Usability Testing with Cognitive Walktrough KORONA Point of Sales

[1]Philip Nathanael, [2]Jonathan Charles Nugraha, [3]Soetam Rizky Wicaksono
[1]Information System Department, Ma Chung University, Malang, Indonesia, 322110007@student.machung.ac.id
[2]Information System Department, Ma Chung University, Malang, Indonesia, 322110004@student.machung.ac.id
[3]Information System Department, Ma Chung University, Malang, Indonesia, soetam.rizky.machung.ac.id

ABSTRACT

This study tests the KORONA POS software in sales transaction scenarios and integration with printers using functional testing methods. The sales transaction testing involves several steps including selecting product categories, deleting items, changing item prices, canceling receipts, logging out, and displaying total payments. All steps were successful as expected, except for invalid inputs which caused the system to fail in providing notifications for unregistered items. Hardware integration testing examines printer connectivity and receipt printing capability. With the initial condition of the printer being on, the device successfully read the connected printer and printed receipts as expected, demonstrating success in integration and printing functions. The testing also includes a scenario where the printer driver is not connected, in which the KORONA POS software successfully provided a notification that the driver could not be connected, ensuring the system responds correctly. The results of these tests indicate that KORONA POS performs well under standard operational conditions but needs improvements in handling invalid inputs to enhance overall system reliability.

Keywords: *Testing, Functional Testing, Point of Sales (POS), Korona POS, Software Testing Documentation*

Corresponding Author:

Philip Nathanael
Information System Department, Ma Chung University
Malang, Indonesia
322110007@student.machung.ac.id

INTRODUCTION

With the increasing development of technology, the need for point of sales (POS) software can help improve business efficiency, operational effectiveness, data accuracy, and provide sales data that can be processed to understand consumer trends or behavior. Point of Sales (POS) systems are used by businesses to facilitate sales transactions. POS includes hardware and software that assist in selling products or services to customers[1]. These systems typically consist of several hardware components such as computers, scanners, printers, and the POS software itself. Generally, POS is used to record and manage sales transactions in a business, process transactions quickly and accurately, manage inventory, and provide useful data for business management. Modern POS systems are often cloud-based and can be accessed through various devices such as computers, tablets, and smartphones[2].

Given the business need for POS software, several vendors provide solutions for businesses such as Square, Shopify, Toast, Lightspeed, and KORONA POS. Each of these POS software has its own advantages and disadvantages[3]. In this case study, we researchers used KORONA POS because of its main features in terms of sales. KORONA POS can record sales transactions in cash, credit cards, and debit cards. KORONA POS is a cloud-based point of sale (POS) system software designed to meet the needs of various types of businesses, from retail to restaurants. KORONA POS offers advanced features such as inventory management, sales analysis, customer loyalty programs, and integration with various payment methods. The main advantage of KORONA POS is its ability to operate offline, so transactions can still be processed even if the internet connection is lost. Additionally, this software is easy to set up and customize, with an intuitive user interface and responsive customer support, making it a popular choice for business owners looking for reliable and flexible POS solutions[4].

The purpose of this research is to test the reliability and effectiveness of KORONA POS software in supporting business operations through usability testing methods. This research aims to ensure that the main features of KORONA POS, such as recording sales transactions in cash, credit cards, and debit cards, sales analysis, and software integration with printers, function well according to the promised specifications. This will provide an overview of the performance and suitability of KORONA POS with the business environment.

LITERATURE REVIEW

Usability testing is a testing technique that emphasizes the importance of ensuring that each software function meets the specified requirements. This method does not require a detailed understanding of the internal structure or source code of the application being tested. In POS systems, functional testing is conducted to verify that each crucial feature, such as sales, inventory management, and financial reporting, functions correctly. This testing involves inputting data into the system and checking whether the results match expectations. Thus, this testing helps identify errors that could potentially affect daily business operations[5] .

Point of Sale (POS) systems have various features and functions that are essential for the sustainability of retail businesses. Functional testing for POS includes testing individual modules and functions, such as transaction processing, sales recording, inventory management, and financial reporting. Functional testing also involves testing various usage scenarios that may occur in daily activities, such as discounts, product returns, and payments using different methods. These tests are conducted to verify the system's ability to handle all scenarios accurately and provide a stable and reliable user experience[6].

