



PERFORMANCE TESTING ON WEB INFORMATION SYSTEM USING APACHE JMETER AND BLAZEMETER

Indrianto^{1,*}

¹ Faculty of Informatika and Computer, ITB STIKOM BALI, Bali, Indonesia

Corresponding author email: 230210103005@mail.unej.ac.id

Article Info

Received: 24 September 2023

Revised: 13 November 2023

Accepted: 2 December 2023

OnlineVersion: 11 December 2023

Abstract :

The Abiansemal Bali Global Vocational School Teacher Information System is a web-based application that makes teachers' work easier, which is the implementation of applied science in the field of technology. As the number of teachers increases, server performance needs to be evaluated to ensure it can handle requests from each teacher. This research tested the performance of an Information System Website using two tools, namely Apache JMeter and BlazeMeter. This test includes four modules: login page, updating teacher profile, uploading images, and creating questions. Testing uses 50 and 100 samples with a ramp-up period of 10 seconds and several loops of 1. Test results show that the system works well on these modules, with a stable average response time, increased throughput, and decreased deviation. This information reveals system performance and can help make improvements and optimizations to improve the speed and quality of the user experience. Performance testing with Apache JMeter and BlazeMeter is a useful method for testing the performance of web-based information systems.

Keywords: Apache JMeter, BlazeMeter, Load Testing, Response Time, Skalability

This is open access article under the [CC BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) licence



INTRODUCTION

Information system is a systematic combination of people, hardware, software, communication networks, and data resources that collect, transform, and disseminate information within an organization. Utilizing information technology support through an information system, the teaching-learning process, exam creation, and input of exam results can be carried out more quickly and cost-effectively (Arul & Asokan, 2014; Memon et al., 2018; Paz & Bernardino, 2018). The information system designed for teachers in Bali Global Vocational School, Abiansemal, Badung, is a solution to ensure that teaching and administrative services can be carried out more effectively and efficiently.

Performance testing of a web-based application is a process conducted to evaluate the performance of a website under high traffic loads. The goal of this testing is to ensure that the website can provide quick responses, high availability, and optimal performance to users (Memon et al., 2018; Mahboubeh, Araban, Paydar, 2020). In the current digital era, users have high expectations for quick response times and a good user experience when accessing websites. Therefore, website performance testing becomes crucial to ensure that the website can meet these expectations. Website performance testing involves simulating traffic loads using testing tools such as Apache JMeter and Blazemeter. These tools allow users to simulate access to the website with different scenarios, including a large number of users and intensive activities (Paz, S., & Bernardino, 2017; Mahboubeh, Araban, Paydar S, 2020; Kołtun & Pańczyk, 2020).



Load Testing is the simplest form of performance testing. A load test is usually conducted to understand the behavior of the application under for a specific expected load. This load can be the expected number of concurrent users on the application performing a specific number of transactions within the set duration. This test will give out the response times of all the important business critical transactions. If the database, application server, etc are also monitored, then this simple test can itself point toward the bottleneck in the application (Markande & Murthy, 2013; Kumar, 2015; Maniyar et al., 2018; Ali, Maghawry, & Badr, 2022).

Apache JMeter is a versatile and open-source load testing tool used to perform performance testing on web applications. Developed by the Apache Software Foundation, JMeter can simulate high user loads by sending HTTP requests to the web server and measuring its performance and responsiveness. JMeter is designed to support comprehensive performance testing. BlazeMeter is a performance testing platform used to measure, analyze, and optimize the performance of web applications, websites, web services, and related infrastructure. It allows users to conduct realistic load and stress testing on their web applications. BlazeMeter offers powerful features for performance testing, including the ability to record and replay testing scenarios, flexible scaling to simulate diverse traffic loads, and real-time performance monitoring. The platform also provides in-depth analysis of key performance metrics, such as response time, transaction speed, request rate per second, and error rate.

In the context of this research, BlazeMeter is used as one of the tools to record testing scenarios in the information system website of Bali Global Vocational School. By using BlazeMeter, researchers can record various user activities, such as login, user profile updates, image uploads, and question creation, and then replay these scenarios to evaluate the system's performance. In website performance testing, several measured parameters include: 1) Response Time: The time it takes for the server to respond to user requests. Slow response times can result in a poor user experience and potentially increased bounce rates (Agnihotri & Phalnikar, 2018). 2) User Load: The number of users simulated to access the website simultaneously. Testing is done by increasing the number of users to see how the website handles high traffic loads. 3) Scalability: The website's ability to handle an increase in the number of users or traffic loads without a decrease in performance. This testing aims to determine the website's maximum limits and its capacity to handle high traffic. 4) Server Load: This testing involves measuring server resource usage such as CPU, memory, and bandwidth during testing. It is essential to ensure that the server can handle high traffic loads without failure (Wang & Wu, 2019).

