information is commonly used to assess a business's financial health.

Nasi Kuning Angel is an MSME that has been operating since 2001. Based on interviews and observations, the business has not previously had clear and structured financial recording that can show the amount of profit or loss generated in a given period. This condition makes it difficult for the owners to obtain a clear picture of the financial condition of the business and also creates difficulties when financial statements, particularly a clear income statement, are required to support bank loan applications. Although the business has been able to survive until the present, the absence of structured financial management makes it more difficult for the owners to evaluate business performance, plan the use of funds, and develop the business in a more measurable manner.

To address these conditions, this study develops an integrated web-based sales accounting information system that connects the recording of sales transactions with accounting information and financial presentation. The system is designed not only to record sales, but also to process transaction data into financial information, present financial statements, and provide a financial dashboard that summarizes important financial indicators. In addition, the system includes a public landing page containing business profile and menu information, with content that can be updated through the management side of the website. Previous studies have developed accounting information systems with varying scopes and outputs. Muthia et al. (2019) developed an accounting information system that focuses on sales activities, while Kustiawan et al. (2022) developed a system that produces trial balance and general ledger information. Wahyuni et al. (2021) focused on the development of a system for cash flow management, whereas Heripracoyo (2009) focused on purchasing activities. Meanwhile, Setyanto and Desanti (2025) developed an accounting information system that has reached profit or loss reporting; however, the system has not yet been integrated with the MSME's own website. Based on these studies, the research gap identified in this study lies in the integration of sales recording, accounting information processing, financial reporting, financial dashboard presentation, and an editable MSME landing page within one web-based platform. Accordingly, this study aims to design and develop a web-based sales accounting information system that can be applied to the operational and financial needs of Nasi Kuning Angel.



METHODS

This research is a design-and-build study conducted at Nasi Kuning Angel as the research object. Data were collected through interviews and observations to identify the business processes, financial recording conditions, user needs, and information requirements of the MSME. The system development method uses the Waterfall model, consisting of requirements analysis, system design, implementation, testing, and maintenance (Azis et al., 2020).

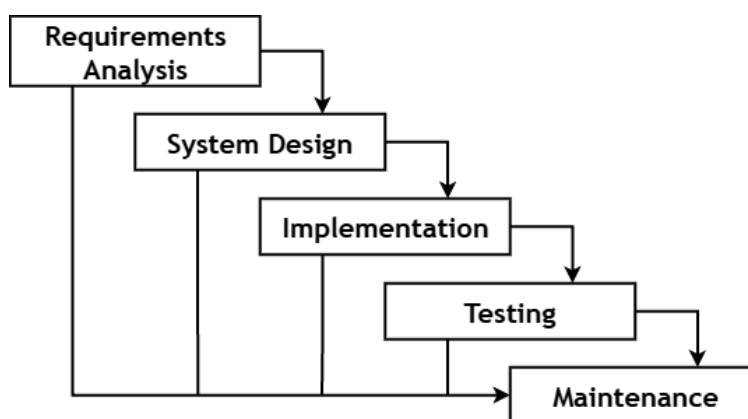


Figure 1. Waterfall Model (A. S & Shalahudin, 2016)

In the requirements analysis stage, the problems and functional requirements of the MSME are identified; in the system design stage, the system processes and data structures are translated into technical designs, use case diagrams, and activity diagrams (Narulita et al., 2024); in the implementation stage, the web application is developed using the Laravel framework (Melyani et al., 2023); the testing stage is planned to use Black-Box Testing to verify system functions based on input and output behavior (Muthia et al., 2019); and the maintenance stage covers corrective or adaptive improvements after system use.

The developed system covers the public landing page, business and menu content management, sales transaction recording, accounting information processing, financial statements, and financial dashboard. Since testing has not yet been conducted, this draft presents testing as a methodological stage and does not yet report test results.

RESULT AND DISCUSSION

The Requirements Analysis phase was conducted to identify the problems, user needs, and functional requirements of the system to be developed for the MSME "Nasi Kuning Angel." This analysis involved observing the existing transaction recording and financial management processes, as well as interviewing individuals involved in operational activities and financial record-keeping. The analysis revealed that the processes for recording transactions and preparing financial reports require greater structure to ensure financial information is presented more systematically. Consequently, the system being developed aims to assist users in managing the entire workflow, from recording transactions to generating financial reports in an integrated manner.

System requirements are then categorized into functional and non-functional requirements. Functional requirements represent the functions that must be available for the system to support financial recording and reporting processes, whereas non-functional requirements relate to system characteristics that facilitate ease of use and system management.

