Journal of Software Engineering Ampera
Vol. 3, No. 2, Junw 2022 e-ISSN: 2775-2488 Published By APTIKOM SUMSEL
https://journal-computing.org/index.php/journal-sea/index

Testing of Integrated Sales Information Systems On UD. Remaja Bumi Raya Based on Black Box Testing with Equivalence Partitioning Method

Muhammad Arifin[1], Fery Ariyana[2], Anteng Widodo[3]

[1,2,3]Information System Departement, Muria Kudus University, Jawa Tengah, Indonesia
Email: arifin.m@umk.ac.id[1], 201753094@std.umk.ac.id[2], anteng.widodo@umk.ac.id[3]

Abstract

UD. Remaja Bumi Raya is a business in the field of selling agricultural materials that provides various types of fertilizers and agricultural medicines. This UD has developed a sales information system that is integrated into stock management and transactions connected between branches so that owners can find out the number of stocks and transactions from each branch in real time. Black Box testing is an application test that covers the outer side of a software, from the display to the input action, while the Equivalence Partitioning method is a testing method in the aspect of input validation from the Valid Class side, observation of input content and input accuracy, so that the error side of the application being tested can be known. The purpose of this study is to determine the quality and errors in the Integrated Sales Information System at UD. Remaja Bumi Raya by using Black Box Testing with the Equivalence Partitioning method. The results of the study show that the selling information system is feasible to implement because it has a minimal error rate. The use of Black Box Testing with the Equivalence Partitioning method can be used to test the software before it is implemented.

Keywords: Sales Information System, Integrated, Black Box Testing, Equivalence Partitioning.

1. INTRODUCTION

UD. Remaja Bumi Raya is a trading business that sells agricultural materials. The business process that is running is still not optimal because all processes are still carried out manually. The sale of goods still uses notes so that there are many note files that take a long time to recap the sale and have the risk that the notes may be tucked away, damaged, and even lost. Because the process that is still manual also makes it difficult for this business to get sales reports and stock reports, it is also difficult to make decisions about inventory, for example in knowing what sales of goods customers often buy. The research

conducted by Asfinoza, discussed PT. Sri Aneka Karyatama there were several problems such as the distance between customers and the company in the fertilizer order process, errors in recording orders, requiring a longer time in making reports, and increasing competitors so that this company needed promotional media. This research confirms the existence of a Fertilizer Sales Information System [1]. Another related study was the one conducted by Dewantara who said UD. Tani Dadi requires the use of information systems for their business, for example as a system for processing transaction data into information in accordance with the needs of consumers, cashiers and administrators more effectively and efficiently [2]. Rahmawati in her research produced an information system for Food Sales and Ordering at The Web-Based Nisma Rasa Resto, which aims to produce a sales and ordering information system that makes it easier for consumers to place orders because so far Nisma Rasa Resto has not utilized an information system in its transaction activities so that there are often problems during the ordering process, lack of information about the food menu and food details as well as reporting that still uses manual recording which causes the risk of losing sales data [3].

Meanwhile, the research conducted by Salim argues that the Sales Management Information System that takes case studies in Kuwawur Village, several important components that are involved and interact with each other in trade are: suppliers, products or services and customers as the goal. Currently, large and small and medium-sized companies urgently need accurate and fast data processing to increase product sales, promotional means and maintain product price stability [4]. After the software is built before it is implemented, it is necessary to carry out a testing process, Software testing is very necessary to ensure that the software that has been or is being created can run according to the expected functionality. Software developers or testers should set up a special session to test the program that has been created so that errors or deficiencies can be detected in the first place and corrected as soon as possible.

Testing itself is a critical element of software quality assurance and is an inseparable part of the software development life cycle as is the case with analysis, design, and coding. Software testing must be carried out in the process of software engineering. Several software testing strategies have been proposed in the literature. All of them provide templates for testing for software creators. In this case, everything should have general characteristics in the form of 1) Testing starts at the module level and works outwards towards

integration on computer-based systems, 2) Different testing techniques according to different points in time, 3) Testing is held by software makers and developers and for large projects by independent testing groups.

