
Design And Construction Of A Web-Based Point Of Sale Application For The Bakso Cirawang Rayu Business Using The Extreme Programming Method

Katresna Asterina Latifa¹⁾, Yudhi Raymond Ramadhan²⁾, Muhammad Rafi Muttaqin³⁾
^{1,2,3)}Wastukencana College of Technology

*Corresponding Author
E-mail: katresna17@gmail.com

Abstract

Manual transaction management in culinary MSMEs still causes various obstacles, such as recording errors, transaction calculations, and sales data management. This study aims to design and build a web-based Point of Sale (POS) application at Bakso Cirawang Rayu using the Extreme Programming method. This study uses the Extreme Programming (XP) method in application development. The scope of the study focused on the process of recording orders, sales transactions, transaction history, sales reports, and estimating soup availability. The research instruments were in the form of interview guidelines, observations, and literature studies. Data were analyzed descriptively to identify user needs as a basis for system design. The results showed that the application was successfully developed with login features, dashboards, menu management, cashier management, sales transactions, reports, and soup availability estimation. Test results showed that all features functioned according to their functions, thus supporting the transaction process and sales data management. In conclusion, the web-based POS application developed using Extreme Programming was able to improve operational efficiency, reduce human error, and assist decision making at Bakso Cirawang Rayu.

Keywords: *Extreme Programming, Point Of Sale, Test-Driven Development, MSMEs, Web-Based Application*

INTRODUCTION

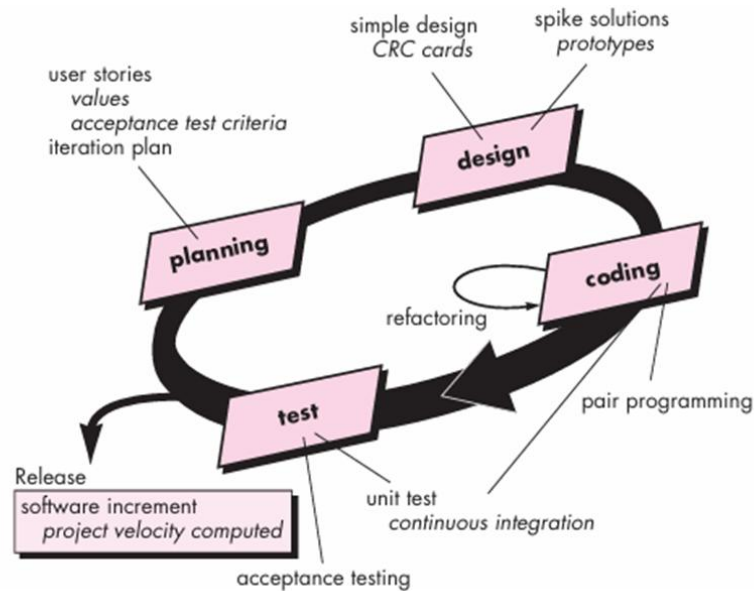
Advances in information technology are driving digital transformation in various sectors, including MSMEs. In the culinary sector, web-based Point of Sale (POS) applications can improve operational efficiency, accelerate service delivery, and reduce recording and transaction errors, although implementation still faces various challenges (Andy & Widiono, 2024; Muslim & Hendri, 2022).

Bakso Cirawang Rayu still processes orders and transactions manually using paper. With around 100 to 150 customers per day, this system results in inefficient service, increases the risk of calculation errors, and complicates sales recording and recapitulation (Andy & Widiono, 2024).

Observations and interviews show that paper use often results in damaged or lost orders, increasing the risk of errors in recording and processing transactions. Furthermore, manual recording results in sales data not being systematically stored, complicating reporting and business evaluation (Muslim & Hendri, 2022).

Another issue is managing soup availability, which is still based on estimates. As a result, sales of soup-based dishes cannot be continued when the soup runs out, even though other ingredients are still available. This situation highlights the need for a system capable of estimating soup availability based on sales data.

Based on these problems, this study aims to design a web-based Point of Sale application using the Extreme Programming (XP) method. This method was chosen because it is adaptive to changing user needs and supports rapid development (Pangestu & Astutik, 2024; Handoko Agustin et al., 2021; Sonhaji Akbar et al., 2025). The research's novelty lies in the addition of a soup availability estimation feature that complements transaction functions, menu management, and sales reporting, thereby improving operational efficiency and reducing human error.



Picture 1. Extreme Programming

RESEARCH METHODS

This study used the Extreme Programming (XP) method to design and build a web-based Point of Sale (POS) application for Bakso Cirawang Rayu. This method was chosen because it supports an iterative and flexible software development process tailored to user needs. The development stages implemented include planning, design, coding, testing, and release (Fachrurozi et al., 2025).

