Jurnal E-Komtek
Vol. 9, No. 2 (2025) pp. 392-407
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Implementation of the RAD Method in a Point of Sales Information System with Payment Gateway Integration at Kedai Ngopi Ngopi

Hafez Nizamuddin Salim[1], Ridho Muktiadi[2], Dimara Kusuma[3], Hakim Agung Purwo Wicaksono[4]

[1-4]Department of Informatics Engineering, Universitas Muhammadiyah Purwokerto, Indonesia, 53182

ridhomuktiadi@ump.ac.id
https://doi.org/10.37339/e-komtek.v9i2.2608

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
28-07-2025
Revised:
04-08-2025
Accepted:
06-08-2025
Online first :
31-06-2025

Abstract

Kedai Ngopi Ngopi faces several operational challenges, including the absence of a digital menu display, manual transaction recording, and limited payment methods. This study aims to develop a web-based information system integrated with sales and digital payment functionalities. The system development was carried out using the Rapid Application Development (RAD) method, which emphasizes speed and active user involvement through iterative processes. The developed system includes product management, transaction recording, order notifications, and integration with a digital payment system. Test results indicate that the system significantly improves operational efficiency, expands the range of available payment methods, and enhances the overall customer service experience.

Keywords: Information System, Point of Sales, Payment Gateway, Rapid Application Development.

Abstrak

Kedai Ngopi Ngopi menghadapi kendala dalam operasionalnya, seperti belum tersedianya media digital untuk menampilkan menu, pencatatan transaksi manual, dan metode pembayaran yang terbatas. Penelitian ini bertujuan mengembangkan sistem informasi berbasis website yang terintegrasi dengan sistem penjualan dan pembayaran digital. Pengembangan dilakukan dengan metode Rapid Application Development, yang menekankan kecepatan dan keterlibatan pengguna dalam proses iteratif. Sistem yang dibangun mencakup manajemen produk, pencatatan transaksi, notifikasi pesanan, serta integrasi dengan sistem pembayaran digital. Hasil pengujian menunjukkan bahwa sistem ini mampu meningkatkan efisiensi operasional dan memperluas metode pembayaran, serta memberikan pengalaman layanan yang lebih baik bagi pelanggan.

***Kata-kata kunci**: Sistem Informasi, Point of Sales, Payment Gateway, Rapid Application Development*

1. Introduction

The development of information technology over the past decade has significantly impacted various sectors, including micro, small, and medium enterprises (MSMEs). The implementation of management information systems has proven to improve operational efficiency and business sustainability by digitizing manual business processes **[1], [2]**. Digital information systems not only enhance efficiency but also support MSMEs in data-driven decision-making and adaptation to increasingly competitive market dynamics.

Studies have shown that web-based Point of Sales (POS) systems positively impact transaction management, inventory control, and reporting efficiency **[3], [4], [5]**. These systems enable the digital transformation of business processes that were previously handled manually. However, many MSMEs still face challenges, including limited access to integrated digital solutions and slow adaptation to non-cash payment technologies **[6], [7], [8].**

Kedai Ngopi Ngopi, a small culinary business, has yet to fully adopt a digital information system. The business faces several issues, such as the absence of an online menu platform, manual transaction recording, and cash-only payment methods. These limitations reduce operational efficiency and lower customer service quality. Therefore, integrating POS systems with payment gateway technology emerges as a strategic solution to enhance user experience and transaction security **[9], [10].**

To address these issues, this research adopts the Rapid Application Development (RAD) method for system development. RAD emphasizes fast development and active user involvement through continuous prototyping iterations. This method has proven effective in web-based application development due to its flexibility in adapting to evolving user needs **[11], [12], [13]**. To address these issues, this research adopts the Rapid Application Development (RAD) method for system development. RAD emphasizes fast development and active user involvement through continuous prototyping iterations. This method has proven effective in web-based application development due to its flexibility in adapting to evolving user needs.

2. Method

This research adopts the Rapid Application Development (RAD) approach because it accelerates system development through iterative prototyping that actively involves users. RAD is known for its speed and flexibility, enabling the system to evolve in response to direct user

feedback [11]. It is particularly suitable for small to medium-scale software development with dynamic requirements, limited budgets, and tight timelines. According to Sondang **[14],** applying the RAD method to web-based information system development has been shown to deliver faster results, adapt effectively to changing user needs, and facilitate user adoption and system sustainability. The process is carried out as illustrated in **Figure 1**, which depicts the flow and steps involved in using the RAD Method.