The importance of documenting software testing results in the testing process cannot be overlooked. This documentation contains details about the test cases conducted, the results achieved, and any issues that arose during the testing process. In POS system usability testing, the documentation of test results records each function tested, the input used, the output produced, and the match between the results and expectations. All bugs or errors found during testing must be documented in detail, including steps to reproduce the issue and its consequences on the system. Documentation is crucial to ensure accurate resolution of all identified issues and serves as a reference for future testing. Testing documentation is important to ensure that the POS system can operate well and meet user needs before being used directly in operations[7] .

METHODOLOGY

To ensure that each application can be used easily, the features and functions of the software operate according to the specified requirements. In this testing, verification and validation of components and modules will be conducted individually to ensure the system can perform its tasks without errors. This method facilitates comprehensive testing of all functional parts, such as inventory management, transaction processing, and sales reporting within the POS system. By thoroughly examining each interface and feature, the risk of disruptions or failures in operational activities can be reduced, which is crucial in the retail industry that emphasizes efficiency and reliability[8].

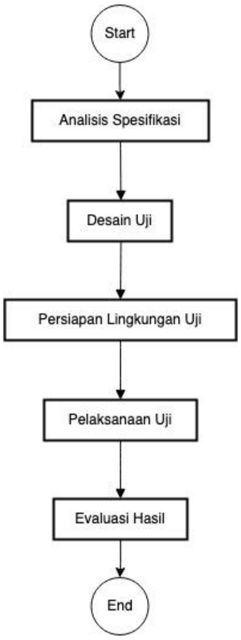


Figure 1. Usability Testing Steps

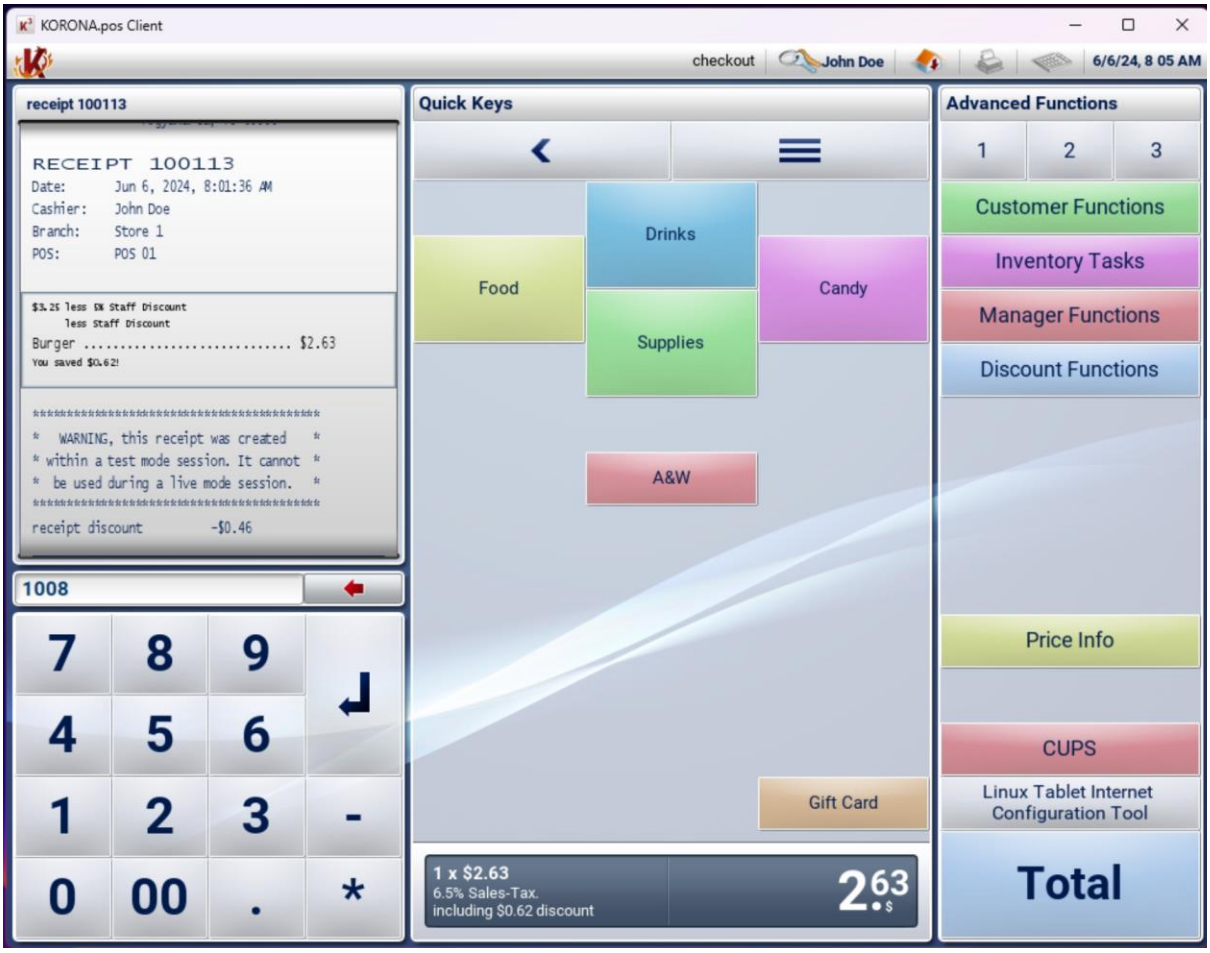
1) Specification Analysis: The initial stage in usability testing involves analyzing specifications, where the functions and features that need to be tested based on the software specification or requirement documents must be well understood. Each feature's specifications will be examined to ensure a clear understanding of what needs to be tested and the success criteria required[9]. This includes understanding the expected inputs, the steps to be taken, and the desired outcomes for each function. The purpose of this phase is to recognize the scope of testing and determine appropriate test cases for each aspect and performance of the system, ensuring that all user and business needs can be confirmed through testing[10].
2) Test Design: After the specification analysis is completed, the next step is to design the tests, where detailed test plans and test cases for each feature and function of the software will be developed. These test cases are designed to cover various situations, such as standard situations, boundary situations, and error situations. Test design includes identifying the inputs to be used, the steps to be followed in conducting the tests, and the desired outcomes to ensure the feature functions according to the specifications[11]. This also involves creating the necessary test data to perform the tests. This stage is crucial to ensure that all system functions are thoroughly tested and that testing can be conducted efficiently and effectively[12].
