MJI
Media Jurnal Informatika
Vol. 18, No. 1, June 2026
p-issn : 2088-2114, e-issn : 2477-2542

Development of an AI-Assisted Mobile Point of Sales and E-Commerce System with Real-Time Inventory Synchronization

Maulana Irfan [a,1], Miftakhul Ushbah Nastaftian [a,2], Hermawan Sentyaki Sarjito [a,3], Jonathan [a,4], Irham Maulana Johani [a,5], Muhammad Nasir [a,6*], Aditya Wicaksono [a,7]

[a] Software Engineering Technology, Vocational College of IPB University, Jl. Kumbang No. 14, Bogor 16128, Indonesia
[1]122maulanairfan@apps.ipb.ac.id; [2]takulnastaftian@apps.ipb.ac.id; [3]senthermawan@apps.ipb.ac.id;
[4]frizy2903jonathan@apps.ipb.ac.id; [5]ipbvirham@apps.ipb.ac.id; [6]m_nasir@apps.ipb.ac.id*;
[7]adityawicaksono@apps.ipb.ac.id;
* corresponding author

ARTICLE INFO

Article history
Received 2026-05-31
Revised 2026-06-24
Accepted 2026-06-27

Keywords
Agile Scrum
Artificial Intelligence
Barcode Scanning
E-Commerce
Point of Sales
Real-Time Synchronization

ABSTRACT

Indonesian micro, small, and medium enterprises (UMKMs) face significant operational bottlenecks due to their reliance on manual and fragmented sales and inventory management systems. To address these challenges, this research designs, develops, and evaluates PasPOS, an integrated mobile Point of Sales (POS) and E-Commerce system tailored for Indonesian micro-retailers, utilizing Toko Mpok Nani as an implementation case study. Developed chronologically using the Agile Scrum framework across eight sprints, the platform implements a decoupled microservices architecture. This infrastructure consists of a centralized Laravel 12 backend API and Flutter-based Admin and Member applications, all synchronized through an 11-table relational database. Furthermore, the Admin App integrates an edge computer vision-based barcode scanning feature to accelerate cashier transactions using standard smartphone cameras. System validation via User Acceptance Testing (UAT) using a structured Likert-scale questionnaire yielded an overall average score of 4.44 out of 5.00 for the Admin App and 4.43 for the Member App, classifying both as "Very Good". These empirical findings demonstrate that PasPOS successfully eliminates cross-channel stock inconsistencies while offering a replicable architectural blueprint for broader UMKM digitalization initiatives.

1. Introduction

Micro, small, and medium enterprises (UMKM) constitute the backbone of Indonesia's national economy, with approximately 30.18 million registered units as of December 2024 representing 99% of all business entities, contributing more than 60% of the national Gross Domestic Product, and absorbing 97% of the total workforce [1], [2]. Despite this critical economic role, the adoption of digital technology for core business operations among UMKM remains critically underdeveloped. According to the BPS Profil Industri Mikro dan Kecil 2024, only 46.83 percent of micro and small industry businesses in Indonesia use the internet for business management, and among those who do, the primary purpose is limited to product marketing at 48.58 percent, while integrated operational functions such as real-time stock control and transaction management receive significantly less attention [3]. This gap is particularly evident in small-scale retail businesses such as Toko Mpok Nani, which continue to rely on conventional paper-based systems for inventory recording and sales transactions. Such manual approaches generate critical operational problems, most notably the absence of real-time data synchronization between the warehouse and the storefront, which leads to stock inconsistency, financial loss, and a decline in customer trust [4], [5]. Research confirms that UMKM competitiveness is still hindered by limited market access, low digital technology utilization, and minimal

adoption of integrated business management tools, underscoring the urgent need for practical digital solutions designed specifically for this sector [6].

To address these challenges, this research proposes a comprehensive, mobile-based Point of Sales and E-Commerce system engineered specifically for the operational context of Indonesian micro businesses, named PasPOS. Point of Sales systems have been widely studied as effective tools for improving transaction accuracy, inventory control, and sales reporting in retail environments, and their integration with E-Commerce channels has been shown to significantly enhance business reach and operational continuity [7], [8]. In addition to inventory synchronization, transaction speed and product input accuracy are also critical factors in micro-retail operations. Manual product searching during cashier transactions may increase processing time and create input errors, especially when product variations increase. Therefore, PasPOS incorporates an edge computer vision-based barcode scanning feature in the Admin App, leveraging an open-source barcode utility to support automatic product identification through the mobile camera. By combining this edge-based pattern recognition library with real-time inventory synchronization, the system improves transaction efficiency while reducing the possibility of human error in product selection. PasPOS is built on a microservices architecture using Laravel 12 as the backend framework and Flutter as the cross-platform mobile frontend, enabling independent scalability of each system component without disrupting overall operations [9]. The microservices architectural pattern is particularly well-suited to applications that require modular service deployment and independent scaling, as each service can be developed, tested, and maintained separately while communicating through a unified API layer [10]. The system integrates three interconnected components: a centralized Backend API, an Admin App for point of sales operations, and a Member App for online customer engagement. This multi-platform design is grounded in the principles of real-time data synchronization and unified database management, ensuring that every transaction and stock update is instantly reflected across all interfaces [11].

