JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 1 | June 2023: 11-18
DOI: 10.15575/join.v8i1.977

Scalability Testing of Land Forest Fire Patrol Information Systems

Ahmad Khusaeri[1], Imas Sukaesih Sitanggang[2], Hendra Rahmawan[3]
[1,2,3] Department of Computer Science, IPB University, Bogor, Indonesia

Article Info

Article history:

Received Nov 18, 2022
Revised April 11, 2023
Accepted April 13, 2023

Keywords:

Fires
Scalability
Software
Testing

ABSTRACT

The Patrol Information System for the Prevention of Forest Land Fires (SIPP Karhutla) in Indonesia is a tool for assisting patrol activities for controlling forest and land fires in Indonesia. The addition of Karhutla SIPP users causes the need for system scalability testing. This study aims to perform non-functional testing that focuses on scalability testing. The steps in scalability testing include creating schemas, conducting tests, and analyzing results. There are five schemes with a total sample of 700 samples. Testing was carried out using the JMeter automation testing tool assisted by Blazemeter in creating scripts. The scalability test parameter has three parameters: average CPU usage, memory usage, and network usage. The test results show that the CPU capacity used can handle up to 700 users, while with a memory capacity of 8GB it can handle up to 420 users. All users is the user menu that has the highest value for each test parameter The average value of CPU usage is 44.8%, the average memory usage is 69.48% and the average network usage is 2.8 Mb/s. In minimizing server performance, the tile cache map method can be applied to the system and can increase the memory capacity used.

Corresponding Author:

Ahmad Khusaeri
Department of Computer Science, IPB University, Bogor, Indonesia
Jl. Meranti Wing 20 Level V, Bogor, Indonesia 16680
khusaerikhusaeri@apps.ipb.ac.id

1. INTRODUCTION

Indonesia is one of the countries that have the largest forest in the world. Indonesia has 94.1 million hectares of forest area [1]. Forest management in Indonesia has many challenges with many forest areas. One of the challenges faced is fire. Forest fires are caused by several things, namely El Nino, government decentralization, and regional economic growth [2]. The significant negative impact of forest fires in Indonesia is the deterioration of public health. Forest fires that occurred in 2019 had an increase from the previous year. The area affected by forest fires reaches 942.484 hectares.

Economic losses experienced by Indonesia amounted to 75 trillion rupiah due to forest fires. The forest fires in 2019 prompted the President of the Republic of Indonesia to issue Presidential Instruction number 3 of 2020 concerning the prevention of Forest and Land Fires (Karhutla) and to revoke Presidential Instruction number 11 of 2015. The President instructs every relevant agency to coordinate the prevention of forest and land fires. In Indonesia. One of the ministries given instructions by the President to tackle forest and land fires is the Ministry of Environment and Forestry (MoF) Indonesia. MoF was instructed to increase patrols of the Forestry Police within the forest area or its jurisdiction. MoF responded to the Presidential Instruction by implementing three strategies to combat forest and land fires in Indonesia: climate analysis and measures, operational control, and landscape management. Active program is carried out by increasing the patrols of the integrated patrol team and

developing the SiPongi Karhutla Monitoring System information system to record hotspots and the fire area from year to year. The integrated patrol team conducts patrols to areas prone to forest and land fires. The Department of Computer Science of IPB, in collaboration with the Center for Climate Change and Forest and Land Fire Control (PPIKHL) and Directorate of Forest and Land Fire Management of MoF, developed the Information System for Prevention Patrols of Forest and Land Fires (SIPP Karhutla) [3]. The development of SIPP Karhutla for the process of data acquisition and data dissemination has been done by Ramdhany et al. [4]. SIPP Karhutla has been used to support forest and land fire patrol prevention in Sumatra and Kalimantan.

Software testing needs to be done because there is a relationship between testing software and the quality of the software [5]. Application testing is also a crucial factor in identifying errors and ensuring that the application functionality is running well [6]. Testing the quality requirements will be challenging in making applications for an enormous scope [7]. Application testing is inseparable from Functional Requirements and Non-Functional Requirements.

