International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

Vol. 39 No. 2 July 2023, pp. 212-218

Performance And Functional Testing With The Black Box Testing Method

[1]Putri Kusuma Ayuningtyas, [2]Dwi Atmodjo WP, [3]Pratiwi Rachmadi

[1]putrikusuma@perbanas.id;[2] dwi.atmodjo@perbanas.id; [3]pratiwi@perbanas.id

Faculty of Information Technology

IKPIA Perbanas Jakarta Indonesia

Abstract – Software testing is an essential stage in software development because it plays a role in maintaining the quality of the software produced. The objective of software testing is for verification and validation, namely ensuring that the software being developed works according to the expected functions when the software is designed. In addition to improving quality, the software being developed must also have minimal errors and test whether the software is feasible to use or not. This research used black box testing method. Black box technical testing was a test of the input or output functionality of the software. The research also described the testing model using automated testing, which was carried out by Apache JMeter tools on performance tests and Katalon Studio to functional tests on the Perbanas portal. The final results of this research depicted the test model from studying business processes, creating test scenarios for automated testing to analyzing the test results. The performance test results aimed to find out how much throughput (transactions per second) and Response Time are. Meanwhile, functional test results aimed to find out whether the expected functions and outputs were appropriate or not. The results of the performance test on the login feature that had been carried out, the best was taken, i.e., sample data that entered approximately 5000 samples for 250 users for a load test per 5 minutes; only 0.06% failed with an average of 15 transactions per second (TPS) by a maximum response time of 68 seconds and a minimum response time of 0.2 seconds.

Keywords – Apache JMeter, automation test, black box test, functional test, Katalon Studio, performance test

I. INTRODUCTION

Along with the development of technology, the number of users of web services is increasing. It causes popular websites to have a high amount of *traffic.* Certain activities or events can also trigger an increase in the amount of web traffic of an organization, e.g., during *an online*-based college or school admissions period, *online* KRS period, and other events that can cause a significant increase in web traffic at a particular time. Traffic growth causes the work of servers that served requests to become increasingly heavy. It results in decreased server performance and frequent interruptions to these web services. If it is untreated, thus it can cause the system or website to be *down.*

The Faculty of Information Technology at the Perbanas Institute has a website that provides information on activities used by students, lecturers, academic admins, and financial admins. The website address of the engineering faculty is *http://portal.perbanas.id.* Perbanas Academic Portal is an information system that acts as an integrator of academic information in various academic units (study programs/ faculties) as well as a means of communication between campus academics. This system was built to integrate educational information to facilitate public access for students and lecturers such as lecture information, financial information, grade transcripts, KRS, the security and accuracy of data and information must be maintained appropriately.

Corresponding Author: Putri Kusuma Ayuningtyas

Testing or measuring the quality of the software is required so that it can be seen to what extent the reliability of a system can be measured. One software testing process uses *performance and functional* tests with *automated testing.* The tests carried out were on the KRS as known as "*Study Plan Card*" and the Perbanas *web portal login*, which sometimes still employs a lot of manual testing, resulting in a lack of time and energy efficiency from the team working on the test. Based on the background above, the researcher conducted research by creating *automated testing that* can be used to test the main functions while determining whether the data or user interface has bugs or not. Functional testing tools automatically use Katalon Studio. Meanwhile, performance test tools use Apache JMeter to see whether the server experiences stress/ problems on a heavy load, e.g., when many users access the portal. Both functional and performance tests for the Perbanas Portal web application are deemed necessary to determine server performance and main portal functions.

II. LITERATURE

Software testing is the process of finding *errors* in each software item, recording the results, evaluating every aspect of each system component, and evaluating all the facilities of the software being developed. There are two significant things on tests, verification and validation. Verification is evaluating a system/ component to determine whether a product completed after the development phase meets the conditions set at the beginning of development when determining software specifications. Meanwhile, validation is the process of evaluating a system or component at the end or during the development period to determine whether the resulting product meets certain needs and requirements requested by the user.

Automated testing is where the tester writes scripts and uses other software to perform tests (Galin, 2009, pp. 235-241). Automated testing is employed to quickly and repeatedly rerun test scenarios and is excellent for extensive software testing such as regression testing by attempting as many possible tests as possible that humans can err (Grm & Grm, 2021; Hao, 2021). This testing includes regression and automation testing and is also employed to test application *load,* performance, and stress testing. It can increase the test's scope, which will impact accuracy and effectiveness compared to manual testing. *Automated tests* are not used to test everything in the software but rather emphasize user activity on a larger scale and be carried out repeatedly.