Table 1. System Functional Requirements

Requirement	Description
Business Profile Management	Users can view and manage business profile data used in the system.
Account Data Management	Users can add, modify, and delete account data used for recording transactions.
Transaction Recording	Users can record financial transactions by entering transaction details, accounts, and debit and credit values.
General Ledger Processing	The system can process recorded transactions and group them by account into the general ledger.
Trial Balance Processing	The system can process the general ledger and group accounts into asset and liability categories.
Profit and Loss Statement	The system can generate an income statement based on recorded revenue and expense data.

Based on these needs, the system is designed with a data processing flow that starts from transaction input by the user. Transaction data is then processed based on the selected account and recorded in a journal. Transactions that have met the recording requirements are then posted to the general ledger and used as the basis for preparing financial reports.

The System Design utilizes use case diagrams to illustrate the relationships between users and the functions available within the system. Activity diagrams are also employed to provide an overview of system boundaries and the primary activities that users can perform (Narulita et al., 2024).

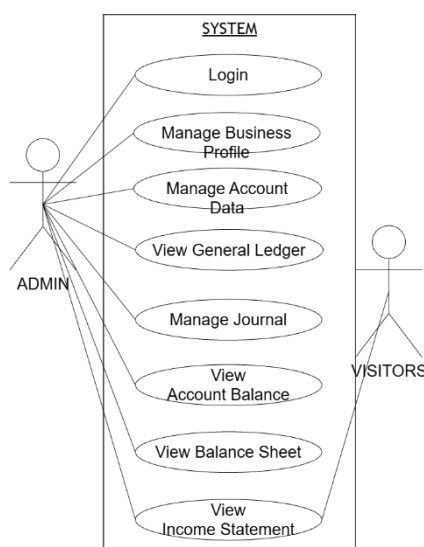


Figure 2. Use Case Diagram

Use Case: Login. The Login use case describes the process by which the administrator accesses the system by providing valid authentication credentials. After successful authentication, the administrator can access the system's available features according to their access rights.

Table 2. Use Case: Login

Component	Description
Use Case Name	Login
Purpose	To authenticate the administrator and provide access to the system.
Precondition	The administrator has an account registered in the system.
Postcondition	The administrator is successfully authenticated and can access the system.
Normal Flow	1. The administrator opens the login page. 2. The administrator enters their username/email and password. 3. The administrator selects the "Login" button. 4. The system verifies the provided credentials. 5. The system grants access to the administrator.
Alternative Flow	4a. Invalid credentials: If the provided credentials are not valid, the system does not grant access and displays an appropriate login error message.

Use Case: Manage Store Profile. The Manage Store Profile use case describes the process of managing the business profile information displayed in the system. The administrator can access the business profile and make changes to the available information.

Table 3. Use Case: Manage Store Profile

Component	Description
Use Case Name	Manage Store Profile
Purpose	To manage and update the store profile information.
Precondition	The administrator has successfully logged into the system.
Postcondition	The store profile information is stored with the latest changes made by the administrator.
Normal Flow	1. The administrator navigates to the Store Profile page. 2. The system displays the existing store profile information. 3. The administrator modifies the required store profile information. 4. The administrator selects "Save Changes." 5. The system saves the updated store profile information. 6. The system displays the updated store profile.

Use Case: Manage Account Data. The Manage Account Data use case describes the process of managing the account information used within the system. The administrator can access the account data and update the available information.

Table 4. Use Case: Manage Account Data

Component	Description
Use Case Name	Manage Account Data
Purpose	To manage and maintain the administrator's account information.
Precondition	The administrator has successfully logged into the system.



Postcondition	The account information is stored according to the changes made by the administrator.
Normal Flow	1. The administrator navigates to the Account Data page. 2. The system displays the existing account information. 3. The administrator modifies the required account information. 4. The administrator saves the changes. 5. The system stores the updated account information. 6. The system displays the updated account information.

Use Case: View General Ledger. The View General Ledger use case describes the process of viewing the general ledger generated from the financial transactions recorded in the system. The administrator can access the ledger to review the recorded financial information.

Table 5. Use Case: View General Ledger

Component	Description
Use Case Name	View General Ledger
Purpose	To allow the administrator to view the general ledger information recorded in the system.
Precondition	The administrator has successfully logged into the system.
Postcondition	The general ledger is displayed to the administrator.
Normal Flow	1. The administrator navigates to the General Ledger page. 2. The system retrieves the available ledger data. 3. The system displays the general ledger information. 4. The administrator reviews the displayed ledger.

Use Case: Manage Journal. The Manage Journal use case describes the process of managing general journal entries in the system. The administrator can enter journal transaction information and save it so that the data can be used in the system's financial records.