The addition of server resources does not have a significant impact on improving the performance of the online exam system if additional settings are not made to the application technology. The existence of hardware limitations makes it necessary to consider using a load balancer and separating the database server from the application server (Ahmad, Mehmood, & Kim, 2019; Çapari, Elmazi, & Prieditis, 2022; Ismail, 2023). Result of Load testing help determine how the software behaves when several users access the software simultaneously. Gorilla Testing techniques are used to ensure that the module functions correctly and there are no bugs. Modules can be tested more than a hundred times, and in the same way. Gorilla Testing is very useful for testing the robustness of an application (Patel et al., 2014; Dangwal et al., 2016; Mukhiya et al., 2019; Permatasari, 2020; Samli & Orman, 2023).

Previous studies on web-based information systems have focused on the functional aspects, such as usability, security, and reliability, but have not paid much attention to the performance aspects, such as response time, throughput, and deviation. Moreover, most of the existing performance testing methods are either too complex, too costly, or too time-consuming to be applied in real-world scenarios. The objective of this research is to assess the present capacity of the web server in managing user requests. Can the web server effectively accommodate up to 100 users? The outcomes of website performance testing offer valuable insights into the website's performance and its capability to withstand significant traffic loads. This data can be utilized to implement enhancements and optimizations, thereby improving responsiveness and the overall quality of the user experience.

RESEARCH METHOD

This research using Observational research methodology, where observe and systematically record the behavior of individuals or groups in natural environment without intervention. This research carried out performance tests on information system websites using Apache JMeter and Blazemeter.

The sample used in this research was a web-based information system at the Bali Global Abiansemal Vocational School. Some of the modules used in this research include: login page, update profile, upload image, and create question page.

The sampling technique is based on user actions, namely when logging in, when updating the user profile, when uploading image, and creating questions. The sampling technique is also based on traffic volume, namely, using 50 samples and 100 samples. The stages conducted are as follows: a) Identification of Objectives: Determine the performance testing objectives to be achieved. For example, objectives may include measuring website response times, determining the maximum load capacity that can be handled, or identifying the causes of performance degradation (Mahboubeh, Araban, & Paydar, 2020); b) Determination of Testing Scenarios: Define the testing scenarios to be used. Testing scenarios may include the number of users, types of activities performed, and the level of generated traffic load. These scenarios should reflect realistic user behavior; c) Preparation of Testing Environment: Prepare the appropriate testing infrastructure and environment. This includes setting up servers, configuring Apache JMeter, and ensuring the availability of sufficient resources to conduct testing; d) Configuration of Apache JMeter: Configure Apache JMeter according to the defined testing scenarios. Set up thread groups to control the number of users and user behavior. Define the performance measurements to be taken, such as response time or server load; e) Execution of Testing: Run the testing scenarios using Apache JMeter. Monitor and record the testing results, including measured performance metrics such as response time, server load, and error rates (Wang & Wu, 2019); f) Analysis of Results: Analyze the testing results to gain insights into the website's performance. Identify factors influencing performance, such as bottlenecks or excessive resource usage. Evaluate whether the website's performance meets the established objectives (Barraood, Mohd, & Baharom, 2021); g) Remediation and Optimization Actions: If there are issues or limitations in the website's performance, take necessary remediation and optimization actions. For example, increase server capacity, fix application code, or implement caching to reduce server load. (Agnihotri & Phalnikar, 2018).

The data analyzed in this study includes the execution time required for user requests, standard deviation, and average throughput of the server. Additionally, it covers the average data reception and transmission by users, simulated with 50 and 100 users. The virtual server configuration comprises 4 GB of RAM, 50 Elastic Processes (EP), 2 CPU Cores, 5096 Input/Output Operations Per Second (IOPS), and an Input/Output (IO) speed of 250 MB/s.

RESULTS AND DISCUSSION

Load testing is conducted by recording the information system website of Bali Global Abiansemal Vocational School using the Blazemeter and Apache JMeter applications. This testing is performed on several pages, including the login page, teacher profile update, photo upload, and question creation.

Login Scenario

In the login scenario, Blazemeter is first executed to record the login process, as shown in Figure 2.



Figure 2. Recording login page using blazemeter

The results recorded on the blazemeter, as shown in Figure 3, are a testing scenario that will be run with Apache JMeter. The testing scenario uses samples of 50 and 100 samples, with a ramp up period of 10 seconds and a loop count of 1 time.