Application developers usually only focus on the functional requirements of an application without considering the non-functional requirements. Non-functional requirements can be the basis for making application functionality. The existence of Non-functional requirements can describe the pattern of system behavior and can make decisions for making functional applications. Non-Functional Requirements can be described by system performance, security, and general system limitations.

Iskandar and Nofiyati (2018) conducted research on performance testing to determine the reliability of the quality of service website of the Faculty of Engineering, Sudirman University (Unsoed) [8]. The study was conducted to test the scalability of the Unsoed Faculty of Engineering website. Website testing was done using the JMeter application. The number of users is set to 1000, with five samples sent every 0.1 seconds. The results of this study indicate that the maximum capacity limit the system can handle simultaneously is 302 users. The highest request that the server can accommodate is 1088.4 transactions/second when the number of users who can access it is 157, with an average of 51.5 transactions/second.

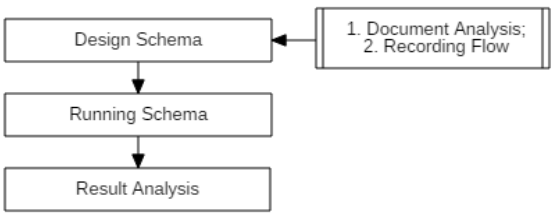
A study conducted by Fathiyah et al [9] tested the stability and ability of the system to receive high traffic. The study was conducted with the number of connections to the server reaching 100 connections and the number of users requesting access as many as 50. The results of the tests are that the server can serve requests of 615.50% per second.

A study by Abbas and Sultan's (2017) compared JMeter, LoadRunner, Siege, and Microsoft Visual Studio applications for automated testing [10]. The results of the questionnaire distribution in this study showed that 80% of correspondents used JMeter compared to other testing software. JMeter gives better results than other test tools because it has a user ratio scale methodology and performs tests consistently. Apache JMeter is most widely used in heavy load testing and has a complete and simulated scenario with multiple test patterns [11].

SIPP Karhutla has been implemented in Sumatra and is in the process of being implemented in Kalimantan. Increasing the number of users and regions requires scalability testing to measure performance trends over a certain period of time with an increase in the number of loads. This study aims to carry out non-functional testing at SIPP Karhutla in the aspects of scalability testing and to formulate recommendations for SIPP Karhutla. Testing is carried out by utilizing automated software testing.

2. METHOD

The research to be conducted has five stages of research. The flow chart of the research stages is shown in Figure 1.



Fugure 1. SEQ Figure * ARABIC Research Steps

2.1. Design Schema

The first step is to design a load test. In this step, we prepared the scope, objectives, features to be tested, features not to be tested, types of testing to be performed, roles and responsibilities of the testing team and criteria.

The next stage is analyzing the necessary needs and the scheme that will be execute. At the analysis and design stages, the schema is adjusted to features to be tested and test data. In addition, at the analysis and design stages, analyze non-functional requirements based on ongoing technical documents and specifications.

At the stage of designing the load test, the analysis process of UML diagrams is also carried out, such as activity diagrams and use case diagrams to describe detailed user interactions in the system. UML diagram analysis was done to determine the processes in the system. In addition to analyze the UML diagram, we identified features frequently accessed by users of SIPP Karhutla. At this stage, checking the time of data traffic often passed is also carried out.

After the ongoing system is analyzed, the next step is to create a schema for testing. Schemas are determined is based on the intensity of users and the average peak usage. The workload will be increased periodically in the process of making the load. After determining the load, the next step is to create a script to run the test stages automatically using the Blazemeter tool. Blazemeter is used to record, browse, upload, and run scripts for activities using a website [13].

This study performed scalability testing in five cases based on the number of users, rump-up period, loop, and total sample. The number of users is the number of users who will access the system simultaneously. The rump-up period is the time lag before the following user starts. The loop is the number of repetitions from each user that will be carried out. The total sample is the number of examples of users who are testing the system as a whole [14]. The sample in the first case is the number of active users currently using SIPP Karhutla. The total sample for each case is multiplied by the stratification according to the case carried out.

