



Design and Development of a Web-Based Inventory Information System Using the Laravel Framework at Pelalawan Pet Shop

Muhammad Raisyan Al Farras¹

¹Program Studi D3 Manajemen Informatika (PSDKU Pelalawan), Jurusan Teknologi Informasi, Politeknik Negeri Padang

Email: m.raisyanalfarras@gmail.com¹

Received Sept 17th 2026; Revised Sept 30th 2026; Accepted Sept 30th 2026; Available Online Sept 30th 2026

Corresponding Author: Muhammad Raisyan Al Farras

Copyright © 2026 by Authors, Published by CV. Alaztha Media Publikasi.

ABSTRACT

Pelalawan Pet Shop still relies on manual, notebook-based recording to manage its inventory, which frequently causes inaccurate stock data, lost records, and delayed stock information, especially when customer demand rises. This study aims to design and develop a web-based inventory information system capable of automating the recording of incoming and outgoing goods and the generation of stock reports at Pelalawan Pet Shop. The study used a Research and Development approach with the Waterfall development model, covering requirement analysis, design, implementation, testing, and maintenance. The system was evaluated through Black Box Testing, a User Acceptance Test (UAT) with six respondents, and a Nielsen Usability Heuristic evaluation conducted by three evaluators. All 39 Black Box Testing scenarios were declared valid. The UAT showed a 100% positive response rate (combined Strongly Agree and Agree) with an average achievement score of 94.17%, although the small number of respondents ($n = 6$) limits the generalizability of this finding. The heuristic evaluation showed a low average severity rating (cosmetic to minor range), indicating that the interface is relatively easy to use, although the help documentation still needs improvement. The developed web-based inventory information system speeds up the recording of goods, minimizes manual recording errors, and provides accurate stock reports that support inventory decision-making at Pelalawan Pet Shop.

Keywords: Black Box; Inventory Information System; Laravel; Usability Heuristic; UAT

1. INTRODUCTION

Small and medium-sized retail businesses, including veterinary-related businesses such as pet shops, rely heavily on the availability and accuracy of inventory data to maintain continuous customer service. Manual inventory recording is generally prone to input errors, data loss, and delays in providing stock information, making it difficult for business owners to make decisions about procurement and inventory management.

Pelalawan Pet Shop is a veterinary-related business located in Pangkalan Kerinci, Pelalawan Regency. Based on interviews with the veterinarian working at the business, inventory, medical equipment, and pet-care products are still managed through simple notebook-based records. This situation creates several problems, including inaccurate stock data, the risk of lost records, and delays in accessing inventory information. These problems become more pronounced when customer demand increases, making service processes less efficient.

Various previous studies have developed web-based inventory information systems to address similar problems in different business settings. Anugrah et al. designed a web-based inventory system to optimize inventory management at PT Bumi Daya Plaza, focusing on real-time stock monitoring and automated reorder processes [1]. Pinatih and Hidayatullah developed a website-based inventory system using the Waterfall model, emphasizing systematic development stages [2], while Widiarta et al.

developed an inventory system using an Agile approach in an electronics store case study [3]. Renaldy and Rustam designed a web-based inventory information system for the warehouse of PT Spin Warriors, focusing on recording incoming and outgoing goods [10], while Hamidy examined inventory management optimization using the First In First Out method for more efficient stock control [11]. Other studies by Rusdianto and Herdiana, as well as Sabiila and Cahyo Utomo, showed that the Laravel framework can be used to build structured and maintainable web-based inventory systems [5].

A review of these studies shows that most developed systems focus on general recording of incoming and outgoing goods, without accommodating the specific needs of pet-product retail businesses involving three user roles with different levels of authority, namely Owner, Admin, and Staff. Few have integrated role-based access control (roles and permissions), low-stock notifications, and expired-product monitoring into a single, comprehensive system. This gap provides the basis for this study to design a system that better fits the operational characteristics of Pelalawan Pet Shop.

Based on the description above, this study aims to design and develop a web-based inventory information system using the Laravel framework at Pelalawan Pet Shop. The system is intended to automate the recording of incoming and outgoing goods, minimize manual recording errors, and provide accurate inventory reports that can be accessed quickly by users according to their respective roles.

The novelty of this study lies in integrating role-based access management (Owner, Admin, and Staff), low-stock notifications, and expired-product monitoring into a web-based inventory system specifically developed for the characteristics of pet-product retail businesses. The system was empirically validated through three testing methods: Black Box Testing, User Acceptance Testing, and an evaluation based on Nielsen's 10 Usability Heuristics. Thus, the study explains not only the development process but also the system's acceptance and suitability from the perspective of end users.

