

Efficiency Development Analysis Use of Automation Testing using Katalon on Mobile-Based Applications (Case Insurance Company)

Shynta Eza Anggrainy¹, Alva Hendi Muhammad²

^{1,2}Universitas Amikom Yogyakarta, Indonesia

Email: Shyntaeza@students.amikom.ac.id¹, alva@amikom.ac.id²

Abstract

The use of automation in mobile application testing has become an important tool in improving the efficiency and quality of applications. Katalon Studio, as one of the automated testing tools, offers convenience for testers in conducting automated testing without having to master any programming language. This tool allows continuous testing to maximize the potential of features and applications as a whole. Katalon Studio also allows the addition of an alert feature on failed scenarios, so that the relevant team will be notified if a failure occurs when the script is run. In addition, the use of automated testing tools such as Katalon Studio can speed up the testing process and increase efficiency. In the context of mobile-based applications, automated testing is very important to ensure the quality of the user experience (usability). By conducting automated testing, the app design and development team can be aware of issues that may arise and fix them immediately. Overall, the use of Katalon Studio in automated testing of mobile applications can help improve the efficiency of the testing process and application quality, as well as maximize the potential of application features.

Keywords: Automated Testing, Catalon Studio, Mobile-Based Application, Efficiency

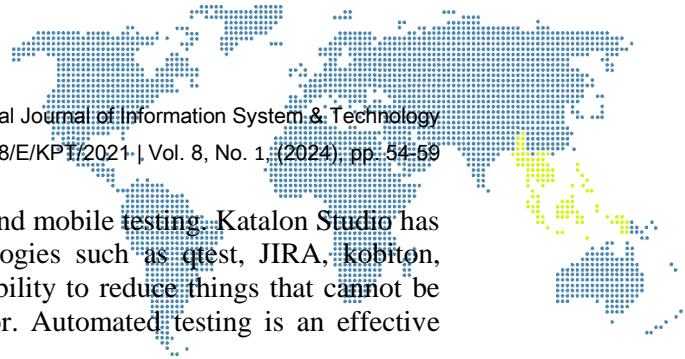
1. Introduction

At this time, mobile-based applications have become an important part of everyday life. From banking, insurance, to healthcare, mobile applications facilitate various aspects of life. However, to ensure that these apps function properly and provide an optimal user experience, app testing is crucial. Manual testing of apps can be a time-consuming process and consume significant resources. Therefore, automated testing has become an increasingly popular solution within the technology industry. One of the tools used for this automated testing is Katalon Studio. Katalon Studio is an automated testing tool that allows testers to test mobile, browser, and API-based applications without having to master any programming language. This allows testing to be done more efficiently and effectively, saving time and resources that would otherwise be used for manual testing.

In this journal, we will analyze how the use of Katalon Studio in automated testing can improve the efficiency of mobile-based application testing. We will discuss how this tool can help in speeding up the testing process, improving the quality of the application, and ultimately, providing a better user experience. System testing is a process used to represent the results of analysis, design, and implementation of a system with the aim of ensuring the quality of the developed system. System testing is carried out to detect errors that may occur in the system as a first step in finding solutions and improving the quality of the system [1].

Black-box testing is a system testing method that aims to find errors and deficiencies in software functionality, such as functions, interfaces, data models, and access to external information sources. This method focuses on testing without paying attention to the internal structure of the software being tested. By using this method, we can identify problems that may occur in the system without paying attention to the internal implementation of the software [2].

Katalon Studio is an open-source software designed to perform automated testing. Katalon Studio was developed by Katalon LLC. The software supports three automated



testing platforms, namely web testing, API testing, and mobile testing. Katalon Studio has also been integrated with several external technologies such as qtest, JIRA, kobiton, github, and others [3]. Automated testing has the ability to reduce things that cannot be backed up by manual testing or avoid human error. Automated testing is an effective method to save cost and time [4].

2. Reseach Methodology

In some previous studies, it was stated that the use of Katalon Studio can improve test efficiency by simplifying test scenarios and providing good integration with various types of applications. These advantages can result in shorter development times and reduce the risk of human error.

However, while Katalon Studio offers significant advantages, some authors also point out some challenges that developers and testing teams may face. These include the initial learning required to master the tool, maintenance of complex test scenarios, and complicated integration with other development tools.