The research instruments included interview guidelines, observations, and literature review. Data were obtained through interviews with business owners, observations of operational processes, and a literature review of Point of Sale applications and software development. The collected data was analyzed descriptively to identify user needs and develop a system design.

The research focus includes the process of recording orders, sales transactions, storing transaction history, compiling sales reports, and estimating the availability of soup which is the scope of application development.

The research procedure included data collection, planning, design, coding, testing, and release according to the Extreme Programming stages. The system was designed using the Unified Modeling Language (UML), and the application was developed using PHP, the Laravel framework, and MySQL. It was then tested using the Test-Driven Development (TDD) approach before being implemented at Bakso Cirawang Rayu.

Research Framework

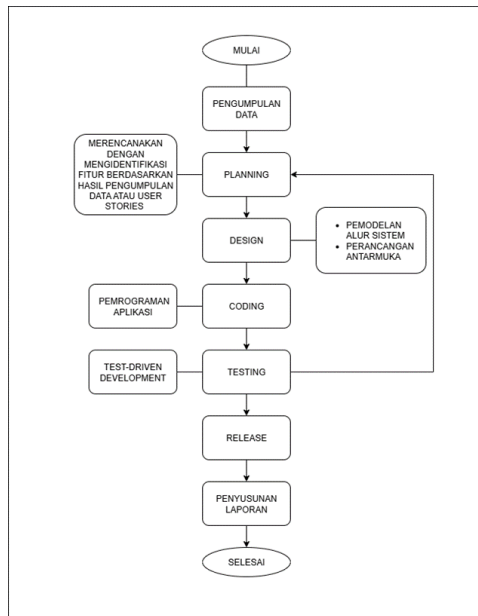


Figure 2. Research Framework

RESULTS AND DISCUSSION

Field Observation

Researchers conducted direct observations of the manual sales transaction process to identify operational barriers related to financial recording and service speed at Bakso Cirawang Rayu. The observations revealed the need for an integrated digital cashier system to improve operational efficiency.

Interview

Researchers conducted direct interviews with the owner of Bakso Cirawang Rayu to gather in-depth information regarding obstacles in the manual sales transaction process, frequently incorrect calculations, the ordering process, and its effects on the buying and selling process.

Literature review

The researchers reviewed previous research on the digitalization of operational management and Point of Sale in MSMEs as a theoretical and methodological basis. Unlike previous studies, this study focuses on integrating the ordering process and reducing transaction errors (human error), tailored to the operational characteristics of Bakso Cirawang Rayu, as the basis for developing application features.

Use Case Diagram

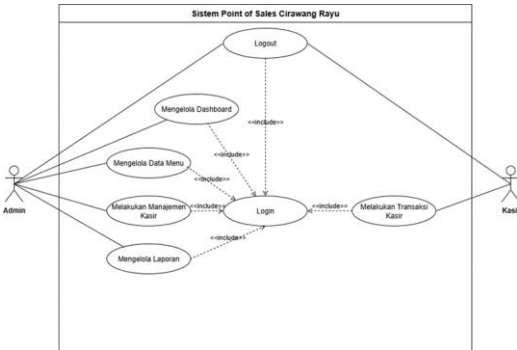


Figure 3. Use Case Diagram

In the Use Case Diagram for the Bakso Cirawang Rayu POS application, there are two actors: the Admin and the Cashier. Both actors can log in and log out to access the system. The Admin has access rights to manage the Dashboard, Menu Data, Cashier Management, and Reports. Meanwhile, the Cashier is responsible for processing sales transactions, from selecting menu items, processing payments, to printing receipts.

Use Case Scenario

The use case scenario for the Bakso Cirawang Rayu POS application involves two actors: an Admin and a Cashier. Both actors can log in to access the system and log out to end the session.

Admins have access to manage the Dashboard, Menu Data, Cashier Management, and Reports. The Dashboard displays a summary of sales statistics for a specific period. Menu Data is used to add, change, and manage menu availability status. Cashier Management manages cashier accounts and monitors user login history. Meanwhile, the Reports menu is used to view transaction data by period and export reports to Excel format.

The cashier is responsible for processing sales transactions, from selecting menu items, managing the shopping cart, making payments, and printing receipts. The system also implements validation at every stage, such as checking credentials during login, validating data when managing menus and cashier accounts, and validating transactions to ensure the cart is not empty and the payment amount is sufficient before the transaction is saved.