3) Test Environment Preparation: When preparing the test environment, it will be checked whether all necessary components are ready and functioning properly. This involves organizing the required hardware and software, installing the system to be tested, and preparing the planned test data [13]. The test environment should be replicated as closely as possible to the production environment to ensure accurate and relevant test results. The availability and functionality of tools and devices for monitoring test results will also be checked. The goal of this step is to create a stable and controlled situation where testing can be conducted consistently, so that test results can be trusted and issues can be detected quickly[15].
4) Test Execution: Test execution is the stage where the designed test cases are run on the system being tested. In this step, test data will be entered into the system, and the steps specified in the test cases will be followed to verify that the functions and features operate according to the specifications[16]. The results of each test are recorded, including whether the results match expectations or if there are discrepancies that require further investigation. Documentation of any errors or bugs found during testing will be conducted[17]. This stage is important to identify and fix issues in the system before the software is released, ensuring that the software can function well under actual operational conditions[18].

5) Result Evaluation: Result evaluation is the final step in functional testing to ensure that all features and functions have been correctly tested and that the system operates according to the specified requirements. The test results will be reviewed to ensure all test cases have been completed and any issues have been addressed. This also involves evaluating the root causes of any detected errors and providing recommendations for improvements to prevent similar issues in the future. Result evaluation is useful to check whether the software meets the desired quality standards and is ready for use by users[19].

RESULTS AND DISCUSSION

KORONA POS provides key features in terms of sales, including the ability to record sales transactions in cash, credit cards, and debit cards. In terms of inventory, KORONA POS is equipped with features to track inventory, stock items, prices, and storage locations. For customers, KORONA POS can store customer information such as names, addresses, and transaction histories[20]. In terms of reporting, KORONA POS can generate sales data reports, inventory, and customer reports. One standout feature of KORONA POS is that it can be accessed via the web, allowing users to set up terminals for each store, download the client desktop application for the computer, and connect the KORONA client application with hardware such as printers and scanners[21].