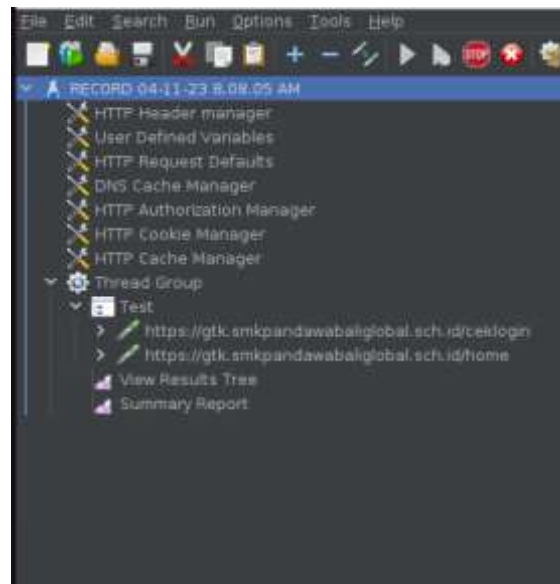


Figure 3. Open recorded file from blazemeter

The test results showed that the shortest time to execute was 228 ms and the longest time to execute was 1368 ms, so the average time to execute this process was 440 ms. Standard deviation 246.59, with a total error of 0%, average throughput 4.95/sec, average data reception 3,395 kb/sec, average data transmission 6.07 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gtk.smkpa...	50	440	228	1368	246.59	0.00%	5.0/sec	3.43	6.08	701.0
Test	50	440	228	1368	246.59	0.00%	4.9/sec	3.38	6.05	701.0
TOTAL	100	440	228	1368	246.59	0.00%	9.8/sec	6.78	12.13	701.0

Figure 4. Test results for 50 samples on Apache JMeter

Next, testing was carried out using 100 samples. The test results are as shown in Figure 5. The minimum time to execute is 186 ms, and the longest time is 2535 ms. The average time required to execute is 411 ms. Standard deviation 452.54, with a total error of 0%, average throughput 9.7/sec, average data reception 6.66 kb/sec, average data transmission 11.92 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gtk.smkpa...	100	411	186	2535	452.54	0.00%	9.7/sec	6.66	11.92	701.0
Test	100	411	186	2535	452.54	0.00%	9.7/sec	6.66	11.92	701.0
TOTAL	200	411	186	2535	452.54	0.00%	19.5/sec	13.33	23.84	701.0

Figure 5. Test results with 100 samples on Apache JMeter

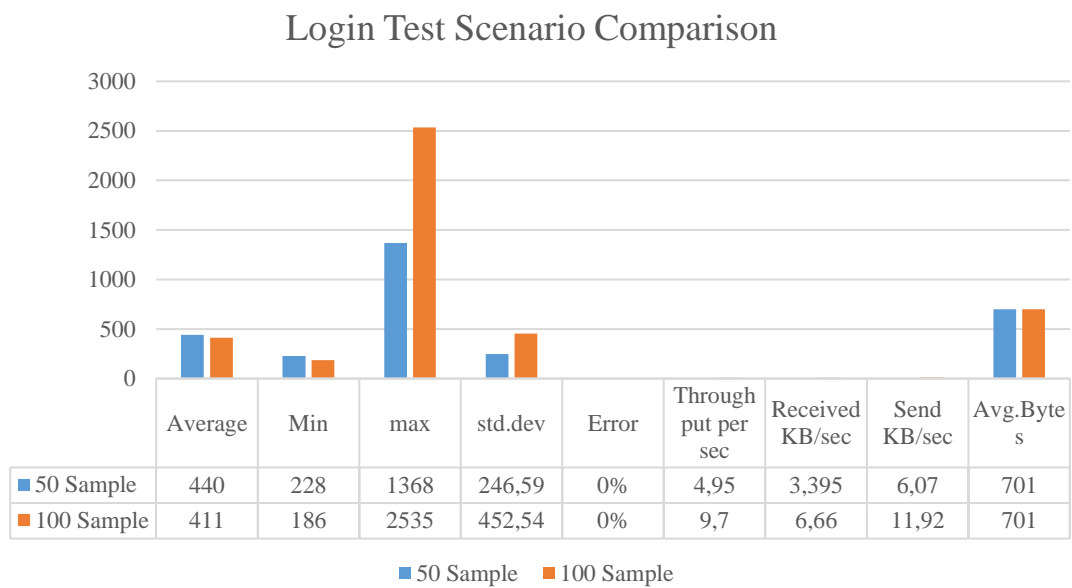


Figure 6. Login Test Scenario Comparison Chart

User Profile Update Page Scenario

The user profile update page scenario is to monitor the user profile update process. The process of recording this page is as shown in Figure 7.

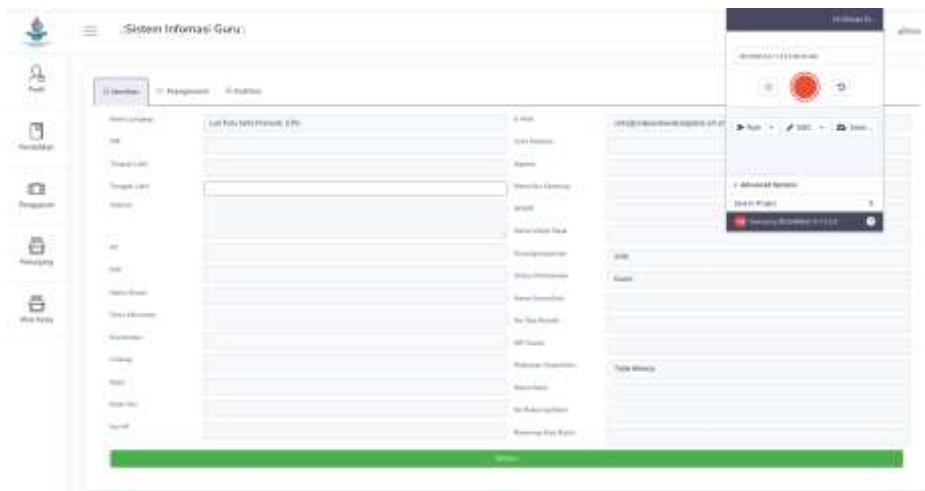


Figure 7. Recording user profile update page by blazemeter

Next is the testing scenario for the user profile update page on Apache JMeter, as shown in Figure 8. This testing scenario uses 50 and 100 samples with a ramp up period of 100 seconds and a loop count of 1 time.