An automated test process can be done using a supporting programming language (Mendiratta & Kumar, 2016). Several tools are available and can be used to write automation scripts. Before selecting tools, it is necessary to identify processes that can be applied for automated tests, such as identifying areas to be automated, selecting tools for automated testing, designing test cases, writing test scripts, developing test suites, running scripts, generating reports, and identifying any bugs or performance issues.

The devices that can be used for automated tests include Katalon Studio and Apache JMeter (Mendiratta & Kumar, 2016; Psujek et al., 2021). Testing is carried out to meet the software quality requirements by executing the program to find out *program syntax errors,* and verifying the software to see the compatibility between the software and the wishes of the customer or *user* (Danendra, 2021; Sasmito & Nishom, 2020).

Black box testing is a complementary approach to the *white box technique* because black box testing is expected to reveal a wider class of errors than the white box technique. Black box testing focuses on testing software functional requirements to obtain a set of input conditions that match the functional requirements of a program (Cholifah et al., 2018; Sasmito & Nishom, 2020).

Black box testing is testing the fundamental aspects of the system without noticing the internal logic structure of the software. This method is employed to find out if the software is working properly. *Black* box testing a method of designing test data based on software specifications. The test data is generated and executed on the software and then the output from the software is checked to determine whether it is as expected. Black box testing attempts to find errors in categories.

System testing, the behavior of the entire system/ product must be considered and has been carried out in the development process. Testing is commonly carried out when the development process is complete to verify the system/ product that qualifies specifications to find as many defects as possible. Usually, this test is carried out by a tester who is familiar with and has specialized in the system. During the testing process, defects found must be immediately reported to the development manager or project manager (Grm & Grm, 2021; Sasmito & Nishom, 2020). System testing can investigate product defects through two testing levels for software, *functional* and *non-functional testing.*

Performance testing is widely used to show and verify common performances or problems. It is not intended to show application errors (bugs) in software (Costa et al., 2020; 'Software Performance Testing', 2020). Several matters frequently become problems

in decreasing application software performance, such as network delay, clienst-side processing, data base transactiona processing, load balancing between servers, and data rendering.

Performance testing can be in the form of qualitative or quantitative testing activities and can be divided into sub-types of testing such as *Load and Stress testing. Load testing* is a testing process that tests software behavior by applying the maximum load in terms of accessing applications and manipulating large amounts of input data. It can be applied under normal and peak load conditions. The test type identifies the maximum capacity of the software and its peak behavior. Commonly, load testing employs tools to carry out tests automatically. The *virtual users* (VUsers) of the *automation tools* are defined in the automated test tools and scripts that are run to verify load testing of the software. The number of users can be manipulated according to needs, either increasing or decreasing.

Meanwhile, *stress testing* is a process of testing software behavior under abnormal conditions. This test uses resources that exceed the normal limits of the *software load.* Therefore, this test is referred to as *stress testing.* The main objective is to test the software by applying a load to the system and taking the resources normally used by the software to identify shutdown points. Stress testing can be done from automation tools such as Apache JMeter with scripts that are run to test the load on the *software/database* with a large number of users.

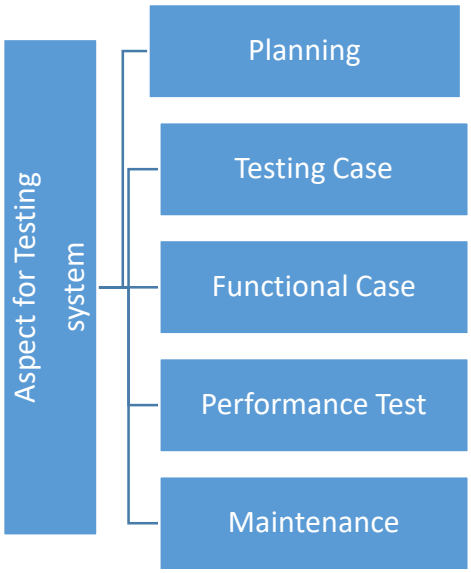

Figure 1. Aspect for Testing System

Katalon Studio is a free automation testing solution developed by Katalon LLC. The software is built on the open-source automation framework Selenium and Appium with a dedicated IDE interface for API, web, and mobile testing. The test automation framework provided in Katalon Studio is developed with a keyword-based approach as the main method of writing tests with data-driven functionality for test execution. The user interface is a complete integrated development environment (IDE) implemented on a client platform such as Eclipse (RCP).