2.2. Running Schema

After designing the load test, the next stage is the load testing process. System load testing is carried out with several schemes based on the number of users and the traffic density of the system usage. System testing was done using the JMeter application. System testing was carried out using the automation testing method. System performance testing aims to identify system bottlenecks and their causes, optimize and adjust the system configuration optimally and verify the system when under pressure [15].

At the execution stage, testing was performed according to the scheme created in the first stage. If bugs or discrepancies are found in the test, a search and resolution of bugs will be carried out to be reported to the application developers. When the application has been repaired, then the next stage is carried out testing again with a predetermined scheme. When the execution of the load test runs for a long time, there is a point in time after the test no longer provides new information about the performance of the system under test, and the execution of the load test is stopped [16].

The scalability test has three assessment parameters: Average CPU Usage, Average Memory Usage, and Average Network Usage[17]. Average CPU Usage is the average percentage of CPU core usage owned by the server. Average memory usage is the average RAM usage of the testing server. Network usage is the average speed of data packets transmitted by the server. During the SIPP Karhutla testing stage, we monitor the testing process and collect test results. The test results obtained from JMeter are exported for analysis.

2.3. Result Analysis

At this stage, an analysis of the current system is executed to determine whether it is following the objectives of the system or not. The test results are made into a report containing an analysis of the test results. Reports and analysis results are then given to the application development team for further improvement of the system. The system testing results will be compared with the system's condition and the number of users. In analyzing the results, recommendations is proposed for the system improvement.

The test results are reported, which ensures that all system, integration, and user acceptance tests pass and a decision is made on whether all requirements are tested. No critical bugs are proposed to be fixed or verified. At this stage, a brainstorming process is also carried out to identify features that are

working properly or not, and areas that need to be improved. The test results need to be submitted to the development team for the improvement of the Karhutla SIPP.

3. RESULT AND DISCUSSION

3.1. Design Schema

The process of designing a scheme is carried out based on the system documents that have been executed. The use case diagram describes that there are six actors in using SIPP Karhutla. The five actors are the operations, the administrator of local fire stations, the head of local fire stations, the head and administrator of regional coordination, the administrator and head of the agency for climate change and forest and land fire management, and all users. SIPP Karhutla has eight main features. The features in SIPP Karhutla are patrolling, reporting, user management, hotspots, data analysis, FAQ, and user profile. Based on these six actors and some features, several access roles can be classified in the SIPP Karhutla. The categories of forest fire and forest fire users can be seen in Table 1[3].

Table 1. User Categories of System

User Category	Menu
The Administrator of Local Fire Stations	Input and Edit Report Create assignment letter Read assignment letter Update assignment letter Delete assignment letter Create patrol member Read patrol members Update patrol member Delete patrol members
Head of Local Fire Stations	Daily reports (in pdf format) Daily reports (in spreadsheet format) Edit the report Delete report View assignment letter and patrol team Download Report
Head and Administrator of Regional Coordination	Create report (in province level) Read the report (in province level) Update report (in province level) Delete report (in province level) View assignment letter (in province level) Download Report (in province level)
Administrator and Head of Agency for Climate Change and Forest and Land Fire Management	Create user Read user Update user Delete user Verify and edit report View assignment letter Download Report
All User	Read report Updates report Delete report Layer Date Layer Filter Print

Determination of the number of sample users is determined based on the actual number of users at that time. The number of Karhutla SIPP users in December 2021 was 140 users. Schematic creation is carried out based on the aggregate of real workloads. Making the test scheme is divided into 5 test schemes. The test scheme is made by adding the total samples to the number of samples in the first scheme. The maximum number of samples in making a scheme is 700 samples. The five schemes are shown in Table 2.

Table 2. Testing Schema

Case	User	Rump Up Period(s)	Loop (x)	Total sample
Case 1	10	10	14	140
Case 2	20	5	14	280
Case 3	30	3,3	14	420
Case 4	40	2,5	14	560
Case 5	50	2	14	700

After making the schematic, then the process of creating a script for testing is carried out. The test script is created using the blazemeter tool. the test script is created by recording every menu activity that is executed and will then be implemented in JMeter.