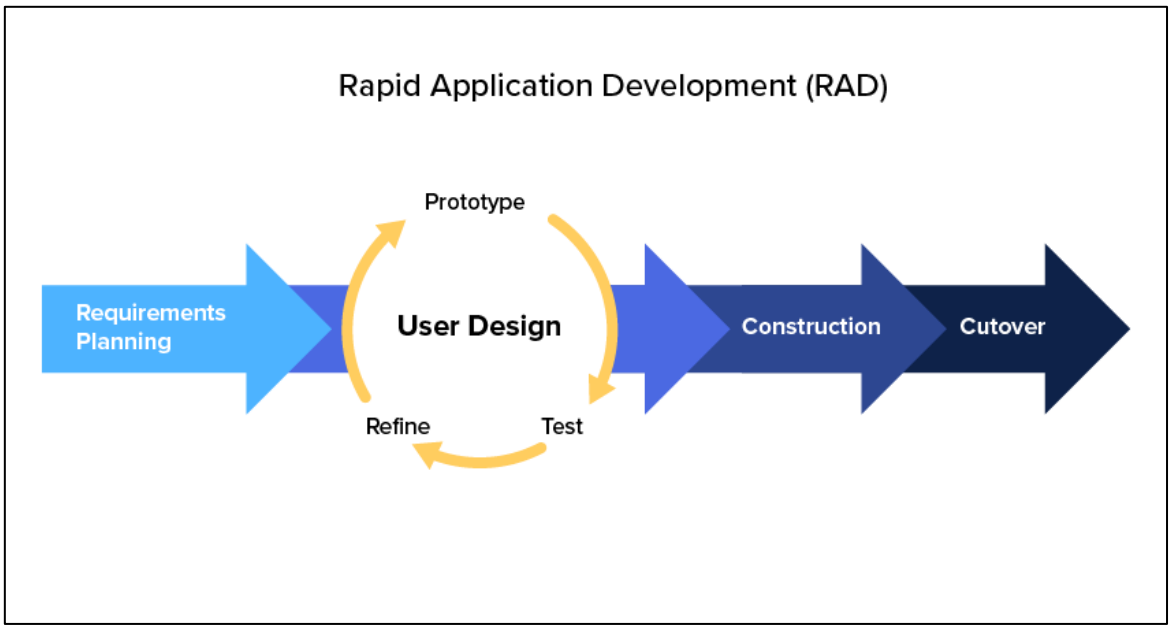

Figure 1. RAD Method Steps

2.1 Requirement Planning

In the initial phase, begins with problem identification, followed by data collection to determine the desired end system objectives **[15]**.

1. Observation

   Direct observation of sales processes at Kedai Ngopi Ngopi to identify problems in manual recording and information delivery.

2. Interview

   Unstructured interviews with the shop owner to explore required features to improve operations.

3. Literature Study

   Literature review of journals, articles, and books related to web-based POS development using RAD and digital payment gateway integration.

2.2 User Design

The User Design is carried out using Unified Modeling Language (UML) to visualize and document the system architecture. The diagrams used include a use case diagram, an activity diagram, and a three-tier class diagram, which support comprehensive system visualization and provide a clear structure for understanding the system's interactions and workflows. According to Nistrina and Sahidah **[16]**, UML is an effective tool for designing web-based systems, particularly in representing object-oriented processes such as those in POS systems. These diagrams simplify complex business processes, ensuring that all stakeholders understand the system's flow and requirements.

In this study, UML diagrams were instrumental in the design of the POS system for Kedai Ngopi Ngopi. The use case diagram describes the system's functional requirements and interactions between the user (admin and customer) and the system. The activity diagram visualizes the workflow of the system, illustrating how each user interacts with the system during transactions **[17]**.

2.3 Construction

This stage implements the system as designed. The development was carried out iteratively, allowing adjustments based on user feedback obtained from the previous stages **[15].**

2.4 Cutover

After development, the system was tested using blackbox testing to validate whether its functionalities align with predefined specifications without examining internal code structure **[18], [19]**. This testing approach focuses on evaluating the system's functional behavior based on inputs and expected outputs, ensuring that it performs as intended without examining the code's internal logic. Blackbox testing is especially effective in identifying interface errors, incorrect functions, and data structure errors in web-based systems **[19]**.

3. Results and Discussion

3.1 Research Result

The initial stage in system development was needs planning, conducted through observation of the sales and record-keeping processes at Kedai Ngopi Ngopi. Based on the results of these observations and unstructured interviews with the cafe manager, several issues were identified: the absence of digital media to display product information, requiring customers to visit in person to view available menus; manual transaction recording using books, which poses

a risk of data loss, damage, and disorganization; and a limited payment method, restricted to cash, which lacks flexibility and security.

As a result of this planning phase, the system requirements were formulated for two user types: administrators and customers. Admins are responsible for managing products, receiving order notifications, and recording sales, while customers place orders through the website with multiple available payment options. The admin features include six main menus: Dashboard, Point of Sale, Category, Item, Order History, and About Us, as illustrated in **Figure 2**. Sitemap Admin Menu.