For the testing conducted in this research, the features used were sales transactions and hardware integration.



```
RECEIPT 100113
Date:       Jun 6, 2024, 8:01:36 AM
Cashier:    John Doe
Branch:     Store 1
POS:        POS 01

$3.25 less 5% Staff Discount
      less Staff Discount
Burger ............................ $2.63
You saved $0.62!

*****************************************
*   WARNING, this receipt was created   *
* within a test mode session. It cannot *
*  be used during a live mode session.  *
*****************************************
receipt discount          -$0.46
```

Figure 2. KORONA POS Interface

Figure 2 shows the user interface of the KORONA POS application used to manage sales transactions in stores. In the upper left corner, detailed transaction information is displayed, such as receipt number, date, time, cashier name, store name, and POS number. Below it, there are details of the purchased items, including name, price, and applied discounts. In the middle of the screen, there is a "Quick Keys" button that allows quick access to product categories such as food, beverages, supplies, candy, and specific items like "A&W." On the right side of the screen, there is an "Advanced Functions" panel consisting of various buttons that direct to advanced functions, such as customer functions, inventory tasks, manager functions, discount functions, price information, and internet configuration for Linux tablets. The numeric keypad, navigation buttons, and confirmation buttons are located at the bottom left. In the middle bottom, detailed information about the current transaction is displayed, including the total items, sales tax, and overall cost after discounts. The total amount to be paid is listed at the bottom right.

Table 1. Sales Transaction Testing Scenarios

Test Scenario ID	Description	Test Case ID	Pre Condition	Steps to Execute	Expected Result	Actual Result	Status	Executed By
1	Handling transactions using KORONA POS	1.1 True	The POS display appears with a large total at the bottom, item category options, numpad, receipt display, and functions like remove item, change item price, change quantity, cancel receipt, and log out	1. Select item category or enter item code directly on the numpad	Items can be selected through item categories or by entering item codes using the numpad	Items appear according to the selected category. Items are found if the entered item code is correct	PASSED	Philip and Charles
				2. Select menu according to the selected item category	Can select items according to the selected category	Only items that match the category can be selected	PASSED	Philip and Charles
				3. Select the remove item option (optional)	Can delete items on the receipt	Only items that match the category can be selected	PASSED	Philip and Charles
				4. Select the option to change item price (optional)	Cannot change the price of items already in the software	Cannot change the price of the selected item	PASSED	Philip and Charles
				5. Select the option to cancel receipt (optional)	Can delete all items already registered on the receipt display	Can delete all items registered on the transaction receipt	PASSED	Philip and Charles
				6. Press the logout button (optional)	Can log out the logged-in employee account	Can log out the logged-in employee account, but if logged in again, the items already added to the transaction are not lost	PASSED	Philip and Charles
				7. Press the total button (optional)	Can display the total amount to be paid and the amount paid by the customer. There are	Can see the total and enter the amount given by the customer and choose other	PASSED	Philip and Charles

					other payment method options	payment options		
		1.2 False	The POS display appears with a large total at the bottom, item category options, numpad, receipt display, and functions like remove item, change item price, change quantity, cancel receipt, and log out	Select item category or enter item code directly on the numpad using numbers mixed with alphabets, unique characters, and symbols	If the entered item code is not registered, or uses symbols, alphabets, and unique characters, a notification will be received that the item is not registered	The entered item code is not found	FAILED	Philip and Charles

The analysis of the sales transaction process shows that the KORONA POS software was tested through a series of scenarios to ensure each function works according to specifications. In the first scenario, the goal was to handle transactions using KORONA POS, with several steps to be followed. Each step had initial conditions and detailed execution steps. The expected results were clearly defined, and the actual results were recorded for each step. All sub-tests in the first scenario, such as selecting item categories, deleting items, changing item prices, canceling receipts, logging out, and displaying total payments, were declared successful (passed) because the actual results matched the expected ones. The second scenario tested the system's response to invalid inputs, such as item codes containing numbers mixed with alphabets and unique symbols. The expected result was that the system would provide a notification that the item was not registered, but the actual result showed a failure (failed), indicating that the system did not handle invalid inputs correctly. This analysis shows that while KORONA POS functions well in normal scenarios, there are weaknesses in handling invalid inputs that need to be addressed.