3.2. Running Schema

In the scalability testing stage, the report used in JMeter is to use the performance module with the assessment matrix being Average CPU Usage, Average Memory Usage and Average Network Usage. The process of running the test is carried out by loading each of the Karhutla SIPP menus made with the Blazemeter file. Testing was done using the Automation Software Testing scheme. The scheme that is run is based on the users of the Karhutla SIPP. In running the load test, the device used is the same and uses an internet connection with the same source in every scheme. Testing is done on a virtual machine (VM) that is stored on the server. The VM specifications are an eight-core CPU, 8 GB RAM and 140 GB HDD. The testing process is run alternately. When the process of one schema has finished running on one user, the following schema will be executed. After the testing of each scheme is carried out, the next stage is the collection of test results. The test schema has been executed, generating 25 reports from each user. Each schema test generates a graph and table report of aggregate features in JMeter. Each schema performed produces multiple results that can be saved in CSV format. The results of each schema test are combined into one file to facilitate the process of reading data and analyzing the test results.

3.3. Result Analysis

Scalability testing was carried out on SIPP Karhutla with five schemes. The schemes that have been made are tested on a series of menus available for each type of user. The user menus tested were on the types the administrator of local fire stations, head of local fire stations, head and administrator of regional coordination, administrator and head of agency for climate change and forest and land fire management and all user. The results of load testing on each user are as follows figure 2,3,4.

