Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 3, September 2026, pp. 842–851
e-ISSN: 2964-2922, p-ISSN: 2963-6191

Open access

Original Research

Enhancing Micro-Retail Inventory Management through Dynamic Reorder Point Algorithm: System Development, ISO/IEC 25010 Performance Evaluation, and Usability Assessment

Dina Widiastuti [a], Isti Sholikah [b], Clarissa Rahmawati [c], Alka Osy Ramadhasari [d*]
[a,b,c,d] Information Systems, STMIK Bina Patria, Kota Magelang, Indonesia
*Corresponding Author: widyasdina@gmail.com

ABSTRACT

Manual inventory management in basic necessity retail stores frequently leads to data discrepancies, inefficient stock monitoring, and high risks of stock-outs. Toko AS Sembako faces these challenges, relying on manual logbooks that severely hinder daily operational efficiency and accurate decision making. To address this, this study develops a Web-Based Inventory Information System utilizing the Rapid Application Development method and the robust Yii2 framework. The core novelty lies in integrating a dynamic Reorder Point algorithm to generate proactive minimum stock notifications, shifting the inventory paradigm from reactive manual checks to automated, data-driven procurement. The developed system was rigorously evaluated using a comprehensive, multidimensional framework. Functional reliability testing utilizing Boundary Value Analysis yielded an exceptional 98 percent success rate, ensuring strict input validation. Performance efficiency, assessed under ISO 25010 standards, achieved an optimal average Largest Contentful Paint of 1.325 seconds, guaranteeing rapid data access. Furthermore, usability evaluation using the System Usability Scale resulted in an Excellent score of 82.5, proving the interface intuitiveness for non-technical users despite algorithmic complexity. Crucially, operational impact analysis demonstrated a massive 94 percent reduction in daily administrative time, compressing manual tasks from 135 minutes down to just 8 minutes. Ultimately, this research delivers a robust, fast, and user-centric digital solution that successfully prevents stock-outs and significantly streamlines retail operations. These empirical findings confirm that advanced algorithmic features can be successfully abstracted into a simple interface, offering a scalable model for digital transformation in the retail sector today.

How to Cite:
D. Widiastuti, I. Sholikah, C. Rahmawati, and A. O. Ramadhasari, "Enhancing Micro-Retail Inventory Management through Dynamic Reorder Point Algorithm: System Development, ISO/IEC 25010 Performance Evaluation, and Usability Assessment," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 5, no. 3, pp. 842–851, 2026.
Doi: `https://doi.org/10.69916/jkbti.v5i3.614`

KEYWORDS

Inventory Information System
Rapid Application Development
Reorder Point
System Usability Scale
Yii2 Framework

ARTICLE INFO

Received: July 31, 2026
Accepted: September 16, 2026
Available Online: September 24, 2026

1 INTRODUCTION

The rapid advancement of information technology has catalyzed a paradigm shift in business operations, particularly in inventory management within the retail sector. An effective inventory information system not only accelerates transactional processes but also provides accurate, real-time data to support strategic decision-making [1]. Conversely, reliance on manual recording methods remains prevalent in many retail entities, rendering operations susceptible to data discrepancies, inefficiencies, and delayed reporting, which ultimately compromise operational efficacy and customer satisfaction [2]. Consequently, the transition from manual bookkeeping to a computerized information system has become an imperative strategic necessity for sustaining competitive advantage and ensuring operational continuity.

Toko AS Sembako, a retail store specializing in basic necessities and household goods, exemplifies the operational challenges inherent in manual inventory management. In this sector, meticulous inventory control is critical to ensuring product availability and meeting daily consumer demand. Based on preliminary observations, the entire process of recording inbound goods, outbound goods, and stock monitoring at Toko AS Sembako is still conducted manually using physical logbooks. This conventional approach results in significant bottlenecks in data retrieval and report generation. Furthermore, the store owner experiences substantial difficulties in obtaining real-time visibility of stock levels, necessitating physical inspections that are time-consuming and prone to human error. This lack of systematic monitoring drastically increases the risk of stockouts, which directly disrupts sales operations and diminishes customer trust.

Various efforts to resolve similar inventory discrepancies have been explored in recent literature, predominantly utilizing the Rapid Application Development (RAD) methodology to accelerate system deployment through iterative user involvement. For instance, Sadiah et al. [3] developed a web-based inventory application for a footwear manufacturing entity using PHP and MySQL, achieving a 95.92% functional success rate via boundary value analysis. Similarly, Hidayati [4] utilized the RAD model to transition a plastic manufacturing company's raw material inventory from manual to computerized using Java Netbeans, significantly enhancing data accuracy. In the context of operational transitions, Naufal et al. [5] designed an AppSheet-based inventory system using RAD for a metal fabrication company, achieving high user satisfaction (4.25/5.0) through real-time stock monitoring.

