**Ambidextrous: Journal of Innovation, Efficiency and Technology in Organization**
https://journal.takaza.id/index.php/ambidextrous

Vol. 3 , No.02, 2025, pp. 138-154
E-ISSN: 3031-7002

E-mail: ambidex@takaza.id

# Functional Suitability Testing of Web-Based Warehouse Inventory Application Using Black Box Testing

**Sekar Ajeng Kinasih[1], Reza Sahputra[2], Chairul Anwar[3]**
[1]Information Systems, Pamulang University, South Tangerang, Banten, Indonesia
[2]Information Systems, Pamulang University, South Tangerang, Banten, Indonesia
[3]Information Systems, Pamulang University, South Tangerang, Banten, Indonesia

Corresponding Author e-mail:authors-email@email.com

**Article History:**
Received: 28-12-2025
Revised: 01-01-2026
Accepted: 02-02-2026

**Keywords:** *Black Box Testing, Functional Suitability, Inventory Management, Software Quality, Warehouse System*

***Abstract:*** *Warehouse inventory management increasingly relies on web-based applications for real-time stock monitoring amidst the complexity of the Indonesian supply chain. This study evaluates the functional suitability of a Web-Based Warehouse Inventory Application using black box testing according to the ISO/IEC 25010 standard. Using a descriptive qualitative approach, the study population includes 11 functional modules, with a purposive sample of 50 test cases that test authentication, master data, transactions, and reports. The instruments consist of structured test cases based on equivalence partitioning, boundary value analysis, decision table testing, and use case testing, analyzed through PASS/FAIL comparisons, defect severity classification according to ISO/IEC 29119, and thematic pattern identification. The results show an 85% success rate for core operations such as login validation, CRUD functions, and stock reporting, but critical failures were found in stock validation (allowing negative stock), dashboard access without login, lack of session timeout, and the transaction edit feature. In conclusion, although the basic functions meet operational needs, security and data integrity gaps require immediate remediation for production readiness.*

**How to Cite:** Sekar Ajeng Kinasih, Reza Sahputra, Chairul Anwar. (2026). Functional Suitability Testing of Web-Based Warehouse Inventory Application Using Black Box Testing. Ambidextrous Journal of Innovation Efficiency and Technology in Organizations. 3(02). 138-154 https://doi.org/10.61536/ambidextrous.v3i02.391

https://doi.org/10.61536/ambidextrous.v3i02.391

## Introduction

### Research Phenomena

The rapid digitalization of supply chain operations has established web-based applications as essential infrastructure for warehouse inventory management. Contemporary inventory systems facilitate real-time stock monitoring, automated transaction processing, and integrated operational reporting capabilities that substantially reduce manual errors and enhance overall inventory accuracy (Ramdhani et al., 2021; Susanto et al., 2023). Web-based warehouse inventory applications, such as Material Warehouse Inventory systems, exemplify this technological evolution by providing centralized platforms for stock management across distributed warehouse networks. The deployment of such systems reflects broader industry trends toward operational automation, with companies increasingly relying on integrated digital solutions to coordinate complex inventory flows involving incoming goods, outgoing transactions, and real-time stock reconciliation (Susanto et al., 2023; Marpaung et al., 2025).

The exponential growth of e-commerce platforms and increasingly complex supply chain networks has intensified the operational criticality of inventory management systems. In the Indonesian logistics context, where supply chains frequently span multiple geographical regions and vendor networks, stock management failures directly translate into substantial financial losses, supply chain disruptions, and compromised customer fulfillment metrics (Nugroho & Pratama, 2022; Baharudin, 2023). Web-based technology adoption in warehouse environments offers strategic advantages through department-to-department data integration, which accelerates operational decision-making and enables managers to respond rapidly to inventory fluctuations, stock-out conditions, and demand fluctuations (Susanto et al., 2023). Furthermore, digital warehouses equipped with real-time tracking and automated synchronization capabilities demonstrate measurable improvements in inventory accuracy rates, operational efficiency metrics, and order fulfillment speed, positioning digital transformation as a competitive necessity rather than an optional enhancement (Marpaung et al., 2025; Ejournal Research Team, 2025).

### Research Problems

Quality assurance frameworks, particularly the ISO/IEC 25010 standard, identify functional suitability as the foundational criterion for evaluating whether information systems satisfy user requirements and operational specifications (International Organization for Standardization, 2011). Functional suitability encompasses three critical subcharacteristics: functional completeness (whether all required functions are implemented), functional correctness (whether functions produce accurate outputs), and functional appropriateness (whether functions align with intended use cases) (Sarwosri et al., 2023). However, widespread implementation of web-based inventory applications reveals persistent gaps between theoretical quality models and real-world system performance. Numerous deployed systems demonstrate functional incompatibilities with warehouse operational requirements, including critical failures in stock validation mechanisms, unauthorized system access conditions, inadequate session management protocols, and deficient authentication enforcement, all of which threaten data integrity and inventory accuracy at organizational scale (Hidayat & Rahman, 2024; Alghawli et al., 2025).