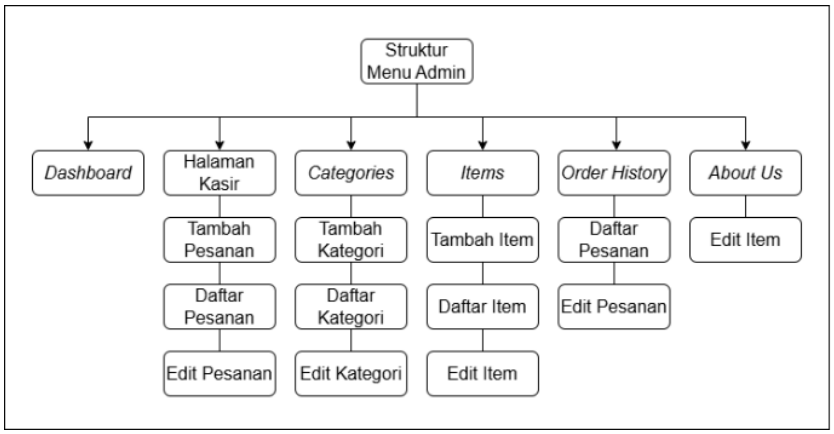

Figure 2. Sitemap Admin Menu

Meanwhile, the customer-facing features consist of four main menus: Homepage, Cart, Order Details, and About Us, as shown in **Figure 3**. Sitemap Customer Menu.

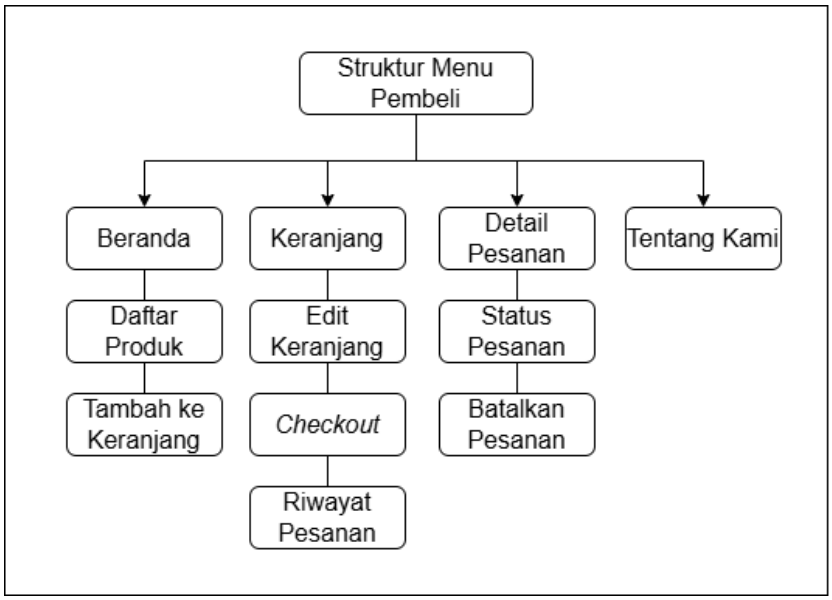

Figure 3. Sitemap Customer Menu

The system was designed using the Unified Modeling Language (UML), comprising a use case diagram, an activity diagram, and a three-tier class diagram. This design employed an iterative approach, with continuous collaboration between developers and users. Each iteration incorporated feedback that shaped the next cycle, ensuring that the design aligned with the business's operational needs and remained sustainable.

The use case diagram represented the interaction between actors and the system. In this system, the two main actors are the admin and the customer. The administrator manages inventory and sales, while the customer interacts with the ordering system, as shown in **Figure 4** (Use Case Kedai Ngopi Ngopi).

Figure 4. Use Case Diagram Kedai Ngopi Ngopi

The activity diagram outlines the flow of activities within the system, particularly the product ordering process. It helps visualize how customers and administrators interact step by step during the transaction. In this study, two main ordering processes are modeled using activity diagrams: ordering via the website interface by customers and ordering via the cashier interface by admins.

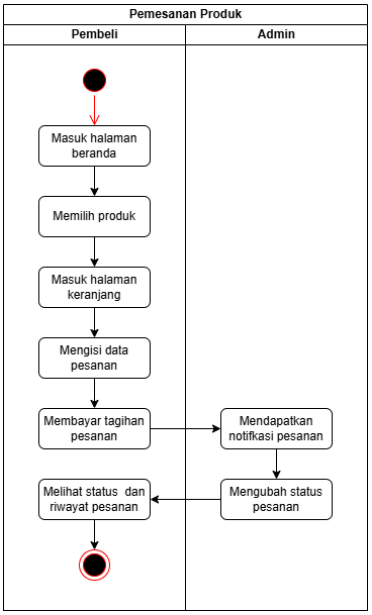

Figure 5. Activity Diagram Order Menu through Website

Figure 5 illustrates the ordering process through the website. In this scenario, the customer browses the product menu, selects items, adds them to the cart, proceeds to checkout, fills in order details, selects a payment method (cash or non-cash), and finally receives a payment confirmation. For non-cash transactions, the system integrates with the Midtrans Snap UI for online payments.

