Vol. 06, No. 02, July 2022, pp. 146-155

INTERNATIONAL JOURNAL OF INNOVATION IN ENTERPRISE SYSTEM

Journal homepage: https://ijies.sie.telkomuniversity.ac.id
e-ISSN: 2580-3050, DOI: https://doi.org/10.25124/ijies.v6i01.165

PERFORMANCE TESTING ANALYSIS OF BANDUNGTANGINAS APPLICATION WITH JMETER

Ekky Novriza Alam[1*], Fitriyana Dewi[2]

[1] S1 Information System
Telkom University, Bandung, Indonesia
ekkynovrizalam@telkomuniversity.ac.id

[2] S1 Information System
Telkom University, Bandung, Indonesia
fitriyanadewi@telkomuniversity.ac.id

*ekkynovrizalam@telkomuniversity.ac.id

ARTICLE INFO

Article history:
Received 04 June 2022
Accepted 19 July 2022
Published 31 July 2022

ABSTRACT

The bandungtanginas.id application was created to serve the needs of data collection on maternal and child health at every posyandu in the city of Bandung. This application is planned to be able to serve around 1,982 posyandu in the city of Bandung, around 10,000 users, who are expected to be able to access the bandungtanginas.id application. In the current condition of the application support infrastructure still has limitations in supporting the bandungtanginas.id application. With specifications that tend to be low, an improvement effort is needed through generating application usage scenarios so that the services provided can be optimally available using the existing infrastructure. This research focuses on analyzing the application performance by using load testing. This research has four stages: determining the test's purpose, creating a test scenario, carrying out scenario testing, and analyzing the results. The analysis will be carried out with the results of load testing experiments based on the scenarios that have been made. Test scenarios are made to obtain experimental data, last further analysis and conclusions will be made. Scenarios are created based on an estimate of the optimal number of users the application should serve. The test scenario will use a target user of between 10 – 200 users who are divided in five groups of load testing scenarios. From this trial, it will be seen how the performance of the bandungtanginas.id application for each scenario group that has been created. The result of this study showed that the best scenario for bandungtanginas.id application could serve users in the range of 30-50 users. It is obtained from the response time results, which show the average results are under 30 seconds according to the SLA (Service Level Agreement) standard for application users.

Keywords:
Load Testing; Response Time; Reliability; Testing; Service Level Agreement.

1. INTRODUCTION

Various web-based applications are generally made to have the ability to be accessed by hundreds to millions of people at one time [1]. This capability will be measured as the performance of an expected application to always be in the best condition. Users must be able to feel the application's existence and have a good experience when using the application, including a feeling that the application can provide information on user needs easily and quickly [1]. Therefore, performance testing is often done to find out whether an application has behaved satisfactorily or not [2]. Load testing is commonly used to measure the performance requirements of software development. The term load testing is often associated with two other terms: strength testing and stress testing. Each type of testing focuses differently even though

the goal is the same, namely measuring software performance. Load testing can assist developers in determining the time required to complete tasks in the software and ensure software stability under different load conditions but still within the parameters of normal working conditions [3], [4].

Several studies related to testing on web-based applications discuss that availability and response time are two things that need to be considered in assessing application performance [1], [2], [5], [6]. Availability for each user varies according to the time of access. There will be a spike in requests or high traffic to the application at certain events. This event will cause the resource to be suppressed to the limit, so the request will be rejected because the resource cannot fulfill the request. The result is a decrease in the availability of the application. Another thing is related to response time, where this measurement will measure user perceptions regarding the length of time it takes a web-based application page to download or perform a keyword search and return the results.

In 2019 tested website performance on a social security organization in Indonesia using JMeter tools. The research aims to test the application capabilities in various scenarios [6]. Similar research was also carried out in 2020 to test the Torepdo7 website using Jmeter [7]. In contrast to research in the previous year, Torepdo7 website testing focuses on analyzing performance when users increase regularly. There are several application performance test tools. Research conducted in 2019 was to carry out load testing using selenium, aiming to get the best conditions for a website [8]. Research conducted by Rabiya Abbas et al. have compared various testing tools. The Research shows that JMeter has advantages over other test tools such as LoadRunner, Microsoft Visual Studio, and Siege [9], [10].