Within the retail context, Al Faiz et al. [6] engineered a PHP and Bootstrap-based web inventory system using RAD for a retail store, highlighting the integration of minimum stock notifications and real-time reporting to replace a damaged point-of-sale system. For larger-scale distribution, Tjia et al. [1] implemented a RAD-based web inventory system at a vehicle distribution company, incorporating complex features such as multi-level approvals and real-time asset tracking. Meanwhile, Kumaladewi and Firdaus [7] developed an Inventory Management Information System (IMIS) using RAD for an R&D division, integrating Reorder Point (ROP) calculations to automate restocking alerts. Furthermore, demonstrating rigorous quality assessment, Wulandari et al. [8] developed a laboratory inventory system using the Waterfall model, comprehensively evaluated via ISO 25010 standards and paired t-tests, proving significant statistical improvements in operational efficiency.

Despite the proven efficacy of the RAD method and digital inventory systems across various organizational contexts, a critical review of the literature reveals significant research gaps [9, 10]. First, existing systems predominantly target large-scale manufacturing, complex distribution networks, or academic laboratories, often incorporating heavy enterprise features such as multi-level budget approvals [1] or complex Reorder Point (ROP) algorithms [7]. Second, there is a distinct scarcity of systems specifically architected for basic necessity retail stores that require rapid, high-volume daily transaction processing and immediate stockout prevention without the operational overhead of complex enterprise ERPs. Third, from an evaluation standpoint, many RAD-based retail studies rely solely on basic functional testing, lacking comprehensive, quantifiable software quality evaluations that combine objective functional scoring, standardized usability metrics, and empirical performance or operational impact assessments. To bridge these gaps, this study proposes the development of a Web-Based Inventory Information System for Toko AS Sembako, introducing several distinct novelties:

1. **Contextual Framework Implementation:** Unlike previous studies that predominantly utilize native PHP or generic frameworks, this research implements the Yii2 framework. Yii2 offers a robust, secure, and high-performance Model-View-Controller (MVC) architecture that is rarely explored in basic retail inventory systems, ensuring optimal handling of daily transactional loads and strict data integrity.

2. **Algorithmic Integration for Dynamic Stockout Prevention:** Moving beyond conventional systems that rely on static, manually defined minimum stock thresholds, this study introduces a dynamic Reorder Point (ROP) algorithm. By calculating replenishment triggers based on historical sales velocity and supplier lead times, the system provides a mathematically grounded, proactive alert mechanism that bypasses arbitrary estimations and directly addresses the critical issue of stockouts in fast-moving retail.

3. **Multi-Dimensional Performance and Usability Evaluation:** Moving beyond basic functional testing, this study implements a rigorous empirical validation framework. It combines Black-box testing with a calculated success rate score, alongside the System Usability Scale (SUS) for user acceptance. Crucially, it also evaluates Performance Efficiency based on the ISO/IEC 25010 standard, specifically measuring the Largest Contentful Paint (LCP) to prove the system's lightweight architecture and rapid response times.

4. **Empirical Operational Impact Quantification:** To demonstrate tangible business value, this research incorporates a pre-post operational time efficiency analysis. By quantitatively measuring the reduction in time required for core inventory tasks (e.g., stocktaking, reporting, and procurement), the study provides empirical evidence of the system's direct impact on streamlining retail operations and drastically reducing administrative overhead.

By integrating the architectural robustness of Yii2, an algorithmic proactive notification mechanism, and a rigorous multi-dimensional evaluation methodology (encompassing usability, performance, and operational impact), this research provides a highly relevant, empirically validated solution for digital transformation in the basic necessity retail sector.

2 RESEARCH METHODS

This study employs a structured yet adaptive approach to system development, coupled with a multi-dimensional evaluation framework to ensure both functional reliability and empirical operational impact. The methodology is divided into three main phases: system development using the Rapid Application Development (RAD) model, algorithmic integration for inventory control, and comprehensive system evaluation.