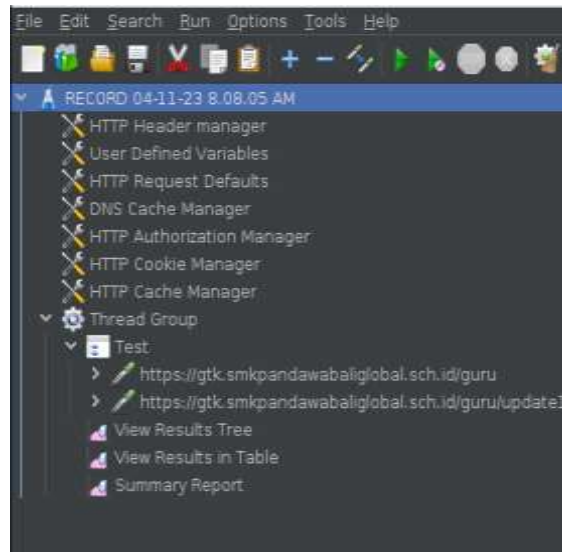


Figure 8. Testing Scenario on Apache JMeter

The results of testing the teacher update testing scenario on 50 samples are shown in Figure 9. The minimum time to execute is 137 ms, and the longest time is 1211 ms. The average time required to execute is 224 ms. Standard deviation 182.39, with a total error of 0%, average throughput 5.0/sec, average data reception 3.47 kb/sec, average data transmission 4.65 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gtk.smkp...	50	224	137	1211	182.39	0.00%	5.0/sec	3.47	4.65	707.0
Test	50	224	137	1211	182.39	0.00%	5.0/sec	3.47	4.65	707.0
TOTAL	100	224	137	1211	182.39	0.00%	10.0/sec	6.93	9.30	707.0

Figure 9. Test Result for 50 samples on Apache JMeter

The results of testing the teacher update testing scenario on 50 samples are as shown in Figure 9. The minimum time to execute is 137 ms, and the longest time is 1211 ms. The average time required to execute is 224 ms. Standard deviation 182.39, with a total error of 0%, average throughput 5.0/sec, average data reception 3.47 kb/sec, average data transmission 4.65 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gtk.smkp...	100	193	120	1020	116.20	0.00%	10.0/sec	6.88	9.24	707.0
Test	100	193	120	1020	116.22	0.00%	10.0/sec	6.88	9.23	707.0
TOTAL	200	193	120	1020	116.21	0.00%	19.9/sec	13.76	18.47	707.0

Figure 10. Test Result for 100 samples on Apache JMeter

User Profile Update Comparison Test

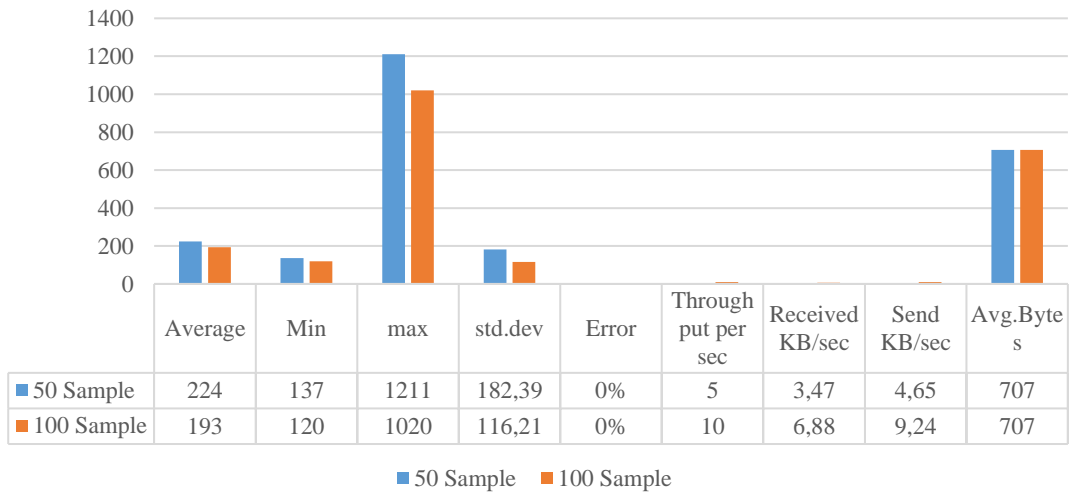


Figure 11. Update Scenario Test Comparison Chart

Image Upload Scenario

The scenario for the image upload page is to monitor the process of uploading teacher profile photos. The process of recording this page is as shown in Figure 12.