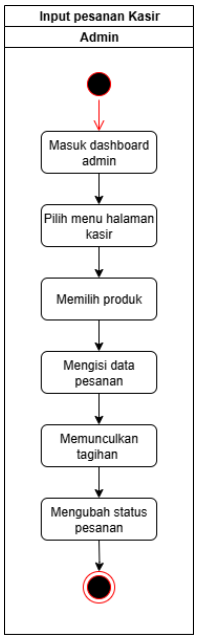

Figure 6. Activity Diagram Order Menu through Cashier Page

Figure 6 presents the ordering process via the cashier page. The admin primarily uses this flow during in-store purchases. The administrator selects the desired items on behalf of the customer, enters the quantity and order details, and completes the transaction using the chosen payment method. The system then stores the transaction data and updates the order status accordingly.

The construction phase was carried out after the needs analysis and system design were agreed upon. The data flow within the Kedai Ngopi Ngopi Point-of-Sale (POS) system was structured to support integrated interactions among the frontend, backend, database, and supporting features. On the admin side, item and category management, as well as order recording, were facilitated through an admin interface built using Filament. The data handled by the administrator is stored in a MySQL database via the Laravel backend using the Eloquent ORM, which simplifies object-oriented data manipulation. Additionally, hashing was applied to secure sensitive data such as user passwords.

Meanwhile, on the customer side, the interface was designed using Blade and Livewire, enabling customers to place orders interactively. Each order submitted by the customer is stored in the orders table in the database and automatically triggers a notification on the admin dashboard using Filament Notification, allowing the admin to respond promptly. The customer-facing frontend is styled using Tailwind CSS as the default in Laravel 11. To provide a comprehensive overview of the system's technologies, a development architecture diagram is shown in Figure 8. Kedai Ngopi Ngopi Technology Diagram.

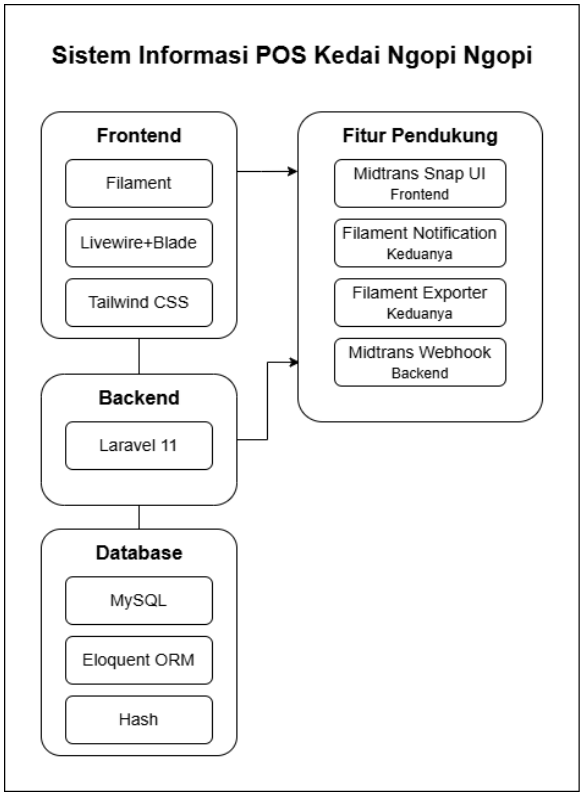

Figure 7. Kedai Ngopi Ngopi Technology Diagram

For payment processing, when a customer selects a non-cash payment method, the system generates a Snap token, stores it in the orders table, and then calls the Midtrans Snap UI to display the billing page. Upon successful payment, Midtrans Webhook sends a callback to the backend to verify the payment status and update the order status accordingly. Admins can monitor transactions marked as "paid" directly from the dashboard.

In addition, the system provides a Filament Exporter feature, which allows admins to export transaction history data into CSV or XLSX format to simplify sales reporting. Some of the main interfaces implemented in the system are described below:

1. POS Page (Admin)

   The admin uses this page to record in-store transactions. The admin can select items, set quantities, view the total amount, and save orders. The data is stored in the orders and order_details tables.