Therefore, through this literature review, we can gain a deep insight into the efficient use of Automation Testing with Katalon Studio. By understanding the advantages and associated challenges, organizations can make better-informed decisions in choosing an automated testing tool that suits their software development needs.

The comparison on automation testing conducted for web-based applications, specifically focuses on the advantages and disadvantages of Selenium and Katalon Studio applications. There are several feature evaluations, such as simulation testing, execution time, reporting, ease of setup, and programming language [5].

Automation testing helps testers avoid repetitive testing due to changes from the development team. This article also discusses the differences in definition and runtime between the two testing methods. Automation testing uses testing tools to execute test cases, while manual testing is done directly by the tester. This article also discusses the advantages of automation testing in optimizing test execution and determining whether the application functions as expected.

From various relevant studies, it can be said that the Katalon application has many benefits that can be implemented for the automation testing process in testing New Business Sales Tools used by AMFS to minimize errors that might occur.

This research begins with reading and analyzing the Business Requirement System (BRS), then proceeds with the design of making test scenarios, designing test cases both in manual and automation testing methods, implementing manual testing, then implementing automation testing after both tests are carried out, then comparing manual and automation testing and the last is writing research results in the form of a final report.

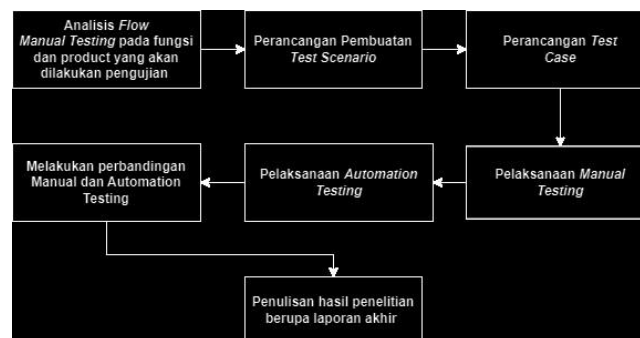
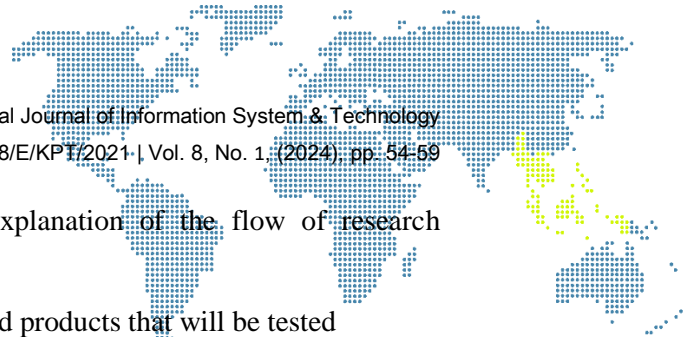


Figure 1. Research Methods

3. Results and Discussion

This application can only be accessed by internal staff or commonly called Financial Advisor (FA), this application should not be accessed by just anyone because it is related



to customer privacy data. The following is an explanation of the flow of research conducted:

a) Analysis of Flow Manual Testing on functions and products that will be tested

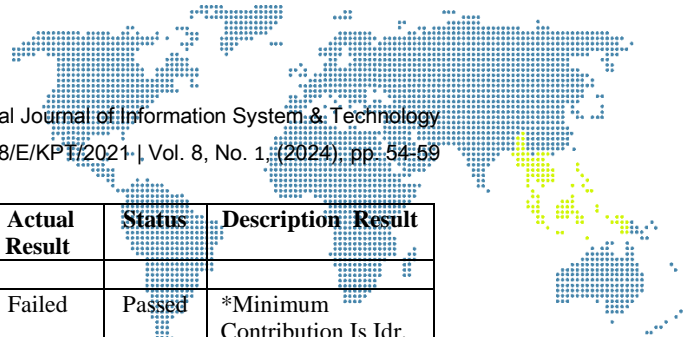
The initial step in this research process is to analyze the flow of functions on the features and products that will be tested. This is done by looking at the Business Requirement System (BRS) document provided by the Business Analyst (BA) as a reference for the manual testing process.

b) Designing Test Scenario Creation

In the second step, the researcher will create a Test Scenario. Test Scenario is a grouping of tests based on the functions that will be tested in the study.