The development of PasPOS follows the Agile Scrum framework, which structures the implementation process into iterative sprints that accommodate evolving requirements and stakeholder feedback [12]. This methodology ensures transparency and adaptability throughout the development lifecycle, from software requirement specification to final system delivery. The primary objective of this research is to design, develop, and evaluate an integrated mobile Point of Sales and E-Commerce system that addresses the inventory management and sales fragmentation challenges faced by Indonesian micro businesses. It is expected that the implementation of PasPOS will produce measurable improvements in three areas: operational efficiency through the elimination of manual recording errors, data accuracy through real-time stock synchronization across all sales channels, and market reach expansion through the integration of an online customer-facing application. Beyond its immediate application to Toko Mpok Nani, this research is expected to contribute a replicable microservices-based system architecture model that other UMKM practitioners and researchers can adapt in similar retail digitalization efforts [13].

2. Method

2.1 Type and Approach of Research

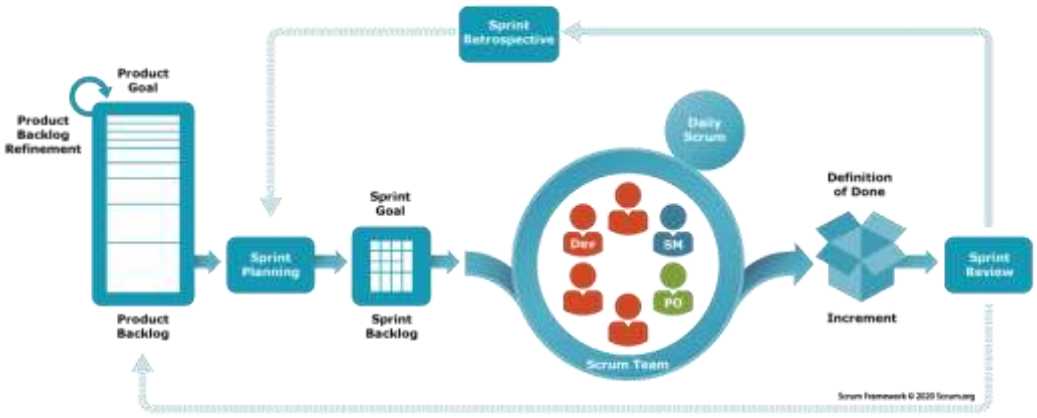


Fig. 1. Scrum Methodology Framework for PasPOS Development

This study employs a Research and Development (R&D) approach coupled with a case study design to design, build, and evaluate PasPOS, an integrated Point of Sales (POS) and e-commerce system tailored for micro-businesses (UMKMs). The entire development process is strictly driven by the Agile Scrum framework, an iterative and incremental methodology that organizes work into short, time-boxed cycles called sprints, enabling the development team to deliver functional software incrementally while continuously incorporating stakeholder feedback [14]. The workflow of the iterative Scrum framework applied in this study is illustrated in Figure 1. The justification for selecting this approach is rooted in the iterative and incremental nature of Scrum, which is highly suited to projects where system requirements evolve throughout the development process, as is common in small business digitalization efforts [15]. Unlike the traditional waterfall model, which requires all requirements to be fully defined before development begins, Scrum enables continuous stakeholder involvement and allows the development team to respond to feedback at the end of each sprint cycle, thereby reducing the risk of delivering a system misaligned with actual user needs [16].

2.2 Object and Scope of Research

The primary object of this research is PasPOS, an integrated digital platform that combines physical cash register transaction capabilities with an online electronic commerce channel. The development of this dual-channel solution follows established trends in web-mobile software architectures designed to expand retail reaching through integrated e-commerce features [17]. The scope of this system architecture encompasses three core components, which are the Backend API serving as the central hub for business logic and data processing, the Admin App designed specifically for the shop owner and staff to manage daily operations, and the Member App dedicated to customers for seamless e-commerce transactions. The domain of this study is situated within retail digitalization and micro-business supply chain management. To ensure the representativeness and generalizability of the findings, the operational boundaries are specifically focused on Toko Mpok Nani, a micro-scale wholesale grocery store (toko grosir kelontong) located in Jakarta Barat, Indonesia. This case study reflects a typical urban micro-enterprise environment, managing a multi-category inventory of approximately 100 to 200 active fast-moving SKUs. The establishment is operated directly by a team of three individuals, comprising the business owner and two active store staff, catering to both local end-consumers and smaller neighborhood retailers. By anchoring the research within these specific boundaries, the system's evaluation directly addresses the high-volume workflow patterns, physical-digital synchronization bottlenecks, and operational challenges common to the broader population of urban micro-retailers.

2.3 Data Collection Techniques

Data for this research were obtained through two primary approaches to ensure comprehensive requirement elicitation and system validation. The first approach involved direct observation and an intensive interview process conducted with the owner and staff of Toko Mpok Nani to identify operational pain points,

existing workflow patterns, and system requirements. This method aligns with established practices in requirements engineering, where direct stakeholder engagement is recommended as the most effective method for eliciting accurate and complete system requirements in small business contexts [18]. The findings from this process were formalized into a Software Requirement Specification (SRS) document and use case documentation, serving as the formal baseline for feature planning, sprint backlog construction, and acceptance criteria definition [19]. The second approach utilized a survey technique for evaluation purposes, where system acceptance data were collected using a structured questionnaire administered to two distinct participant groups consisting of the management and staff of Toko Mpok Nani representing administrative users of the Admin App, and selected customers representing end-users of the Member App.