The BandungTanginas.id application is a website-based application to help reduce the number of stunting cases in Indonesia. This application is used for monitoring and providing education related to health and posyandu programs in Bandung. The hope is that it can simplify the data collection process on maternal and child health in Bandung City through the Bandung City PKK network. Posyandu location points are in every kelurahan in Bandung City. The appearance of the Bandungtanginas website can be seen at Figure 1.

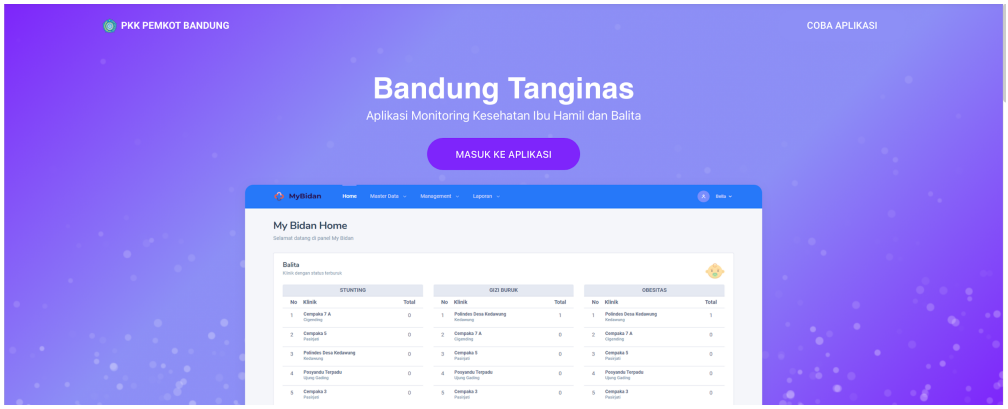

Figure 1 - BandungTanginas.id Website

Bandungtanginas.id will be used for every posyandu activity where mothers and children will have their health checked and weighed. This process will not go on all the time, but when implemented, it will invite multiple users at once. The total number of posyandu in Bandung is 1982 units. Each posyandu has an average of 9 cadres [11]. The availability of the Bandung Tanginas application during the activity will be crucial. The ability of the Bandung Tanginas application to meet the demands of both the number of users and the amount of data that enters the application must be good.

This study will analyze the object's performance to meet stakeholders' needs. The analysis will be conducted based on the experimental results to get the system treatment in different situations. Experiments will be carried out using load testing, which will test the ability of the object based on several experimental scenarios. This research gives recommendations for the best situation so that the object can provide the best service to stakeholders.

2. METHOD

The research methodology adopts the stages of load testing on the software (Figure 2). The process is preceded by determining the purpose of testing, then followed by three main stages, namely designing, executing, and analyzing the results [5].

*ekkynovrizalam@telkomuniversity.ac.id

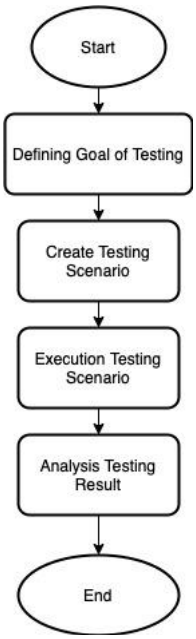

Figure 2 - Research Methodology

In the first stage, the purpose of the test will be determined. Determining this goal is very important to specify the testing scenario's needs at a later stage. This test aims to find out the most significant number of users that can be served by the application when accessed simultaneously. This amount will be used as input to determine a strategy so that the application continues to provide the best service when accessed simultaneously. Next, make a testing scenario in the form of a detailed test plan, such as determining the variables and the value of these variables. Variables can be used, such as the number of users, time, and page to be tested. Load testing design can be divided into two minds: (1) focusing on realistic requirements and (2) focusing on maximizing capabilities until functionality fails [2], [5]. This research will focus on realistic needs. This was chosen because the bandungtanginas.id application has limited operational funds for the provision of server infrastructure. So the goal should be focused on solutions to meet the realistic needs of the application owner first. The test scenarios carried out were five scenarios as shown in Table 1.