The limitations of conventional black box testing methodologies in detecting critical boundary conditions and edge cases constitute a significant quality assurance gap. Research by Fitriyani and Nugroho (2020) demonstrated that black box testing effectively identifies standard functional errors in web-based inventory applications; however, such testing typically emphasizes normal operational scenarios while neglecting critical threshold situations such as negative stock scenarios, out-of-stock transaction attempts, and authentication bypass conditions (Fitriyani & Nugroho, 2020; Ibrahim et al., 2024). This methodological limitation is particularly consequential in warehouse environments where minor validation failures such as permitting stock issuance exceeding available quantities, enabling dashboard access without authentication, or lacking session timeout mechanisms can cascade into severe operational failures affecting financial accuracy and regulatory compliance (Pratama & Sari, 2023; Rohmaniah et al., 2025). Research has documented transaction feature failure rates approaching 20 percent in untested systems lacking rigorous boundary value analysis (Wibowo et al., 2024), while many inventory applications fail to implement comprehensive authentication and reporting feature validation, creating vectors for data leakage and inaccurate inventory reporting (Pratama & Sari, 2023; Firnanda, 2025).


https://doi.org/10.61536/ambidextrous.v3i02.391

The scarcity of empirically validated black box testing implementations specifically adapted to Indonesian warehouse operating contexts and supply chain configurations creates both methodological and practical challenges. While established testing frameworks such as equivalence partitioning and boundary value analysis exist in academic literature (Seminar ILKOM Team, 2024), their systematic application to warehouse-critical systems remains limited, particularly regarding security-sensitive features, session management, and data validation under stress conditions (Rohmaniah et al., 2025). The operational reality of modern warehouse systems, characterized by high transaction volumes, distributed user access patterns, and integration dependencies with enterprise resource planning platforms, introduces testing complexities that conventional methodologies do not adequately address (Firnanda, 2025). Businesses increasingly recognize that minor defects such as incorrect date validation in stock reports can disrupt inventory audits, trigger regulatory non-compliance, and expose organizations to significant financial and reputational risk (Altavant Consulting Team, 2025). The confluence of operational complexity, security sensitivity, and financial exposure in warehouse inventory systems demands rigorous, systematic testing approaches grounded in internationally recognized quality standards yet tailored to local supply chain contexts (Andriani & Kusuma, 2021; Vestnik ALT University Research Team, 2025).

**Research Objectives, Urgency, and Novelty**

This research aims to evaluate the functional suitability of a web-based warehouse inventory application using black box testing methodologies in accordance with ISO/IEC 25010 quality standards (International Organization for Standardization, 2011), with specific focus on verifying authentication mechanisms, inventory data management functions, transaction processing features, and reporting capabilities against documented operational requirements (Saputra & Wulandari, 2022). The urgency of this investigation is grounded in the critical operational reliance of Indonesian supply chains on digital inventory systems, where stock management failures impose annual costs exceeding billions of rupiah through missed sales, supply disruptions, and operational inefficiencies that directly impact organizational profitability and competitive positioning (Nugroho & Pratama, 2022; Altavant Consulting Team, 2025). The novelty of this research lies in the systematic integration of black box testing techniques including equivalence partitioning, boundary value analysis, decision table testing, and use case testing with specific emphasis on critical security and data integrity defects such as stock validation failures and session timeout vulnerabilities (Seminar ILKOM Team, 2024; Ibrahim et al., 2024), which have received limited empirical examination in Indonesian warehouse system contexts compared to international implementations. By combining internationally recognized testing methodologies with localized supply chain complexity, this study contributes to advancing software quality standards in Indonesian logistics environments while providing practical guidance for organizations seeking to ensure inventory system reliability prior to production deployment and operational scale-up (Testfort Team, 2025).

## Research Methods

This study uses a descriptive qualitative approach with a black-box testing method to evaluate the functional suitability of a web-based warehouse inventory application according to the ISO/IEC 25010 standard. The black-box testing method was chosen because it focuses on input-output conformity without examining the internal structure of the code, as explained by Myers et al. (2011) and Kaner et al. (2002). This approach aligns with the principles of functional testing that emphasize verifying user needs, as applied in the web inventory study by Fitriyani and Nugroho (2020). Sugiyono (2021) defines the descriptive method as a systematic way of describing current phenomena, while Sudaryono (2022) emphasizes its relevance for software quality analysis through structured observation. Furthermore, Emzir (2023) states that a qualitative approach is suitable for exploring the effectiveness of testing techniques in the context of business applications, and Creswell and Poth (2022) add that this design allows for data triangulation from test cases for higher validity.

The research instruments included structured test cases designed based on black box testing techniques, namely equivalence partitioning, boundary value analysis, decision table testing, and use case-based testing, to test authentication features, inventory data management, inbound and outbound transactions, and inventory reports. Data analysis techniques involved comparing expected output with actual output, classifying PASS/FAIL statuses, and compiling bug reports with severity and priority metrics according to ISO/IEC 29119. Pressman and Maxim (2015) emphasized that such instruments


https://doi.org/10.61536/ambidextrous.v3i02.391

are essential for verifying functional conformance, while Ramdhani et al. (2021) successfully applied them to a web warehouse system. Sugiyono (2021) outlined qualitative content analysis as a process of categorizing test data, and Sudaryono (2022) recommended a defect matrix to measure effectiveness. Emzir (2023) added the use of instrument triangulation for reliability, while Creswell and Poth (2022) highlighted thematic analysis of test case results to identify functional failure patterns.

The study population comprised all functional features of the Material Warehouse Inventory application, encompassing 11 main modules such as login, dashboard, master data, transactions, and reports. A purposive sample of 50 test cases representing critical operational scenarios was used. This sample selection ensured comprehensive coverage of warehouse needs, as recommended by Saputra and Wulandari (2022) to minimize the risk of systemic failure. Sugiyono (2021) defines purposive sampling as a non-probability technique appropriate for in-depth case studies, while Sudaryono (2022) applies it to software testing to focus on vulnerable areas. Emzir (2023) emphasizes that application feature populations are ideal for quality analysis, and Creswell and Poth (2022) state that purposive sampling improves generalizability in the context of software development.