2.4 Tools and Materials Used

The technical foundation of PasPOS is built on a microservices architecture, in which application functionality is decomposed into independently deployable services that communicate through well-defined API contracts [20]. Laravel 12 was selected as the backend framework due to its mature ecosystem, built-in support for RESTful API development, robust authentication mechanisms, and strong community support that makes it particularly appropriate for rapid application development in resource-constrained projects [21]. On the frontend, Flutter was selected as the mobile framework because of its ability to produce natively compiled applications for Android from a single codebase, significantly reducing development time and ensuring UI consistency across both the Admin App and Member App [22]. The Admin App also implements a barcode scanning feature using a mobile_scanner library that utilizes the smartphone camera to detect and decode product barcodes. This feature applies computer vision-based recognition to capture barcode patterns and match the detected barcode value with product records stored in the centralized database. When a barcode is successfully recognized, the system automatically retrieves the corresponding product name, price, and stock information through the Backend API, enabling faster cashier transactions and reducing manual input errors. Data storage is managed through a centralized relational model consisting of 11 tables that serves all system components, addressing the underlying data synchronization and inventory management problems. Communication between the Laravel 12 backend and Flutter mobile frontends is enabled via RESTful API principles to maintain consistent and stateless communication [23], [24]. The primary material used for system evaluation is a structured Likert-scale questionnaire covering aspects of ease of use, feature completeness, system reliability, and overall satisfaction [25].

2.5 Research Procedures or Stages

The development of PasPOS was carried out chronologically through eight Scrum sprints, each systematically incorporating sprint planning, daily development activities, a sprint review, and a retrospective session. The first sprint focused entirely on establishing the core project infrastructure, including repository setup, service decomposition into microservices, database initialization, and shared authentication mechanisms across all components. Sprints two through seven incrementally delivered core functional modules, covering product and inventory management, point of sales transaction processing, order management for the e-commerce channel, and member account features. During the POS transaction sprint, the AI-assisted barcode scanning module was integrated into the cashier workflow. The module was designed to capture barcode input through the mobile camera, validate the detected barcode value, request product data from the Backend API, and automatically add the matched product into the transaction cart. Each sprint produced a working software increment that was reviewed against the SRS to verify alignment with defined requirements before progressing to the next cycle. This chronological and iterative delivery procedure allowed functional increments to be deployed independently, ensuring that as new features were integrated across the microservices, the overall development cycle remained adaptable, consistent with the established best practices of scaling Agile methodologies in distributed software architectures [24]. The procedure concluded with the execution of User Acceptance Testing (UAT) based on predefined test scenarios derived from the use case documentation.

2.6 Data Analysis Techniques

Data analysis was performed to transform the qualitative feedback from user evaluations into a quantifiable and verifiable measure of system acceptance. The responses gathered from the UAT questionnaire were measured on a scale of 1 to 5, where 1 represents strongly disagree and 5 represents strongly agree [25]. The average scores computed from the participant responses were then evaluated and interpreted against standard classification categories. Based on this established mathematical baseline, scores between 1.00 and

1.80 are classified as very poor, 1.81 to 2.60 as poor, 2.61 to 3.40 as sufficient, 3.41 to 4.20 as good, and 4.21 to 5.00 as very good [26]. The application of this analysis technique successfully yielded a quantifiable measure of system acceptance, interface usability, and functional completeness across different user roles for the PasPOS platform [25], [26].

3. Results and Discussion

3.1 System Implementation and Evaluation Results

The development of PasPOS was initiated by analyzing the qualitative data from pre-development stakeholder interviews to map out the system requirements. The thematic analysis of these interview sessions with the store owner and staff highlighted critical operational pain points, specifically severe data fragmentation between the main store and its branches, manual stock tracking inefficiencies, delayed transaction processing during peak hours, and an absolute lack of automated sales or purchase reporting. Consequently, these qualitative findings directly generated the key requirements and functional use cases of the platform, including real-time inventory synchronization across multi-branch networks, an integrated product catalog for direct member ordering, automated stock movement tracking, secure user role management, and dynamic sales reporting with data export and import capabilities.

The development of PasPOS successfully yielded three core components functioning as an integrated ecosystem, which consists of a centralized Backend API, an Admin application for mobile-based point of sales operations, and a Member application for e-commerce transactions. The technical architecture is structured around a microservices pattern, allowing each component service to execute independently while establishing communication through a centralized RESTful API powered by Laravel 12.

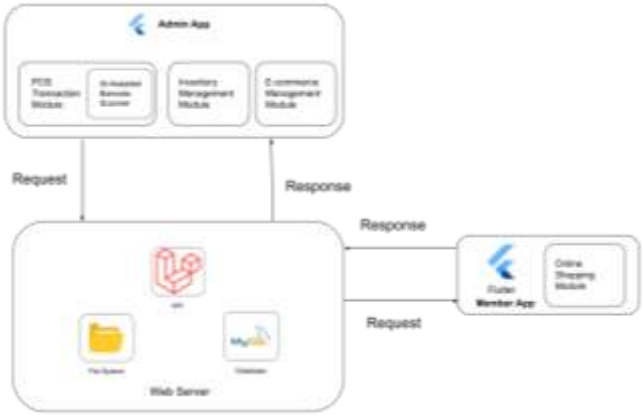


Fig. 2. System Architecture of PasPOS

Figure 2 illustrates the overall system architecture of PasPOS, demonstrating the Backend API as the central orchestration layer that connects the Admin App and Member App while managing dedicated services for inventory management, transaction processing, order handling, and user authentication. Supporting this architecture design is a centralized relational database schema comprising 11 tables that ensures absolute data consistency across both mobile interfaces.In the Admin App, an AI-assisted barcode scanning module is integrated into the POS transaction workflow. This module utilizes the smartphone camera and a Flutter-based barcode recognition library to detect and decode barcode patterns from product packaging. The detected barcode value is then sent to the Backend API to retrieve the corresponding product data, including product name, selling price, and stock availability. By embedding this recognition module into the cashier workflow, PasPOS reduces manual product search, minimizes input errors, and accelerates transaction processing while maintaining real-time synchronization with the centralized inventory database.