Table 6. Use Case: Manage Journal

Component	Description
Use Case Name	Manage Journal
Purpose	To manage and record general journal transactions in the system.
Precondition	The administrator has successfully logged into the system.
Postcondition	The journal data entered by the administrator is stored in the system.
Normal Flow	1. The administrator navigates to the General Journal page. 2. The administrator selects "Add Journal." 3. The administrator enters the journal data, including the date, proof number, description, debit/credit journal, and debit/credit balance. 4. The administrator selects "Save Journal." 5. The system saves the journal data. 6. The system displays the saved journal entry.

Use Case: View Account Balance. The View Account Balance use case describes the process of viewing the balance of accounts recorded in the system. The administrator can access the account balance information to review the financial position of each account.

Table 7. Use Case: View Account Balance

Component	Description
Use Case Name	View Account Balance
Purpose	To allow the administrator to view account balance information.
Precondition	The administrator has successfully logged into the system.
Postcondition	The account balance information is displayed by the system.
Normal Flow	1. The administrator navigates to the Account Balance page. 2. The system retrieves the available account balance data. 3. The system displays the account balance information. 4. The administrator reviews the displayed account balances.

Use Case: View Balance Sheet. The View Balance Sheet use case describes the process of accessing the balance sheet generated by the system. The balance sheet provides financial information that can be viewed by the administrator and visitor according to the access provided in the system.

Table 8. Use Case: View Balance Sheet

Component	Description
Use Case Name	View Balance Sheet
Purpose	To provide access to the balance sheet generated by the system.
Precondition	The balance sheet data is available in the system.
Postcondition	The balance sheet is displayed to the user.
Normal Flow	1. The user selects the Balance Sheet menu. 2. The system retrieves the required financial data. 3. The system processes the available data to display the balance sheet. 4. The system displays the balance sheet. 5. The user reviews the displayed balance sheet.

Use Case: View Income Statement. The View Income Statement use case describes the process of viewing the income statement generated by the system. The report presents information related to the business's income and expenses for the available financial data.

Table 9. Use Case: View Income Statement

Component	Description
Use Case Name	View Income Statement
Purpose	To provide access to the income statement generated by the system.
Precondition	The financial data required for the income statement is available in the system.
Postcondition	The income statement is displayed to the user.



- Normal Flow**
1. The user selects the Income Statement menu.
 2. The system retrieves the required financial data.
 3. The system processes the available data to generate the income statement.
 4. The system displays the income statement.
 5. The user reviews the displayed income statement.

Activity diagram; Edit store profile. The diagram describes the process of customizing and saving a store profile in the application.

1. The process begins when the user enters the shop profile.
2. The user navigates to the store profile page to manage the information and appearance of their store.
3. The user makes changes or customizes the store profile according to their needs, such as modifying the store information or appearance.
4. After finishing the customization, the user selects “Save Changes.”
5. The application receives the changes and saves the updated store profile data.
6. Once the profile data has been successfully saved, the user proceeds to the home page.
7. The process ends after the user reaches the home page.

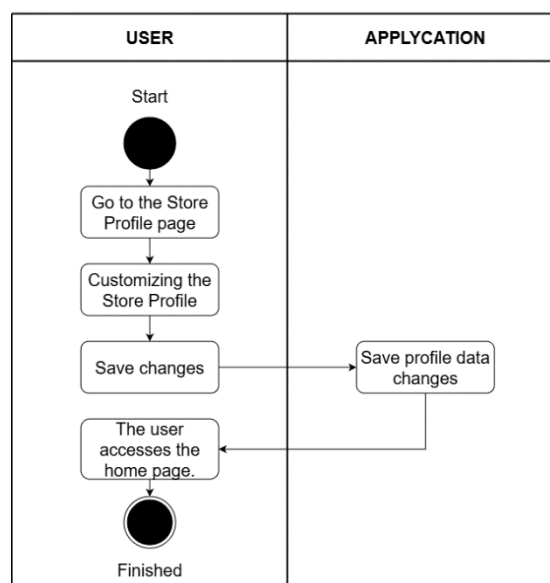


Figure 3. Activity diagram: Edit store profile

Activity diagram; Add General Journal. The diagram describes the process of adding and saving a general journal in the application.

1. The process begins when the user starts the general journal process.
2. The user navigates to the General Journal Page to manage and add journal entries.
3. The user selects “Add Journal” to create a new general journal entry.
4. The user enters the required journal data, including the date, proof number, description, journal (debit/credit), and balance (debit/credit).
5. After completing the journal data, the user selects “Save Journal.”
6. The application receives the journal data and saves the new journal entry.
7. Once the journal data has been successfully saved, the application displays the saved journal to the user.
8. The process ends after the saved journal is displayed.