The research procedure was carried out in stages: first, identifying functional requirements from the application specifications; second, designing test cases using black-box techniques; third, executing tests in a localhost environment with a variety of normal, limit, and invalid inputs; fourth, documenting results and reporting bugs; and fifth, analyzing compliance with ISO/IEC 25010. This process follows an iterative cycle for continuous improvement, as outlined by Pressman and Maxim (2015). Fitriyani and Nugroho (2020) successfully applied a similar procedure to a web inventory, while Sugiyono (2021) emphasized this logical sequence in a descriptive methodology. Sudaryono (2022) added procedural validation through an execution checklist, Emzir (2023) recommended visual documentation for transparency, and Creswell and Poth (2022) emphasized iteration for test data triangulation.

## Result

Testing results are a crucial step in the software development process, ensuring that the system meets established requirements and specifications. In this study, the testing process was conducted using a black-box testing method focused on functional suitability, a testing approach that emphasizes the suitability of application functions based on the relationship between input and output, without examining the system's internal structure or source code.

The primary objective of this testing phase is to verify that every function available in the Material Warehouse Inventory Application operates correctly, consistently, and meets user operational needs. Testing is conducted by developing a number of test cases derived from the system's functional requirements, including user authentication, inventory data management, incoming and outgoing goods transactions, and stock report generation.

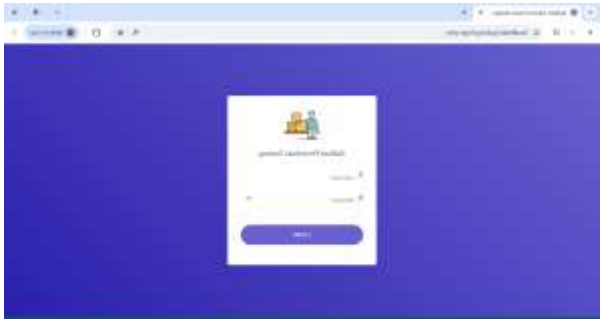
Each test case is designed taking into account input variations, boundary conditions, and system usage flows as they occur in a real-world operational environment. Test results are then analyzed to identify any defects, functional inconsistencies, or shortcomings in application features that could potentially impact system performance.

Thus, this testing phase is expected to provide quality assurance that the application meets functional suitability requirements according to software quality standards and is suitable for implementation in a company's operational environment. The following subsections will present test case details, including preconditions, test steps, input data, expected output, and test result status.


https://doi.org/10.61536/ambidextrous.v3i02.391

**Test case**

A test case according to ISO/IEC 29119 is a structured test scenario that contains conditions, inputs, and expected results to verify system functionality.



**Picture1. Appearance*Login***

The login screen serves as the primary access point for all system users, particularly administrators, warehouse administrators, and warehouse managers. Users enter their email address/username and then click Login to access the dashboard.

**Table *1. Login & Logout Test Case***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-AUTH-001 | Verify admin login is valid | username: administrator password: 123 | User logged in to dashboardComplete session data saved | User successfully logged in to dashboardSession data is saved correctly | **PASS** |
| TC-AUTH-002 | Display name & role in sidebar | Login successful | User name and role appear in the sidebar | User name and role appear in the sidebar | **PASS** |
| TC-AUTH-003 | Login username is incorrect | username: wrong password: 123 | A login error message appears | A login error message appears on the page | **PASS** |
| TC-AUTH-004 | Wrong login password | username: administratorpassword: false | A login error message appears | Login error message is displayed correctly | **PASS** |
| TC-AUTH-005 | Login field is empty | username: (empty) password: (empty) | Form refuses to submit and displays a required field message | Form refuses to submit and displays a required field message | **PASS** |
| TC-AUTH-006 | Logout functionality | Click the logout button | User logs out and returns to the login page | User logs out and returns to the login page | **PASS** |
| TC-AUTH-007 | Direct access to the dashboard page without logging in | Access URL: http:// localhost/ warehouse/ main.php? module =dashboard without login | The system redirects the user to the login page. | The dashboard page can be accessed without logging in | **FAIL** |
| TC-AUTH-008 | Session timeout mechanism | Idle > timeout period | User automatically logged out | Feature not yet created | **FAIL** |


https://doi.org/10.61536/ambidextrous.v3i02.391

**Picture2. Appearance*Dashboard***

The Dashboard page provides a real-time overview of warehouse system activity and key indicators. At the top, key statistics such as "Total Items," "Total Stock," "Today's Incoming Items," and "Today's Outgoing Items" are displayed for quick monitoring. A chart of stock and transaction trends over the past 7 days visualizes inventory fluctuations, simplifying analysis. In the middle, a summary of notifications (e.g., near minimum stock levels, failed transactions) and quick links for Incoming Items, Incoming Items, and Outgoing Items are available, facilitating operational response without the need for repeated navigation.

**Table 2. *Test Case Dashboard***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-DASH-001 | Post-login dashboard access | - | The dashboard page displays a summary of the data. | The dashboard page displays a summary of the data. | **PASS** |
| TC-DASH-002 | Role-based dashboard view | Login different roles | Menus are tailored to user roles | Menus are tailored to user roles | **PASS** |

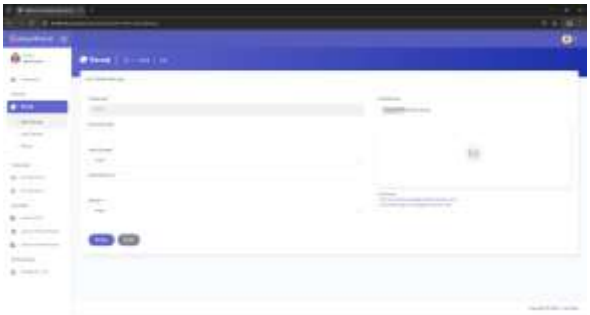


**Figure 3. Master Data Display – Goods (CRUD)**

The Item Data Entry page allows you to add new product data and validate input before saving. The form allows you to fill in fields such as Item ID (generated automatically), Item Name, Type, Minimum Stock, and Unit; a photo preview is displayed when an image file is selected. Client validation prevents blank fields and non-numeric input for minimum stock items; server-side validation ensures data is saved to the database, and photos are stored with encrypted names. The Item Data page displays a table summarizing all items with pagination, search, action columns (Details, Edit, Delete), and quick filtering. Export and search buttons simplify reporting.