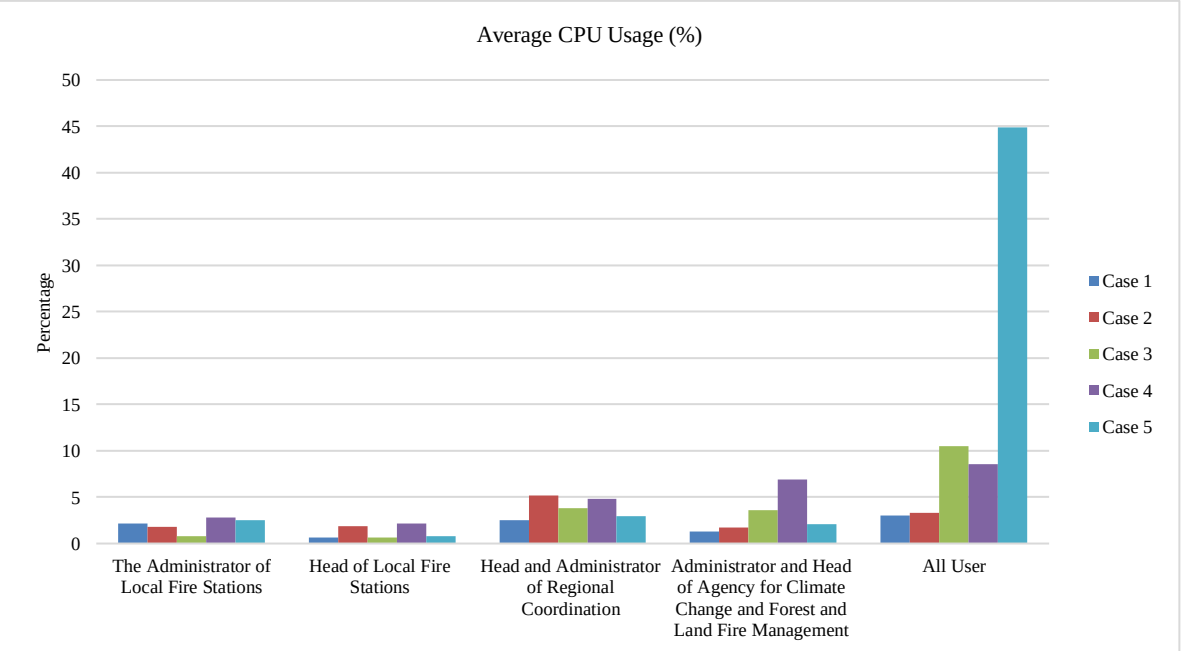


Figure 2.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

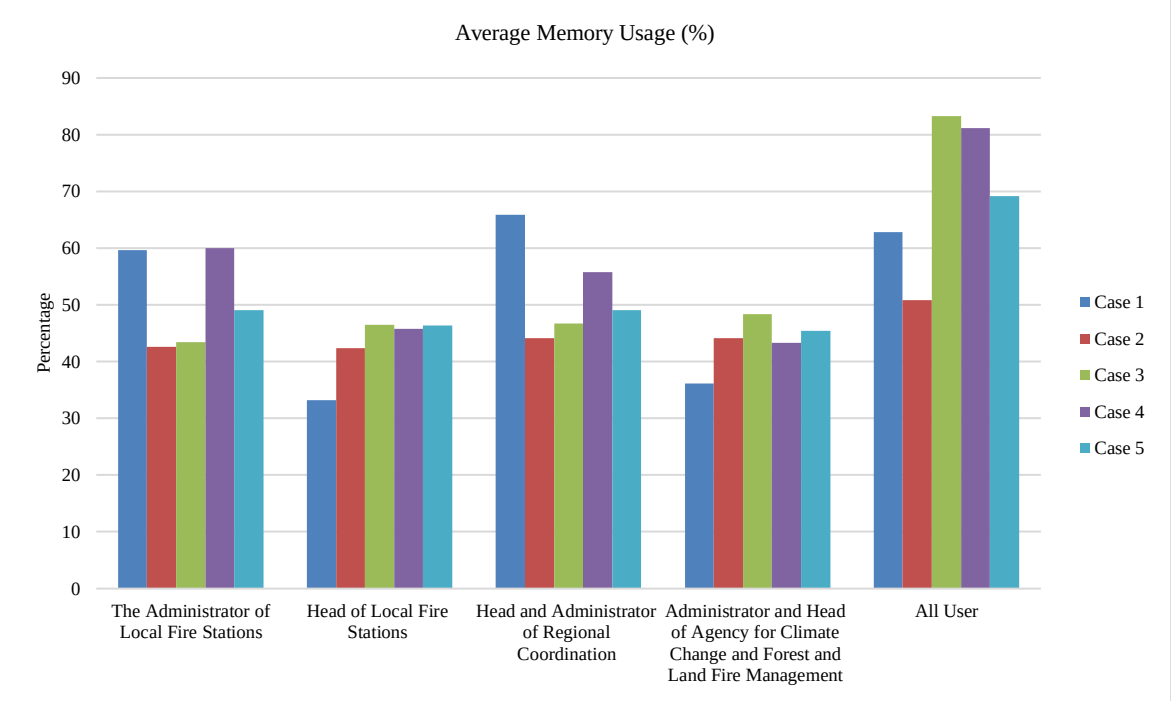


Figure 3. Average Memory Usage

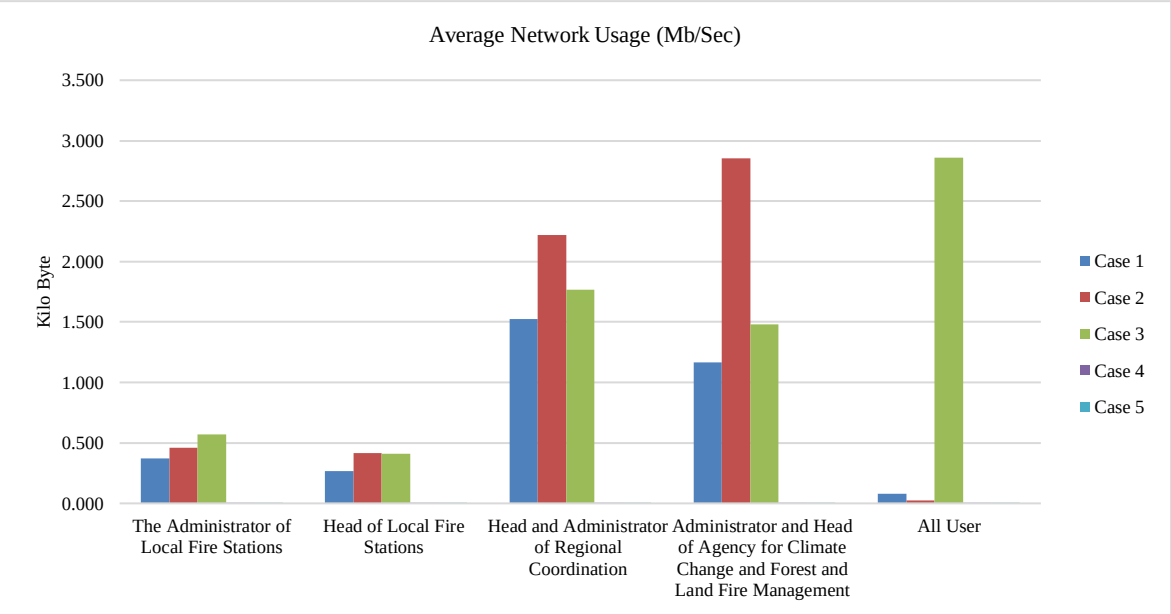


Figure 4. Average Network Usage

In scalability testing, testing is carried out on each user actor. The total user actors are five: the administrator of local fire stations, head of local fire stations, head and administrator of regional coordination, administrator and head of agency for climate change and forest and land fire management and all user. The results of scalability testing show that the Average CPU Usage matrix, which has the most significant value, is in data analysis with the fifth scheme with a value of 44.86%. As for the RAM usage matrix, the most significant value is in the data analysis menu in the third scheme, with a RAM usage value of 83.23%. The five users' most significant average data usage is in data analysis, with the fifth scheme with a value of 2.8 Megabyte per second.

In the data all user, the five schemes contained in the menu have a significant CPU core usage value, especially in the fifth scheme. The fifth scheme in the data analysis menu requires 44% of core usage due to a large number of user samples, up to 700. As for the other menus, it does not use too many large CPU cores. The RAM usage of the five menu actors is relatively the same even though the value of RAM usage

in the data analysis menu is greater than the other menus. This is due to a large number of data processing, searching and loading data on the data analysis menu.

At the stage of the all user, each parameter is checked. each parameter is checked for the highest value of each scheme. The discussion of SIPP Karhutla scalability testing can be summarized according to the Table 3.

Table 3. Result of Scalability Testing

Information	Average CPU Usage	Average Memory Usage	Average Network Usage
User	All user	All user	All user
Value	44.87%	83.24%	2.85 Mb/s