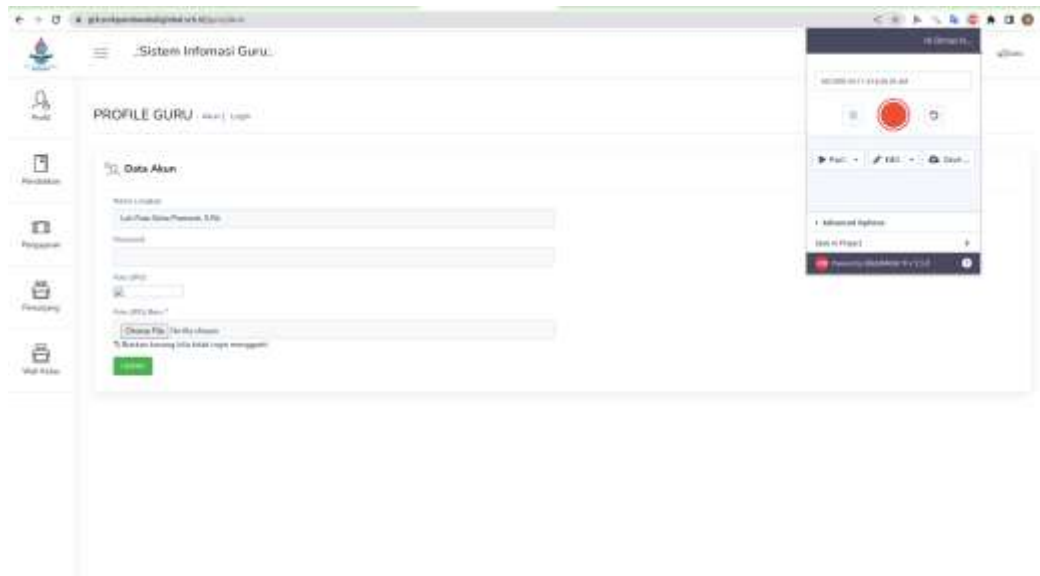


Figure 12. Recording process on image upload page using blazemeter

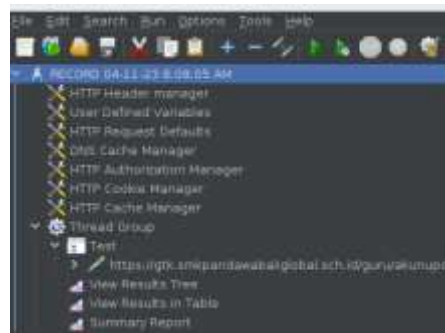


Figure 13. Upload image testing scenario on Apache JMeter

Next is the photo upload testing scenario on Apache JMeter, as shown in Figure 13. This testing scenario uses 50 and 100 samples with a ramp up period of 10 seconds and a loop count of 1. The results of testing the 50 testing scenario are as follows.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gk.smp...	50	371	110	1091	314.15	0.00%	23.8/sec	16.44	15.72	707.0
Test	50	371	110	1091	314.15	0.00%	23.8/sec	16.44	15.71	707.0
TOTAL	100	371	110	1091	314.15	0.00%	47.6/sec	32.86	31.42	707.0

Figure 14. Test Result by 50 samples on Apache JMeter

Test results using 50 samples on Apache JMeter, as shown in Figure 14, state that the minimum time to execute is 110 ms, and the longest time is 1091 ms. The average time required to execute is 371 ms. Standard deviation 314.15, with a total error of 0%, average throughput 23.8/sec, average data reception 16.44 kb/sec, average data transmission 15.72 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
https://gk.smp...	100	493	115	1324	397.89	0.00%	47.4/sec	32.74	31.30	707.0
Test	100	493	115	1324	397.89	0.00%	47.4/sec	32.74	31.30	707.0
TOTAL	200	493	115	1324	397.89	0.00%	94.8/sec	65.47	62.60	707.0

Figure 15. Test Result by 100 samples on Apache JMeter

The results of testing 100 samples on Apache JMeter are as shown in Figure 15. The minimum time to execute is 115 ms, and the longest time is 1324 ms. The average time required to execute is 493 ms. Standard deviation 397.89, with a total error of 0%, average throughput 47.4/sec, average data reception 32.74 kb/sec, average data transmission 31.3 kb/sec.