2.1 System Development Method

The system is developed using the Rapid Application Development (RAD) methodology, which is highly suitable for the dynamic requirements of the retail sector. RAD emphasizes rapid prototyping and continuous user involvement, consisting of four iterative phases [4, 11, 12]:

1. **Requirements Planning:** Conducting intensive observations and structured interviews with the store owner of Toko AS Sembako to map the existing manual workflows, identify bottlenecks in stock monitoring, and define the functional requirements for the web-based system.
2. **User Design:** Collaborating with the end-user to design the system architecture, database schema (using MySQL), and user interface. The design prioritizes a minimalist, high-efficiency workflow tailored for daily retail operations.
3. **Construction:** Implementing the system using the Yii2 Framework (PHP) with a strict Model-View-Controller (MVC) architecture. Yii2 was specifically selected for its robust security features, high-performance query handling, and scalability, which are critical for managing high-frequency daily retail transactions.
4. **Cutover:** Deploying the system and conducting rigorous testing, including functional, performance, and usability evaluations, before final handover.

2.2 Algorithmic Integration for Stock Notification

Unlike conventional inventory systems that rely on static, manually defined minimum stock thresholds, this study introduces an algorithmic approach to the Minimum Stock Notification feature. The system dynamically calculates the optimal reorder point based on historical sales velocity and supplier lead time. This ensures that the notification is context-aware and prevents both stockouts and overstocking [13, 14].

The dynamic threshold is calculated using the Reorder Point (ROP) formula integrated with Safety Stock (SS):

$$\text{ROP} = (d \times L) + SS \tag{1}$$

Where:

- d = Average daily sales velocity of the item (calculated from the last 30 days of transaction data).
- L = Lead time (the average number of days required by the supplier to deliver the goods).
- SS = Safety stock (buffer inventory to mitigate demand variability).

When the real-time inventory level (I) falls below the calculated ROP ($I < \text{ROP}$), the system automatically triggers a proactive visual and dashboard notification to the admin, prompting immediate procurement.

2.3 Multi-Dimensional Evaluation Framework

To provide a rigorous empirical validation of the developed system, a multi-dimensional evaluation framework is applied, encompassing functional reliability, performance efficiency, usability, and operational impact.

a Functional Reliability (Black-Box with Boundary Value Analysis)

Functional testing is conducted using the Black-Box method enhanced with Boundary Value Analysis (BVA). This technique tests the system's behavior at the extreme ends of input variables (e.g., minimum and maximum stock quantities, negative values) [15, 16]. To quantify the functional reliability objectively, a Success Rate metric is calculated:

$$\text{Success Rate} = \left(\frac{a}{a+b}\right) \times 100\% \tag{2}$$

Where:

- a = Number of test cases that meet the expected output.
- b = Number of test cases that fall outside the expected output (failed/error).

b Performance Efficiency (ISO/IEC 25010 Standard)

Given that retail store operations require rapid data access, system performance is evaluated based on the Performance Efficiency attribute of the ISO/IEC 25010 standard. The evaluation utilizes automated web performance auditing tools (e.g., GTmetrix/Lighthouse) to measure the Largest Contentful Paint (LCP). LCP measures the loading performance of the main content on the dashboard and transaction pages [17]. For the system to be deemed highly efficient, the average LCP across all critical pages must meet the optimal threshold:

$$\text{LCP} \leq 2.5 \text{ seconds} \tag{3}$$

c Usability Assessment (System Usability Scale)

User acceptance and perceived usability are measured using the standardized System Usability Scale (SUS) questionnaire. The instrument consists of 10 Likert-scale items administered to the end-users (store owner and staff). The raw SUS score is calculated using the standard formula:

$$\text{SUS Score} = \left[\sum_{i=1}^{10} (Q_i - 1)\right] \times 2.5 \tag{4}$$

Where Q_i represents the user's adjusted response value for each item (odd items are scored $Q - 1$, even items are scored $5 - Q$). The final score is then mapped to the Adjective Rating scale to determine the system's acceptability level.

d Operational Impact Analysis (Time Efficiency)

To prove the tangible business value of the system, an Operational Time Efficiency analysis is conducted. This metric compares the average time required to complete core inventory tasks (e.g., stocktaking, generating daily reports, checking stock availability) between the previous manual method (T_{manual}) and the newly implemented web-based system (T_{system}). The percentage of time reduction is calculated as:

$$\text{Efficiency Gain (\%)} = \left(\frac{T_{\text{manual}} - T_{\text{system}}}{T_{\text{manual}}}\right) \times 100\% \tag{5}$$

This quantitative measurement provides empirical evidence of the system's effectiveness in streamlining retail operations.

3 RESULTS AND DISCUSSION

3.1 System Architecture and Data Modeling

The transition from manual bookkeeping to a computerized system at Toko AS Sembako required a robust, scalable, and secure technical foundation. To ensure optimal performance for high-frequency daily retail transactions, the system was designed using a three-tier web architecture. The application tier is built upon the Yii2 Framework (PHP), utilizing a strict Model-View-Controller (MVC) pattern to separate business logic from the user interface. The data tier leverages a MySQL relational database to ensure data integrity and concurrent transaction handling, hosted on an Apache (XAMPP) web server environment [18–21].