Table 1 - Scenario Testing

No	Number of Users	Ramp-up Period
1	10	1 – 10 sec
2	20	5 – 20 sec
3	30	15 -30 sec
3	50	1 – 100 sec
4	100	50 – 200 sec
5	200	200 – 400 sec

The maximum number of users set in the scenario will be 200. The scenario was based on the estimated number of users who will use it from each kelurahan in the city of Bandung. Meanwhile, ramp-up is the time it takes to run the total target number of users [12]. For example, a ramp-up of 50 seconds for a target of 50 users means that the distance between one user and another is 1 second. Because the formula is ramp-up/number of users, this time can be used as a simulation of conditions when several users access the application.

After that, execute the test scenario that has been created. Execution will be carried out with the scenario settings that have been made previously. The scenario will be a guide during execution. Finally, an analysis of the existing test results will be carried out. The analysis was carried out to see the existing pattern of these results. The pattern will be used to determine suggested actions that can be taken to maximize service. The last stage will be making conclusions to answer the objectives of the test. The flow chart for this method is shown in Figure 2

2.1. Load Testing

Software testing is not simple and very dynamic. Therefore, the testing process can exceed 30% for budget, effort and time [13]. But the use of tools and technical approaches can be a way to speed up the testing process. Load testing is a

test that focuses on the ability of the system to serve requests from consumers [14], [15]. Load testing is the process of assessing the behavior of the software under load to detect load-related problems that the software can accommodate. Load-related problems can be in the form of functional or non-functional problems defined previously, such as software stability or robustness. So that the load given in the load test can be varied as needed, whether to carry out a load test according to the expected load when the software is operated or a load that exceeds the expected standard limit.

Load Testing is needed to simulate simultaneous access to website applications. Simulations are better and more precise than testing by inviting hundreds of thousands of people to access the website application simultaneously. Load Testing has a testing method that tests based on actual conditions in the real world so that websites, applications, or software built and developed can be utilized and work optimally when the user already uses them.

2.2. Testing Tools

Many tools can accommodate load testing needs, but for this research using Apache JMeter.

2.2.1. Apache JMeter

Apache JMeter is an open-source Java-based Apache Software Foundation product. This app is designed to load test functional behavior and measure performance on that behavior. Apache JMeter was originally designed for testing web applications but can now be used for performance testing on both static and dynamic resources. The distributed testing feature is one of the advantages of Apache JMeter.

This application is considered quite good in measuring the ability of an application with a variety of features that are quite a lot and complete, especially if one of the factors is finance [16]. In addition, several studies have proven that jMeter has advantages over other testing tools [7], [16], [17].

The use of this application can be started by configuring the thread group configuration. The thread properties section can be filled according to the prepared scenario. For example, a scenario of 50 users with a ramp-up period of 100 seconds as in Figure 3.