Test Comparison on Image Upload

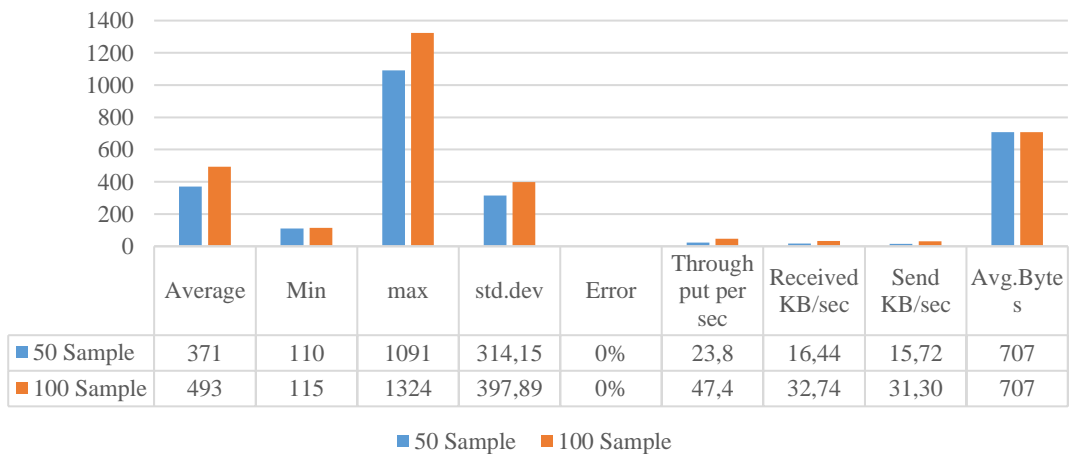


Figure 16. Image Upload Comparison Test Chart

Create Question Page Scenario

The create question page testing scenario is to monitor the question creation process. The process of recording this page is as shown in Figure 17.

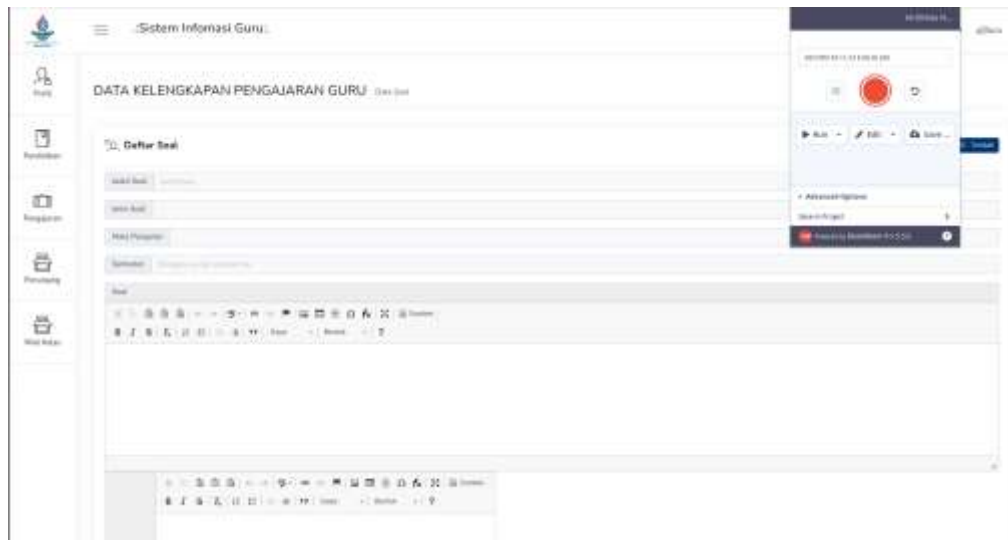


Figure 17. Recording Create Question Page by blazemeter

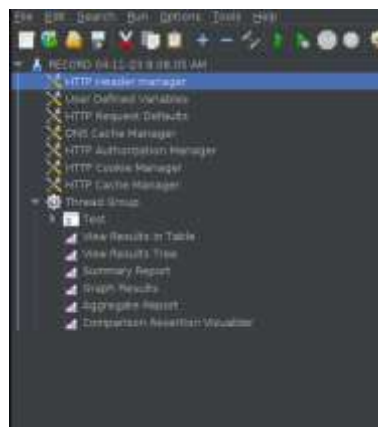


Figure 18. Testing Scenario for create question page on Apache JMeter

Next, the testing scenario for create question page on Apache JMeter, as shown in Figure 18. This testing scenario uses 50 and 100 samples with a ramp up period of 10 seconds and a loop count of 1. The results of testing the 50 testing scenario are as follows.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Average Data Rec.	Average Data Trans.	Req. Bytes
http://g2.2162...	50	161	121	775	114.36	0.00%	5.0/sec	3.48	4.48	137.0
Test	50	161	121	775	114.36	0.00%	5.0/sec	3.48	4.48	137.0
TOTAL	50	161	121	775	114.36	0.00%	5.0/sec	3.48	4.48	137.0

Figure 19. Test Result for 50 samples on Apache JMeter

Test results using 50 samples on Apache JMeter, as shown in Figure 19, state that the minimum time to execute is 121 ms, and the longest time is 775 ms. The average time required to execute is 161 ms. Standard deviation 114.36, with a total error of 0%, average throughput 5.0/sec, average data reception 3.48 kb/sec, average data transmission 4.48 kb/sec.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Average Data Rec.	Average Data Trans.	Req. Bytes
http://g2.2162...	100	219	117	1484	261.35	0.00%	10.0/sec	6.89	8.88	137.0
Test	100	219	117	1484	261.35	0.00%	10.0/sec	6.89	8.88	137.0
TOTAL	100	219	117	1484	261.35	0.00%	10.0/sec	6.89	8.88	137.0

Figure 20. Test Result for 100 samples on Apache JMeter

Test results using 100 samples on Apache JMeter, as shown in Figure 20, state that the minimum time to execute is 117 ms, and the longest time is 1484 ms. The average time required to execute is 219 ms. Standard deviation 261.35, with a total error of 0%, average throughput 10.0/sec, average data reception 6.89 kb/sec, average data transmission 8.88 kb/sec.