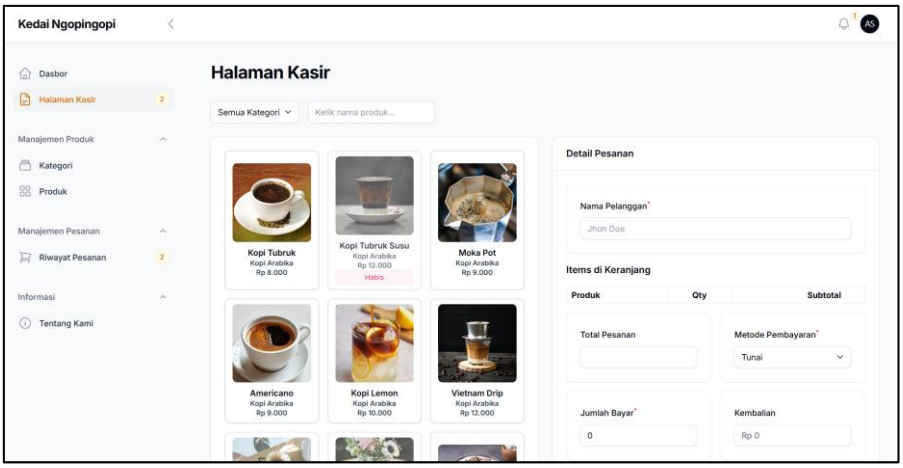

Figure 8. POS Page (Admin)

2. Payment Page via Midtrans (Customer)

This page appears when a customer selects a non-cash payment method. The system displays the billing interface via the Midtrans Snap UI, and upon successful payment, the order status is automatically updated using the Midtrans Webhook.

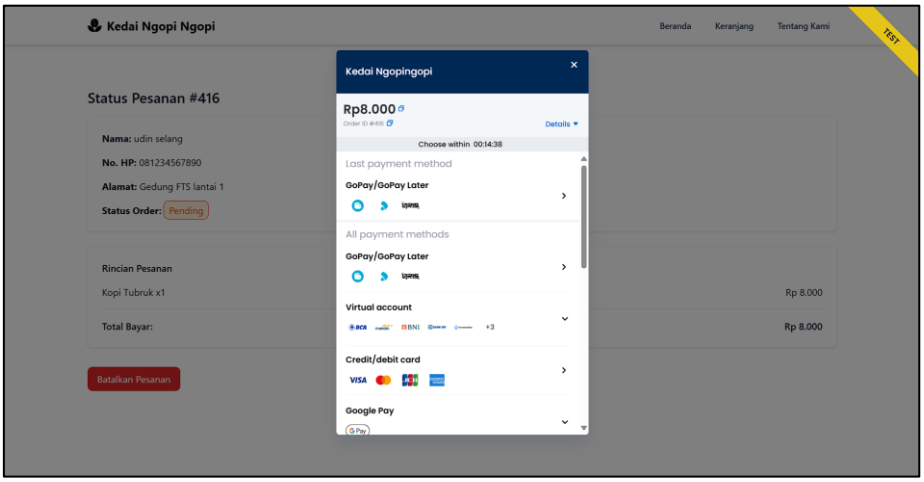

Figure 9. Payment Page via Midtrans (Customer)

3. Order Notification via Filament (Admin)

This notification appears automatically when a new customer order is received. This feature helps administrators monitor orders in real time.

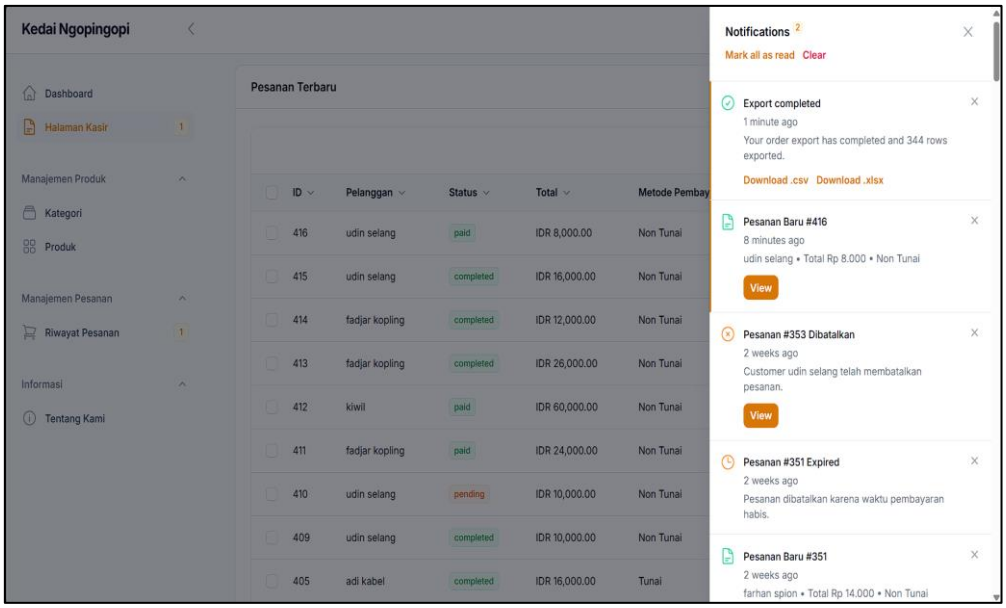

Figure 10. Order Notification via Filament (Admin)

4. Export Order History Page (Admin)

This page displays all completed transactions. Admins can view detailed orders, payment status, and export data in CSV or XLSX format using the export feature.