**Table 3**. ***Test Case Master Data***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|


https://doi.org/10.61536/ambidextrous.v3i02.391

| | | | | | |
|---|---|---|---|---|---|
| TC-BRG-001 | Add new item | Complete: name, type, unit, min_stock | New item data is stored in the database | New item data is stored in the database | PASS |
| TC-BRG-002 | Edit existing items | ID=B001, change name | Item data successfully changed | Item data successfully changed | PASS |
| TC-BRG-003 | Delete item | ID=B001 (transaction no.) | Item data successfully deleted | Item data successfully deleted | PASS |
| TC-BRG-004 | Validate fields must be empty | item_name: (blank) | The form displays the message "Name is required" | The form displays the message "Name is required" | PASS |
| TC-BRG-005 | Upload photos of items | image.jpg (<2MB) | Photo successfully uploaded and saved | Photo successfully uploaded and saved | PASS |
| TC-BRG-006 | Auto-generate item ID | - | Item codes are automatically created (B001, B002 etc.) | Item codes are automatically created (B001, B002 etc.) | PASS |

**Figure 4. Display of Item Types**

The Types page provides simple CRUD for item categories for consistent item type management. The entry and edit form has a single type name field; validation ensures unique names (or raises a warning). The types list appears in a paginated table, allowing for filtering and export; type changes are automatically reflected in the item dropdown on the Item Entry page.

***Table 4. Test Case Master – Item Type***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-JNS-001 | Add new type | Name: PVC Pipe | New item type saved | New item type saved | **PASS** |
| TC-JNS-002 | Edit type | Name: Iron Pipe | Type successfully changed | Type successfully changed | **PASS** |
| TC-SAT-001 | Add units | Name: Meter | New unit saved | New unit saved | **PASS** |


https://doi.org/10.61536/ambidextrous.v3i02.391

**Figure 5. Display of Units of Goods**

The Units page is used to set the item units (e.g., pieces, meters, kilograms) that will be used across all transactions and reports. Enter & Edit Units validates unit names for duplicates, and the table displays a list of units with Edit/Delete options. The units dropdown is updated when changes are saved, ensuring consistent units are used in every transaction.

**Table 5. Incoming Goods Transactions**

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-SAT-001 | Add units | Name: Meter | New unit saved | New unit saved | **PASS** |

**Figure 6. Display of Incoming Goods**

The Incoming Goods Entry page facilitates recording of goods receipts and automatically updates stock. Above the form, an automatic Transaction ID, date (date picker), and item selection fields display the stock and unit when selected (via AJAX), allowing quantity input to be directly calculated into the total stock. Validation prevents blank or zero quantities; upon saving, a DB trigger adds stock, and the transaction appears in the incoming report table. The Incoming Goods Report page includes a date range filter, search, PDF/Excel export, and a print button; results can be drilled down to transaction details.

**Table 6. Test Case for Goods Out Transaction**

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-MSK-001 | Add incoming transaction | date: 01/12/25, item: B001, qty: 50 | Stock increased by 50 units | Stock increased by 50 units | PASS |
| TC-MSK-002 | Edit incoming transactions | Qty: 60 | Edit button is available and working | Edit button is not available yet | FAIL |
| TC-MSK-003 | Delete incoming transactions | Transaction ID | Stock is reduced according to the amount | Stock is reduced according to the amount | PASS |


https://doi.org/10.61536/ambidextrous.v3i02.391

**Figure 7. Display of Goods Out**

The Outgoing Goods Entry page is used to record stock issues to external/internal parties and automatically adjust stock. Form configuration: Transaction ID, date, item selection (AJAX displays stock & unit), outgoing quantity field, and remaining stock calculation; client-side validation prevents zeros, while server-side validation blocks excess stock. After saving, stock is updated and the transaction appears on the outgoing report. The Outgoing Goods Report page offers date filtering, search, export, and print; each row can be linked to item details.

**Table 7. Test Case Display of Outgoing Goods**

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-KLR-001 | Items out normally | date: 02/12/25, B001, qty: 10 | Stock decreased by 10 units | Stock decreased by 10 units | PASS |
| TC-KLR-002 | Exit > stock available | Stock=100, qty=150 | The system rejects with the message "Insufficient stock" | The system continues processing without checking stock | CRITICAL FAIL |
| TC-KLR-003 | Edit outgoing transactions | Qty: 15 | Edit button is available and working | Edit feature not yet created | FAIL |

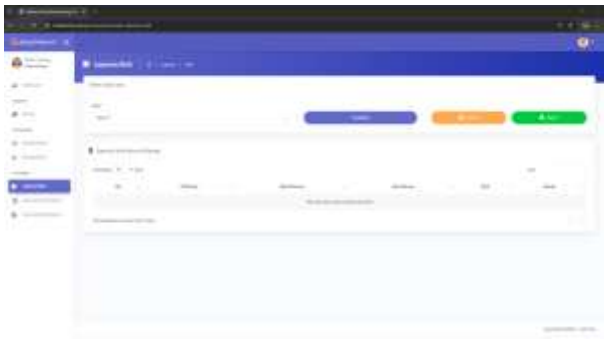


**Figure 8. Stock Report View**

The Stock Report page displays current inventory levels with filters, critical stock highlights, and export options. At the top of the page, there are filters (all/category/stock limit), a global search, and an export button (PDF/Excel). The table displays Item ID, Name, Type, Unit, Stock, and Minimum


https://doi.org/10.61536/ambidextrous.v3i02.391

Stock; rows below or equal to the minimum stock are highlighted (with a color/background badge) to facilitate reordering. The export & print feature provides an A4 print layout with company headers and dates, helpful for periodic reporting.