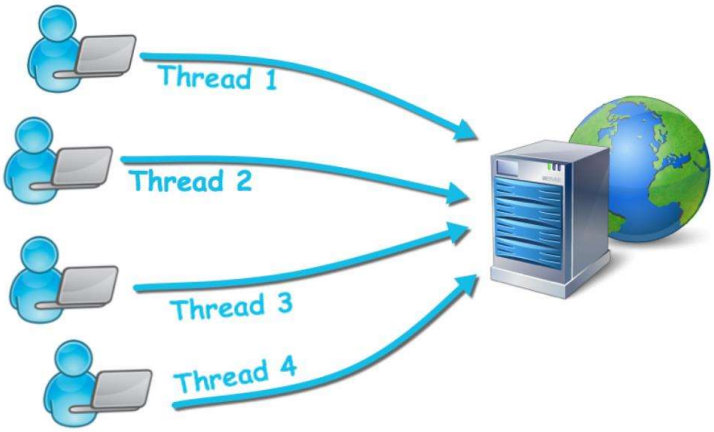

Figure 2. Thread Group Architecture

The thread group element is the initial of any test plan. As the name implies, the thread group (users)controls the number of users JMeter will be used to run tests. Hence, the thread group element is crucial. Usually, only a few thread group configurations are used, such as the number of threads (users) or the number of loops and ramp-up period (in seconds). Indeed, it was enough to run the test. However, other configurations are also useful and might help with testing in some cases.

III. Problem Definition/ Experimental Work

This paper will focus on Automation testing in the form of functional and performance testing on the Perbanas academic portal using the black box testing method. The business process from the Perbanas Portal is studied first, the Requirement file, the process flow, and the user interfaces display. The next stage is creating a testing scenario, making test cases, and recording scripting in software testing Katalon Studio, and is followed by validating each test case as the fulfillment requirements of the functional testing process. The successful or failed tests can be seen from the script validation and the results can be integrated directly into the katalon analytic platform. The Apache JMeter software is used for performance tests. It is similar to the functional test for performance tests; there is also script validation so that the test can verify whether the test has failed or not. The performance test is run with a non-GUI and the report is automatically generated.

When carrying out functional testing, if a bug is found, it will be recorded in the Bug List, which will be submitted to the IT team. If there are no bugs during testing, then there is no need to write them in the bugs list. Data reports that fail or not can be found in the integrated analytical catalog live. Due to the performance test, the results will be listed in the form of data reports such as Throughput and Response time for various testing methods and several concurrent users in the form of tables that have generated reports immediately when the testing is complete.

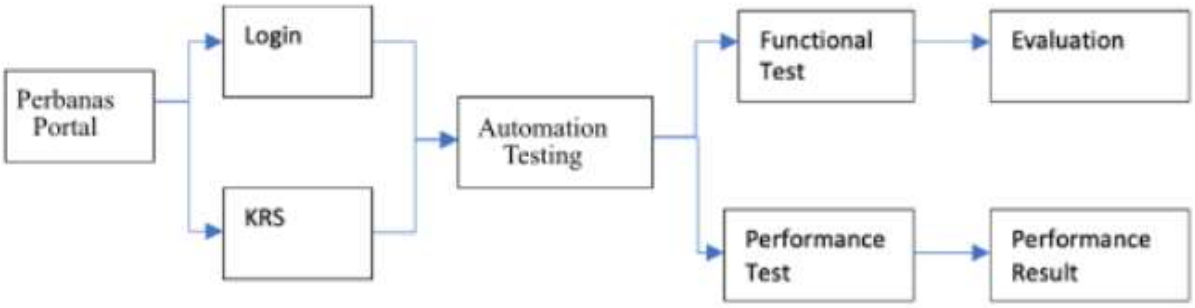

Figure 3. Framework of Research

IV. RESULTS AND DISCUSSION

Based on the research and sorting results, the automation test on the Perbanas portal is as follows:

1. Login Test
    - Successful Login on Putri Kusuma user with a positive line
    - A failure Login on Putri Kusuma user with a negative line, an incorrect input of password and username.
2. KRS Online Test
    - Additional KRS was successful for Putri Kusuma users with a positive line
    - Additional KRS failed on Putri Kusuma's user with a negative line

Table 1. Positive Test Cases

Test Suite ID	**Test Suite Description**	**Test Case ID**	**Expected Result**
A.1	Login Portal Perbanas	A.1.1	Login Success and go to the homepage user

Table 2. Negative Test Cases

Test Suite ID	**Test Suite Description**	**Test Case ID**	**Expected Result**
B.1	Login Portal Perbanas Failed (Different Password)	B.1.1	Login Failed and redirect to Error Message "**No Empty for User Name and Password, Try Again**"
B.2	Login Portal Perbanas Failed (Different User Name and Password)	B.2.1	Login failed and redirect to Error Message "**No Empty for User Name and Password, Try Again**"
B.3	Login Portal Perbanas Failed (Different User Name)	B.3.1	Login failed and redirect to Error Message "**No Empty for User Name and Password, Try Again**"