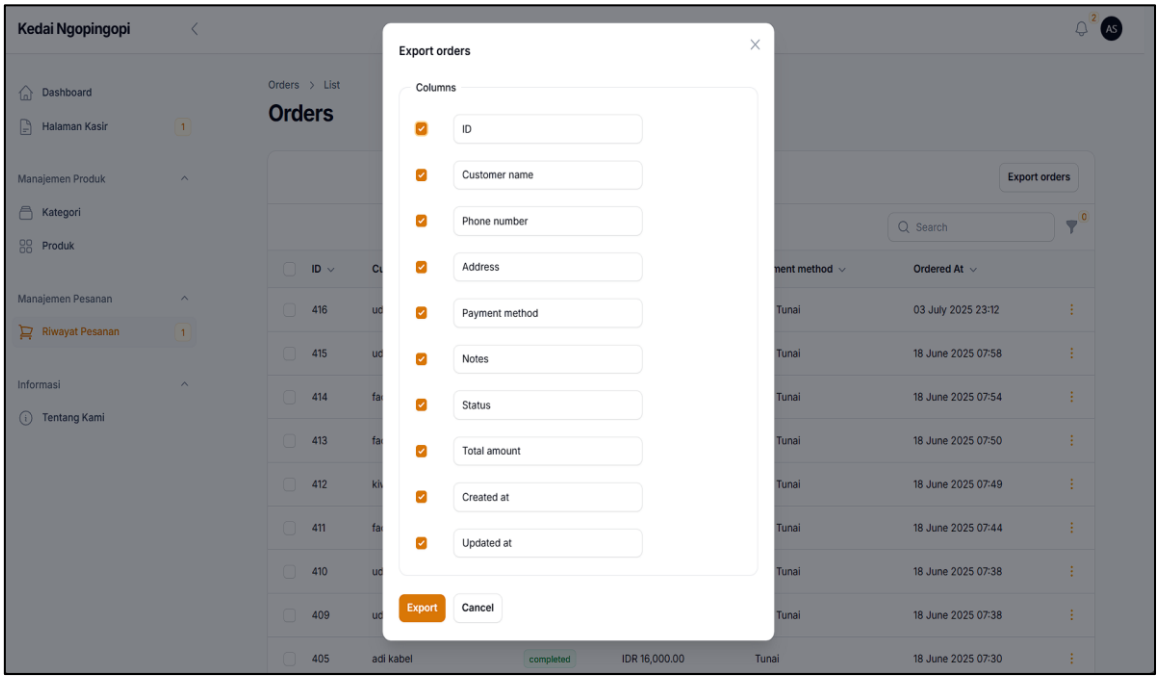

Figure 11. Export Order History Page (Admin)

The cutover stage is the final step of system development and aims to ensure that the system operates in accordance with the initial design and user requirements. During this stage, the system was tested to validate its functionality and guarantee that all developed features work as intended.

The testing method used was black-box testing, which focuses on testing the functional aspects of the system without examining its internal structure or code logic. Inputs were provided to the system, and the corresponding outputs were observed and compared against expected results. The testing process was conducted at Kedai Ngopi Ngopi and involved the developer and user representatives (the administrator and the customer).

By including users in the testing process, direct feedback was obtained regarding usability, response speed, and the alignment of system features with operational needs. The test results confirmed the system's ability to enhance transaction recording, expand payment options, and improve service efficiency.

Table 1. Blackbox Testing

Process	Actor	Action	Result
Login Admin	Admin	Enter a valid email and password, then click the login button	Admin successfully accesses the admin dashboard
Login Admin (invalid)	Admin	Enter an incorrect email or password	System denies login and displays an error message
Item Management	Admin	Add, update, and delete item data	Item data is saved/updated/deleted in the database
Category Management	Admin	Add, update, and delete category data	Category data is saved/updated/deleted in the database
Order via POS	Admin	Select items, set quantity, fill customer details, then save	Order is saved to the orders and order_details tables
Incoming Order Notification	Admin	Wait for orders submitted by customers through the website	Notification appears automatically on the admin page
Order History	Admin	Open the order history page	All orders are displayed with complete status and details
Export Order Data	Admin	Export order history data to CSV/XLSX format	CSV/XLSX file successfully downloaded
Cash Payment	Admin	Process order using the cash payment method	The order is marked as paid
Non-Cash Payment (Midtrans)	Customer	Place an order and choose a non-cash payment method	Snap UI appears, and payment can be made
Successful Non-Cash Payment	Customer /Admin	Make payment through Snap UI	Order status changes to paid

Process	Actor	Action	Result
Failed Non-Cash Payment	Customer /Admin	Cancel or fail to complete payment in Snap UI	Order status changes to cancelled, and the admin receives notification
Purchase History (Customer)	Customer	View purchase history on the customer page	All orders are displayed with their respective status
Order Status Validation	Admin	Check the order status after payment	Status reflects notification from Midtrans
Data Security	Admin	Verify that passwords are stored as a hash in the database	Sensitive data is not visible in plain text

3.2 Discussion

The results of this research indicate that the developed POS web-based system successfully addressed the issues identified during the planning phase. Problems related to manual sales recording, lack of digital media, and limited payment options were effectively resolved through the implemented features. This achievement was made possible by applying the Rapid Application Development (RAD) method, which enabled frequent user involvement and iterative feedback throughout development.