2. RESEARCH METHODS

This study employed a Research and Development (R&D) approach using a case study strategy at Pelalawan Pet Shop. The software development model used was Waterfall, selected because its systematic and sequential stages facilitate planning, development, and quality control at each phase of system development [4]. The stages of the Waterfall method used in this study are as follows:

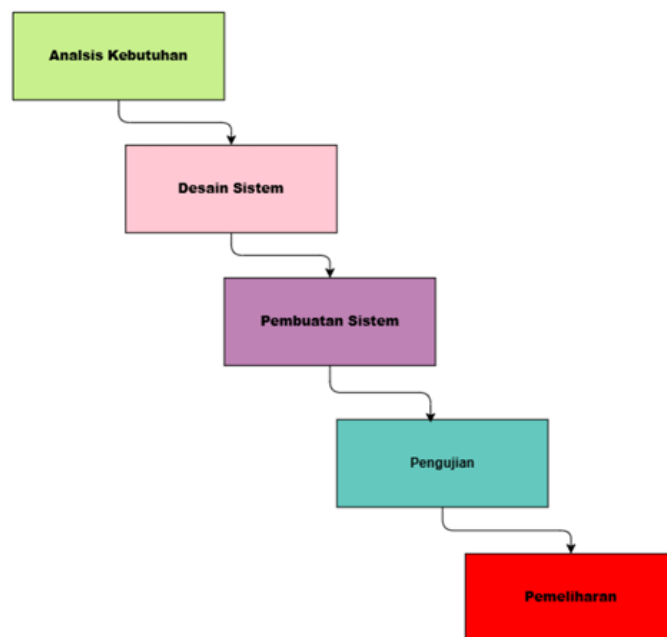


Figure 1. Waterfall Method

1. Requirements Analysis. Data were collected through observation and interviews at Pelalawan Pet Shop to obtain information about the current inventory management process, required data types,

workflows, and existing constraints. The results of this requirements analysis served as the basis for system design to ensure that it met user needs.

2. **System Design.** System design included database design for inventory data storage, user interface (UI/UX) design using Laravel and Blade Template, and the use of technologies such as PHP and Laravel. The system workflow was also designed to facilitate digital inventory management by the Admin.
3. **System Development.** The system was coded using Visual Studio Code and the Laravel framework based on the completed system design. This stage included developing features for data entry, editing, deletion, and searching.
4. **Testing.** After the features had been developed, the system was tested to ensure that they functioned properly and according to their intended purposes, and that the system operated stably without errors before use.
5. **Maintenance.** System maintenance involved fixing any errors or bugs and making improvements and further developments to add features in the future, so that the application could continue to support the smooth operation of Pelalawan Pet Shop.

The research object was the inventory management business process at Pelalawan Pet Shop, including item, category, attribute, and supplier data, as well as incoming and outgoing stock transactions. System users were grouped into three actors: the Owner, who has full authority over reports and access control; the Admin, who manages master data and transactions; and the Staff, who records daily transactions.

Data analysis during the testing stage was conducted in three ways. First, Black Box Testing was used to verify system outputs against the inputs provided for each module without examining the program code structure. Second, User Acceptance Testing (UAT) was conducted with six users using a five-point Likert-scale questionnaire consisting of 12 statements grouped into several system-acceptance variables. The percentage was calculated using the formula $P = (S / \text{Ideal Score}) \times 100\%$, where P is the percentage, S is the total score of respondents' answers, and the Ideal Score is the maximum score multiplied by the sample size. Third, usability was evaluated using Nielsen's 10 Usability Heuristics to assess the ease of use of the system interface in terms of information clarity and user support [9]. A complete summary of the three testing results is presented in the Results and Discussion section.

3. RESULTS AND DISCUSSION

3.1 Initial Identification

Based on direct observation and interviews with the veterinarian working at Pelalawan Pet Shop, inventory management—from recording incoming and outgoing goods to stock reporting—was found to still be performed manually using notebooks. This process involved three user roles with different needs: the Owner, who requires comprehensive reports for supervision and decision-making; the Admin, who is responsible for managing item master data, transactions, and users; and the Staff, who records daily stock transactions.

The identification process found that manual recording often resulted in stock data that did not match the physical inventory, records that were easily damaged or lost, and lengthy searches for specific item data, particularly when many customers arrived. In addition, there was no automatic alert mechanism when pet food or medicines were running low or when products were approaching their expiration dates, creating a considerable risk of losses due to damaged or expired goods. These findings formed the basis for defining the system requirements in the next stage.