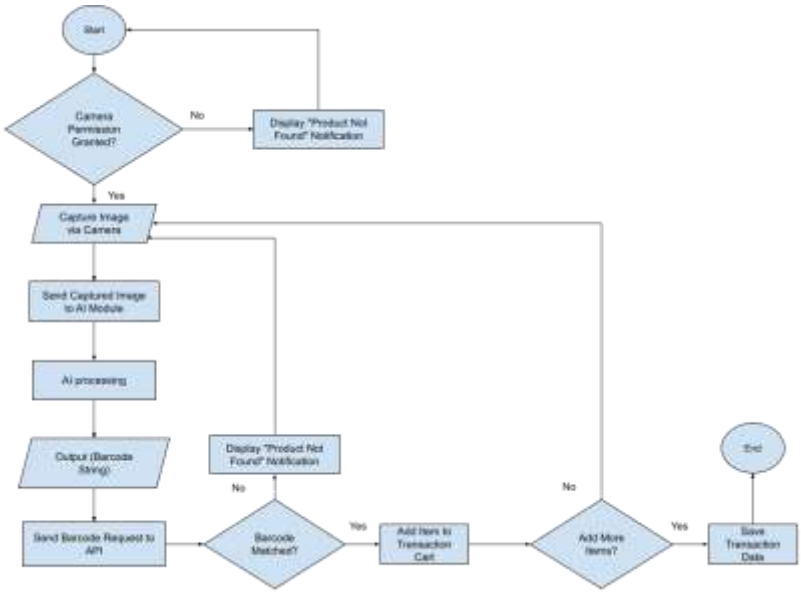


Fig. 3.AI-Assisted Barcode Scanner Workflow

The step-by-step operational workflow of the embedded AI-assisted barcode scanning module is further detailed in Figure 3. The process initiates when the cashier triggers the scanning function within the POS transaction module, prompting the system to request camera access permission. If the permission is denied, the application displays a "Camera Permission Required" notification and halts the scanning workflow. Conversely, when permission is granted, the system actively captures live frames through the smartphone camera.

Once a clear frame is captured, the image data is instantly channeled to a Flutter-based barcode recognition library for localized processing. Through edge recognition and pattern intelligence, this library extracts the visual features of the barcode and decodes them into a raw data string. This decoded output, representing the unique barcode sequence, is subsequently dispatched via an asynchronous API request to the central Laravel web server.

The backend server queries the MySQL database to verify whether the incoming barcode string matches an existing product record. If no match is found, the system delivers a "Product Not Found" notification to the cashier dashboard and loops back to reactivate the camera capture frame. If the barcode is successfully matched, the backend retrieves the detailed item parameters such as name, current stock, and unit pricing, and then appends the item straight into the active transaction cart. Finally, the system prompts the user to either capture additional items or conclude the workflow by securely saving the final transaction records into the centralized database.

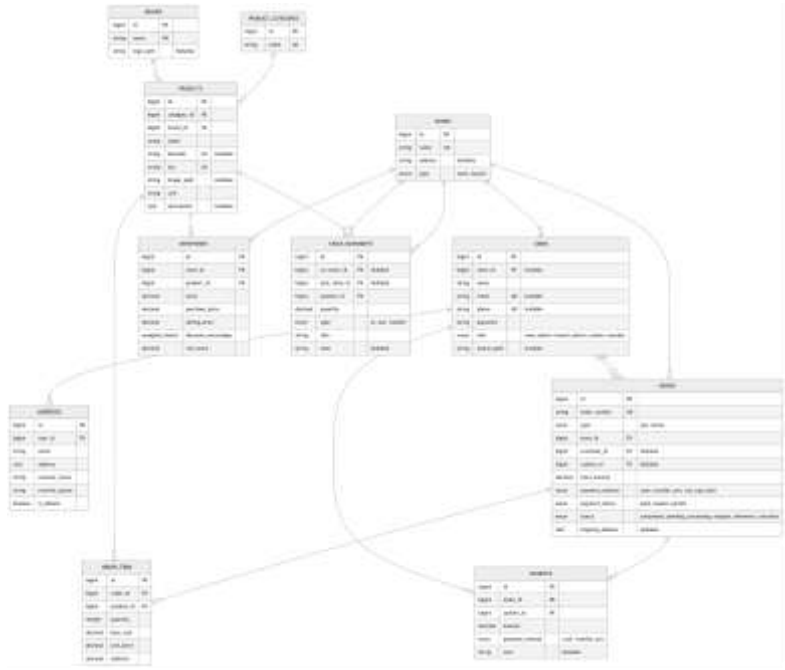
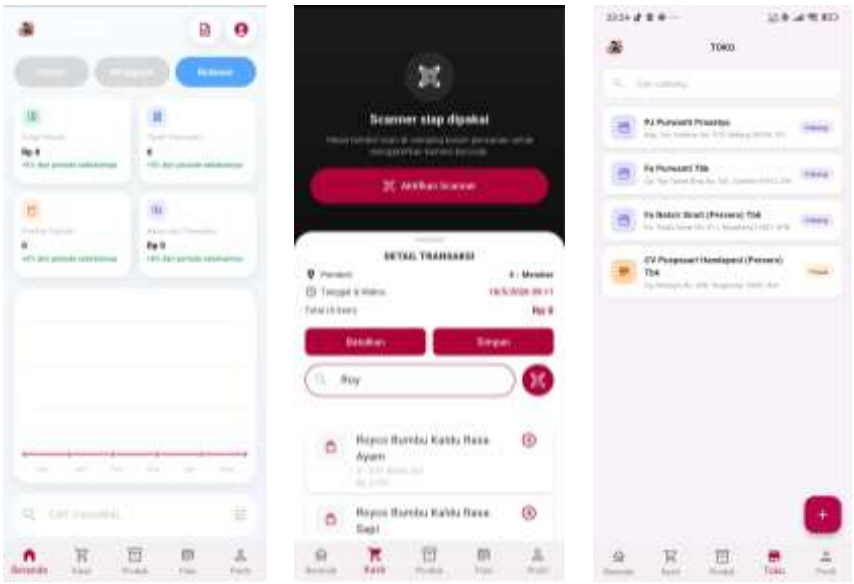