The use of Unified Modeling Language (UML) diagrams significantly contributed to the successful development of the web-based POS system. UML served as a standardized modeling language that effectively communicated system behavior and structure throughout the development process.

The use case diagram provided a clear visualization of how actors interact with the system. It mapped the functional requirements for both administrative and customer roles, enabling stakeholders to understand and validate expected behaviors early in the development phase. According to Khilda & Lisna [16]Use case diagrams are effective in defining system scope and user interactions, making them essential tools during requirements analysis in web-based systems.

Meanwhile, the activity diagrams illustrated detailed workflows for key system operations, such as placing orders via the website and the cashier interface. These diagrams helped the development team analyze and optimize business processes by clearly representing conditional flows and user actions. As supported by Nugroho et al. [17]Activity diagrams are particularly

useful for modeling real-time business processes because they enhance clarity and accuracy in understanding system logic prior to implementation.

By employing these UML tools, the development process was more structured and responsive to user needs. They enabled early detection of ambiguities and reduced the risk of misalignment between the functional design and the implemented system.

During the construction phase, Laravel 11, combined with Filament, was used to develop the administrative interface, while Blade and Livewire facilitated smooth interaction with customer orders. The integration with Midtrans as the payment gateway was critical in expanding payment methods and reducing transaction errors. This integration process aligns with the study by Waluyo et al. **[1]**, which demonstrated how Snap API streamlines payment verification and enhances the user experience. Additional features, such as real-time Filament Notifications for incoming orders and data export capabilities in CSV/XLSX formats, further demonstrate the system's value to operational tasks. The Blackbox Testing conducted confirmed that the system's functionality met specifications and user expectations, particularly with respect to usability and responsiveness.

The cutover phase focused on system testing using Blackbox Testing, confirming that all functionalities worked as expected. Engaging users (admin and customers) in this testing phase provided valuable feedback and validated system usability in a real environment. The system's ability to handle both cash and non-cash transactions, generate real-time notifications, and export reports effectively demonstrates that it meets the operational needs of the business. Blackbox testing, especially with the Equivalence Partitioning technique, is an effective method for validating software functionalities without considering internal code structure. This technique has proven effective in other systems as well, such as in the study conducted by Wijaya and Astuti [20], which demonstrated that Blackbox Testing was able to detect incorrect functionalities and confirm the readiness of an information system for production use.

Overall, the system successfully combines ease of use, secure data handling (via password hashing and token-based payment validation), and administrative efficiency. These outcomes reinforce the importance of using a structured and participative approach, such as RAD, in the development of digital transaction systems for micro and small enterprises.

4. Conclusion

Based on the research findings and discussion, it can be concluded that the development of the web-based information system at Kedai Ngopi Ngopi successfully addressed the operational issues faced by the café. The system integrates a point-of-sale (POS) feature and a payment gateway, facilitating product management, transaction recording, and payment processing in a more efficient, secure, and flexible manner.

The system provides two distinct user roles: admin and customer, with features tailored to their respective needs. The administrator can manage product data, categories, and sales records, monitor transaction history, and receive real-time order notifications. Meanwhile, customers can access the menu, place orders, select payment methods, and use digital payment services via the Midtrans integration.

Therefore, the development of a web-based POS system integrated with a payment gateway has proven effective in improving transaction management efficiency, enhancing data management, expanding payment options, and elevating service quality for Kedai Ngopi Ngopi's customers.

References

[1] Waluyo, I. Ningrum Resmawa, S. Masruroh, and R. Kurniawati, "Penerapan Sistem Informasi Manajemen untuk UMKM yang Berfokus Pada Pengelolaan Sumber Daya Alam Secara Berkelanjutan," *Journal of Human And Education*, vol. 5, no. 1, pp. 48–55, 2025.

[2] Y. Fatman, N. K. Nafisah, and P. B. J. Pambudi, "Implementasi Payment Gateway dengan Menggunakan Midtrans pada Website UMKM Geberco," *Jurnal KomtekInfo*, vol. 10, pp. 64–72, 2023, doi: 10.35134/komtekinfo.v10i2.364.

[3] D. E. T. Salim, D. David, G. Syarifuddin, S. Kosasi, and I. D. A. E. Yuliani, "Implementation of Point of Sales Using Laravel Framework on Matahari Motor," *CCIT Journal*, vol. 16, no. 1, pp. 111–123, 2023, doi: 10.33050/ccit.v16i1.2557.