Testing and Debugging are different activities, but debugging must be accommodated on each testing strategy. If you want to implement a successful software testing strategy Tom Gilb suggests that the procedures used are: 1) Determine all the needs of the software product in the calculation before starting testing, 2) The status of the testing object must be clear, 3) Understand the software user and develop a profile for each user category, 4) Develop a testing plan that emphasizes the "rapid cycle testing", 5) Build the perfect software designed to test itself alone, 6) Using a formal review as a filter before testing, 7) Conducting a formal review to assess the test strategy and the test case itself, 8) Developing a continuous improvement approach to the testing process. The black box method is a method commonly used to test a program by not having to pay attention to every detail of the program to be tested. In this black box test, it only checks the value of each input. And there is no step or attempt to search for the program code used for output.

The advantage in using the Black Box method is that in the implementation of its tests it is not necessary to have a deep knowledge of a particular programming. Testing is carried out from a user point of view so that programmers and testers are both interdependent on each other [5] Equvalence Partitioning (EP) is one of the many Black Box Testing testing techniques and this research will use this technique, to test the input process, we can divide an input into groups based on its function, so that later we will get an accurate test case. Equivalent partitions themselves are dividing inputs into data classes to generate test cases. Testing using the equivalent partitioning technique can be done on a form that has been created and has been contained in the book sales information system then by inputting a data that does not match the data type or by using random data [5]. There will be many stages in this study, where the first thing to do is to determine the input form to be tested and then determine the value or sample input into the form based on the equivalent partition technique after that determine the Test Case software to be tested where here the software tested is a website-based book sales information system, after specifying the Test Case then initialize the standard grade partition input and output. The goal is so that we can produce a data set in the form of documentation. testing with the Equivalent partitions

method and the value of the effectiveness level of the Equivalent partitions method [6].

2. METHODS

The first thing to do is to determine the input form to be tested then determine the value or sample input into the form based on the equivalent partition technique after that determine the Test Case software to be tested where here the software tested is a website-based book sales information system, after determining the Test Case then initialize the standard grade partition input and ouput. The goal is so that we can produce a data set in the form of documentation. testing with the Equivalent partitions method and the value of the effectiveness level of the Equivalent partitions method. The Test Case design table which will later be used to check the program whether it is in accordance with the desired needs or there are still some errors or errors that occur so that improvements are still needed so that the quality of the program created will improve and in accordance with what is desired.

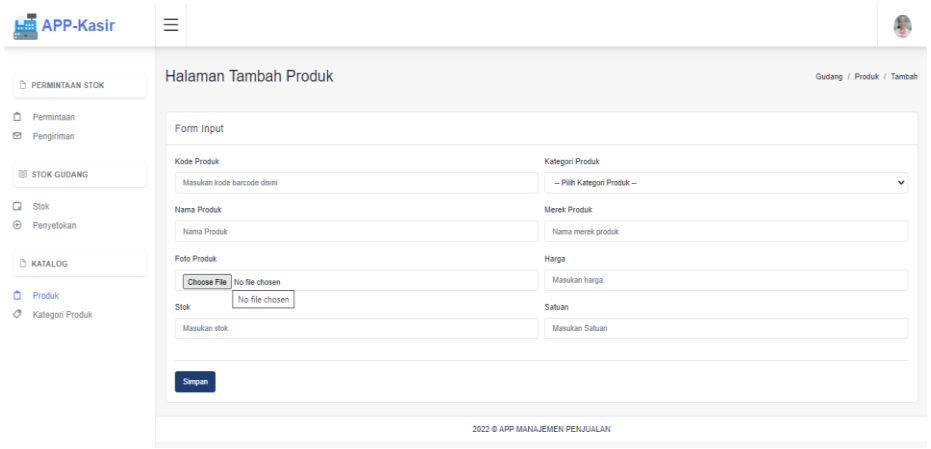

Figure 1. Form Insert Product

The form above has several fields that must be filled in so that product additions can be done, namely filling in the product code, product name, product photo, stock amount, product category, product brand, price and number of units. The SAVE button works so that the entered data can be stored in the database. To test each field in the form above, several test scenarios have been prepared. Especially for product photo fields and product categories, they