Table 1. Test Scenario

No	Detail Testcase	Type Test	Expected Result	Actual Result	Status	Description Result
1	Create An Insurance Policy With Less Than The Minimum Insurance Benefit.	Negative	Failed	Failed	Passed	*Insurance Benefits In Accordance With Recommendations
2	Make An Insurance Policy With Provisions Less Than The Maximum Insurance Benefit	Negative	Failed	Failed	Passed	*Insurance Benefits In Accordance With Recommendations
3	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
4	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
5	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
6	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
7	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
8	Make An Insurance Policy In Accordance With The Provisions	Positive	Passed	Passed	Passed	
9	Make Insurance Policy According To Filling Rbp Mix Less Than 20% - 60%	Negative	Failed	Failed	Passed	*Contains 20% - 60% Rbp Mix
10	Make Insurance Policy According To Filling Rbp Mix More Than 20% - 60%	Negative	Failed	Failed	Passed	*Contains 20% - 60% Rbp Mix
11	Create An Insurance Policy In Accordance With The Percentage Of Subfund Allocation Not Yet 100%	Negative	Failed	Failed	Passed	Percentage Of Subfunding Allocation Not Yet 100%
12	Create An Insurance Policy In Accordance With Sub-Funding Allocation Of Less	Negative	Failed	Failed	Passed	*Minimum Fund Allocation Must Be 10%



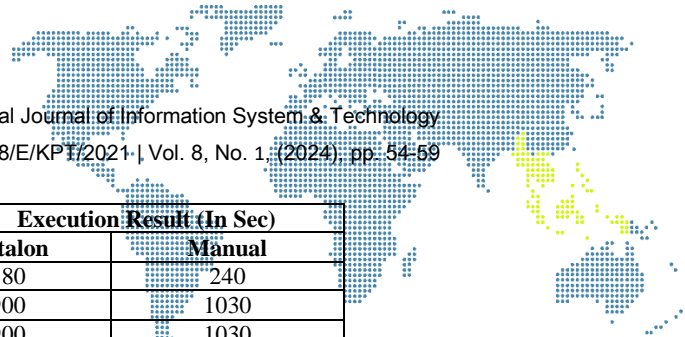
No	Detail Testcase	Type Test	Expected Result	Actual Result	Status	Description Result
	Than 10%					
13	Make An Insurance Policy With Provisions Less Than A Minimum Of Idr. 10.000.000	Negative	Failed	Failed	Passed	*Minimum Contribution Is Idr. 10.000.000
14	Make An Insurance Policy With The Condition That The Entry Age Of The Prospective Participant Is More Than 70 Years Old	Negative	Failed	Failed	Passed	*Maximum Entry Age Of Candidates Is 70 Years Old
15	Make An Insurance Policy With The Condition That The Entry Age Of The Prospective Participant Is Less Than 18 Years Old	Negative	Failed	Failed	Passed	*Minimum Entry Age Of Prospective Participants Is 18 Years Old
16	Make An Insurance Policy With The Provision Of Lump Sum Nominal Less Than Idr 1,000,000 / 100 Usd.	Negative	Failed	Failed	Passed	* Minimum Lump Sum Amount Idr 1,000,000 / 100 Usd.
17	Make An Insurance Policy With A Lump Sum Nominal Provision Of More Than Idr 1,000,000 / 100 Usd.	Negative	Failed	Failed	Passed	* Minimum Withdrawal Amount Idr 1,000,000 / 100 Usd.
18	Make An Insurance Policy With The Maximum Age Of Entry Of The Payor Rider Is 64 Years Old	Negative	Failed	Failed	Passed	Plan Not Available
19	Make An Insurance Policy With The Entry Age Of The Prospective Participant Below 18 Years Old	Negative	Failed	Failed	Passed	*Minimum Entry Age Of Prospective Participants Is 18 Years Old
20	Create Insurance Policy With Rbp Mix 1% - 40%	Negative	Failed	Failed	Passed	*Contains 1% - 40% Rbp Mix

c) Test Case Design

This step is the step of making test cases that are part of the test scenario. Test cases will be created and divided into two parts, positive test cases and negative test cases. Positive test cases are tests that use the correct parameters and in accordance with the purpose of the existing function. The goal is to test the main flow of the function. Negative test case is a test that uses parameters that are not correct and do not match the purpose of the existing function. The purpose of the negative test case is to test the exception handling of the application.