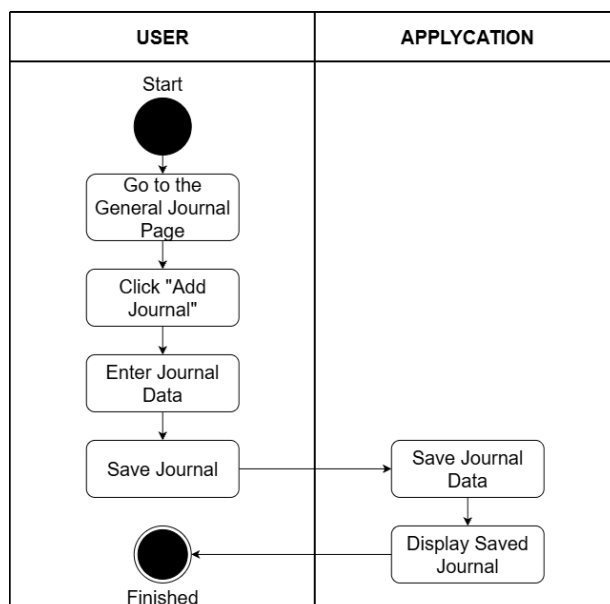


Figure 4. Activity diagram; Add General Journal.

Activity diagram; Print Profit and Loss Report. The diagram describes the process of generating and downloading a profit and loss report in the application.

1. The process begins when the user accesses the Report Menu.
2. The user navigates to the Report Menu to view and generate the available financial reports.
3. The user selects the report period according to the period for which the profit and loss report is needed.
4. After selecting the report period, the user selects “Download Selected Report” to download the report.
5. The application processes the selected period and generates the Profit and Loss Report based on the available financial data.
6. The application provides the generated report in PDF format, allowing the user to download and view the report.
7. The process ends after the profit and loss report has been provided in PDF format.

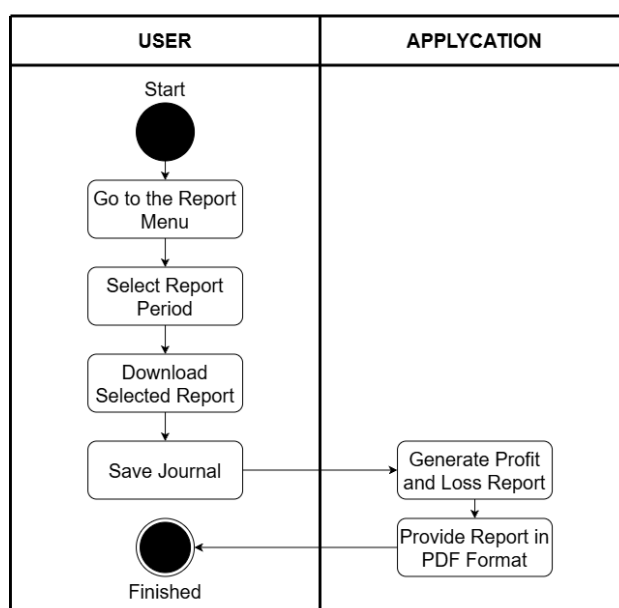


Figure 5. Activity diagram; Print Profit and Loss Report.

The Implementation stage represents the realization of the system design that has been developed in the previous stage. At this stage, the logical architecture, visual models, functional specifications, and system requirements are translated into program code to produce a functional web-based application that meets the needs of Nasi Kuning Angel MSME. The implementation is based on the functions and processes that have been previously defined through the Use Case Diagram and Activity Diagram, ensuring that the developed system corresponds to the proposed design.

In developing the application, the Laravel Framework is used as the main framework to support the implementation process. Laravel provides several features that facilitate application development, including shorter and more efficient code, easier layout management, comprehensive libraries, the Artisan command-line tool, and the implementation of the Model-View-Controller (MVC) architecture. The MVC structure separates the application into models, views, and

controllers, allowing the files and program components to be organized systematically. This structure also supports a more efficient and manageable development process, particularly when implementing and maintaining the various functions of the website.

The implementation process integrates the system designs and functional requirements that have been established during the design stage. Through this process, the planned features can be realized and operated within the application. The implemented features primarily support the financial recording and reporting activities of Nasi Kuning Angel MSME, including transaction journaling and the presentation of financial information. Each feature is developed according to the previously defined system flow so that users can carry out financial management activities through an integrated web-based system.

Login Page. The image below shows the login page, which serves to verify user access rights when logging into this web-based doctor consultation application. Users then enter their username and password.