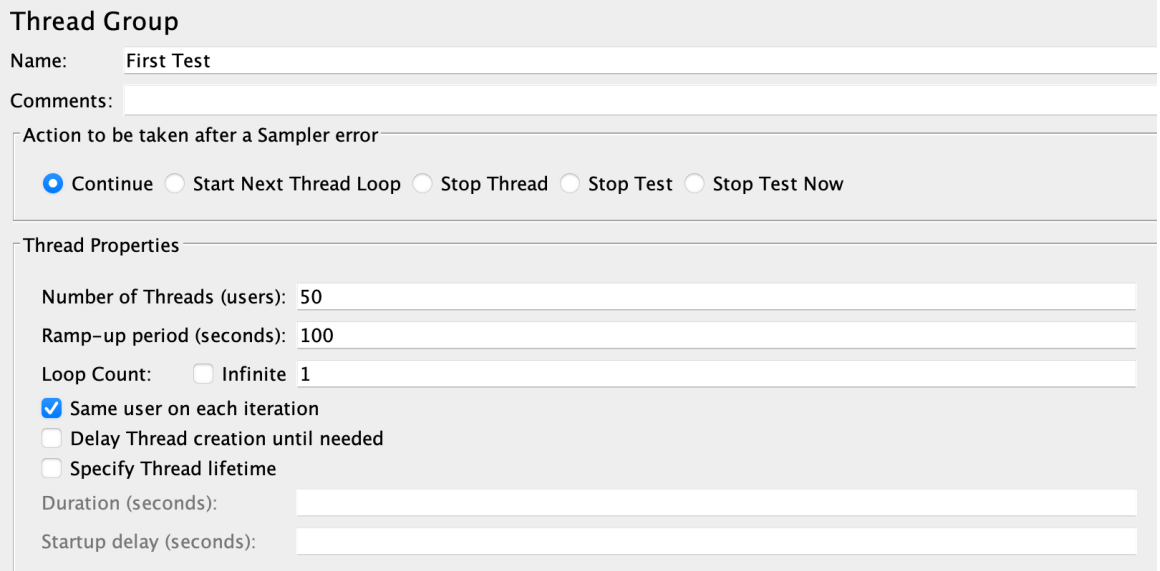

Figure 3 - Setup JMeter Load Test Configuration

2.3. Web Performance Testing Indicators

The performance of an application is measured with the aim of providing the best service for consumers. Thus, consumers will feel satisfied and will not experience disturbances such as errors or slow access services. Tests are carried out to ensure application performance can meet these expectations. The following are indicators that can generally be used as a reference for measuring performance [2], [18]:

1. Response Time

Response time is the time it takes the application to serve a request from the time the request is received until the request is completed. The time will continue to be counted until the last byte is received by the client. The response time should theoretically increase as the number of people accessing the API at the same time grows.

2. Concurrent User

Concurrent users are the number of simultaneous online users accessing system applications in a formal operational environment

3. Throughput:

The amount of customer requests processed by the system per unit time is known as throughput, and it directly represents the software system's carrying capacity. Bits per second is the most common unit of measurement for throughput. During performance testing using Jmeter, however, throughput is measured in requests per second (rps). Theoretically, when the offered load increases, throughput will increase as well, up to the network's maximum capacity.

4. Error Rate

The error rate is a calculation of the percentage of transactions that fail during the test. The error rate that should be generated based on the load test method is 0%. This is since the test is conducted with a normal number of people making simultaneous queries (below the maximum point of the number of requests simultaneously)

3. RESULT AND DISCUSSION

This section will present the experimental results and discussion based on the results obtained. The experiment will be conducted using a cloud server with 1 CPU Core and 1 GB Memory based on Ubuntu OS. Bandungtanginas.id was built using the CodeIgniter 3 framework, which runs using PHP 8.0, and the database is supported by MySQL ver 8.0.

3.1. Result

3.1.1. Execution Testing Scenario

Testing is carried out by setting parameters on the Jmeter component according to the scenario in Table 2. Jmeter has the advantage of being able to record user behavior while on a single feature. Then Jmeter will perform a simulation based on the parameters that have been set using the existing behavior records. The test results can be seen in the following table:

The first scenario group is done with a target number of 10 users with a ramp-up between 1-10 seconds. When the ramp-up is below 10 seconds, it can be seen that the response time reaches more than 12 seconds, while when the ramp-up reaches 10 seconds, the response time drops to 7 seconds. In the first scenario group, the error rate does not occur. Results can be seen in Table 2.