Class Diagram

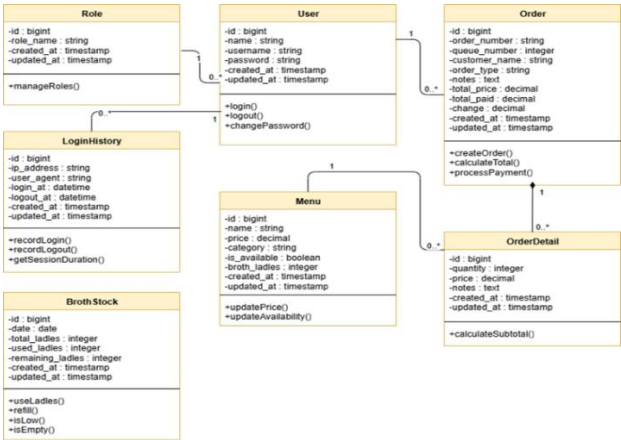


Figure 4. Class Diagram

Interface Design Planning

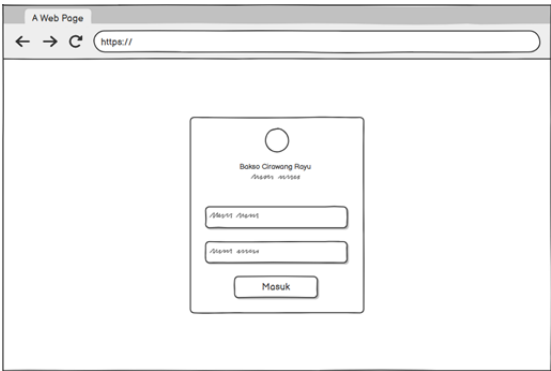


Figure 5. Login Interface Design

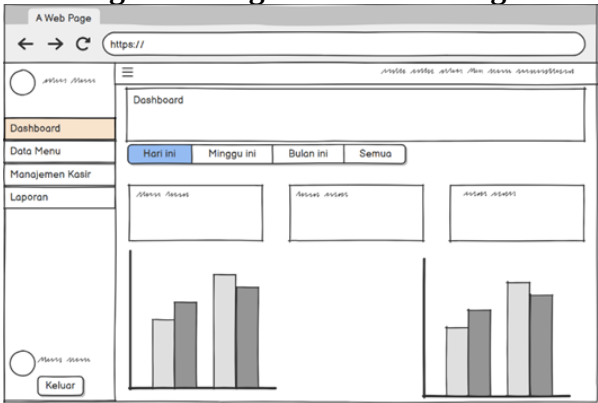


Figure 6. Dashboard Interface Design

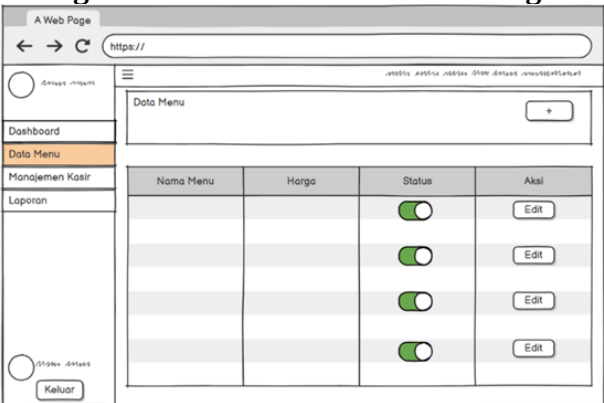


Figure 7. Menu Data Interface Design

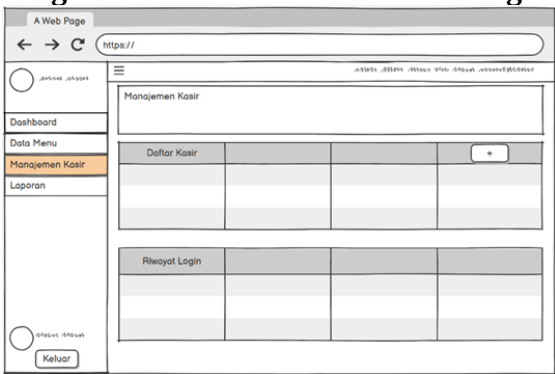


Figure 8. Cashier Management Interface Design

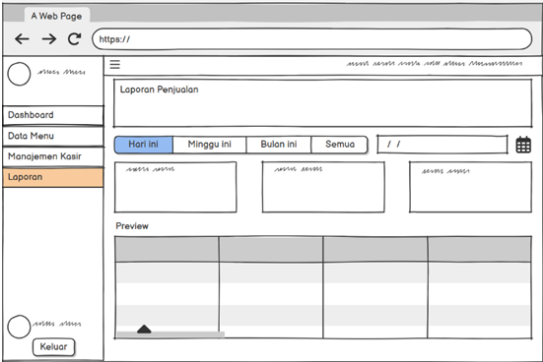


Figure 9. Report Interface Design

Coding



Figure 10. Login Interface

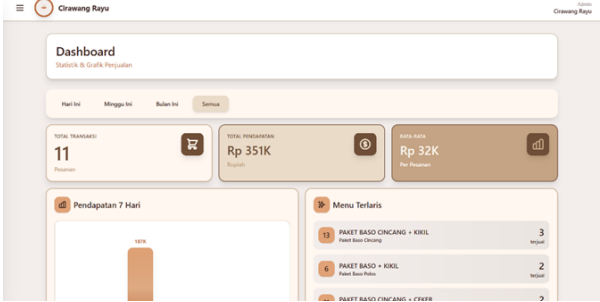


Figure 11. Dashboard Interface

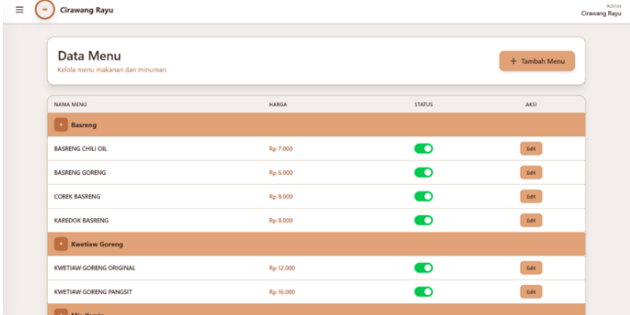


Figure 12. Data Menu Interface

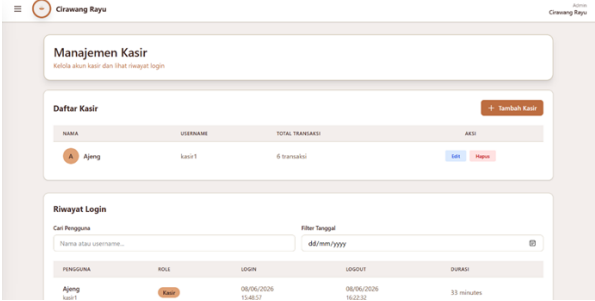


Figure 13. Cashier Management Interface

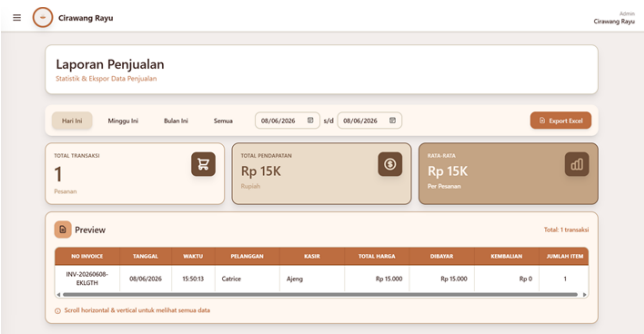


Figure 14. Report Interface

Testing

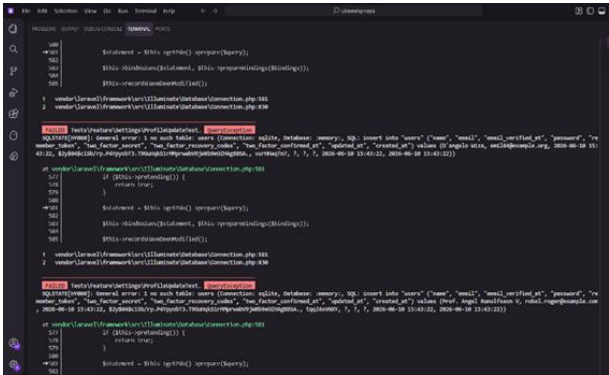


Figure 15. Red Test-Driven Development Cycle

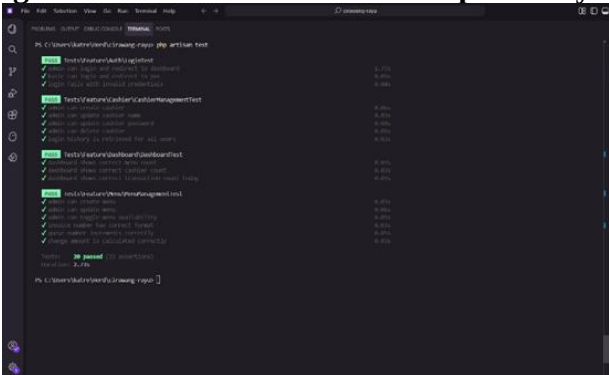


Figure 16. Green Test-Driven Development Cycle

Table 1 Test-Driven Development Testing

No	Test Case	Red Cycle (Failed)	Green Cycle (Success)	Refactor Cycle
1	Admin and cashier can login according to their role	Route dashboard and redirect role do not match	Success after implementing LoginController, authentication route, and role-based LoginResponse	Added eager loading load('role'), logging, role validation, and Indonesian error messages
2	Login failed with incorrect credentials	No error message is displayed	Success after adding validation in FortifyServiceProvider	Error messages using Indonesian
3	Admin can manage cashier data	Email column not found, update method does not	Success after adjusting username schema, creating	Added conditional password update,

		exist, password is not encrypted, and foreign key constraint violation	updateCashier, Hash::make(), and onDelete('cascade')	role validation before delete, and relationship loading optimization.
4	User login history can be displayed	Cashier_name column is not available	Success after migration drop cashier name	Added eager loading user.role and date filter