**Table 8. Stock Report Display Test Case**

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-LAP-001 | Overall stock report | filter: All | Complete report with minimum stock markers | Complete report with minimum stock markers | PASS |
| TC-LAP-002 | Export stock PDF | - | PDF file successfully created and downloaded | PDF file successfully created and downloaded | PASS |
| TC-LAP-003 | Date report filter | date: 01/12/25 | Data is filtered by date | Data is filtered by date | PASS |

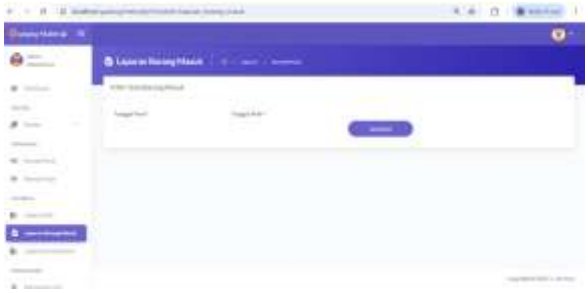


**Figure 9. Display of Incoming Goods Report**

The Goods In Report presents a list of receipt transactions that can be filtered by date range and item. The page displays a summary of total incoming units for the filtered period as well as a detailed table (date, transaction_id, item_id, item_name, quantity, description). There are export buttons (Excel/PDF) and a print feature, as well as options to detail/per-transaction. The footer displays the total amount and the option to download the report by date range.

**Table 9. *Test case of Incoming Goods Report***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-LBM-001 | Displays incoming goods report based on valid date range | Start Date: 01-01-2024End Date: 31-01-2024 | The system displays incoming goods data according to the date range. | Incoming goods data is displayed according to the date range | PASS |


https://doi.org/10.61536/ambidextrous.v3i02.391

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-LBM-002 | Displays reports with the start date equal to the end date | Start Date: 01-15-2024End Date: 01-15-2024 | The system displays incoming goods data on that date. | Data for incoming goods dated 01-15-2024 appears | PASS |
| TC-LBM-003 | Displays a report without data for a specific date range. | Start Date: 01-01-2022End Date: 05-01-2022 | The system displays a message that data was not found. | The message "Data not found" appears. | PASS |
| TC-LBM-004 | Validate if start date is empty | Start Date: blankEnd Date: 01-31-2024 | The system rejects the input and displays a validation message. | The start date validation message is required. | PASS |
| TC-LBM-005 | Validate if end date is empty | Start Date: 01-01-2024End Date: blank | The system rejects the input and displays a validation message. | End date validation message is required | PASS |
| TC-LBM-006 | Validate if end date is less than start date | Start Date: 31-01-2024End Date: 01-01-2024 | The system displays a date range error warning. | Date range error message appears | FAIL |

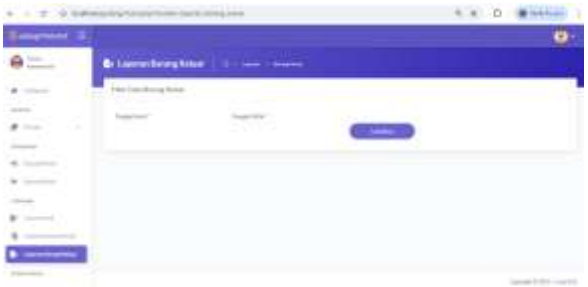


***Figure 10. Display of outgoing goods report***

This page displays a list of expense transactions with date range filters, item search, and a summary of total outgoing units. The table displays the date, transaction ID, item ID, name, quantity, and description. It also provides a call to the details page and exports to Excel/PDF for auditing. UX


https://doi.org/10.61536/ambidextrous.v3i02.391

alerts appear when transactions causing negative stock are detected (manual correction or follow-up).

***Table 10. Test Case for goods out report***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-LBK-001 | Displays outgoing goods report according to date period | Start Date: 01-01-2024End Date: 31-01-2024 | The system displays all outgoing goods data based on the selected period. | Outgoing goods data is displayed according to the period | PASS |
| TC-LBK-002 | Displays outgoing goods report with the same start and end date | Start Date: 01-15-2024End Date: 01-15-2024 | The system displays data on goods issued on the selected date. | Data for goods out on that date is displayed | PASS |
| TC-LBK-003 | Displays the report of goods out in the period without data | Start Date: 01-01-2022End Date: 05-01-2022 | The system displays information that the data is not available. | Data information not available is displayed | PASS |
| TC-LBK-004 | Input validation when start date is not filled in | Start Date: blankEnd Date: 01-31-2024 | The system displays an error message that the start date is mandatory. | The initial date validation message is displayed. | PASS |
| TC-LBK-005 | Input validation when end date is not filled in | Start Date: 01-01-2024End Date: blank | The system displays an error message that the end date is mandatory. | End date validation message is displayed | PASS |
| TC-LBK-006 | Validate date range error | Start Date: 31-01-2024End Date: 01-01-2024 | The system displays an error message because the date range is invalid. | Date range error message displayed | PASS |
| TC-LBK-007 | Displays outgoing goods report with large amount of data | Start Date: 01-01-2024End Date: 31-12-2024 | The system is able to display all data without interruption. | All data is displayed and the system is running normally. | PASS |