Sample	700	420	700

The scalability test shows that all user dominates each assessment matrix in the scalability test. The all user gets the highest average CPU usage value of 44.87% with a total sample of 700 and produces an average memory usage of 83.24% with a total sample of 420. This indicates a need for an increase in CPU and memory quality to support the increase in the number of users. Apart from the hardware aspect, methods in the all user are needed to reduce the hardware load. The all user displays several layers selected by the user. Displaying each of these layers makes the process of using CPU and memory bigger. If the number of layers accessed by the user increases, server processes will also increase. Therefore, it is necessary to use a method to display each layer called by the user. One method that can be used is a catching layer or tile cache.

4. CONCLUSION

Based on the results of the SIPP Karhutla scalability test it shows that menus with high average CPU usage, memory usage and network usage are menus for all users, this is due to the process of displaying layered data which is visualized with a map. When the user requests more map layers to display, it will affect the performance in processing to display the user's request. This forest and land fire information system can handle up to 700 users with relatively good CPU usage performance, but with 8GB of memory it can only handle up to 420 users. If the number of users increases to more than 420, there will be a decrease in system performance. This is due to the effect of an increase in the number of users on the use of the menu for all users. In minimizing server performance that is too high, ways can be applied to reduce server load and can increase the capacity of the memory provided. One method that can be applied is the tile cache website method. Map website tile cache is the method of downloading the entire map library which is done when initial accessing the website.