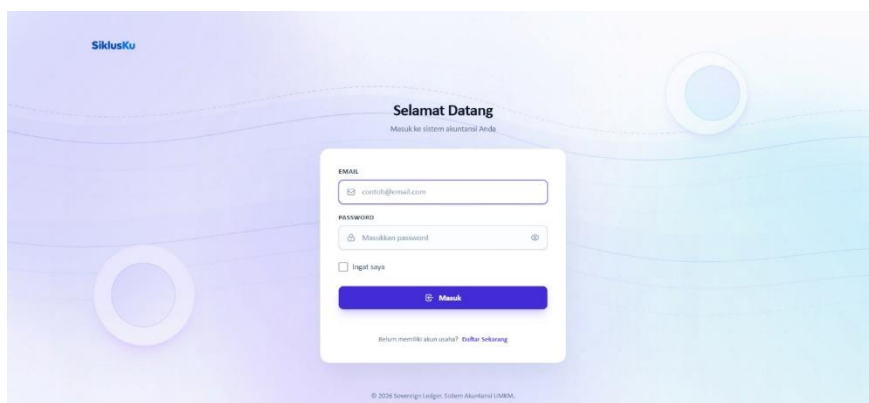


Figure 6. Login Page.

Overview section consists of the Dashboard and Profil Toko menus. The Dashboard displays a summary of the business's financial condition, including key metrics such as Total Assets, Total Liabilities, Total Equity, and Net Profit, alongside a revenue versus expense chart and fiscal health indicators. The Profil Toko menu manages operational information through three tabs: the Profil tab for basic details like the store name and address, the Menu Minuman tab for managing the beverage catalog and pricing, and the Laporan Tahunan tab for uploading annual PDF reports.

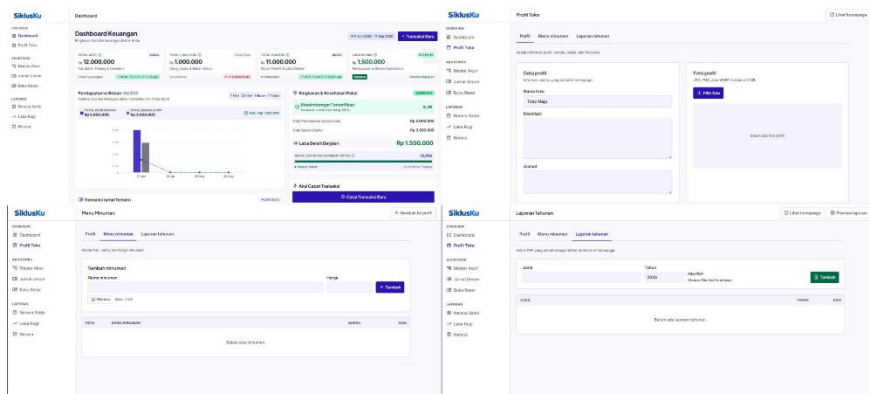


Figure 7. Overview Section (Dashboard, Store Profile)



Accounting section includes the Master Account, Journal, and General Ledger menus. Master Account is used to manage the transaction accounts categorized into Assets, Liabilities, Equity, Revenue, and Expenses, showing details like account codes and normal balance positions. The Journal serves to record new transactions and displays a chronological history of daily entries, including dates, voucher numbers, descriptions, and Debit or Credit amounts. The Ledger menu provides specific transaction details for each account, allowing users to monitor Debit and Credit movements as well as the running balance updated by each transaction.

The screenshot displays three panels of the SikkuKu Accounting module. The left panel shows the 'Master Akun' (Master Account) menu with a list of accounts categorized by type (Assets, Liabilities, Equity, Revenue, Expenses). The middle panel shows the 'Jurnal' (Journal) menu, displaying a table of transactions with columns for date, voucher number, description, and debit/credit amounts. The right panel shows the 'Ledger' menu, displaying a detailed view of a specific account's transactions, including a running balance column.

Figure 8. Accounting (Master Account, Journal, and General Ledger)

Reports section includes the Trial Balance and Profit and Loss menus. Balance Sheet presents a summary of the final balances of all active accounts in the current period, verifying that the total Debit and Credit amounts are perfectly balanced. The Profit and Loss menu displays the financial performance report by detailing Revenue and Expense accounts to calculate the Net Profit or loss, and it provides options to preview or download completed reports from previous months.

The screenshot displays two panels of the SikkuKu Reports module. The left panel shows the 'NERACA SALDO' (Trial Balance) report, which is a table summarizing the final balances of all active accounts, categorized by type (Assets, Liabilities, Equity, Revenue, Expenses). The right panel shows the 'Laba Rugi' (Profit and Loss) report, which is a table detailing the financial performance by listing revenue and expense accounts, and calculating the net profit or loss.

Figure 9. Reports (Trial Balance, Income Statement)

The Landing Page provides a simple and user-friendly platform for visitors to interact with the business. On this page, users can read a brief introduction about the culinary business's commitment to quality home-cooked meals and directly browse the available food menu, such as their signature Nasi Kuning, complete with pricing details. Additionally, the site allows users to easily view and download the business's annual reports in PDF format. Finally, visitors can locate the physical store using the integrated interactive map or reach out directly using the provided WhatsApp number and email address.