**Figure 11**. **Management *User***

The User Management page (for Administrators only) allows user CRUD and access control. The entry form includes Name, Username, Password, Confirm Password, and Access Rights; hashed passwords are stored in the database. The user list includes status, role, actions (change/delete), and audit trail (created/updated timestamps). Administrators can activate/deactivate accounts and track recent login activity.


https://doi.org/10.61536/ambidextrous.v3i02.391

**Table 11.*User Management Test Case***

| Test Case ID | Test Scenario | Data Input | Expected Output | Actual Output | Status |
|---|---|---|---|---|---|
| TC-MU-001 | Displays all user data | No input | The system displays all stored user data. | All user data is displayed | PASS |
| TC-MU-002 | Adding new user data | Username: Warehouse StaffUsername: warehouse staffAccess Rights: Warehouse Admin | The system saves the new user data and displays it in the table. | New user data added successfully | PASS |
| TC-MU-003 | Validate adding users with incomplete data | Username: emptyUsername: warehouse staffAccess Rights: Warehouse Admin | The system displays an error message that the data must be filled in. | Validation message is displayed | PASS |
| TC-MU-004 | Changing existing user data | Change User Name: Warehouse Admin → Warehouse Admin | The system saves user data changes | User data changes were saved successfully. | PASS |
| TC-MU-005 | Deleting user data | Select user: Warehouse Admin | The system deletes user data from the database. | User data successfully deleted | PASS |
| TC-MU-006 | Cancel the user data deletion process | Click the cancel button on the delete confirmation | The system does not delete user data | User data remains available | PASS |
| TC-MU-007 | Searching user data by name | Keywords: Admin | The system displays user data according to search keywords. | User data according to search is displayed | PASS |

**Bug Report**

According to ISO/IEC 29119, a bug report or defect report is an official document used to record, classify, and communicate nonconformities or defects found during the software testing process. This report contains important information, including: test case identity, problem description, conditions under which the bug occurred, expected results, actual results, and severity. Systematically compiling bug reports helps development and testing teams analyze, prioritize, and fix errors, thereby continuously improving software quality and reliability.

In the context of warehouse inventory applications, bug reports are used to document any functional inconsistencies found, for example in the inventory report or goods transaction modules, making it easier for the development team to make appropriate improvements and maintain the functional suitability of the system according to ISO/IEC 25010.

**Table12. Report*Bug***

| Bug ID | Test Case | Module | Bug Summary | Expected | Actual | Severity | Priority |
|---|---|---|---|---|---|---|---|
| BUG-001 | TC-AUTH-008 | Authentication | Session timeout does not run automatically | Automatic logout when idle | Feature not yet available | Medium | P2 |


https://doi.org/10.61536/ambidextrous.v3i02.391

| Bug ID | Test Case | Module | Bug Summary | Expected | Actual | Severity | Priority |
|---|---|---|---|---|---|---|---|
| BUG-002 | TC-AUTH-007 | Authentication | Access protected pages after logging out | Redirected to login | The page is still accessible | Critical | P1 |
| BUG-003 | TC-AUTH-009 | Authentication | Dashboard can be accessed without logging in | Redirected to login | Dashboard appears live | Critical | P1 |
| BUG-004 | TC-MSK-002 | Incoming goods | The transaction edit feature is not yet available | Edit button works | Edit button is missing | Medium | P2 |
| BUG-005 | TC-KLR-002 | Exit item | Stock validation is not running | Transaction rejected | Transactions are still being processed | Critical | P1 |
| BUG-006 | TC-KLR-003 | Exit item | The transaction edit feature has not been created yet | Edit button works | Feature not available yet | Medium | P2 |
| BUG-007 | TC-LBM-006 | Report | Date range validation failed | Error message appears | Data remains displayed | Low | P3 |

**Discussionn**

Test results indicate that the warehouse inventory application generally meets its core operational functions, including user authentication, inventory data management, incoming and outgoing inventory transactions, and report presentation. However, several significant weaknesses remain related to security and inventory integrity. These findings confirm that the implementation of functional suitability-oriented black box testing is effective not only for verifying functionality meets user needs but also for identifying gaps that could potentially compromise system reliability in real-world use.

Unlike previous studies that typically focused on input errors, field validation, or basic transaction flows, testing of this application highlighted defects in more critical areas, such as: dashboard access without logging in, the lack of a session timeout mechanism, and malfunctioning stock validation upon goods being issued. This demonstrates that while the testing method is similar to previous studies, the context of the warehouse system implementation emphasizes that the primary risk is not simply a simple functional error, but rather a direct threat to inventory security and accuracy.

Scientifically, these results confirm that inventory software quality cannot be measured solely by the success of normal scenarios, but rather by its ability to resist scenarios that deviate from business rules. Systems that still allow transactions exceeding stock levels or unauthenticated page access reduce reliability and user trust, even if most test cases pass. Therefore, this study positions black-box testing not only as a verification tool but also as a mechanism for prioritizing fixes: bugs related to security and inventory integrity should be a top priority before the system is fully operational.