Table 2. Test Case

No	Test Case		Execution Result (In Sec)	
	Katalon	Manual	Katalon	Manual
1	Failed	Failed	180	240



No	Test Case		Execution Result (In Sec)	
	Katalon	Manual	Katalon	Manual
2	Failed	Failed	180	240
3	Passed	Passed	900	1030
4	Passed	Passed	900	1030
5	Passed	Passed	900	1030
6	Passed	Passed	900	1030
7	Passed	Passed	900	1030
8	Passed	Passed	900	1030
9	Failed	Failed	150	210
10	Failed	Failed	150	210
11	Failed	Failed	200	260
12	Failed	Failed	200	260
13	Failed	Failed	180	240
14	Failed	Failed	130	190
15	Failed	Failed	130	190
16	Failed	Failed	250	310
17	Failed	Failed	250	310
18	Failed	Failed	200	260
19	Failed	Failed	130	190
20	Failed	Failed	180	240

From the test results above using the same testcase results in different durations, where the results of automated testing are faster than manual testing. In addition, automated testing reduces "human errors" that may occur when using manual testing.

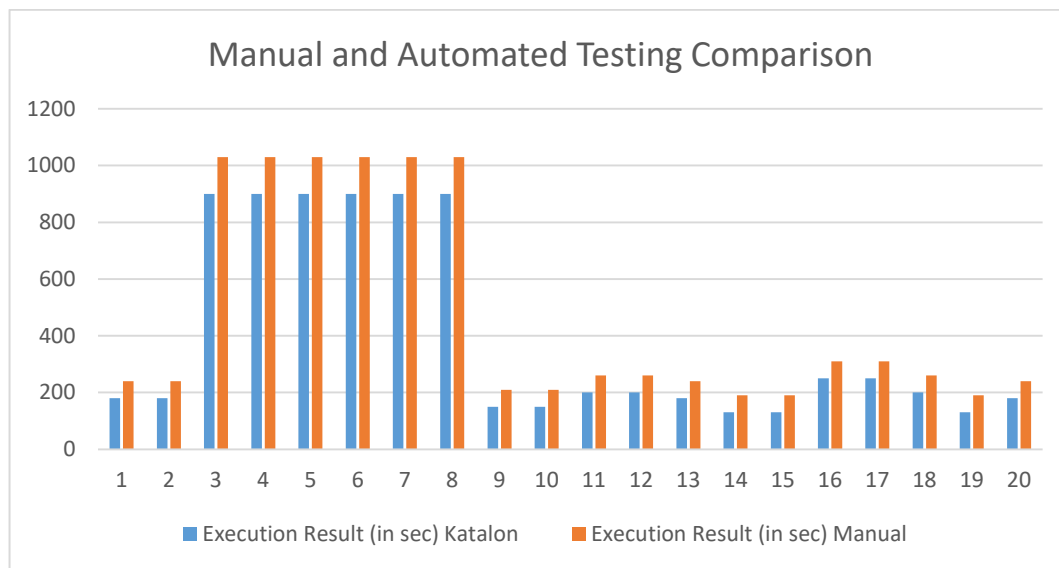
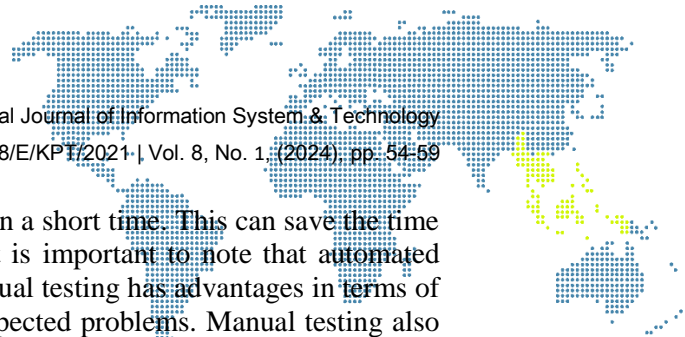