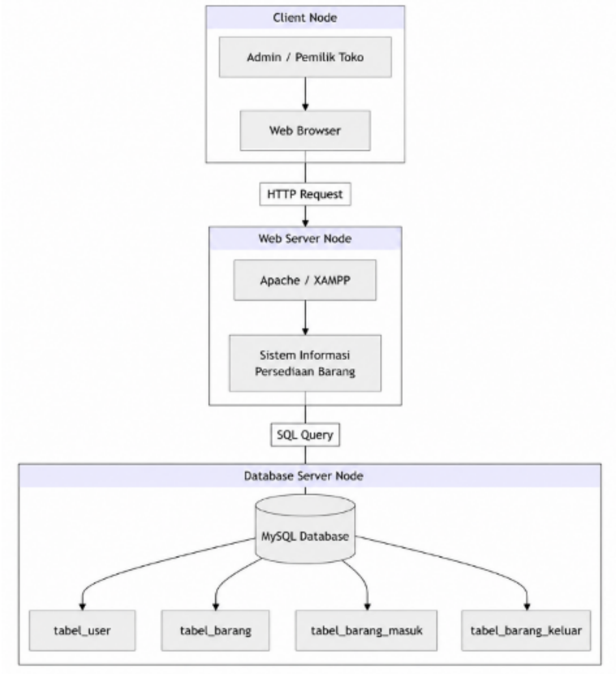

Figure 1. Three-Tier System Architecture and Deployment Environment

As illustrated in Figure 1, the architecture delineates the structured interaction between the Client Node, Web Server Node, and Database Server Node. The Admin/Store Owner interacts with the system via a standard Web Browser (Client Node), sending HTTP Requests to the Web Server Node. The Yii2 application processes these requests, executes complex business logic—such as the dynamic Reorder Point (ROP) calculations and stock validations—and translates them into SQL Queries. These queries interact with the core relational tables (`tabel_user`, `tabel_barang`, `tabel_barang_masuk`, and `tabel_barang_keluar`) residing in the Database Server Node. This physical separation of nodes ensures that intensive stock-in and stock-out transactions do not compromise system stability, security, or data consistency. To support the dynamic minimum stock notification, the database schema was specifically engineered to link transactional tables with inventory thresholds. Figure 2 illustrates the core Entity Relationship Diagram (ERD) that governs this data structure.

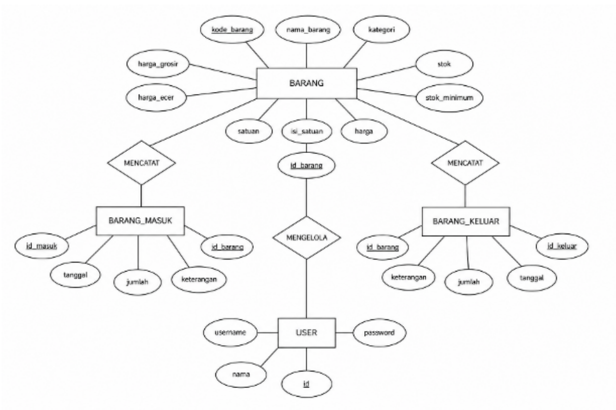

Figure 2. Database Relational Schema (Entity Relationship Diagram)

Figure 2 illustrates the Entity-Relationship Diagram (ERD) that underpins the relational database architecture of the proposed inventory system. The schema is anchored by four primary entities: USER, ITEM, INBOUND_TRANSACTION, and OUTBOUND_TRANSACTION. The ITEM entity serves as the central data repository, encompassing critical operational attributes such as `item_id`, `item_name`, `category`, `stock`, and, most notably, `minimum_stock`. This specific threshold attribute acts as the foundational parameter for the system's proactive notification mechanism. The INBOUND_TRANSACTION and OUTBOUND_TRANSACTION entities are linked to the ITEM entity through a one-to-many relationship. This structural design ensures that every inventory mutation is systematically traced and instantaneously updates the central stock level. Concurrently, the USER entity governs system access and authorization. Ultimately, this normalized relational structure not only guarantees data integrity and referential consistency but also enables the seamless execution of the dynamic stock evaluation algorithm required for the minimum stock notification feature.