Table 2 - First Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 10, Ramp-up : 1 sec	12779	38.5/min	0 %
2	Users : 10, Ramp-up : 5 sec	12473	33.4 / min	0 %
3	Users : 10, Ramp-up : 10 sec	7794	35.0 / min	0%

The second scenario group is done with a target number of 20 users with a ramp-up of 5-20 seconds. In this scenario group, the user will access the application with the composition of four users/second, two users/second, and one user/second. In this scenario, it can be seen that the lowest response time is obtained from the test scenario with a ramp-up of 20 sec, which is 14.3 sec. There was an increase in response time when the scenario test with a ramp-up of 5 sec was 26.5 sec. Each Error Rate in the second scenario group is still 05 or does not occur. Results can be seen in Table 3.

Table 3 - Second Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 20, Ramp-up : 5 sec	26561	33.8/min	0 %
2	Users : 20, Ramp-up : 10 sec	17974	39.1 / min	0 %
3	Users : 20, Ramp-up : 20 sec	14311	36.4 / min	0%

*ekkynovrizalam@telkomuniversity.ac.id

The third scenario group is done with a target number of 30 users with a ramp-up of 15-30 seconds. In this scenario group, the composition is two users/second and one user/second. The results show that the scenario with a ramp-up of 30 seconds reaches a response time of 16.1 seconds and increases when the ramp-up is lowered to 15 seconds. Results can be seen in Table 4.

Table 4 - Third Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 30, Ramp-up : 15 sec	25933	39.0 / min	0 %
2	Users : 30, Ramp-up : 30 sec	16122	38.0 / min	0%

The fourth scenario group is done with a target number of 50 users with a ramp-up between 1-100 seconds. In this scenario, the composition of the access settings are fifty users/second, two users/second, one user/second, and one user/two seconds. There is a new scenario that is one user/two seconds to see the application's behavior when a user accesses the application with a time lag of 2 seconds but reaches fifty users. The lowest response time results are obtained in the one-user/two-second scenario, which is 2.1 seconds. Response time in one user/second scenario reaches 25.9 seconds, while in other scenarios, it has reached more than 30 seconds. Results can be seen in Table 5.

Table 5 - Fourth Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 50, Ramp-up : 1 sec	71388	37.1/min	0 %
2	Users : 50, Ramp-up : 25 sc	37562	42.5 / min	0 %
3	Users : 50, Ramp-up : 50 sc	25919	36.2 / min	0 %
4	Users : 50, Ramp-up : 100 sc	2127	29.7 / min	0%

The fifth scenario group is carried out with a target number of 100 users with a ramp-up of 50-200 seconds. In this scenario group, the composition of users accessing the application are two users/second, one user/second, and one user/two seconds. All response times are more than 50 seconds except for the one user/two-second scenario, which is 2.6 seconds. In this scenario, the response time starts to reach more than 50 seconds, but the error rate does not occur. Results can be seen in Table 6.

Table 6 - Fifth Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 100, Ramp-up : 50 sec	97811	34.3/min	0 %
2	Users : 100, Ramp-up : 100 sc	58777	33.6 / min	0 %
3	Users : 100, Ramp-up : 200 sc	2642	29.9 / min	0%

The sixth scenario group is done with a target number of 200 users with a ramp-up of 100-400 seconds. In this group scenario, the composition of user access to the application is the same as in the fifth group scenario. The results show that the response time is more than 110 seconds except for the scenario of one user per two seconds. Even though the response time is more than 110 seconds, the error in the application does not occur. It can be seen in the error rate column, which shows 0%. Results can be seen in Table 7.

*ekkynovrizalam@telkomuniversity.ac.id

Table 7 - Sixth Scenario Group Result

No	Test Scenario	Response Time(ms)	Throughput	Error Rate
1	Users : 200, Ramp-up: 100 sc	137257	42.1/min	0 %
2	Users : 200, Ramp-up: 200 sc	110460	33.9/min	0 %
3	Users : 200, Ramp-up : 400 sc	1909	30.0 / min	0 %

3.2. Analysis and Discussion

Application performance will reflect user satisfaction. Users will need sufficient waiting time if the application performance is good. The Service Level Agreement (SLA) on the waiting time will be used as a baseline for whether the application performance is good or not. Waiting time can be categorized as good enough, between 1 – 30 seconds [19].