Figure 2. Test Execution Results

4. Conclusion

Manual and automated testing are two methods used in software testing. The following are conclusions regarding the comparison between manual and automated testing, with a focus on the speed of automated testing. Manual testing is a testing method that is performed manually by humans. In manual testing, testing is done by following predefined steps and observing the results directly. Manual testing requires significant time and effort, especially when testing is thorough and repetitive. On the other hand, automated testing uses tools and scripts to run tests automatically. In automated testing, test scripts are created to automate the test steps. Automated testing can be done quickly and efficiently, as tests can be run automatically without human intervention. In terms of testing speed, automated testing is generally faster than manual testing. Automated tests



can run test scripts quickly and generate test results in a short time. This can save the time and effort required in software testing. However, it is important to note that automated testing is not always better than manual testing. Manual testing has advantages in terms of flexibility and the ability to detect complex or unexpected problems. Manual testing also allows for more in-depth testing and direct observation of software behavior. Therefore, manual testing is still necessary in some cases, especially when testing requires human understanding and subjective judgment. In conclusion, automated testing is generally faster than manual testing. However, it is important to consider the advantages and disadvantages of each testing method and choose the method that best suits the software testing needs.

References

- [1] Arief Zulianto, A. P. (2021). Pemanfaatan Katalon Studio Untuk Otomatisasi Pengujian Black Box Pada Aplikasi Iposyandu. *Jurnal Edukasi Dan Penelitian Informatika*, 1-9.
- [2] Eland Vaskita Nugraha, R. Y. (2022). Uji Black Box Tes Aplikasi Sotware Development System Information (Sodevi) Pt. Dimata Sora Jayate Menggunakan Katalon Studio. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (Snast)*, 1-6.
- [3] Fitriyan Ardi, H. P. (N.D.). Pengujian Black Box Aplikasi Mobile Menggunakan Katalon Studio (Studi Kasus: Acc Partner Pt. Astra Sedaya Finance). 1-6.
- [4] Mario Anugraha Agung Eka Tjandra, R. P. (2024). Pengujian Kualitas Perangkat Lunak Menggunakan Pendekatan Manual Dan Automation Testing Pada Aplikasi “Nuzantara” Menggunakan Katalon. 1-21.
- [5] Rusli, H. (2020). Analisa Perbandingan Black-Box . 1-36.
- [6] Chunhui Wang, 2015, Automatic Generation Of System Test Cases From Use Case Specifications, *Issta, Acm*. 978-1-4503-3620-8/15/07.
- [7] Lindawani Siregar, 2020, Review Pengujian Keamanan Perangkat Lunak Dalam Software Development Life Cycle (Sdlc), *Applied Sciences, Electrical Engineering And Computer Technology (Aseect)*, Issn: 2746-7422, Vol. 1 Issue 31 December 2020.
- [8] Sang Hun Lee, 2020, Optimization Of Software Development Life Cycle Quality For Npp Safety Software Based On A Risk-Cost Model, *Annals Of Nuclear Energy*, 2020.
- [9] A. Yudi Permana, 2019, Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode Sdlc Pada Pt. Mandiri Land Prosperous Berbasis Mobile, *Sigma – Jurnal Teknologi Pelita Bangsa*, Issn : 2407-3903, Volume 10 Nomor 2 Desember, 2019 .
- [10] Vu Nguyen, 2021, Generating And Selecting Resilient And Maintainable Locators For Web Automated Testing, *Software Testing, Verification And Reliability*, Published Online In Wiley Online Library, 31 January, 2021.
- [11] Muralidhar Yalla, 2020, Ai-Driven Conversational Bot Test Automation Using Industry Specific Data Cartridges, *International Conference On Automation Of Software Test (Ast)*, Ieee/Acm 1st, 2020.
- [12] Ashok Sivaji, 2020, Software Testing Automation: A Comparative Study On Productivity Rate Of Open Source Automated Software Testing Tools For Smart Manufacturing, *Ieee Conference On Open Systems (Icos)*, 2020.
- [13] Rizal Broer Bahaweres, 2020, Behavior-Driven Development (Bdd) Cucumber Katalon For Automation Gui Testing Case Cura And Swag Labs, *International Conference On Informatics, Multimedia, Cyber And Information System (Icimcis)*, 2020.
- [14] Elis Pelivani, 2021, A Comparative Study Of Automation Testing Tools For Web Applications, *10th Mediterranean Conference On Embedded Computing (Meco)*, 7-10 June, 2021.