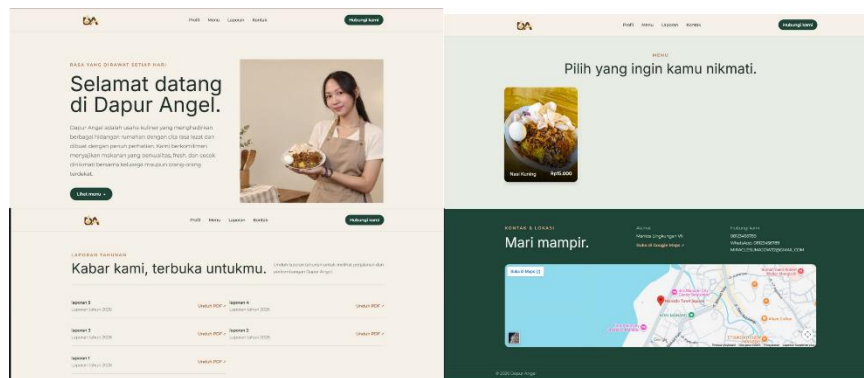


Figure 10. Landing Page

Testing. In this study, the system testing process will be conducted using the Use Case Testing technique. Use case testing is a software testing technique that focuses on interaction scenarios between users (actors) and the system. This technique is employed to ensure that the application can execute tasks or business processes from start to finish (end-to-end) in accordance with user requirements.

The testing was conducted in a single testing stage involving the development team and the business partner. Although Selenium was used as an automated testing tool, the test cases remained the same as the predefined Use Case Testing scenarios. The testing was performed by executing each scenario directly with the partner to evaluate the system's functionality in an operational context. The test activities included entering data, navigating between pages, clicking buttons, and verifying the system's responses. The actual results from each test scenario were then compared with the expected results to determine whether the system functioned as intended and met the requirements defined during the system design stage.

Login Testing. The Login testing was conducted to verify whether the authentication process works according to the designed use case. Three test scenarios were performed, consisting of successful login using valid credentials, failed login using an incorrect password, and failed login using an invalid email format. These scenarios were used to determine whether the system could properly authenticate valid users and reject invalid input.

Table 10. Login Testing.

Scenario ID	Test Scenario	Test Steps	Expected Result	Status
UC-LOG-01	Successful Login with Valid Credentials	1. Enter a valid email.2. Enter a valid password.3. Click the "Login" button.	The system validates the credentials, displays a success message, and redirects the user to the Dashboard page.	Passed
UC-LOG-02	Failed Login with Incorrect Password	1. Enter a valid email.2. Enter an incorrect password.3. Click the "Login" button.	The system rejects the login attempt, remains on the Login page, and displays an error message indicating that the credentials are incorrect.	Passed

UC-LOG-03	Failed Login with Invalid Email Format	1. Enter an invalid email format, such as admin@.2. Enter any password.3. Click the "Login" button.	The system displays an inline validation message indicating that the email format is invalid and prevents the form from being submitted.	Passed
-----------	--	---	--	--------

Based on the testing results, the Login use case successfully handles both valid and invalid authentication scenarios. The system allows access when valid credentials are provided, while incorrect passwords are rejected. The system also performs input validation when an invalid email format is entered. Therefore, the authentication process operates according to the expected behavior defined in the system design.

Store Profile Testing. The Manage Store Profile testing was conducted to verify the process of updating the business profile and logo. The testing focuses on whether the administrator can modify the profile information, upload a logo, process the image, and save the changes successfully. An additional scenario was performed to verify the system's response when an uploaded image exceeds the permitted size.

Table 11. Store Profile Testing.

Scenario ID	Test Scenario	Test Steps	Expected Result	Status
UC-STF-01	Update Business Profile and Store Logo	1. Open the Store Profile page.2. Change the store name and select a JPG/PNG logo file.3. Crop the image using the image editor.4. Click "Save Profile."	The profile is successfully updated. The public store homepage displays the updated store name and the cropped logo.	Passed
UC-STF-02	Add a New Beverage Product	1. Open the Beverage menu.2. Enter the product name and price, and upload the product image.3. Click "Save."	The new product is successfully added and displayed in the store catalog.	Passed
UC-STF-03	Image Upload Validation Exceeding the Size Limit	1. Open the product form.2. Upload an image larger than 2 MB.3. Allow the system to process the uploaded file.	The system displays a warning indicating that the image size is too large and rejects the image for processing/upload.	Passed

The testing results show that the store profile and related storefront functions operate according to the expected scenarios. The administrator can update the business profile and logo, while the system also displays the updated information on the public store page. The product addition process also successfully stores and displays a new beverage product. In addition, the system performs file-size validation by rejecting images that exceed the specified limit.