3.2 Requirements Design

3.2.1 Functional Requirements

The functional requirements of the inventory information system were grouped into three main aspects: system inputs, processes, and outputs.

- a. Input requirements: item data (name, category, type, supplier, purchase price, selling price, unit, weight, photo, and description); category data and additional item attributes; supplier data; incoming and outgoing stock transaction data, including expiration dates; stock-taking data; and user, role, and permission data.
- b. Process requirements: validation of input completeness in each form; automatic increases and decreases in stock when incoming or outgoing stock transactions are saved; processing of expired goods to automatically reduce stock; reconciliation of system stock with physical stock in the stock-taking module; and calculation of report summaries based on the period selected by the user.
- c. Output requirements: a dashboard displaying stock summaries and low-stock notifications; lists and details of items, suppliers, categories, and attributes; incoming and outgoing stock reports; expired-goods reports, including estimated losses; and comprehensive inventory reports that can be exported to PDF and Excel formats.

3.2.2 Non-Functional Requirements

- a. Software: Windows 11 Pro 64-bit operating system; PHP programming language with the Laravel framework; MySQL database; Laragon as the local server; Git for version control; Visual Studio Code as the text editor; Draw.io for diagram design; and Google Chrome as the testing browser.
- b. Hardware: a laptop with a processor equivalent to a 6th-generation Intel Core i5, 8 GB RAM, and 512 GB SSD storage, used for both system development and testing.

3.2.3 System Architecture

Architecturally, the system was developed using the Model-View-Controller (MVC) pattern provided by the Laravel framework, allowing business logic, views, and request controllers to be clearly separated and facilitating future system maintenance. The three user roles—Staff, Admin, and Owner—access different modules through a web interface according to their respective permissions. All data, however, are stored in a centralized MySQL database consisting of twelve main tables, including users, roles, permissions, barang, kategori, atribut, supplier, stok_masuk, stok_keluar, and laporan. Figure 2 presents the architectural design of the inventory information system at Pelalawan Pet Shop.

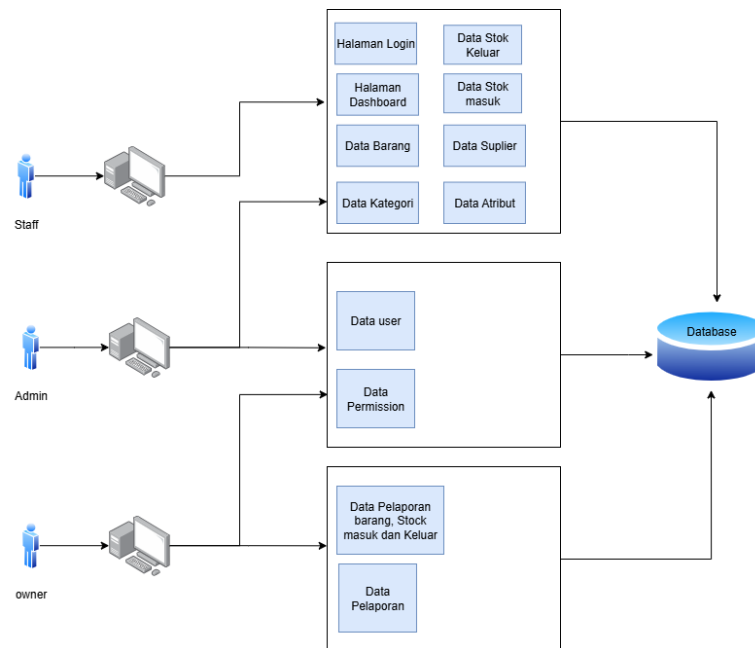


Figure 2. Inventory Information System Architecture

3.3 System Design

This study used Unified Modeling Language (UML) visual modeling, particularly use case diagrams and class diagrams, for system design. These diagrams were selected because they can illustrate interactions between actors and the system, as well as relational data structures [7]. The design results served as a reference for writing the program code using the Laravel framework.

3.3.1 Use Case Diagram

The use case diagram illustrates the interactions between actors and the developed inventory information system. The system involves three main actors—Owner, Admin, and Staff—with their respective authorities, as summarized in Table 1. Figure 3 shows the use case diagram of the inventory information system at Pelalawan Pet Shop, covering login, management of item data, categories, attributes, suppliers, incoming stock, outgoing stock, stock taking, users, roles and permissions, as well as inventory and expired-goods reports.