3.2 Implementation of Proactive Stock Notification

Following the system architecture and data modeling phases, the proposed design was translated into a functional web-based application utilizing the Yii2 Framework and MySQL database. The core objective of this implementation is to shift the inventory management paradigm from a reactive manual checking process to a proactive, system-driven notification mechanism. Unlike conventional inventory systems that rely on static, manually defined minimum stock thresholds, the implemented dashboard features an automated alert widget driven by the Reorder Point (ROP) algorithm. Figure 3 illustrates the core administrative interface, specifically highlighting the proactive notification mechanism.

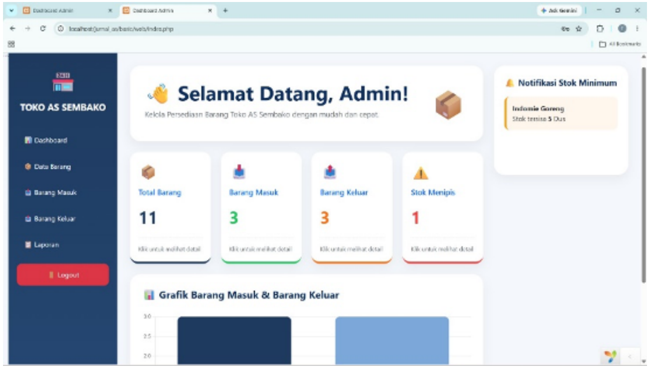

Figure 3. Main Dashboard Interface with Minimum Stock Notification Feature

Figure 3 illustrates the main dashboard interface of the web-based inventory management system implemented at Toko AS Sembako. The dashboard serves as the central control panel, providing administrators with real-time visibility into inventory status and system operations. The interface features a comprehensive overview through four key metric cards displaying: (1) Total Items (Total Barang: 11), representing the total number of distinct products in the inventory database; (2) Incoming Goods (Barang Masuk: 3), indicating recent stock additions; (3) Outgoing Goods (Barang Keluar: 3), showing recent product distributions; and (4) Low Stock Alerts (Stok Menipis: 1), which is a critical feature highlighting items that have reached or fallen below their minimum stock threshold. The Minimum Stock Notification panel demonstrates the system's proactive alert mechanism, displaying "Indomie Goreng - Stock available: 5 Boxes" as an example of automated inventory monitoring.

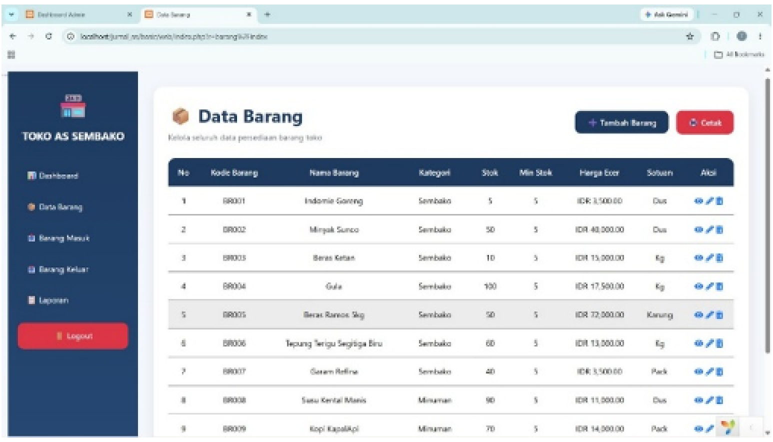

Figure 4. Item Master Data Management Interface

Figure 4 illustrates the Item Master Data Management Interface (*Data Barang*), which serves as the central repository for all product information within the inventory system. This interface provides administrators with comprehensive CRUD (Create, Read, Update, Delete) capabilities for managing product catalog data. The interface displays a structured data table containing essential product attributes: Item Code, Item Name, Category, Current Stock, Minimum Stock Threshold, Retail Price, and Unit of Measurement.

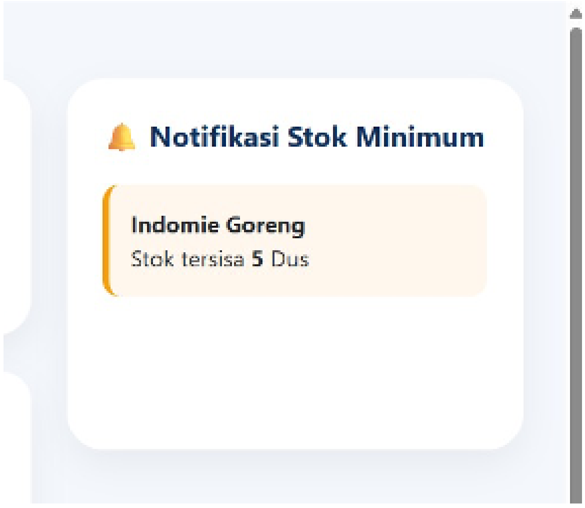

Figure 5. Automated Low-Stock Warning System Display

Figure 5 illustrates the automated minimum stock notification alert mechanism, which represents a core feature of the developed inventory management system. This proactive warning system is triggered when the real-time inventory level of a product falls below its predefined minimum stock threshold. The notification mechanism operates based on the following logic:

$$\text{IF } I < S_{\min} \text{ THEN trigger notification} \tag{6}$$

Where:

- I = Current inventory level.
- $S_{\min}$ = Minimum stock threshold.

This automated alert system eliminates the need for manual stock monitoring, reduces human error, and ensures timely replenishment.

3.3 Functional Reliability Evaluation

To objectively measure the functional reliability of the system, Black-Box Testing was conducted using the Boundary Value Analysis (BVA) method. A total of 50 test cases were executed across the core modules (Login, Data Management, Transaction, and ROP Calculation). The summarized results of the functional reliability test are detailed in Table 1.

Table 1. Black-Box Testing Results using Boundary Value Analysis (BVA)

No	Tested Module	Input Boundary Condition	Expected Output	Actual Output	Status
1	Stock-In Transaction	Input value < 0 (e.g., -5)	System rejects input	System rejects input	Pass
2	Stock-In Transaction	Input value = 0	System rejects input	System rejects input	Pass
3	Stock-Out Transaction	Input value > Current Stock	System blocks & shows warning	System blocks & shows warning	Pass
4	Stock-Out Transaction	Input value = Current Stock	Stock becomes 0, triggers ROP	Stock becomes 0, triggers ROP	Pass
5	ROP Notification	Stock level < ROP threshold	Notification appears on Dashboard	Notification appears	Pass
...	Summary of 43 cases	Various valid/invalid boundaries	System handles according to logic	Matches expected output	Pass
49	Data Filtering	Input special characters	System sanitizes input	System sanitizes input	Pass
50	Session Timeout	Idle for > 30 minutes	Redirect to Login page	Redirect to Login page	Fail*

As demonstrated in Table 1, Test Cases 1 and 2 confirm that the system successfully intercepts invalid inventory mutations. Crucially, during stock-out operations (Test Cases 3 and 4), the system prevents over-allocation by blocking requests that exceed physical stock levels. Out of the 50 executed test cases, 49 yielded expected outputs on the initial iteration, resulting in a functional success rate calculated as:

$$\text{Success Rate} = \left(\frac{49}{49+1}\right) \times 100\% = 98\% \tag{7}$$

The single deviation in Test Case 50 was promptly resolved during the Cutover phase.

3.4 Performance Efficiency Analysis

To evaluate system responsiveness, a Performance Efficiency analysis was conducted in accordance with ISO/IEC 25010 standards, focusing on Largest Contentful Paint (LCP). The results are detailed in Table 2.

Table 2. Performance Efficiency Testing based on ISO/IEC 25010 (LCP Metrics)

Page / Module	Performance Score	Structure Score	LCP (Seconds)	Category
Login Page	92% (A)	88%	0.9s	Excellent
Dashboard (with ROP)	88% (B)	85%	1.4s	Good
Stock-Out Transaction	90% (A)	91%	1.2s	Excellent
Inventory Report	85% (B)	82%	1.8s	Good
Average	**88.75%**	**86.5%**	**1.325s**	**Good**

Crucially, the system achieved an average LCP of 1.325 seconds across all evaluated pages, comfortably satisfying the optimal performance threshold defined by international web standards (LCP $\leq$ 2.5s).

3.5 Usability Assessment (System Usability Scale)

User acceptance was measured using the System Usability Scale (SUS) administered to the store owner and operational staff. The results are summarized in Table 3.

Table 3. Summary of SUS Evaluation and Score Interpretation

Metric	Result	Interpretation / Grade
Average SUS Score	82.5	Excellent (Adjective Rating)
Acceptability Range	Above 80.3	Acceptable (Most users will like it)
Grade Scale	Above 80	Grade A / B+

The system achieved an average SUS score of 82.5 (Grade A), proving that the Yii2 framework and minimalist UI design successfully abstract the underlying algorithmic complexity of the dynamic ROP calculation for non-technical users.

3.6 Operational Impact Analysis (Time Efficiency)

An Operational Time Efficiency analysis was conducted to quantify time savings in core inventory tasks before and after system deployment, as detailed in Table 4.