5	Dashboard displays statistics correctly	Count menu, cashier, and transactions do not match	Success after implementing where('is_available', true), user role filter, and whereDate()	Added computed property, withCount('orders'), and multi-period filter
6	Admin can manage menu data	Validation failed, menu not found, entire menu displayed on POS, and toggle status not saved	Success after implementation of Livewire validation, find(), is_available filter, and toggleAvailability()	Added default is_available, category sorting, and feedback message
7	Cashier can create new transactions	Orders table, subtotal column, invoice format, queue number, and return calculation have errors.	Successful after creating migration orders, fixing subtotal schema, invoice format, queue increment, and fixing payment formula	Adding foreign key constraints, orderDetails() relation, resetting daily queue numbers, and payment validation

Release

In the Release phase, the web-based Point of Sale (POS) application that passed testing was handed over to the owner of Bakso Cirawang Rayu for implementation. The application ran locally using Laravel Herd and MySQL, starting with database configuration, initial data entry, and admin and cashier account creation. Once all functions were confirmed to be working properly, the system was used live to process sales transactions.

CONCLUSIONS

This research successfully designed and built a web-based Point of Sale (POS) application for Bakso Cirawang Rayu using the Extreme Programming (XP) method. The developed application provides login, dashboard, menu management, cashier management, sales transactions, reports, and soup availability estimation features to support business operations. The implementation and testing results show that all features function according to their intended function, thus assisting the process of recording orders, calculating transactions, storing sales data, and presenting reports more effectively. Practically, this application has implications for increasing operational efficiency, reducing human error, and facilitating decision-making for business owners.