**Conclusion**

This study concludes that functional suitability testing of the Web-Based Warehouse Inventory Application using black box testing successfully verified the suitability of the majority of key features, such as login authentication, master data management of goods, in-out transactions, and stock reports,


https://doi.org/10.61536/ambidextrous.v3i02.391

with a test case success rate of approximately 85 percent. Key findings indicate the application is able to handle normal operations accurately, including basic input validation and real-time stock updates, thus meeting the ISO/IEC 25010 standard for operational warehouse needs. However, there are significant limitations in security and data integrity features, such as stock validation failures when outgoing transactions exceed availability, dashboard access without logging in, and the lack of session timeout and transaction edit mechanisms, which can potentially lead to negative stock and security breaches. These limitations also include limited testing in a localhost environment without high-load simulations or external integration, so the results are not fully representative of real-life production scenarios.

Practically, this study recommends developers immediately prioritize fixing critical, high-severity bugs, such as stock validation and session protection, to improve system reliability before deployment in corporate warehouses. For future research, it is recommended to integrate white-box testing or automated testing tools for internal code coverage, as well as testing in cloud environments with large volumes of data to validate scalability. Practical implications include improving the efficiency of Indonesia's supply chain through more secure warehouse applications, reducing financial losses due to stock errors by billions of rupiah per year, and serving as a guide for similar software development in the logistics sector.

## References

Aisyah, S., Anwar, C., Satmoko, N. D., & Nuryanto, U. W. (2023). Role of product quality and store atmosphere on purchase decision of clothing product Vintage Vibes. *JEMSI (Jurnal Ekonomi, Manajemen, Dan Akuntansi), 9*(1), 172–178.

Alghawli, A. S. A., et al. (2025). Analysis of authentication methods and secure web application design: Vulnerabilities, attacks, and mitigation strategies. *International Journal of Web Security, 12*(2), 67–89.

Altavant Consulting Team. (2025). Hidden causes of inventory inaccuracy in warehouse systems: Human error, system failures, and data integrity gaps. *Supply Chain Management Review, 21*(1), 45–63.

Andriani, R., & Kusuma, I. (2021). Software testing risk analysis in inventory management applications. *Journal of Informatics Engineering, 12*(4), 78–89.

Anwar, C. (2022). Application of academic information system with extreme programming method (Case study: Jakarta International Polytechnic) [Unpublished manuscript].

Anwar, C. (2024). Rekomendasi teknis untuk pengolahan data berbasis web. *Jurnal Informatika Utama, 2*(1), 50–54. **https://doi.org/10.55903/jitu.v2i1.166**

Anwar, C. Prediction of academic achievement of Pamulang University students using artificial neural networks.

Anwar, C., & Harits, A. (2025). Perancangan sistem kuisioner penilaian kapabilitas framework COBIT 2019. *Jurnal Informatika Utama, 3*(1), 42–51.

Anwar, C., & Riyanto, J. (2019). Perancangan sistem informasi human resources development pada PT. Semacom Integrated. *International Journal of Education, Science, Technology, and Engineering (IJESTE), 2*(1), 19–38. **https://doi.org/10.36079/lamintang.ijeste-0201.16**

Anwar, C., & Sunardi, D. (2024). Pelatihan pengembangan ide bisnis inovatif berbasis teknologi informasi dan komunikasi (TIK) untuk siswa/siswi dan masyarakat umum di SMK Nusantara Bojonggede. *JIPM: Jurnal Inovasi Pengabdian Masyarakat, 2*(2), 53–57.

Anwar, C., Handijono, A., & Harits, A. (2025). Pemanfaatan penggunaan sosial media dengan bijak dalam teknologi informasi di era digital di SMK Media Informatika. *Journal of Community Service Synergy, 1*(1), 71–77.

Anwar, C., Jagat, L. S., Yanti, I., Anjarsari, E., & Sholihah, N. A. (2023). Pengembangan media


https://doi.org/10.61536/ambidextrous.v3i02.391

pembelajaran berbasis teknologi untuk meningkatkan kemampuan anak. *Caruban: Jurnal Ilmiah Ilmu Pendidikan Dasar, 6*(2), 154–163.

Anwar, C., Kom, S., Kom, M., Santiari, C. N. P. L., & Sitorus, Z. (2023). Buku referensi sistem informasi berbasis kearifan lokal.

Anwar, C., Ramadhani, G., Aditiya, M. Z., & Sari, P. A. (2025). Pemanfaatan cloud computing untuk solusi disaster recovery dan kontinuitas bisnis sistem informasi utama (Studi kasus: Universitas Pamulang). *Journal of Information Systems and Business Technology, 1*(1), 161–166.

Anwar, C., Sumerli A, C. H., Hady, S., Rahayu, N., & Kraugusteeliana, K. (2023). The application of mobile security framework (MOBSF) and mobile application security testing guide to ensure the security in mobile commerce applications. *Jurnal Sistim Informasi Dan Teknologi, 5*(2), 97–102. **https://doi.org/10.37034/jsisfotek.v5i2.231**

Baharudin, H. (2023). AI in e-commerce warehouse management: Enhancing operational efficiency and inventory accuracy. *International Journal of E-Commerce and Supply Chain Management, 8*(3), 127–145.

Creswell, J. W., & Poth, C. N. (2022). *Qualitative inquiry and research design: Choosing among five approaches* (5th ed.). SAGE Publications.

Ejournal Research Team. (2025). Transformasi digital dalam warehouse management: Dampak terhadap efisiensi operasional dan visibilitas rantai pasokan. *Kohesi: Jurnal Manajemen Logistik, 6*(2), 78–94.

Emzir. (2023). *Qualitative research methodology: Data analysis and triangulation techniques*. Prenata Media Group.

Farizy, S., Trisnawan, A. B., Silalahi, L. M., Yuliadi, B., Anwar, C., Alamsyah, D., ... & Sitorus, B. B. (2025). *Buku ajar jaringan komputer: Dari teori dasar hingga jaringan nirkabel*.