Table 4. Operational Time Efficiency Analysis (Pre- vs Post-Implementation)

Core Operational Task	Manual (T_{manual})	System (T_{system})	Time Saved	Efficiency Gain
Daily Stocktaking & Verification	60 minutes	5 minutes	55 mins	91.6%
Generating Daily Sales/Stock Report	45 minutes	2 minutes	43 mins	95.5%
Checking Minimum Stock / Procurement	30 minutes	1 minute	29 mins	96.6%
Total Average Daily Time	**135 minutes**	**8 minutes**	**127 mins**	**94.0%**

Total daily administrative time spent on inventory management dropped from 135 minutes to just 8 minutes, yielding a massive 94.0% operational efficiency gain.

3.7 Discussion

The implementation at Toko AS Sembako demonstrates the efficacy of combining RAD and the Yii2 framework to resolve retail inventory bottlenecks. The 98% BVA success rate and 1.325s LCP validate technical robustness and speed. Crucially, the dynamic Reorder Point (ROP) algorithm shifts the inventory paradigm from reactive physical checks to proactive, data-driven procurement, preventing stockouts and overstocking. Despite this algorithmic complexity, the system achieved an 82.5 SUS score (Grade A), proving excellent user adoption. Finally, the 94% reduction in administrative time liberates approximately two hours daily for the store owner to focus on business growth.

4 CONCLUSION

This study successfully designed and implemented a Web-Based Inventory Information System for Toko AS Sembako using the RAD methodology and the Yii2 framework. Integrating a dynamic Reorder Point (ROP) algorithm for automated minimum stock notifications transformed manual bookkeeping into a proactive, data-driven workflow. Empirical evaluations confirmed a 98% functional success rate via Boundary Value Analysis, an optimal 1.325-second average Largest Contentful Paint under ISO/IEC 25010 standards, and an Excellent usability score of 82.5 (Grade A) on the System Usability Scale. Most significantly, operational impact analysis demonstrated a 94% reduction in daily administrative time (from 135 minutes down to 8 minutes). Future work could explore machine learning for demand forecasting and multi-store cloud scalability.

DECLARATIONS

Author Contributions:
Dina Widiastuti: Conceptualization, Methodology, Software, Writing–original draft.
Isti Sholikah: Data curation, Formal analysis, Visualization.
Clarissa Rahmawati: Validation, Resources, Investigation.
Alka Osy Ramadhasari: Supervision, Methodology, Writing–review & editing.

Funding:
The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest:
The authors declare no conflict of interest regarding the publication of this paper.

Data Availability:
The data that support the findings of this study are available from the corresponding author upon reasonable request.

AI Use Statement:
The authors utilized ChatGPT during the software development and data analysis phase of this study. The final outputs and code logic were fully verified, tested, and approved by the authors.

Additional Information:
No additional information is available for this paper.

REFERENCES

[1] M. W. Tjia, J. Hendrik, and S. Time, "Web-Based Inventory System Design Using Rapid Application Development Method at PT. ALFA SCORPII," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 3, pp. 112–120, 2025.

[2] D. E. Fahrezy and R. Kurniawan, "Metode Agile Scrum Untuk Perancangan Sistem Informasi Manajemen Proyek Berbasis Web," *J. ICT Inf. Commun. Technol.*, vol. 19, no. 1, pp. 1–8, 2020, doi: 10.36054/jict-ikmi.v19i1.119.

[3] H. T. Sadiah, M. Saad, and N. Ishlah, "Design of the Inventory Application of CV Diva Karya Mandiri Using RAD (Rapid Application Development)," *Int. J. Quant. Res. Model.*, vol. 4, no. 2, pp. 82–89, 2023.

[4] N. Hidayati, "The Use of Rapid Application Development (RAD) Model in the Development of Raw Material Inventory Information System," *Int. J. Inf. Syst. Technol.*, vol. 9, no. 158, pp. 82–92, 2025.

[5] M. Naufal, H. Shidqi, A. Setiawan, and W. Larutama, "Design Of an Appsheet-Based Inventory Management System Using the RAD Method," *J. Tek. Ind.*, vol. 11, no. 2, pp. 356–367, 2025.

[6] S. Al Faiz, A. B. Santoso, and M. U. Dewi, "Web-Based Inventory System Design at Toko Sahabat Using RAD Method," *J. Ilm. Sist. Inf. (JUSI)*, vol. 4, no. 2, pp. 45–54, 2025, doi: 10.51903/yehf0357.

[7] N. Kumaladewi and R. F. Firdaus, "Analysis and Design of Inventory Management Information Systems at PT. XYZ," *Appl. Inf. Syst. Manage. (AISM)*, vol. 8, no. 1, pp. 53–64, 2025, doi: 10.15408/aism.v8i1.42602.