Test Comparison on Create Question Page

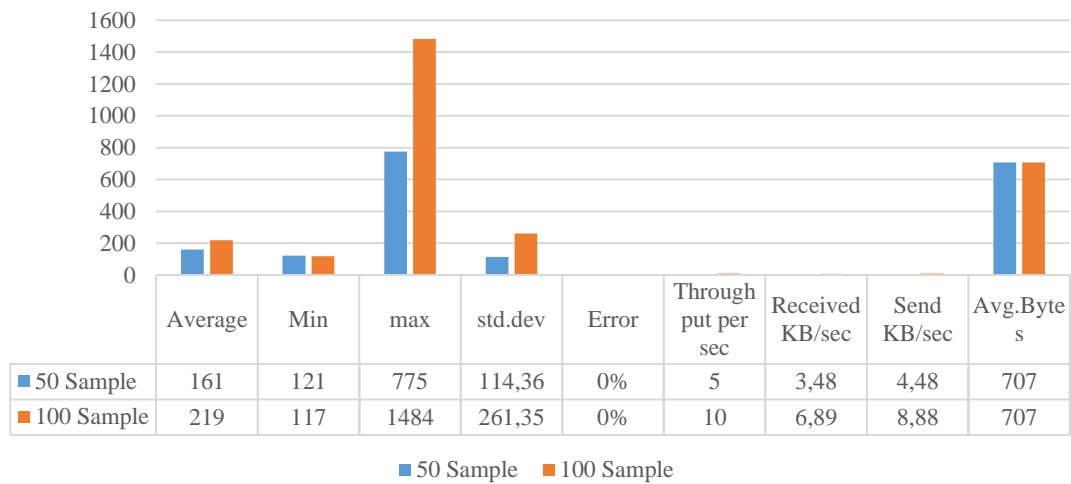


Figure 21. Create Question Page Scenario Comparison Chart

The test results show that the average test processing time for 50 samples is 299 ms, while for 100 samples it is 329 ms. The average standard deviation for a test of 50 samples is 214,37, while for a test of 100 samples is 306,997. The average throughput for testing 50 samples is 5,6875 per second, while for testing 100 samples the average is 19,275 per second. The average data received for testing 50 samples was 6,69625 kb/sec, while for 100 samples it was 13,2925 kb/sec. The average data that can be sent for a 50 sample test is 7,73 kb/sec, while for a 100 sample test it is 15,3334 kb/sec.

Table 1. Average Test Processing Comparison.

Num Sample	Execution Time	Std Dev	throughput	Received	Send
50	299 ms	214.37	5.6875/sec	6.69625 KB/sec	7.73 KB/sec
100	329 ms	306.997	19.275/sec	13.2925 KB/sec	15.3334 KB/sec.
Increase	91%	70%	30%	50%	50%

The research results show that a server with specifications of 4 GB RAM, 50 EP, 2 CPU Core 5096 IOPS, 250 MB/s IO, is able to serve 100 users very well. However, based on data that the trend of increasing execution time from 50 users to 100 users is 91%, standard deviation increases 70%, throughput increases 30%, data reception increases 50%, data transmission increases 50%, so it can be concluded that if later the number of users exceeds 100 users, then the server specifications need to be increased.

This study introduces a novelty approach to performance testing of web-based information systems using Apache JMeter and Blazemeter, two open-source and cloud-based tools that are easy to use, affordable, and efficient. The study also applies the performance testing to a specific case of the Teacher Information System of Bali Global Abiansemal Vocational School, a web-based application that serves the teachers in their duties. The study has some limitations that need to be addressed in future research. First, the study only tests four modules of the Teacher Information System, which may not represent the whole system functionality. Second, the study only uses two sample sizes, 50 and 100, which may not capture the full range of possible user scenarios. Third, the study does not compare the performance results with other web-based information systems or other performance testing tools, which may limit the generalizability and validity of the findings.



CONCLUSION

The Web Server is capable of handling requests from 50 users simultaneously well. However, when the number of users increases to 100, there is a decrease in server performance, although in general the server is still able to serve user requests. When the number of user requests is more than 100, it is necessary to increase server capacity, such as memory and processor.