[4] F. Azizah, L. Novyanti, N. O. Amri, A. Oktaviani, and R. Nurfalah, "Perancangan Aplikasi Point of Sale Pandita Coffee Berbasis Web Dengan Kombinasi Model Sdlc Waterfall," *Jurnal Sistem Informasi Kaputama (JSIK)*, vol. 6, no. 1, pp. 69–74, 2022, doi: 10.59697/jsik.v6i1.184.

[5] C. Arig, E. R. Borbon, T. C. Guzman, K. A. Reynaldo, and M. Melendez, "Point-of-Sale System with Card Reader for HTM Department Stephanie," vol. VII, no. 2454, pp. 1175–1189, 2023, doi: 10.47772/IJRISS.

[6] O. R. Runda, S. Kosasi, and G. Syarifudin, "Implementasi Progressive Web Application Pada Toko Online Widman Store Pontianak," *e-Jurnal JUSITI (Jurnal Sistem Informasi dan Teknologi Informasi)*, vol. 10, no. 2, pp. 170–179, 2021, doi: 10.36774/jusiti.v10i2.892.

[7] S. Ake, "Security Review of Payment Gateway," vol. 13, no. 3, pp. 34–49, 2022, doi: 10.17605/OSF.IO/C7QAY.

[8] B. S. Prayogi, I. Fitri, and R. Nuraini, "Aplikasi Point of sale Berbasis Website pada Toko Sembako Tegar," *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 6, no. 2, pp. 260–266, 2022, doi: 10.35870/jtik.v6i2.411.

[9] C. Gibran, A. R. Dewi, and E. Hadinata, "Implementasi Framework Laravel Untuk Pengembangan Website Penjualan Ayam Potong Dengan Pemanfaatan Midtrans Menggunakan Metode Fast," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 246–253, 2024, doi: 10.55338/jikomsi.v7i1.2920.

[10] E. R. Djuwitaningrum, I. Budi, and W. Jati, "Implementasi Payment Gateway Midtrans pada Website E-commerce Toko Buah dan Sayur," *Jurnal IPTEK*, vol. 9, pp. 19–24, 2025.

[11] A. D. Hernindyaputra, M. E. Johan, and D. Tjahjana, "Integrated System Design Of Sales And Production Module Using RAD Method (Case Study: PT Shafira Putri Kreatif)," *International Journal of New Media Technology*, vol. 10, no. 1, 2023.

[12] R. W. Dewantoro, A. Ichsan, and P. P. Hariani, "Design and Build Point of Sales Applications with Rapid Application Development (RAD) for Sales Effectiveness," *Indonesian Journal of Applied Technology*, vol. 1, no. 2, pp. 64–71, 2024.

[13] F. Q. Khan, S. Rasheed, M. Alsheshtawi, T. M. Ahmed, and S. Jan, "A comparative analysis of RAD and agile technique for management of computing graduation projects," *Computers, Materials and Continua*, vol. 64, no. 2, pp. 777–796, 2020, doi: 10.32604/CMC.2020.010959.

[14] Sondang, "Penerapan Metode RAD Dalam Pengembangan Sistem Informasi Pemesanan Jasa Percetakan Berbasis Web pada Percetakan Karya Sehati Jaya," *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 8, pp. 871–881, 2024.

[15] M. S. P, M. D. Irawan, and A. P. Utama, "Implementasi RAD (Rapid Aplication Development) dan Uji Black Box pada Administrasi E-Arsip," *sudo Jurnal Teknik Informatika*, vol. 1, no. 2, pp. 60–71, 2022, doi: 10.56211/sudo.v1i2.19.

[16] K. Nistrina and L. Sahidah, "Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil," *Jurnal Sistem Informasi, J-SIKA*, vol. 4, no. 1, pp. 17–23, 2022.

[17] Siska Narulita, Ahmad Nugroho, and M. Zakki Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS)," *Bridge : Jurnal publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 3, pp. 244–256, 2024, doi: 10.62951/bridge.v2i3.174.

[18] R. Rifany and N. Pratiwi, "Perancangan Dan Implementasi Sistem Point Of Sales (POS) Berbasis Web Menggunakan Framework Laravel 11 (Studi Kasus Toko Umkm Nasution)," *Seminar Nasional TEKNOKA*, vol. 9, no. 2502, pp. 82–90, 2024.

[19] U. Nugraha and T. Sianturi, "Blackbox Testing On E-Commerce System Web-Based Evermos (Feature: Registration Experiment & Revamp)," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 8, pp. 1026–1037, 2021.

[20] Y. D. Wijaya and M. W. Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions," *Jurnal Digital Teknologi Informasi*, vol. 4, no. 1, p. 22, 2021, doi: 10.32502/digital.v4i1.3163.