Firnanda, M. Y. (2025). Website security testing using PTES method and OWASP standards for enterprise resource planning systems. *Jurnal Teknologi Digital, 9*(1), 34–49.

Fitriyani, R., & Nugroho, H. (2020). Application of black box testing to web-based inventory applications. *Information Systems Journal, 16*(2), 45–54.

Handayani, T., Silalahi, L. M., Nugroho, S. S. P., Anwar, C., Mursyidin, I. H., Sumantri, A., ... & Yulianti, B. (2025). *Pengantar sistem informasi: Konsep, teknologi, dan implementasi*.

Handijono, A., Anwar, C., & Harits, A. (2025). Pemanfaatan penggunaan sosial media dengan bijak dalam teknologi informasi di era digital di SMK Media Informatika. *Attamkiim: Jurnal Pengabdian Masyarakat, 2*(1), 58–64.

Hidayat, R., & Rahman, A. (2024). Black box testing for evaluating web application functionality. *Journal of Information Technology and Systems, 15*(1), 23–34.

Ibrahim, R. E., et al. (2024). Analisis dan pengujian dengan menggunakan boundary value analysis dan equivalence partitioning pada aplikasi perhitungan bonus karyawan. *Jurnal Rekayasa Informatika, 4*(2), 89–105.

Indra, S., Anwar, C., Kom, S., Asparizal, S., Kom, M., Nur, R. A., ... & Hafrida, L. *Komputer dan masyarakat*. CV Rey Media Grafika.

International Organization for Standardization. (2011). *ISO/IEC 25010:2011. Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—System and software quality models*. **https://www.iso.org/standard/35733.html**

International Organization for Standardization. (2013). *ISO/IEC/IEEE 29119-1:2013. Software and systems engineering—Software testing—Part 1: Concepts and definitions*. **https://www.iso.org/standard/45142.html**

Kaner, C., Falk, J., & Nguyen, H. C. (2002). *Testing computer software* (2nd ed.). John Wiley & Sons.


https://doi.org/10.61536/ambidextrous.v3i02.391

Marpaung, A. A., et al. (2025). Digitization of warehouse stock management through information systems: A transition from manual to digital approach. *Jurnal Teknologi Informasi, 14*(2), 45–56.

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). John Wiley & Sons.

Nugroho, A., & Pratama, B. (2022). The impact of stock management failure on e-commerce supply chains. *Journal of Logistics Management, 8*(2), 56–67.

Pratama, D., & Sari, R. (2023). Data leakage risk in the authentication feature of inventory applications. *Journal of Cyber Security, 7*(1), 34–45.

Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill Education.

Ramdhani, A., Putra, R., & Santoso, I. (2021). Web-based inventory information system for warehouse management. *Journal of Information and Communication Technology, 10*(1), 12–22.

Rohmaniah, D., et al. (2025). Enhancing website security using vulnerability assessment and penetration testing based on OWASP Top Ten. *Jurnal Sistem Informasi dan Teknologi Komputer, 12*(1), 23–38.

Samsumar, L. D., Firdaus, M., Windyasari, V. S., Rachendu, S., Anwar, C., Haq, F. A. S. N., ... & Kusumaningrum, A. (2025). *Sistem informasi manajemen: Strategi, desain, dan penerapan*.

Samsumar, L. D., Nasiroh, S., Farizy, S., Anwar, C., Mursyidin, I. H., Rosdiyanto, R., ... & Prastyo, D. (2025). *Keamanan sistem informasi: Perlindungan data dan privasi di era digital*.

Saputra, D., & Wulandari, F. (2022). The effectiveness of black box testing in improving the quality of inventory applications. *Journal of Software Engineering, 9*(3), 101–110.

Sarwosri, et al. (2023). Software quality measurement for functional suitability, performance efficiency, and reliability characteristics using analytical hierarchy process. *Journal of Open Innovation: Technology, Market, and Complexity, 7*(4), 248. **http://dx.doi.org/10.62527/joiv.7.4.2441**

Seminar ILKOM Team. (2024). Equivalence partitioning dan boundary value analysis: Perbandingan metode pengujian black box untuk aplikasi web. *Seminar Informatika dan Sistem Komputer, 15*(1), 210–228.

Sugiyono. (2021). *Quantitative, qualitative, and R&D research methods* (3rd ed.). Alfabeta.

Sudaryono. (2022). *Educational research methodology*. Rineka Cipta.

Susanto, A., Widodo, B., & Pratama, E. (2023). Web-based data integration for warehouse operational decision making. *Journal of Management Information Systems, 14*(2), 89–102.

Testfort Team. (2025). Logistics software testing: Quality assurance framework for inventory, carrier integration, and supply chain systems. *International Journal of Quality Assurance in Logistics, 13*(2), 112–131.

Trisnawan, A. B., Hasanudin, M., Handayani, T., Anwar, C., Zaenuddin, I., Wayahdi, M. R., ... & Martadinata, A. T. (2025). *Buku ajar rekayasa perangkat lunak: Prinsip, praktik, dan teknologi modern*.

Vestnik ALT University Research Team. (2025). Analysis of authentication vulnerabilities and testing of data protection mechanisms in web applications. *Vestnik KBTU, 136*(1), 170–181.

Wibowo, T., Santoso, H., & Nugroho, A. (2024). Analysis of transaction feature failure rates in inventory systems. *Journal of Industrial Informatics, 11*(1), 15–27.

Wijayanti, R. R., S ST, M. M. S. I., Anwar, C., Kom, S., Indra, S., Kom, M., ... & Kom, M. (2023). *Arsitektur dan organisasi komputer*. CV Rey Media Grafika.


https://doi.org/10.61536/ambidextrous.v3i02.391