This research still has limitations because the application was only implemented in a local, offline environment and does not yet support the integration of digital payments, stock management, or accounting systems. Furthermore, the scope of the research was only applied to one MSME, so the results cannot be generalized to other types of businesses. Therefore, further research is recommended to develop an online-based application, add digital payment integration features, and stock management, and expand implementation to various types of MSMEs so that the system's benefits and effectiveness can be evaluated more broadly.

REFERENCES

- Andy, FAM, & Widiono, S. (2024). Technological innovation in sales management: Web-based Point of Sales application for MSMEs. *Infomatek Journal*, 26(2), 161–174. <https://doi.org/10.23969/infomatek.v26i2.19007>
- Handoko Agustin, Y., Latifah, A., & Fikri Nugraha, A. (2021). Design of a web-based cashier application information system at Restorasi Kopi Cafe. *Jurnal Algoritma*, 302–312. <http://jurnal.sttgarut.ac.id/>
- Muslim, MI, & Hendri. (2022). Web-based Point of Sales (POS) information system using the Laravel 8 framework at Draquatic Store. *Journal of Accounting Information System*, 2(2), 6–11. <http://jurnal.bsi.ac.id/index.php/jais>
- Pangestu, SD, & Astutik, IRI (2024). Design of a website-based grocery store cashier application using the Waterfall method. *JIPI (Scientific Journal of Informatics Research and Learning)*, 9(1), 125–135. <https://doi.org/10.29100/jipi.v9i1.4311>
- Sonhaji Akbar, M., Agustin Yasa, S., Rusprameswari, V., Adisti Rizki Insyirah, J., Atheyla, R., & Fadliana, A. (2025). Design and implementation of a web-based cashier website for a traditional toy store using the ADDIE and Black Box Testing methods. *Journal of Computers, Information and Technology*, 5(1), 1–11. <https://doi.org/10.53697/jkomitek.v5i1.25>