REFERENCES

[1] Kemenlhk, "Hutan dan Deforestasi Indonesia Tahun 2019," Kemenlhk, 2020. http://ppid.menlhk.go.id/siaran_pers/browse/2435.

[2] R. B. Edwards, R. L. Naylor, M. M. Higgins, and W. P. Falcon, "Causes of Indonesia's forest fires," World Dev., vol. 127, 2020, doi: 10.1016/j.worlddev.2019.104717.

[3] I. S. Sitanggang, R. Trisminingsih, and Wulandari, "Panduan Penggunaan SIPP Karhutla," 2021.

[4] D. Ramdhany, I. S. Sitanggang, I. Kurniawan, and Wulandari, "Modul Front-End Sistem Informasi Geospasial Patroli Terpadu Kebakaran," J. Resti (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 2, pp. 272–280, 2021.

[5] D. Kumar and K. K. Mishra, "The Impacts of Test Automation on Software's Cost, Quality and Time to Market," Procedia Comput. Sci., vol. 79, pp. 8–15, 2016, doi: 10.1016/j.procs.2016.03.003.

[6] F. Okezie, I. Odun-Ayo, and S. Bogle, "A Critical Analysis of Software Testing Tools," J. Phys. Conf. Ser., vol. 1378, no. 4, 2019, doi: 10.1088/1742-6596/1378/4/042030.

[7] J. Larsson, M. Borg, and T. Olsson, "Testing quality requirements of a system-of-systems in the public sector - Challenges and potential remedies," CEUR Workshop Proc., vol. 1564, 2016.

[8] D. Iskandar and N. Nofiyati, "Performa Testing untuk mengetahui reabilitas QoS (Quality of Service) website Fakultas Teknik Universitas Jenderal Soedirman," Din. Rekayasa, vol. 14, no. 1, p. 39, 2018, doi: 10.20884/1.dr.2018.14.1.206.

[9] A. C. Fathiyah, S. F. S. Gumilang, and D. Witarsyah, "PENGUJIAN FUNGSIONAL DAN NON FUNGSIONAL APLIKASI WEB BORONGAJAYUK," vol. 44, no. 12, pp. 2–8, 2019.

[10] R. Abbas, Z. Sultan, and S. N. Bhatti, "Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege," Int. Conf. Commun. Technol. ComTech 2017, vol. 1, no. 2, pp. 39–44, 2017, doi: 10.1109/COMTECH.2017.8065747.

[11] A. K. Chandrasekhar and D. A. S. ; Chandran, "Comparative Analysis of Load Testing Tools Sahi And Selenium," Int. J. Creat.

Res. Thoughts, vol. 5, no. 7, pp. 55–60, 2016.

[12] Z. M. Jiang and A. E. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," IEEE Trans. Softw. Eng., vol. 41, no. 11, pp. 1091–1118, 2015, doi: 10.1109/TSE.2015.2445340.

[13] S. Suryadevara and S. Ali, "Preperformance Testing of A Website," pp. 33–52, 2020, doi: 10.5121/csit.2020.100703.

[14] N. Husufa and I. Prihandi, "Optimizing JMeter on Performance Testing Using the Bulk Data Method," J. Inf. Syst. Informatics, vol. 4, no. 2, pp. 205–215, 2022, doi: 10.51519/journalisi.v4i2.244.

[15] S. Sharmila and E. Ramadevi, "Analysis of Performance Testing on Web Applications," Int. J. Adv. Reserach Comput. Commun. Eng., vol. 3, no. 3, pp. 2021–2278, 2014.

[16] H. M. AlGhamdi, C. P. Bezemer, W. Shang, A. E. Hassan, and P. Flora, "Towards reducing the time needed for load testing," J. Softw. Evol. Process, pp. 1–18, 2020, doi: 10.1002/smr.2276.

[17] A. Kumar and M. N. Yarlagadda, "Performance, Scalability, and Reliability (PSR) challenges, metrics and tools for web testing : A Case Study," 2016, [Online]. Available: www.bth.se.