Fig. 4. Entity-Relationship Diagram of PasPOS Database

Figure 4 presents the entity-relationship diagram of the PasPOS database, detailing the structural relationships between the following entities: Stores, Users, Addresses, Product Categories, Brands, Products, Inventories, Stock Movements, Orders, Order Items, and Payments. This schema handles both physical POS workflows and digital online orders through a unified Orders table that distinguishes transaction types via a type attribute, whereas the Inventories table tracks per-store stock levels using purchase price, selling price, and minimum stock threshold fields to support real-time inventory monitoring across distinct store branches.
The user-facing layers were evaluated directly through interface verification and user testing. The Admin App provides store owners and staff with mobile point of sales capabilities, including product management, inventory control, transaction processing, and order monitoring.

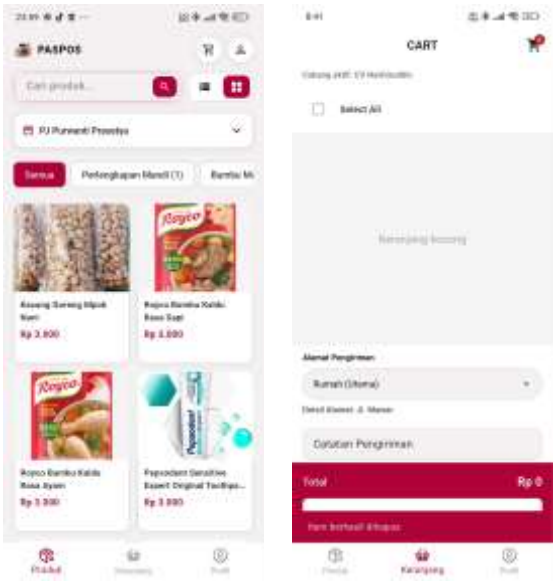


(a) (b) (c)

Fig. 5. Admin App Interface: (a) Dashboard overview, (b) POS transaction with barcode scanner, (c) Multi-branch store management

Figure 5 displays the Admin App interface across three key screens. First, the dashboard presents daily, weekly, and monthly sales summaries, including total omzet, total transactions, products sold, and

average transaction value. Second, the cashier screen enables efficient checkout operations through product search and barcode scanning, which serves as the AI-assisted component of the Admin App. Instead of requiring the cashier to search and input product data manually, the system utilizes camera-based barcode recognition to identify items. Once a barcode is detected, the application sends the barcode value to the Backend API to retrieve the corresponding product information from the centralized database, thereby improving transaction speed, minimizing errors, and strengthening the integration between physical cashier operations and real-time inventory control. Third, the store management screen displays all registered branches and the main store under a unified multi-branch view. Concurrently, the Member App enables customers to browse products, place orders, and track delivery status from their mobile devices.



(a) (b)

Fig. 6. Member App Interface: (a) Product catalog with branch selection and category filter, (b) Shopping cart with delivery address and shipping notes

Figure 6 displays the Member App interface across two key screens, including the product catalog screen allowing customers to browse available items by category with branch selection capability, showing real product images and prices; and the shopping cart screen enabling customers to review selected items, choose a delivery address, and add shipping notes before completing their order. Both frontends were constructed using Flutter to maintain consistent user interface behavior across Android devices while communicating with the same Laravel 12 Backend API. The functional integration is further validated through synchronized data states, where a stock update performed through the Admin App is immediately reflected in the product availability displayed on the Member App.

System acceptance was quantified through User Acceptance Testing administered to two primary participant groups. The first group consisted of 5 administrative users comprising the store owner and staff of Toko Mpok Nani who evaluated the Admin App, while the second group consisted of 13 member users who evaluated the Member App. Each participant completed a predefined set of test scenarios derived from use case documentation before completing a structured Likert-scale questionnaire.