Experimental results have shown the performance of the bandungtanginas.id application to serve users. In some scenarios, there is a similar pattern with different total target numbers. In the first 1 User / Second pattern, the application can serve well with a maximum total of 50 users. The response time when the user reaches 50 is 25.9 seconds. Meanwhile, for a total of 100 users, the application cannot provide the best performance because the response time reaches 58.7 seconds. However, in the experimental results, the Error Rate column all shows 0%, which means there is no error in the application, and the application can still meet the availability for users. The data graph can be seen in Figure 4.

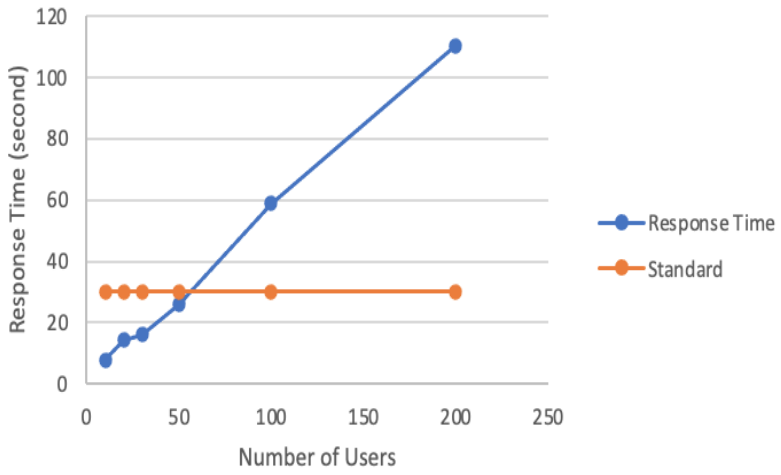

Figure 4 - Response Time Case 1 User/Second

In the second pattern of two users/second, the application can serve well with a maximum total of 30 users. Response time when users reach 30 users is 25.9 seconds. While for a total number of 50 users, the application cannot provide the best performance because the response time has reached 37.5 seconds. It is seen that the response time increases sharply when the user ranges from 50-100 users. However, the experimental results of the Error Rate column all show 0%, meaning there is no error in the application. The data graph can be seen in Figure 5.

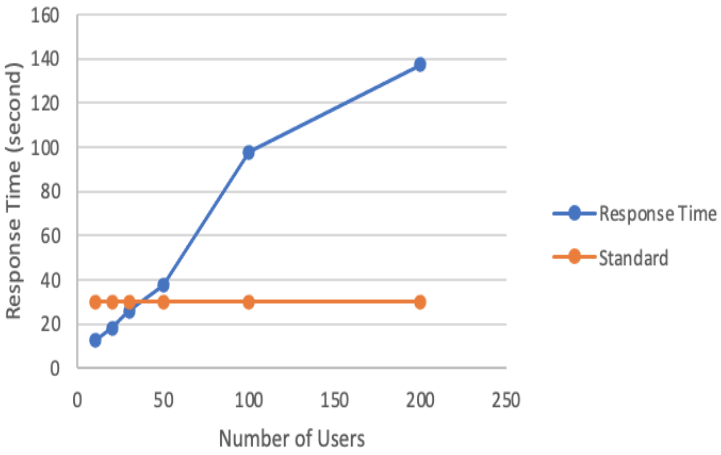

Figure 5 - Response Time Case 2 User/Second

In the one-user/two-detail pattern, the number of users above and equal to 100 shows different things. Even if the user is over 100, the response time tends to decrease as the number of users increases. If this pattern is applied, it is seen that the response time decreases as the number of users increases. It is also seen that the average response time is 2.25 seconds. This response time is very far compared to the one-user/one-second scenario pattern. Based on this behavior, it can be concluded that the average system takes between 1 and 2 seconds to serve requests from each user. The data graph can be seen in Figure 6.