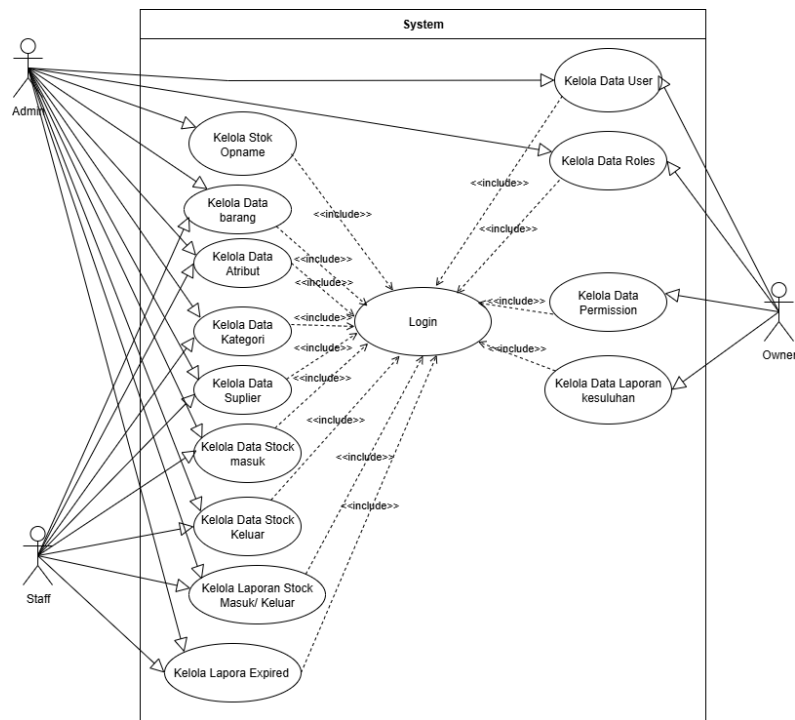


Figure 3. Use Case Diagram

Table 1. Use Case Diagram Actor Definitions

No.	Actor Name	Description
1	Owner	The actor responsible for supervision, overall reports, and control of roles and permissions.
2	Admin	The actor with full access to manage item master data, transactions, users, roles, and reports.
3	Staff	The actor responsible for recording item data, incoming stock, outgoing stock, and operational reports.

3.3.2 Class Diagram

The class diagram represents the structure and relationships among the classes involved in the inventory information system, including attributes, operations, and relationships among classes such as Item, Category, Attribute, Supplier, IncomingStock, OutgoingStock, User, and Role, along with their respective controllers. This diagram was used as a reference for implementing an integrated and well-structured system, as shown in Figure 4.

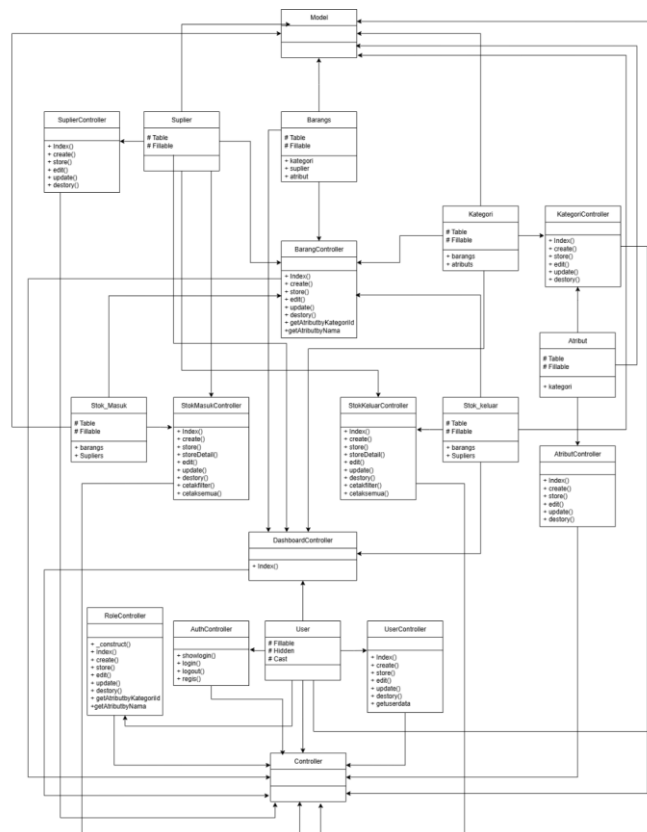


Figure 4. Class Diagram

3.4 System Implementation

Implementation involved translating the design results from the previous stage into program code using the Laravel framework. The login page in Figure 5 is used by the Owner, Admin, and Staff to access the system using their registered email addresses and passwords, with access rights and menu displays differing according to their respective roles.