Table 1. Summary of User Acceptance Testing (UAT) Scores

Aspect	*Admin App*	*Member App*	*Average*
Ease of Use	4.50	4.60	4.55
Feature Completeness	4.30	4.25	4.28
System Reliability	4.40	4.35	4.38
Overall Satisfaction	4.55	4.50	4.53
Overall Average	**4.44**	**4.43**	**4.43**

The quantitative summary derived from participant evaluations across the four core dimensions is thoroughly broken down within Table 1. When these data points are fully aggregated, the Admin App achieved an Overall Average score of 4.44 out of 5.00, falling within the Very Good category, and the Member App achieved an Overall Average score of 4.43, which is also classified as Very Good. Combined, the final aggregate average UAT score across both participant groups stands at 4.43 out of 5.00, confirming a highly positive acceptance rate across the entire platform ecosystem. However, this study is subject to a limitation regarding its highly localized and small sample size, consisting of only 18 total users across both applications. This constraint limits the broader statistical generalizability of the User Acceptance Testing (UAT) findings.

3.2 Analysis of Findings

The empirical results compiled from the user acceptance testing validate the design decisions made throughout the engineering process and demonstrate that PasPOS achieves an exceptional level of user acceptance appropriate for deployment in micro-business environments. The overall average scores of 4.44 for the Admin App and 4.43 for the Member App fall well within the predefined very good classification category, which spans from 4.21 to 5.00. This high level of performance across both administrative and member user groups indicates that the multi-platform mobile approach is highly effective in delivering an accessible, functional system to users who possess varying levels of digital literacy. In the UMKM context, user technical capabilities range widely from store owners with moderate technical familiarity to customers who interact primarily through standard consumer applications. The high ease of use scores, which reached 4.50 for the Admin App and 4.60 for the Member App, indicate that the unified Flutter user interface successfully accommodates these diverse literacy levels by providing intuitive, responsive navigation and clear visual hierarchies.

The technical integration between the separate applications directly addresses the data fragmentation and stock inconsistency problems caused by manual recording systems. By maintaining a single source of truth through a centralized relational database and a microservices architecture, PasPOS eliminates the discrepancies between warehouse logs and digital storefront availability previously experienced during manual operations at Toko Mpok Nani. The Inventories table serves as the primary technical mechanism ensuring synchronization in practice because it actively reflects stock adjustments from both physical POS transactions and online e-commerce channels. The integration of AI-assisted barcode scanning further strengthens the usability of PasPOS in real cashier operations. The feature reduces dependence on manual product search and supports faster checkout processing by automatically identifying products through barcode recognition. In the context of micro-business users, this functionality is important because it simplifies cashier workflows without requiring additional dedicated scanning hardware. The use of the smartphone camera as a barcode recognition device also makes the system more affordable and practical for UMKM adoption. This arithmetical behavior corresponds with established patterns in distributed software engineering, which emphasize that stateless RESTful communication combined with localized inventory tracking significantly reduces data collisions and cross-channel delays.

Furthermore, the successful delivery of these functional modules reinforces the efficacy of the Agile Scrum framework in managing the complexities of multi-component systems under strict resource and time constraints. Each of the eight sprints yielded a functional, reviewable software increment that was systematically validated against the Software Requirement Specification document. This chronological and iterative deployment cycle kept the final system aligned with stakeholder expectations, avoiding the requirement misalignment and scope creep that frequently disrupt traditional waterfall-based development models in small-scale digitalization projects.

3.3 Implications of the Results

The practical implications of PasPOS focus on the structural modernization of micro-businesses, providing a viable blueprint for transitioning manual retail operations into integrated, omni-channel digital ecosystems. In practical terms, the system equips micro-retailers with the software tools necessary to operate physical points of sale and online storefronts simultaneously without increasing administrative overhead or requiring separate inventory managers. This synchronization protects small business owners from overselling stock, streamlines order fulfillment, and provides clear, data-driven insights through the administrative dashboard regarding overall turnover, transaction volumes, and product performance.

Theoretically, this research contributes to the literature on software engineering application by demonstrating how advanced architectural patterns, such as microservices and centralized relational sync models, can be successfully scaled down to the micro-business level using accessible frameworks like Laravel

12 and Flutter. It proves that microservices are not exclusively beneficial for enterprise-scale platforms but can be leveraged to establish fault-tolerant, modular architectures for resource-constrained development projects.

For target stakeholders, the deployment of this platform increases operational efficiency for store staff while elevating the consumer shopping experience for local customers by introducing convenient online ordering, structured product catalogs, and transparent branch-based stock availability. For policy makers, the demonstrable success of PasPOS at Toko Mpok Nani supports the broader national agenda of accelerating digital adoption among the massive network of over 30 million registered UMKM units that currently form the structural backbone of the Indonesian economy. The system architecture, database layout, and API workflows verified through this investigation successfully serve as an actionable, referenceable framework for subsequent developers and investigators targeted at advancing the digital transformation of the micro-retail sector.

4. Conclusion

This research has successfully designed, developed, and evaluated PasPOS, an integrated mobile Point of Sales (POS) and e-commerce system engineered specifically to resolve the operational challenges of manual recording and data fragmentation within Indonesian micro-businesses (UMKMs). Developed chronologically through eight Agile Scrum sprints, the project delivered an interconnected ecosystem consisting of a centralized Laravel 12 Backend API, a mobile Admin App for cash register and store management, and a mobile Member App for online customer transactions. The system successfully addresses the initial research problem by establishing a single source of truth through a microservices arithmetical framework, allowing stock adjustments from both physical and online channels to synchronize in real time. The integration of AI-assisted barcode scanning also enhances the cashier workflow by enabling faster product identification and reducing manual input errors during POS transactions. The quantitative evaluation through User Acceptance Testing (UAT) confirmed an exceptional acceptance rate, yielding an overall average score of 4.43 out of 5.00 across both administrative and member user groups, which positions the platform firmly within the very good category. These results validate that a scaled-down microservices architecture built with accessible frameworks can effectively deliver high feature completeness, ease of use, and reliability to users with varying levels of digital literacy.