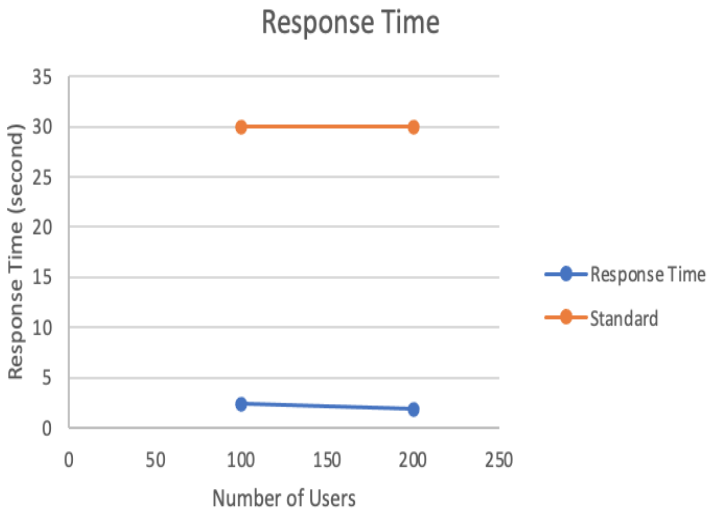

Figure 6 - Response Time Case 1 User/2 Seconds

Based on the analysis of the experimental data above, an access rule can be made so that the performance of the bandungtanginas.id application can serve users well. These rules include 30 users if accessed within the range of 2 users/second and 50 users if accessed within one user/second. The server used by bandungtanginas is the lowest server category in the VPS category that can be provided for the bandungtanginas program. So, it is not possible to suggest upgrading the server infrastructure capabilities. Therefore, this research aims to get the maximum number that the bandungtanginas.id application can serve with the existing infrastructure.

The experimental results and the state of the Bandung Tanginas application infrastructure show conditions that meet the needs. The bandungtanginas application has a target number of users between 5-30 users at one time. Although bandungtanginas is an application implemented at the Bandung city level, each user is relatively small because, in the early stages, the target users are posyandu cadres from certain sub-districts.

*ekkynovrizalam@telkomuniversity.ac.id

The results of this experiment can still be used for website-based applications that use infrastructure support with the proper specifications because the test configuration is carried out in the range of 10-200 users. If the user requires more than 100 concurrent users, the one user / two-second rule can be used according to the experimental results obtained.

4. CONCLUSION

The bandungtanginas.id application with a server infrastructure of 1 CPU Core and 1 GB Memory can serve 30-50 users in a close time span. If using this range, the application can serve users with a target service time per request below or equal to 30 seconds. The service time results obtained are still within the range that is categorized as good when referring to the SLA (Service Level Agreement) target.

The pattern of application performance that can be concluded is that the application experiences a decrease in performance as users increase, but the application can still serve requests without experiencing errors. The decrease is felt when the number of users exceeds 50 users at the same time. The same thing is seen in the throughput indicator which increases based on the number of users who make requests.

Overall, the performance can be said to be good because it does not experience error conditions when demand increases. The solution of restricting user access at the same time can be a solution when resources to provide better infrastructure are not possible.

In the future, the same test pattern can be carried out but with a larger target number of users. But this must also be supported by a larger and more diverse infrastructure in order to get results that can be used in more diverse situations.

Online License Transfer

All authors are required to complete the Procedia exclusive license transfer agreement before the article can be published, which they can do online. This transfer agreement enables Elsevier to protect the copyrighted material for the authors, but does not relinquish the authors' proprietary rights. The copyright transfer covers the exclusive rights to reproduce and distribute the article, including reprints, photographic reproductions, microfilm or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder, the permission to reproduce any figures for which copyright exists.

Disclaimer

The authors whose names are written certify that they have no conflict of interest

REFERENCES

[1] A. Z. Musthafawi, A. Mas'adah, Sukmadiningtyas, and F. Ramdani, "Performance testing on the shopee website in the pandemic period of COVID-19," *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, pp. 195–199, 2020, doi: 10.1145/3427423.3427457.