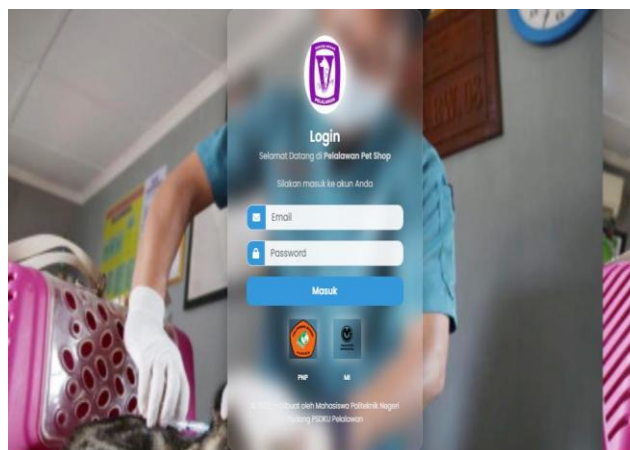


Figure 5. Login Page

After logging in successfully, users are directed to the dashboard page, as shown in Figure 6. The dashboard displays summaries of the total number of items, suppliers, and categories; automatic notifications when items are running low; and a list of low-stock items that require immediate follow-up.

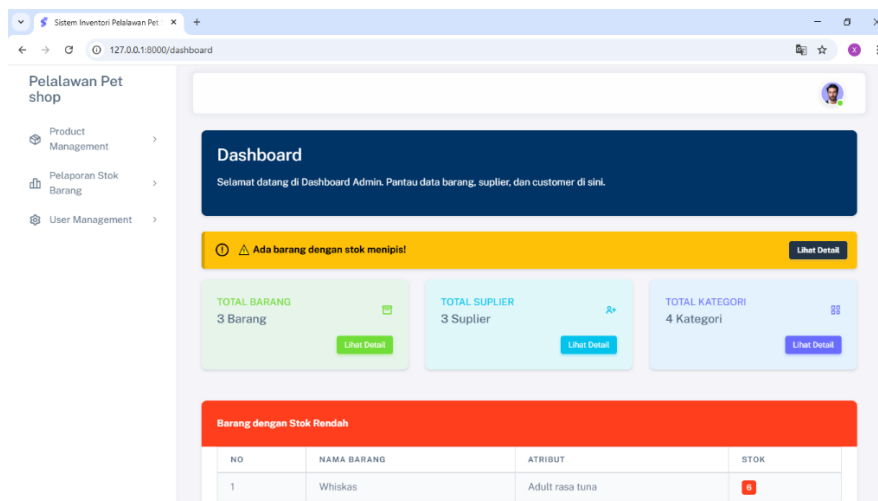


Figure 6. Dashboard Page

The expired-goods management module in Figure 7 allows the Admin to process goods that have passed their expiration dates, automatically reducing stock. It also includes filters by month and year and a report-printing feature. This module addresses the need identified during the initial analysis for monitoring expired goods, a feature that was previously unavailable in the manual process.

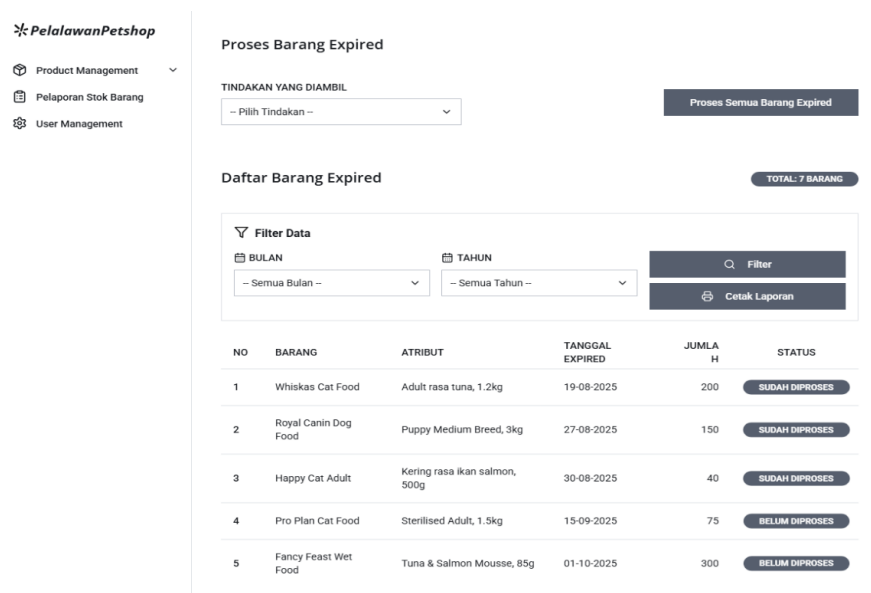


Figure 7. Expired-Goods Processing Page

In addition to the modules described above, the system also provides modules for recording incoming and outgoing stock that are directly linked to item data, a stock-taking module for reconciling system stock with physical stock, and an inventory reporting module that can be filtered by a specific period and exported to PDF or Excel.