The significance of this study spans across practical, theoretical, and societal dimensions. Practically, PasPOS provides micro-retailers with a low-cost, unified tool to operate dual sales channels simultaneously without inflating administrative overhead, thereby preventing cross-channel stock discrepancies and enabling data-driven tracking via automated sales dashboards. Theoretically, it contributes a referenceable system architecture, database schema, and API model that demonstrates how modern enterprise software patterns can be scaled affordably for resource-constrained development projects. Societally, the success of this platform at Toko Mpok Nani directly aligns with the broader national objective of accelerating digital transformation among the millions of UMKM units that form the structural backbone of the Indonesian economy.

Nevertheless, this study carries structural limitations, as PasPOS was built exclusively for a single brand, preventing immediate multi-tenant adoption by other businesses without significant architectural re-engineering, and the frontend applications were developed and evaluated solely on the Android platform. To overcome these constraints, future research directions should focus on migrating the current infrastructure into a multi-tenant platform design to host multiple independent UMKM brands under a single, data-isolated architecture. Additionally, subsequent expansions should include the integration of automated payment gateways and shipping rate APIs to diversify payment channels, alongside building cross-platform compatibility for iOS devices to broaden user accessibility in larger real-world deployments.

Declarations

Author contribution. Equal contributions were made by all authors to the design, development, and testing of the PasPOS system with real-time data synchronization. The development, integration, and testing phases of the backend API, admin app, and member app were collectively managed by the entire author team. Additionally, all authors were involved in analyzing data, drafting the manuscript, and conducting reviews. The final manuscript has been approved by all authors for publication.
Funding statement. This research did not receive funding from any institution.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The source code supporting the results and structural design of the PasPOS platform is hosted across three separate repositories on GitHub. The cross-platform mobile frontend application for customers is available via the Member App repository at https://github.com/dark-hermes/paspos-app. The mobile application developed for store operators is hosted within the Admin App repository at https://github.com/mirfan777/paspos-admin. The central orchestration and service layer is accessible through the Backend API repository located at https://github.com/dark-hermes/paspos-api. To protect the integrated store's operational privacy and prevent unauthorized access to the live system, the executable application packages (APKs) and administrative access credentials are restricted from public distribution. Testing credentials and sandboxed demo environments for system verification are not included in the repositories but can be provided by the corresponding author upon reasonable academic inquiry.

References

[1] "Data UMKM, Jumlah dan Pertumbuhan Usaha Mikro, Kecil, dan Menengah di Indonesia - UKMINDONESIA.ID." Accessed: May 04, 2026. [Online]. Available: https://ukmindonesia.id/baca-deskripsi-posts/data-umkm-jumlah-dan-pertumbuhan-usaha-mikro-kecil-dan-menengah-di-indonesia

[2] "Kementerian Koordinator Bidang Perekonomian Republik Indonesia." Accessed: May 04, 2026. [Online]. Available: https://www.ekon.go.id/publikasi/detail/6152/pemerintah-dorong-umkm-naik-kelas-tingkatkan-kontribusi-terhadap-ekspor-indonesia

[3] B. P. S. Indonesia, "Profil Industri Mikro dan Kecil 2024." Accessed: May 04, 2026. [Online]. Available: https://www.bps.go.id/id/publication/2025/09/16/a83f105e49377d0a7434e62a/profil-industri-mikro-dan-kecil-2024.html

[4] C. Aguedo, J. Espinoza, and A. Pacheco, "Improving Inventory Control Through a Web-Based System in a Retail Company," *F1000Research*, vol. 13, p. 252, Apr. 2024, doi: 10.12688/f1000research.145178.1.

[5] S. Kumar, B. E. Narkhede, and K. Jain, "Revisiting the warehouse research through an evolutionary lens: a review from 1990 to 2019," *Int. J. Prod. Res.*, vol. 59, no. 11, pp. 3470–3492, Jun. 2021, doi: 10.1080/00207543.2020.1867923.

[6] F. Kurniasari, E. D. Lestari, and H. Tannady, "Pursuing Long-Term Business Performance: Investigating the Effects of Financial and Technological Factors on Digital Adoption to Leverage SME Performance and Business Sustainability—Evidence from Indonesian SMEs in the Traditional Market," *Sustainability*, vol. 15, no. 16, p. 12668, Aug. 2023, doi: 10.3390/su151612668.

[7] M. Yokeshwaran and J. Murugachandravel, "Enhancement of business using e-commerce with point of sale," *Int. J. Health Sci.*, pp. 3250–3260, Apr. 2022, doi: 10.53730/ijhs.v6nS2.5811.

[8] E. C. Lewis, K. M. Harper, L. K. Poirier, and J. Gittelsohn, "Feasibility of using Mobile Point-of-Sale Technology in Baltimore City Corner Stores for Tracking Sales: A Brief Report," *J. Public Health Res.*, vol. 11, no. 1, p. jphr.2021.2607, Jan. 2022, doi: 10.4081/jphr.2021.2607.