Account Data Testing. The Manage Account Data testing was performed to verify the process of adding and managing account data used in the financial recording system. The test scenarios cover successful account creation, rejection of duplicate account codes, and changing the status of an active account.

Table 12. Account Data Testing.

Scenario ID	Test Scenario	Test Steps	Expected Result	Status
UC-COA-01	Add New Account Successfully	1. Click "Add Account." 2. Enter the account code, such as 1101, account name, such as Cash, and account type. 3. Click "Save."	The new account is successfully saved. A success notification is displayed, and the account appears in the account list.	Passed
UC-COA-02	Add Account with Duplicate Code	1. Click "Add Account." 2. Enter an account code that already exists in the system. 3. Click "Save."	The system rejects the submission and displays a validation message indicating that the account code has already been used.	Passed
UC-COA-03	Deactivate an Active Account	1. Find an account with an Active status. 2. Click the "Deactivate" or Toggle button.	The account status changes to Inactive, and the account is no longer displayed in the journal account selection dropdown.	Passed

The testing of the Manage Account Data use case demonstrates that the system can manage account data according to the defined scenarios. A new account can be added successfully and displayed in the account list. The system also prevents the use of duplicate account codes through validation. Furthermore, an active account can be deactivated, after which the account is no longer available for selection when recording journal transactions.

Journal Testing. The Manage Journal testing was conducted to verify the process of recording financial transactions in the General Journal. The testing focuses on the fundamental accounting requirement that the total debit must be equal to the total credit. In addition, a deletion scenario was performed to verify whether a selected journal transaction can be removed and whether the related financial records are updated accordingly.

Table 13. Journal Testing.

Scenario ID	Test Scenario	Test Steps	Expected Result	Status
-------------	---------------	------------	-----------------	--------



This open access article is distributed under a
Creative Commons Attribution (CC-BY-NC) 4.0 licence

UC-JRN-01	Add Balanced Journal Successfully	1. Click "Add Journal."2. Enter the date and description.3. Add Account 1: Cash, Debit 100,000.4. Add Account 2: Revenue, Credit 100,000.5. Click "Save."	The system successfully saves the journal because the total debit equals the total credit. A success notification is displayed and the data appears in the General Journal table.	Passed
UC-JRN-02	Add Unbalanced Journal	1. Click "Add Journal."2. Enter the date and description.3. Add Account 1: Cash, Debit 100,000.4. Add Account 2: Revenue, Credit 50,000.5. Click "Save."	The system rejects the journal because the total debit does not equal the total credit. An error message indicating that the journal is not balanced is displayed.	Passed
UC-JRN-03	Delete Journal Transaction	1. On the General Journal table, click "Delete" for a selected transaction.2. Confirm the deletion.	The selected transaction is removed from the General Journal table, and the related General Ledger and balance calculations are automatically updated.	Passed

The results of the Journal testing indicate that the system can properly handle the recording of financial transactions. A balanced journal, in which the total debit and credit are equal, can be successfully saved. Conversely, the system rejects an unbalanced journal and displays an appropriate validation message. The deletion scenario also demonstrates that removing a journal transaction updates the related financial records. These results indicate that the journal management process operates according to the defined accounting rules and system requirements.

Report Testing. The Report testing was conducted to verify the process of displaying financial reports and exporting them into PDF format. The testing consists of two scenarios: viewing the income statement based on a selected period and downloading the generated financial report as a PDF file.

Table 14. Report Testing.

Scenario ID	Test Scenario	Test Steps	Expected Result	Status
UC-RPT-01	View Income Statement	1. Open the Income Statement page.2. Select the desired date range, if	The system displays the calculated total revenue and expenses for the selected period based on the available financial data.	Passed



This open access article is distributed under a Creative Commons Attribution (CC-BY-NC) 4.0 licence

UC-RPT-02	Export Financial Report to PDF	applicable.3. Click "Filter/View." 1. Open the report preview page.2. Click "Print PDF" or "Download PDF."	The system generates and downloads the financial report in PDF format with the appropriate report structure and table layout.	Passed
-----------	--------------------------------	--	---	--------

The Report testing demonstrates that the system can present financial information based on the available transaction data. When a reporting period is selected, the system displays the calculated revenue and expense information for that period. The system also provides an export function that generates the financial report in PDF format, allowing the resulting report to be stored or used as a formal financial document.