3.5 System Testing

3.5.1 Black Box Testing

Black Box Testing was conducted to verify that the system outputs matched the expected results for each module without examining the program code structure. Testing covered twelve modules with a total of 39 scenarios, including login, mandatory-input validation, data storage, data modification and deletion, and report export. Table 2 summarizes the test results for each module.

Table 2. Summary of Black Box Testing Results

No.	Testing Module	No. of Scenarios	Conclusion
1	Login for all actors (Owner, Admin, Staff)	2	Valid
2	User data management	5	Valid
3	Role and permission management	4	Valid
4	Item data management	5	Valid
5	Supplier data management	4	Valid
6	Category data management	3	Valid
7	Attribute data management	3	Valid
8	Incoming stock recording	3	Valid
9	Outgoing stock recording	3	Valid
10	Inventory reports	3	Valid
11	Report export by Admin/Staff	2	Valid
12	Inventory report export by Owner	2	Valid

The test results showed that all 39 scenarios across the twelve modules were valid. This means that the system outputs were consistent with the expected results for each test case, including negative scenarios, such as leaving mandatory fields blank, which were handled appropriately with validation messages.

3.5.2 User Acceptance Test (UAT)

UAT was conducted with six user respondents using a five-point Likert-scale questionnaire (SA = Strongly Agree, A = Agree, N = Neutral, D = Disagree, SD = Strongly Disagree). The questionnaire consisted of 12 statements covering the user interface, ease of data entry, stock information accuracy, low-stock notifications, report suitability, system stability, and overall system benefits. The percentage for each statement was calculated using the formula $P = (S / \text{Ideal Score}) \times 100\%$, where the Ideal Score = the maximum score (5) multiplied by the number of respondents (6) = 30. Table 3 presents the complete UAT questionnaire results.

Table 3. User Acceptance Test (UAT) Results

No.	Statement	SA	A	N	D	SD	Score	Percentage
1	The application interface is easy to understand and comfortable to use.	4	2	0	0	0	28	93.3%
2	The menus and features in the application are well organized and easy to access.	4	2	0	0	0	28	93.3%
3	The login process runs smoothly and securely.	5	1	0	0	0	29	96.7%
4	User data can be entered easily.	4	2	0	0	0	28	93.3%
5	Item master data can be entered easily and accurately.	4	2	0	0	0	28	93.3%

No.	Statement	SA	A	N	D	SD	Score	Percentage
6	Incoming/outgoing stock data can be entered easily and quickly.	4	2	0	0	0	28	93.3%
7	The system displays item stock lists accurately.	4	2	0	0	0	28	93.3%
8	The system provides clear information/notifications when stock is running low.	4	2	0	0	0	28	93.3%
9	Item, incoming stock, outgoing stock, and inventory reports meet user needs.	4	2	0	0	0	28	93.3%
10	The system operates stably without disruptive errors.	5	1	0	0	0	29	96.7%
11	The application helps accelerate inventory recording at the Pet Shop.	4	2	0	0	0	28	93.3%
12	Overall, the application is useful, suitable for use, and meets requirements.	5	1	0	0	0	29	96.7%

A total of 72 responses (12 statements $\times$ 6 respondents) were recorded: 51 responses were in the Strongly Agree category (70.83%) and 21 in the Agree category (29.17%), while the Neutral, Disagree, and Strongly Disagree categories each received 0 responses (0%). Thus, all respondents gave positive assessments of the system (Strongly Agree and Agree combined), resulting in a 100% positive response rate and an average achievement percentage of 94.17% across statements. This indicates that the developed Inventory Information System was considered suitable for use, beneficial, and aligned with the needs of users at Pelalawan Pet Shop. The highest scores were for login performance, system stability, and overall system usefulness (96.7%), while the user interface and data-entry aspects received slightly lower scores (93.3%), which were still classified as very good.

3.5.3 Nielsen Usability Heuristic Evaluation

The usability evaluation was conducted independently by three evaluators (the researcher and two academic supervisors) using Nielsen's Usability Heuristics method, with a severity rating scale from 0 (not a problem) to 4 (usability catastrophe). Table 4 presents the average severity rating from the three evaluators for each of the ten heuristic principles, along with the category of the findings.