Functional Test Execution

Table 3. Details of the Load Test Continue – 250 Users

Label	Execution			Response Time (ms)			Throughput TPS	Error
	Sample	KO	Error	Average	Min	Max		
Login	5097	3	0.06%	9992.15	193	53850	16.71	Non HTTP response code: org.apache.http.comm.HttpHostConnectException/Non HTTP response message : Connect to portal.perbanas.id:433 failed: connection time out: connect

Table 4. Details of the Load Test Continue – 100 Users

Label	Execution			Response Time (ms)			Throughput TPS	Error
	Sample	KO	Error	Average	Min	Max		
Login	827	98	11.85%	7070.17	1139	3870	2.74	Test failed: text expecte4d to contain /PortalAkademikPerbanas Institute

Table 5. Detail Staging Continue – 100 Users

Label	Execution			Response Time (ms)			Throughput TPS	Error
	Sample	KO	Error	Average	Min	Max		
Login	3502	0	0.00%	408.46	244	7307	11.23	

Table 6. Detail Staging Continue – 250 Users

Label	**Execution**			**Response Time (ms)**			**Throughput TPS**	**Error**
	Sample	KO	Error	Average	Min	Max		
Login	5184	3	0.06%	4909.47	203	68563	15.75	Non HTTP response code: org.apache.http.conn.HttpHostConnectException/ Non HTTP response message: Connect to portal.perbanas.id:433

V. CONCLUSION

Based on the test results and analysis of the functional test using the Katalon Studio tools and the performance test using the Apache JMeter tool, these can meet user needs in testing. Especially for functional tests and performance tests on the Perbanas Portal for login and KRS features. The best performance test results on the login feature are taken by the sample data that enters approximately 5000 samples for 250 users for a load test per 5 minutes, only 0.06% fails by an average of 15 transactions per second (TPS) with a maximum response time of 68 seconds and a minimum response time of 0.2 seconds. Suppose the KRS feature that has been carried out is taken. In that case, the best is the incoming sample data of 4732 samples for 100 users for a load test per 5 minutes and those that fail are 2.24% by an average of 15 transactions per second (TPS) with a maximum response time of 7 seconds. When there is a hit on 250 concurrent users, the sample can be reduced due to quite a lot of errors due to failed login. Furthermore, the response time obtained is much longer if hits 250 concurrent users. Hence, it can be stated that the Perbanas portal login feature is relatively good because the TPS obtained is quite high and the response time is fast enough for a campus portal.

REFERENCES

[1]. Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. STRING (Satuan Tulisan Riset Dan Inovasi Teknologi), 3(2). https://doi.org/10.30998/string.v3i2.3048

[2]. Costa, V., Girardon, G., Bernardino, M., MacHado, R., Legramante, G., Neto, A., Basso, F. P., & de MacEdo Rodrigues, E. (2020). Taxonomy of performance testing tools: A systematic literature review. Proceedings of the ACM Symposium on Applied Computing. https://doi.org/10.1145/3341105.3374006

[3]. Danendra, K. P. W. (2021). Black box testing aplikasi point of sales post. KURAWAL Jurnal Teknologi, Informasi Dan Industri, 4(1).

[4]. Grm, G., & Grm, A. (2021). Testing the Functionality and Applicability of Smart Devices for a Handheld Celestial Navigation System. Naše More, 68(3). https://doi.org/10.17818/nm/2021/3.3

[5]. Hao, D. (2021). Research on DDS-based Portable Signal Generation Testing Device. Journal of Physics: Conference Series, 1971(1). https://doi.org/10.1088/1742-6596/1971/1/012001

[6]. Mendiratta, N., & Kumar, R. (2016). Relative Study of Automated Testing Tools: Selenium, Quick Test Professional and Test complete. IMS Manthan (The Journal of Innovations), 11(01). https://doi.org/10.18701/imsmanthan.v11i01.6879

[7]. Psujek, M., Radzik, A., & Kozieł, G. (2021). Comparative analysis of solutions used in Automated Testing of Internet Applications. Journal of Computer Sciences Institute, 18. https://doi.org/10.35784/jcsi.2373

[8]. Sasmito, G. W., & Nishom, M. (2020). Testing the Population Administration Website Application Using the Black Box Testing Boundary Value Analysis Method. 2020 IEEE Conference on Open Systems, ICOS 2020. https://doi.org/10.1109/ICOS50156.2020.9293645

[9]. Software performance testing. (2020). Bulletin of V.N. Karazin Kharkiv National University, Series Mathematical Modeling. Information Technology. Automated Control Systems, 45. https://doi.org/10.26565/2304-6201-2020-45-07