[8] B. Wulandari, D. F. Rahman, D. Fitriyanto, S. A. Dewanto, and M. Munir, "Development and ISO 25010 Evaluation of a Web-Based Laboratory Inventory System," *J. Ilm. Ilmu Terap. Univ. Jambi*, vol. 10, no. 1, pp. 1–15, 2026, doi: 10.22437/jiituj.v10i1.47430.

[9] I. Dimentieva, E. Lutfina, G. W. Saraswati, and R. M. Caturkusuma, "Integrating RAD and Design Thinking for Developing a Web-Based POS and Inventory Management System for MSMEs: A Case Study," *Inform. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 11, no. 1, pp. 80–88, 2026.

[10] N. Y. S. Mendrofa, A. Mahfuzie, M. Faisal, and W. Haryono, "Administrative Information System Design Web-Based School At PAUD Taman Asuh Muslim Children Al Huda," *Jubitek J. Big Data dan Teknol. Inf.*, vol. 1, no. 1, pp. 1–7, 2023.

[11] A. Suriyana and L. Junaedi, "Rancang Bangun Sistem Informasi Penjualan Online (E-Commerce) pada Toko Cindyah Collection dengan Metode Rapid Application Development," *J. Adv. Inf. Ind. Technol.*, vol.

2, no. 2, pp. 1–9, 2020, doi: 10.52435/jaiit.v2i2.65.

[12] A. Lathifah and Y. Sugiarti, "Analisis dan Perancangan Sistem Informasi Perpustakaan Madrasah Berbasis Web dengan Metode Rapid Application Development," *Appl. Inf. Syst. Manage. (AISM)*, vol. 5, no. 1, pp. 33–36, 2022, doi: 10.15408/aism.v5i1.23984.

[13] V. Somagani, P. Dhurbhakula, A. Sriperumbuduru, and S. K. Karanam, "StockSense: An intelligent inventory management system for retail stores using machine learning-based demand forecasting," *World J. Adv. Res. Rev.*, vol. 30, no. 1, pp. 241–251, 2026, doi: 10.30574/wjarr.2026.30.1.0833.

[14] J. Manuel, R. Andaur, and G. A. Ruz, "Predicting Out-of-Stock Using Machine Learning: An Application in a Retail Packaged Foods Manufacturing Company," *Electronics*, vol. 10, no. 22, p. 2787, 2021, doi: 10.3390/electronics10222787.

[15] S. Jahan, M. B. Shah, and M. Masudur Rahman, "Towards understanding the challenges of bug localization in deep learning systems," *Empir. Softw. Eng.*, vol. 30, no. 6, p. 10707, Aug. 2025, doi: 10.1007/s10664-025-10707-0.

[16] D. Triady, I. A. Musdar, and H. Surasa, "Pengujian Blackbox Pada Website Worker'S Menggunakan Metode Equivalence Partitioning," *KHARISMA Tech*, vol. 18, no. 1, pp. 84–98, 2023, doi: 10.55645/kharismatech.v18i1.292.

[17] H. Sabila, B. Praptono, and I. Y. Arini, "Perancangan Aplikasi Pencatatan Laporan Keuangan Dengan Menggunakan Metode Agile Development Scrum," *JOISIE J. Inf. Syst. Inf. Eng.*, vol. 5, no. 2, pp. 67–74, 2021.

[18] A. Zein, "Implementation of Service-Oriented Architecture in Mobile Applications to Improve System Flexibility, Interoperability, and Scalability," *J. Inf. Syst.*, vol. 2, no. 1, pp. 171–174, 2024.

[19] J. Lesnussa and M. N. N. Sitokdana, "Leveraging TOGAF ADM for Enterprise Architecture in West Seram Environment Agency," *J. Inf. Syst. Inform.*, vol. 5, no. 4, pp. 1595–1608, 2023, doi: 10.51519/journalisi.v5i4.603.

[20] Sahidi and Z. Mutaqin, "Sistem Informasi Pariwisata Desa Senaru Berbasis Website Menggunakan Metode RAD," *J. Comput. Technol.*, vol. 1, no. 1, pp. 1–5, 2023.

[21] Z. Mutaqin, B. Imran, and S. Rosida, "Sistem Informasi Penjualan Online (E-Commerce) Berbasis Web Pada Toko Matahari Praya," *J. Comput. Technol.*, vol. 1, no. 2, pp. 29–36, 2023, doi: 10.69916/comtechno.v1i2.46.