are not tested because they include the types of elements in the form that are filled with photo files and product categories have been set on other forms. The product code field can only be filled with 10 digits of letters, numbers and spaces. Based on the Boundary Value Analysis Technique, for product code fields, a test case is made as in Table 1. The product name field can only be filled with 20 digits of letters, numbers, and spaces. Based on the Boundary Value Analysis Technique, for the product name field, a test case is made as in Table 2. Stock fields can only be filled with a maximum of 5digit numbers. Based on the Boundary Value Analysis Technique, for stock fields, a testcase is made as in Table 3. The product brand field can only be filled with 30 digits of letters, numbers, and spaces. Based on the Boundary Value Analysis Technique, for the product brand field, a testcase is made as in Table 4. The price field can only be filled with a maximum of 8digit numbers. Based on the Boundary Value Analysis Technique, the price field is made a testcase as in Table 5. Unit fields can only be filled with a maximum of 6digit numbers. Based on the Boundary Value Analysis Technique, for unit fields, a testcase is made as in Table 6.

Table 1. Test Case Table for Product Code Fields

ID	Test Scenarios	Expected results
TC01	Entering the character "abc12345" the next thing to do is click SAVE.	The data is successfully stored in the database.
TC02	Entering the character "abc-12345" the next thing to do is click SAVE.	Can store "-" characters and data successfully stored in the database

Table 2. Test Case Table for Product Name Field

ID	Test Scenarios	Expected results
TC03	Entering the character "Herbicide" the next thing to do is click SAVE.	The data is successfully stored in the database.
TC04	Entering the character "Herbicide1234567891011" the next thing to do is click SAVE.	Refuse to store such data in the database because the number of characters exceeds the limit.

Table 3. Test Case Table for Stock Field

ID	Test Scenarios	Expected results
TC05	Entering the character "20000" next all that has to be done is click SAVE.	The data is successfully stored in the database.
TC06	Entering the character "200@" the next thing to do is click SAVE.	Refuses to store the data in the database because it can only hold numerical characters.

Table 4. Test Case Table for Product Brand Field

ID	Test Scenarios	Expected results
TC07	Entering the character "farm source" the next thing to do is click SAVE.	The data is successfully stored in the database.
TC08	Entering the character "source & tani" next all that has to be done is click SAVE.	Can store "&" characters and data is successfully stored in the database.

Table 5. Test Case Table for Price Field

ID	Test Scenarios	Expected results
TC09	Entering the character "50000" next all that has to be done is click SAVE.	The data is successfully stored in the database.
TC10	Entering the character "500#" the next thing to do is click SAVE.	Unable to save the character "#" and the data was not successfully stored in the database.
TC11	Entering the character "500F" next all that must be done is click SAVE.	Unable to save character "F" and data not successfully stored in the database.

Table 6. Test Case Table for Unit Fields

ID	Test Scenarios	Expected results
TC12	Entering the character "pcs" next all that has to be done is click SAVE.	The data is successfully stored in the database.
TC13	Entering the character "/pcs" the next thing to do is click SAVE.	Can store the character "/" and the data is successfully stored in the database.

3. Results and Discussion

Integrated Sales Information Systems On UD. Remaja Bumi Raya sales activities, purchases, listings, etc. In this test, one of the functions, namely "Add Product" which is felt to be able to represent other functions in the system. Based on the test case table that has been created, it is continued with testing the system. The results of the tests of the systems carried out are shown in Table 7.