After analyzing the testing of sales transactions with KORONA POS, the following is the analysis of the driver integration process with hardware (printer):

Table 2. Hardware Integration Testing Scenario

Test Scenario ID	Description	Test Case ID	Pre Condition	Steps to Execute	Expected Result	Actual Result	Status	Executed By
2	Integrating KORONA POS software with the printer	2.1	Printer is on	1. Connect the printer to the device	The device can read the connected printer	The device can read the connected printer	PASSED	Philip and Charles
				2. Open KORONA POS software	KORONA POS software can read the	The software can read the printer connected to	PASSED	Philip and Charles

					printer connection	the device		
		2.2	Printer driver not connected	Select printer connection with device driver	KORONA POS software will receive a notification that the driver cannot be connected	KORONA POS software cannot connect to the printer	PASSED	Philip and Charles
3	Printing sales transaction receipts	3.1	Sales transaction has been conducted and recorded	Select the printer icon on KORONA POS	KORONA POS will send the transaction receipt to the printer	The printer can print the sales transaction receipt	PASSED	Philip and Charles
					The printer will receive and can print the sales receipt			Philip and Charles

The hardware integration analysis shows that the KORONA POS software was tested for integration with the printer and the ability to print receipts. The second scenario emphasized the process of integrating the software with the printer. In one of the sub-tests, with the initial condition of the printer being on, the steps taken included connecting the printer to the device and opening the KORONA POS software. The expected result was that the device could read the connected printer, and the actual result matched the expectation, so this test was declared successful (passed). The next sub-test tested the condition where the printer driver was not connected. The step taken was to try to connect the printer with the device driver, and the expected result was that the KORONA POS software would receive a notification that the driver could not be connected. The actual result showed that the software could not connect to the printer, so this test was also declared successful (passed). The third scenario tested the ability to print sales transaction receipts. In the last sub-test, with the initial condition that the sales transaction had been conducted and recorded, the steps taken included selecting the printer icon on KORONA POS to send the receipt to the printer. The expected result was that the printer could print the sales transaction receipt, and the actual result matched the expectation, so this test was declared successful (passed). This analysis shows that KORONA POS functions well in scenarios of integration with the printer and receipt printing, ensuring that this system can effectively support business operations.

CONCLUSION

The results of the KORONA POS testing show that this system has strong quality to meet basic needs in sales, inventory, and hardware integration. The tests on the sales process demonstrate that KORONA POS can effectively handle item selection, transaction processing, and connectivity with various payment methods. However, the detected weakness in handling invalid inputs indicates that the system can be improved in terms of resilience to non-standard inputs. At the same time, interactions with hardware such as printers have proven effective, with KORONA POS able to print sales transaction receipts without interruption.

In conclusion, the implementation of KORONA POS in a retail environment should be considered due to its success in transaction management and essential hardware integration. Although there is still room for improvement in handling invalid inputs, core functions such as inventory management, transaction tracking, and receipt printing run smoothly. Similarly, by making certain adjustments to the response to invalid inputs, KORONA POS can become a reliable solution to support operational efficiency and accuracy in today's retail environment.

REFERENCES

[1] A. R. Mendoza, "POINT OF SALE SYSTEM WITH INVENTORY FOR ARM'S FOOD AND DELICACIES," *International Journal of Advanced Research in Computer Science*, 2019, doi: 10.26483/IJARCS.V10I2.6378.

[2] I. A. Dewi, Y. Miftahuddin, M. A. Fattah, C. B. Palenda, and S. F. Erawan, "Sistem Point of Sales Pada Website InHome Café Menggunakan Metodologi Agile," *Jurnal Inovasi dan Keterlibatan Komunitas*, doi: 10.28932/JICE.V1I1.3321.

[3] P. Raharjo, W. Ananta Kusuma, and H. Sukoco, "UJI USABILITY DENGAN METODE COGNITIVE WALKTHROUGH PADA SITUS WEB PERPUSTAKAAN UNIVERSITAS MERCU BUANA JAKARTA."