[9] P. V. R. Pagidipalli, "Designing Scalable It Automation Through Microservices Architecture," *J. Int. Crisis Risk Commun. Res.*, pp. 315–324, Dec. 2025, doi: 10.63278/jicrcr.vi.3541.

[10] G. Quattrocchi, D. Cocco, S. Staffa, A. Margara, and G. Cugola, "Cromlech: Semi-Automated Monolith Decomposition Into Microservices," *IEEE Trans. Serv. Comput.*, vol. 17, no. 2, pp. 466–481, Mar. 2024, doi: 10.1109/TSC.2024.3354457.

[11] E. Kamau, T. Myllynen, S. D. Mustapha, G. O. Babatunde, and A. A. Alabi, "A Conceptual Model for Real-Time Data Synchronization in Multi-Cloud Environments," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 5, no. 1, pp. 1139–1150, 2024, doi: 10.54660/.IJMRGE.20247.5.1.1139-1150.

[12] C. Verwijs and D. Russo, "A Theory of Scrum Team Effectiveness," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, pp. 1–51, Jul. 2023, doi: 10.1145/3571849.

[13] A. Pangarso, K. Sisilia, R. Setyorini, Y. Peranginangin, and A. A. Awirya, "The long path to achieving green economy performance for micro small medium enterprise," *J. Innov. Entrep.*, vol. 11, no. 1, p. 16, Dec. 2022, doi: 10.1186/s13731-022-00209-4.

[14] W.-T. Lee and C.-H. Chen, "Agile Software Development and Reuse Approach with Scrum and Software Product Line Engineering," *Electronics*, vol. 12, no. 15, p. 3291, Jul. 2023, doi: 10.3390/electronics12153291.

[15] S. Merzouk, B. Jabir, A. Marzak, and N. Sael, "Best Agile method selection approach at workplace," *Bull. Electr. Eng. Inform.*, vol. 13, no. 3, pp. 1868–1876, Jun. 2024, doi: 10.11591/eei.v13i3.5782.

[16] A. Mishra and Y. I. Alzoubi, "Structured software development versus agile software development: a comparative analysis," *Int. J. Syst. Assur. Eng. Manag.*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.

[17] S. Sutono, A. Musrifah, and H. L. Fauzy, "METODE KNOWLEDGE BASED RECOMMENDATION DENGAN BACKWARD CHAINING UNTUK PERANCANGAN APLIKASI E-COMMERCE," *Media J. Inform.*, vol. 14, no. 2, p. 63, Dec. 2022, doi: 10.35194/mji.v14i2.2555.

[18] M. Hidalgo, F. Yanine, R. Paredes, J. Frez, and M. Solar, "What Is the Process? A Metamodel of the Requirements Elicitation Process Derived from a Systematic Literature Review," *Processes*, vol. 13, no. 1, p. 20, Dec. 2024, doi: 10.3390/pr13010020.

[19] K. Rączkowska-Gzowska and Anita Walkowiak-Gall, "What Should a Good Software Requirements Specification Include? Results of a Survey," *Found. Comput. Decis. Sci.*, vol. 48, no. 1, pp. 57–81, Mar. 2023, doi: 10.2478/fcds-2023-0004.

[20] R. Kanishka Jayalath, H. Ahmad, D. Goel, M. Shuja Syed, and F. Ullah, "Microservice Vulnerability Analysis: A Literature Review With Empirical Insights," *IEEE Access*, vol. 12, pp. 155168–155204, 2024, doi: 10.1109/ACCESS.2024.3481374.

[21] L. Nebiu Hyseni, A. Dermaku, and Z. Dika, "Evaluating Web Frameworks for Personal Learning Decision-Making: A Comparative Analysis," *Int. J. Comput. Exp. Sci. Eng.*, vol. 11, no. 2, May 2025, doi: 10.22399/ijcesen.1845.

[22] G. Jošt and V. Taneski, "State-of-the-Art Cross-Platform Mobile Application Development Frameworks: A Comparative Study of Market and Developer Trends," *Informatics*, vol. 12, no. 2, p. 45, Apr. 2025, doi: 10.3390/informatics12020045.

[23] S. Karlsson, R. Jongeling, A. Čaušević, and D. Sundmark, "Exploring behaviours of RESTful APIs in an industrial setting," *Softw. Qual. J.*, vol. 32, no. 3, pp. 1287–1324, Sep. 2024, doi: 10.1007/s11219-024-09686-0.

[24] S. Bimonte, E. Gallinucci, P. Marcel, and S. Rizzi, "Logical design of multi-model data warehouses," *Knowl. Inf. Syst.*, vol. 65, no. 3, pp. 1067–1103, Mar. 2023, doi: 10.1007/s10115-022-01788-0.

[25] I. Huh and J. Gim, "Exploration of Likert scale in terms of continuous variable with parametric statistical methods," *BMC Med. Res. Methodol.*, vol. 25, no. 1, p. 218, Sep. 2025, doi: 10.1186/s12874-025-02668-1.

[26] I. Kusmaryono, D. Wijayanti, and H. R. Maharani, "Number of Response Options, Reliability, Validity, and Potential Bias in the Use of the Likert Scale Education and Social Science Research: A Literature Review," *Int. J. Educ. Methodol.*, vol. 8, no. 4, pp. 625–637, Nov. 2022, doi: 10.12973/ijem.8.4.625.