Table 4. Nielsen Usability Heuristic Evaluation Results

No.	Nielsen Heuristic Principle	Average Severity (0–4)	Finding Category
1	Visibility of system status	1.7	Cosmetic–Minor
2	Match between system and the real world	0.3	Cosmetic
3	User control and freedom	2.0	Minor
4	Consistency and standards	0.3	Cosmetic
5	Error prevention	0.7	Cosmetic
6	Recognition rather than recall	0.3	Cosmetic
7	Flexibility and efficiency of use	0.7	Cosmetic
8	Aesthetic and minimalist design	0.3	Cosmetic
9	Help users recognize and recover from errors	0.7	Cosmetic

No.	Nielsen Heuristic Principle	Average Severity (0–4)	Finding Category
10	Help and documentation	2.3	Minor

The overall average severity rating was 0.93, within the cosmetic-to-minor range, indicating that the system interface generally did not create significant barriers for users. The two principles with the highest severity ratings were help and documentation (2.3) and user control and freedom (2.0), indicating the need for additional documentation or help tooltips and an undo/restore feature for data deletion. Other principles, such as interface consistency, input-error prevention, and matching the users' mental models (match between system and the real world), received low severity ratings, indicating that these aspects had been adequately addressed.

3.6 Discussion

The evaluation using Nielsen's 10 Usability Heuristics showed that the system interface generally fulfilled ease-of-use principles, including consistent add, edit, and delete buttons, input validation, and dropdown support for searching categories and suppliers. Nevertheless, the evaluation also identified several aspects requiring improvement, including a status indicator during data export, simpler technical terms such as "Roles" and "Permission" to make them easier for non-technical users to understand, an undo or restore feature for data deletion, and user guides or tooltips to support system use.

Compared with the studies by Anugrah et al. and Renaldy and Rustam, which focused on general stock recording without detailed role separation [1], the system developed in this study adds role-based access management and expired-goods monitoring features relevant to the characteristics of pet products and medical equipment. Compared with the studies by Rusdianto and Herdiana and by Sabiila and Cahyo Utomo, which also used the Laravel framework for inventory systems [5], this study reinforces the finding that Laravel can be used to develop structured systems, while adding empirical validation through UAT and heuristic usability evaluation. These findings also reinforce Hamidy's argument regarding the importance of systematic stock control in the operations of small and medium-sized businesses [11].

The developed system has several limitations: the database is still implemented locally and can only be accessed through an internal network; access is limited to desktop browsers without a mobile interface; and the system is not yet integrated with external systems such as hosting services or instant messaging notifications. Therefore, implementing a similar system in other pet-product retail businesses should consider adding cloud-based access and a mobile interface to enable more flexible stock monitoring.

4. CONCLUSION

This study produced a web-based inventory information system using the Laravel framework that helps Pelalawan Pet Shop manage incoming and outgoing goods more quickly, accurately, and systematically than the previous manual recording process. Black Box Testing showed that all 39 test scenarios were valid; the User Acceptance Test showed a 100% positive response rate (Strongly Agree and Agree combined), with an average achievement of 94.17%; and Nielsen's Usability Heuristic evaluation showed a low average severity rating (0.93 on a 0–4 scale), indicating that the system interface is relatively easy to use. These findings address the research objective by demonstrating that the developed system can minimize manual recording errors and accelerate inventory report preparation, thereby supporting procurement and stock-management decisions. Several limitations should be considered when interpreting the results: the small number of UAT respondents ($n = 6$) means that the findings are indicative and cannot yet be statistically generalized; the heuristic usability evaluation involved only three internal evaluators, without an external evaluator for comparison; and the

measurement of practical impact (time efficiency and error reduction) remains qualitative, based on user perceptions, and has not been measured quantitatively under controlled conditions.