[4] M. Rifki Maulana, N. Rahaningsih, and D. Pratama, "ANALISIS USABILITY APLIKASI POINT OF SALES (POS) BERBASIS WEB MENGGUNAKAN METODE SYSTEM USABILITY SCALE (STUDI KASUS: WARUNG BUYUT SEMAR)," 2023.

[5] A. Kibet, S. Karume, and N. Masese, "Functional Testing on A software Model For Creating A Stable Cryptocurrency Using Fiat Currency For Global Electronic Commerce," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 10, pp. 236–240, Oct. 2022, doi: 10.22214/ijraset.2022.46682.

[6] Y. Shi, M. Gillenson, and X. Zhang, "Value Estimation of Software Functional Test Cases," *J. Database Manag.*, vol. 33, pp. 1–33, 2019, doi: 10.4018/jdm.299559.

[7] H. Tang and S. Nadi, "Evaluating Software Documentation Quality," *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, pp. 67–78, 2023, doi: 10.1109/MSR59073.2023.00023.

[8] S. R. Wicaksono, *Usability Testing*. Zenodo, 2023. doi: 10.5281/zenodo.7705056.

[9] R.-F. Wang, Y. Sato, and S. Liu, "Specification-based Test Case Generation with Genetic Algorithm," *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1382–1389, 2019, doi: 10.1109/CEC.2019.8790233.

[10] S. Masuda, Y. Nishi, and K. Suzuki, "Complex Software Testing Analysis using International Standards," *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 241–246, 2020, doi: 10.1109/icstw50294.2020.00049.

[11] D. A. Hasan, B. K. Hussan, S. R. M. Zeebaree, D. M. Ahmed, O. S. Kareem, and M. A. Sadeeq, "custom: The Impact of Test Case Generation Methods on the Software Performance: A Review," vol. 5, pp. 33–44, 2021, doi: 10.5281/ZENODO.4623940.

[12] Y. Aoyama, T. Kuroiwa, and N. Kushiro, "Executable Test Case Generation from Specifications Written in Natural Language and Test Execution Environment," *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, 2021, doi: 10.1109/CCNC49032.2021.9369549.

[13] M. Aniche, C. Treude, and A. Zaidman, "How Developers Engineer Test Cases: An Observational Study," *IEEE Transactions on Software Engineering*, vol. 48, pp. 4925–4946, 2021, doi: 10.1109/TSE.2021.3129889.

[14] P. S. Kochhar, X. Xia, and D. Lo, "Practitioners' Views on Good Software Testing Practices," *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 61–70, 2019, doi: 10.1109/ICSE-SEIP.2019.00015.

[15] E. Aghajani *et al.*, "Software Documentation: The Practitioners' Perspective," *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 590–601, 2020, doi: 10.1145/3377811.3380405.

[16] Z. Peng, T. Chen, and J. Yang, "Revisiting Test Impact Analysis in Continuous Testing From the Perspective of Code Dependencies," *IEEE Transactions on Software Engineering*, vol. 48, pp. 1979–1993, 2022, doi: 10.1109/tse.2020.3045914.

[17] P. Jung, S. Kang, and J. Lee, "Reducing Redundant Test Executions in Software Product Line Testing—A Case Study," *Electronics (Basel)*, p., 2022, doi: 10.3390/electronics11071165.

[18] E. Jabbar, H. Hemmati, and R. Feldt, "Investigating Execution Trace Embedding for Test Case Prioritization," *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS)*, pp. 279–290, 2023, doi: 10.1109/QRS60937.2023.00036.

[19] F. Pudlitz, F. Brokhausen, and A. Vogelsang, "What am I testing and where? Comparing testing procedures based on lightweight requirements annotations," *Empir Softw Eng*, vol. 25, pp. 2809–2843, 2020, doi: 10.1007/s10664-020-09815-w.

[20] F. G. N. P. Putra, Jaswadi, and L. Djajanto, "Design and build a Point of Sales (POS) application model using the agile method for super mama frozen food," *World Journal of Advanced Engineering Technology and Sciences*, 2023, doi: 10.30574/wjaets.2023.9.2.0212.

[21] M. D. M. Tatuhas and S. S. Wulandari, "Sales Planning and Strategy for Enterprise Resources Planning Application at PT. Kolega Moeda Berkarja," *Management Research Studies Journal*, 2023, doi: 10.56174/mrsj.v4i2.550.