Table 7. Test Results

ID	Test Scenarios	Expected results	Test Results	Conclusion
TC01	Entering the character "abc12345" the next thing to do is click SAVE.	The data is successfully stored in the database.	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC02	Entering the character "abc-12345" the next thing to do is click SAVE.	Can store "-" characters and data successfully stored in the database	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC03	Entering the character "Herbicide" the	The data is successfully	The data is successfully saved in the database and	Appropriate

	next thing to do is click SAVE.	stored in the database.	the message "Saved Data" appears	
TC04	Entering the character "Herbicide1234567891011" the next thing to do is click SAVE.	Refuse to store such data in the database because the number of characters exceeds the limit.	Data failed to be successfully stored in the database and an "Error" message appears	Appropriate
TC05	Entering the character "20000" next all that has to be done is click SAVE.	The data is successfully stored in the database.	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC06	Entering the character "200@" the next thing to do is click SAVE.	Refuses to store the data in the database because it can only hold numerical characters.	Data failed to be successfully stored in the database and an "Error" message appears	Appropriate
TC07	Entering the character "farm source" the next thing to do is click SAVE.	The data is successfully stored in the database.	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC08	Entering the character "source & tani" next all	Can store "&" characters and data is successfully	The data is successfully saved in the database and	Appropriate

	that has to be done is click SAVE.	stored in the database.	the message "Saved Data" appears	
TC09	Entering the character "50000" next all that has to be done is click SAVE.	The data is successfully stored in the database.	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC10	Entering the character "500#" the next thing to do is click SAVE.	Unable to save the character "#" and the data was not successfully stored in the database.	Data failed to be successfully stored in the database and an "Error" message appears	Appropriate
TC11	Entering the character "500F" next all that has to be done is click SAVE.	Unable to save character "F" and data not successfully stored in the database.	Data failed to be successfully stored in the database and an "Error" message appears	Appropriate
TC12	Entering the character "pcs" next all that has to be done is click SAVE.	The data is successfully stored in the database.	The data is successfully saved in the database and the message "Saved Data" appears	Appropriate
TC13	Entering the character "/pcs" the next thing to do is click SAVE.	Can store the character "/" and the data is successfully stored in the database.	The data is successfully saved in the database and the message	Appropriate

"Saved Data"
appears

4. CONCLUSION

Based on the testing process that has been carried out, several conclusions can be obtained, namely that the documentation of the test results is easier to observe. Test results for the functionality of each form are 100% successful. Testing using the Black Box Testing method along with the relevant Boundary Value Analysis technique to be applied to the sales application by determining the number of minimum limits and maximum limits of the data structure that has been designed.

REFERENCES

[1] Asfinoza dan Puspasari. S. (2018). "Sistem Informasi Penjualan Pupuk Berbasis Web Pada PT. Sri Aneka Karyatama". Jurnal Media Infotama Vol. 14 No. 1, Februari 2018.

[2] Dewantara. P dan Nugrahanti. F. (2018). "Rancang Bangun Sistem Informasi Pembelian dan Penjualan Pupuk Bersubsidi Berbasis Web Pada UD. Tani Dadi" Jurnal Seminar Nasional Teknologi Informasi dan Komunikasi 2018.

[3] Rahmawati, S. (2017). "Sistem Informasi Penjualan Dan Pemesanan Makanan Pada Nisma Rasa Resto Berbasis Web". Skripsi Program Studi Sistem Informasi, Universitas Muria Kudus.

[4] Salim. (2020) "Sistem Informasi Manajemen Penjualan Buah Berbasis Web" Skripsi Program Studi Teknik Informatika, Universitas Muria Kudus.

[5] Hidayat, T., & Putri, H. D. (2019). Pengujian Portal Mahasiswa Pada SistemInformasi Akademik Menggunakan Black Box Testing Dengan Metode Equivalence Partitioning dan Boundary Value Analysis. JUTIS Vol.7 No.1, 83 -92.

[6] Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. Jurnal Informatika Universitas Pamulang Vol.4 No.2, 125 -130.