CONCLUSION

Based on the design and testing phases conducted, it can be concluded that a web-based sales accounting information system using the Laravel framework has been successfully developed for the MSME "Nasi Kuning Angel." The system effectively addresses financial management issues by modernizing workflows—ranging from daily transaction input and automated general ledger posting to the presentation of metrics on a financial dashboard. Implementing this system enables the MSME to generate automated income statements and balance sheets to evaluate its financial health. Furthermore, system testing validated that all core functionalities operate according to the design specifications. As a continuation of this research, future studies are encouraged to integrate modules for expense management and raw material inventory (Cost of Goods Sold/COGS), add features for calculating Final Income Tax (PPh Final) obligations, or develop the system into a mobile application to enable efficient monitoring of financial performance.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to Politeknik Negeri Manado, particularly the Department of Accounting, for the support provided throughout the implementation of this research. This research was supported by PNPB Politeknik Negeri Manado under Grant Number 429/PL12.11/PL/2026. The authors also extend their appreciation to the owner and management of Nasi Kuning Angel for their cooperation, willingness to provide information, and support throughout the data collection and system development process. Finally, the authors would like to thank all parties who contributed to the completion of this research.

REFERENCES

- A. S, R., & Shalahudin, M. (2016). *REKAYASA PERANGKAT LUNAK TERSRUKTUR dan BERORIENTASI OBJEK* (pp. 1-278). Informatika Bandung.
- Azis, M. S., Hakim, L., & Walim. (2020). Perancangan Aplikasi Berbasis Desktop Dengan Microsoft Visual Basic (Studi Kasus: Aplikasi Absensi Anak Magang 1.0). *JURNAL RESPONSIF*, 2(1), 44-52.
- Farina, K. (2022). PENGARUH PEMANFAATAN SISTEM INFORMASI AKUNTANSI DAN PENGGUNAAN TEKNOLOGI INFORMASI TERHADAP KINERJA UMKM. *Jurnal Ekonomi*



This open access article is distributed under a Creative Commons Attribution (CC-BY-NC) 4.0 licence

& *Ekonomi Syariah*, 6(1), 704–713.

Heripracoyo, S. (2009). ANALISIS DAN PERANCANGAN SISTEM INFORMASI AKUNTANSI PEMBELIAN DAN PERSEDIAAN PADA PT. OLISER INDONESIA. *Seminar Nasional Aplikasi Teknologi Informasi*, 93–100.

Kustiawan, D., Cholifah, W. N., Destriana, R., & Heriyani, N. (2022). Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode Extreme Programming. *Jurnal Teknologi Dan Informasi*, 12(1), 78–92. <https://doi.org/10.34010/jati.v12i1>

Melyani, R. I., Rosita, & Aji, S. (2023). Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Framework Laravel dengan Metode Agile Software Development. *JASIKA (Jurnal Sistem Informasi Akuntansi)*, 03(01), 31–36.

Muthia, N., Amalia, H., Puspita, A., & Lestari, A. F. (2019). RANCANG BANGUN SISTEM INFORMASI AKUNTANSI PENJUALAN DENGAN MODEL WATERFALL BERBASIS JAVA DESKTOP. *JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER*, 5(1), 15–22.

Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, 2(1), 38–50. <https://doi.org/https://doi.org/10.62951/bridge.v2i3.174>

Nasution, E. D. P. (2023). *KETERKAITAN UMKM DALAM MENGURANGI KEMISKINAN*.

Nento, M. H. A. H., Patilima, D., Laiya, N. F. P., Mamonto, D., Mohehu, F., & Ahmad, R. (2026). IMPLEMENTASI PENCATATAN AKUNTANSI DALAM PENGELOLAN KEUANGAN USAHA MIKRO DI KOTA GORONTALO. *JURNAL ILMIAH MANAJEMEN DAN BISNIS*, 8(3), 1455–1462.

Setyanto, E., & Desanti, A. (2025). RANCANG BANGUN SISTEM INFORMASI AKUNTANSI (SIA) PENJUALAN BERBASIS VBA DI KLINIK INGGIT GARNASIH. *JURNAL INTEGRASI AKUNTANSI DAN KEUANGAN DIGITAL*, 1(1), 1–15.

Wahyuni, A. T., Saefudin, D. F., & Hakim, L. (2021). Rancang Bangun Sistem Informasi Akuntansi Berbasis Desktop Penerimaan dan Pengeluaran Kas Sekolah. *Profitabilitas*, 1(2), 140–148.

Wahyuningrum, T., & Wibowo, P. A. (2025). Evaluasi faktor penentu kinerja UMKM: Peran kompetensi SDM, literasi keuangan, modal keuangan, dan modal sosial. *Journal of Accounting and Digital Finance*, 5(1), 45–59. <https://doi.org/https://doi.org/10.53088/jadfi.v5i1.1624>