[2] Arlinta Christy Barus, Johannes Harungguan, and Efren Manulu, "PENGUJIAN API WEBSITE UNTUK PERBAIKAN PERFORMANSI APLIKASI DITENUN," *Journal of Applied Technology and Informatics Indonesia*, vol. 1, no. 2, pp. 14–21, Feb. 2022, doi: 10.54074/jati.v1i2.33.

[3] M. Kalita and T. Bezboruah, "Investigation on performance testing and evaluation of PReWebD: A.NET technique for implementing web application," *IET Software*, vol. 5, no. 4, pp. 357–365, Aug. 2011, doi: 10.1049/iet-sen.2010.0139.

[4] A. Avritzer and E. J. Weyuker, "The Automatic Generation of Load Test Suites and the Assessment of the Resulting Software," 1995.

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

[6] R. Hidayanto and P. Sawitri, "Performance Testing of e-Payment Website Using JMeter," *International Research Journal of Advanced Engineering and Science*, vol. 4, no. 3, pp. 350–352, 2019.

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

[8] S. M. Shariff, H. Li, C. P. Bezemer, A. E. Hassan, T. H. D. Nguyen, and P. Flora, "Improving the testing efficiency of SELENIUM-based load tests," in *Proceedings - 2019 IEEE/ACM 14th International Workshop on Automation of Software Test, AST 2019*, May 2019, pp. 14–20. doi: 10.1109/AST.2019.00008.

*ekkynovrizalam@telkomuniversity.ac.id

[9] R. Abbas, Z. Sultan, and S. N. Bhatti, "Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege," in *International Conference on Communication Technologies, ComTech 2017*, Oct. 2017, pp. 39–44. doi: 10.1109/COMTECH.2017.8065747.
[10] M. Musa, I. Shuaibu, and M. Musa Machina, "Investigation onto the Soware Testing Techniques and Tools: An Evaluation and Comparative Analysis," 2019.
[11] Pemerintah Kota Bandung, "Portal Data Kota Bandung," 2020. http://data.bandung.go.id/beta/index.php/portal/dataset?keywords=kader+posyandu (accessed Jul. 02, 2022).
[12] "Apache JMeter - User's Manual: Elements of a Test Plan." https://jmeter.apache.org/usermanual/test_plan.html (accessed Mar. 18, 2022).
[13] A. A. S. Ahmad, P. Brereton, and P. Andras, "A Systematic Mapping Study of Empirical Studies on Software Cloud Testing Methods," *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, pp. 555–562, 2017, doi: 10.1109/QRS-C.2017.94.
[14] R. Khan and M. Amjad, "Performance testing (load) of web applications based on test case management," *Perspectives in Science*, vol. 8, pp. 355–357, 2016, doi: 10.1016/j.pisc.2016.04.073.
[15] U. Khaira Latif and T. F. Kusumasari, "COMPARISON BETWEEN YII FRAMEWORKS AND LARAVEL IN 3 DIFFERENT VERSION FOR VIEWING LARGE DATA OF SHIPYARD INDUSTRY IN INDONESIA," 2018.
[16] N. Srivastava, U. Kumar, and P. Singh, "Software and Performance Testing Tools," *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, vol. 2, no. 1, pp. 1–12, 2021, doi: 10.54060/jieee/002.01.001.
[17] M. A. Putri, H. N. Hadi, and F. Ramdani, "Performance testing analysis on web application: Study case student admission web system," in *Proceedings - 2017 International Conference on Sustainable Information Engineering and Technology, SIET 2017*, Feb. 2018, vol. 2018-January, pp. 1–5. doi: 10.1109/SIET.2017.8304099.
[18] J. Wang and J. Wu, "Research on performance automation testing technology based on JMeter," in *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, Jun. 2019, pp. 55–58. doi: 10.1109/ICRIS.2019.00023.
[19] K. F. Salmawati, T. F. Kusumasari, and E. N. Alam, "Carte server implementation for improving data quality management application performance in profiling module," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1010, no. 1. doi: 10.1088/1757-899X/1010/1/012012.