REFERENCES

- Agnihotri, J., & Phalnikar, R. (2018). Development of performance testing suite using apache JMeter. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-981-10-7245-1_32
- Ahmad, S., Mehmood, F., & Kim, D. H. (2019). A DIY approach for the design of mission-planning architecture using autonomous task-object mapping and the deployment model in mission-critical IoT systems. *Sustainability*, *11*(13), 3647. <https://doi.org/10.3390/su11133647>
- Ali, A., Maghawry, H. A., & Badr, N. (2022). Performance testing as a service using cloud computing environment: A survey. *Journal of Software: Evolution and Process*, *34*(12), e2492. <https://doi.org/10.1002/smr.2492>
- Arul, D. P., & Asokan, M. (2014). Load testing for query based e-commerce web applications with cloud performance Testing Tools. *International Journal of Computer Engineering & Technology (IJCET)*, *5*(10), 01-10.
- Barraood, S. O., Mohd, H., & Baharom, F. (2021). Test case quality factors: content analysis of software testing websites. *Webology*, *18*(SpecialIssue). <https://doi.org/10.14704/WEB/V18SI01/WEB18007>
- Çapari, K., Elmazi, D., & Prieditis, M. (2022). Efficiency performance evaluation on multi-user web application platforms in cloud computing. *International Journal of Innovative Technology and Interdisciplinary Sciences*, *5*(3), 1014-1032. <https://doi.org/10.1515/IJITIS.2022.5.3.1014-1032>
- Dangwal, N., Dewan, N. M., & Sachdeva, S. (2016). Testing the cloud and testing as a service. *Encyclopedia of Cloud Computing*, 338-348.
- Ismail, A. (2023). Performance testing sistem ujian online menggunakan jmeter pada lingkungan virtual. *Jurnal Informatika Polinema*, *9*(2). DOI: <https://doi.org/10.33795/jip.v9i2.1190>
- Kořtun, A., & Pańczyk, B. (2020). Comparative analysis of web application performance testing tools. *Journal of Computer Sciences Institute*, *17*, 351-357. <http://dx.doi.org/10.35784/jcsi.2209>
- Kumar, M. (2015). A comparative study of black box testing and white box testing techniques. *International Journal of Advance Research in Computer Science and Management Studies* <https://ijarcsms.com/docs/paper/volume3/issue10/V3I10-0018.pdf>
- Mahboubeh, D., Araban S., & Paydar S. (2020). A systematic literature review on semantic web enabled software testing. *Journal of Systems and Software*, *162*, 110485, <https://doi.org/10.1016/j.jss.2019.110485>
- Maniyar, M. A. R., Hakeem, M. A., & Zafar, M. K. M. (2018). Bottom-up approach for performance testing of software applications or products. *International Journal of Innovative Research in Computer and Communication Engineering*, *6*(9), 7812-7817. <http://dx.doi.org/10.15680/IJIRCCE.2018.0609013>
- Markande, K., & Murthy, S. J. (2013). Leveraging potential of cloud for software performance testing. *Cloud Computing: Methods and Practical Approaches*, 293-322.
- Memon, P., Dewani, A., Bhatti, S., & Hafiz, T. (2018). Scrutinizing automated load testing via blazmester, apache jmeter and hp loadrunner. *International Journal of Advanced Studies in Computers, Science and Engineering*, *7*(3), 35-42.
- Memon, P., Hafiz, T., Bhatti, S., & Qureshi, S. S. (2018). Comparative study of testing tools Blazemeter and Apache JMeter. *Sukkur IBA Journal of Computing and Mathematical Sciences*, *2*(1), 70-76. <https://doi.org/10.30537/sjcms.v2i1.66>
- Mukhiya, S. K., Rabbi, F., Pun, V. K. I., Rutle, A., & Lamo, Y. (2019). A GraphQL approach to healthcare information exchange with HL7 FHIR. *Procedia Computer Science*, *160*, 338-345. <http://dx.doi.org/10.1016/j.procs.2019.11.082>



-
- Permatasari, D. I. (2020). Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 8(1) DOI:<https://dx.doi.org/10.26418/justin.v8i1.34452>.
- Patel, B., Parikh, J., & Shah, R. (2014). A review paper on comparison of SQL performance analyzer tools: Apache JMeter and HP LoadRunner. *International Journal of Current Engineering and Technology*, 4(5), 3642-3645.
- Paz, S., & Bernardino, J. (2018). Web platform assessment tools: An experimental evaluation. In *Web Information Systems and Technologies: 13th International Conference, WEBIST 2017, Porto, Portugal, April 25–27, 2017, Revised Selected Papers 13* (pp. 45-63). Springer International Publishing. https://doi.org/10.1007/978-3-319-93527-0_3
- Paz, S., & Bernardino, J. (2017). Comparative analysis of web platform assessment tools. In *WEBIST* (pp. 116-125). <https://doi.org/10.5220/0006308101160125>
- Samli, R., & Orman, Z. (2023). A comprehensive overview of web-based automated testing tools. *İleri Mühendislik Çalışmaları ve Teknolojileri Dergisi*, 4(1), 13-28.
- Wang, J., & Wu, J. (2019). Research on performance automation testing technology based on JMeter. *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*. <https://doi.org/10.1109/ICRIS.2019.00023>