REFERENCES

- [1] R. E. Anugrah, Y. A. Saputra, and W. Haryono, "Perancangan Sistem Inventory Berbasis Web untuk Optimalisasi Manajemen Persediaan Barang di PT Bumi Daya Plaza," *Bridge: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, vol. 2, no. 4, pp. 342–363, 2024, doi: 10.62951/bridge.v2i4.317.
- [2] G. P. Pinatih and D. Hidayatullah, "Rancang Bangun Inventory System Menggunakan Model Waterfall Berbasis Website," *Jurnal Ilmiah Teknologi Informasi (JATISI)*, vol. 9, no. 1, 2022, doi: 10.35957/jatisi.v9i1.1561.
- [3] I. M. Widiarta, Y. Mulyanto, and A. Sutrianto, "Rancang Bangun Sistem Informasi Inventory menggunakan Metode Agile Software Development (Studi Kasus Toko Nada)," *Digital Transformation Technology (Digitech)*, vol. 3, no. 1, 2023, doi: 10.47709/digitech.v3i1.2549.
- [4] A. A. Kadim, L. Hadjaratie, and M. Muthia, "Implementasi Framework Laravel Dalam Pembuatan Sistem Pencatatan Notula Berbasis Website," *Jurnal Sistem Informasi Bisnis*, vol. 13, no. 1, pp. 45–51, 2023, doi: 10.21456/vol13iss1pp45-51.
- [5] D. Rusdianto and Y. Herdiana, "Rancang Bangun Sistem Informasi Inventori Menggunakan Framework Laravel 8 Di PT. Cemara Kwangjin Tekstil," *Jurnal Sistem Informasi J-SIKA*, vol. 5, no. 2, pp. 71–76, 2023.
- [6] Y. J. Sabiila and I. Cahyo Utomo, "Sistem Inventori Persediaan Produk Berbasis Web di Toko Nanik Tri Menggunakan Framework Laravel," *Jurnal Ilmiah Komputasi*, vol. 24, no. 3, pp. 369–378, 2025, doi: 10.32409/jikstik.24.3.3834.
- [7] S. M. Pulungan, R. Febrianti, T. Lestari, N. Gurning, and N. Fitriana, "Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database," *Jurnal Ekonomi Manajemen dan Bisnis (JEMB)*, vol. 1, no. 2, pp. 143–147, 2023, doi: 10.47233/jemb.v1i2.533.
- [8] A. B. Hikmah, H. Faqih, J. M. Hudin, L. S. Ramdhani, and Y. S. Mulyani, "Sistem Informasi Penjadwalan Maintenance Peralatan Menggunakan Model Waterfall," *Swabumi*, vol. 10, no. 2, pp. 141–145, 2022, doi: 10.31294/swabumi.v10i2.13015.
- [9] K. T. Nugroho, B. Julianto, and D. F. Nur MS, "Usability Testing pada Sistem Informasi Manajemen AKN Pacitan Menggunakan Metode System Usability Scale," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 1, p. 74, 2022, doi: 10.23887/janapati.v11i1.43209.
- [10] Renaldy and A. Rustam, "Perancangan Sistem Informasi Inventory Berbasis Web Pada Gudang Di PT. Spin Warriors," *Aisyah Journal of Informatics and Electrical Engineering*, vol. 4, no. 1, pp. 27–32, 2022.
- [11] F. Hamidy, "Optimalisasi Sistem Manajemen Persediaan untuk Pengendalian Stok yang Efisien Menggunakan Metode FIFO," *CHAIN: Journal of Computer Technology, Computer Engineering, and Informatics*, vol. 2, no. 4, pp. 171–180, 2024, doi: 10.58602/chain.v2i4.150.
- [12] N. Saputra and B. Mulyawan, "Web-Based System Inventory at PT Sapta Tunas Teknologi," *International Journal of Science, Engineering, and Information Technology (IJSEIT)*, vol. 3, no. 3, 2023, doi: 10.35870/ijsecs.v3i3.1831.
- [13] A. Rahman and D. F. Suyatno, "Rancang Bangun Sistem Informasi Persediaan Stok Barang Menggunakan Framework Laravel Dan Metode Lifo," *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, vol. 3, no. 3, pp. 77–83, 2022, doi: 10.26740/jeisbi.v3i3.47082.
- [14] F. Putri, A. Koko, and A. Yayuri, "Rancang Bangun Sistem Informasi Inventaris menggunakan Metode Prototype pada SMK Muhammadiyah 3 Pekanbaru," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 143–153, 2024.

- [15] V. B. Hadikin and J. Wiratama, "Web-Based ERP System Prototype for Enhanced Inventory and Supply Chain Management in the Logistics Industry," *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, vol. 15, no. 2, pp. 138–145, 2024, doi: 10.31937/si.v15i2.3839.
- [16] I. Hilmi, A. Latifa, D. H. Pertiwi, Subhiyanto, and D. Kasoni, "Sistem Informasi Stok Barang Gudang MAMA Mart Berbasis Web," *Jurnal Sistem Informasi LPPM STMIK Antar Bangsa*, vol. 14, no. 1, 2025, doi: 10.51998/jsi.v14i01.598.
- [17] A. Yuliana, M. I. R. Pratama, and A. Sudrajat, "Sistem Manajemen Distribusi Suplai Usaha Kecil Menengah Berbasis Web Menggunakan Framework Laravel," *Jurnal Jecatama*, vol. 5, no. 1, 2026, doi: 10.70428/jecatama.v5i1.1560.