Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Performance Analysis of Nginx and Apache2 Web Servers on Debian Using Stress Testing Methods

Jusnan Panggabean[1]*, Adri Muliadi Pasaribu [2], Lotar Mateus Sinaga [3]

[1,2,3]*Faculty of Computer Science, Informatics Engineering Study Program, Santo Thomas Catholic University Medan*
jusnanpanggabean@gmail.com

Abstract

Web servers play a crucial role in supporting web-based services by receiving and processing user requests. The performance of a web server is one of the key factors that determines service quality, especially when the server is required to handle a large number of concurrent users. Therefore, selecting an appropriate web server is essential to ensure optimal system performance.
This study aims to compare the performance of two widely used web servers, Nginx and Apache2, running on the Debian operating system using a stress testing approach. The testing process was conducted by generating simultaneous access loads using several benchmarking tools, including Apache Benchmark (ab), Autocannon, and JMeter. The performance parameters evaluated in this study include throughput, latency, requests per second (RPS), response time, and server resource utilization such as CPU and memory usage.
The results show that Nginx is able to handle a large number of concurrent connections more effectively than Apache2. This is indicated by its lower latency values and higher requests-per-second performance. Meanwhile, Apache2 demonstrates relatively stable throughput across various testing scenarios. Based on the findings, Nginx is recommended for environments with high traffic demands and resource efficiency requirements. On the other hand, Apache2 remains a suitable choice for applications that require greater configuration flexibility and extensive module support.

Keywords: *Web Server, Nginx, Apache2, Debian, Stress Testing.*

1. Introduction

The development of information technology in recent years has brought about significant changes in various aspects of life. Currently, almost all activities, whether in education, government, healthcare, or business, utilize web-based services as the primary means of delivering information and services to users. As the number of internet users increases, the need for fast, stable web services capable of serving many users simultaneously also increases.
In a web-based system, the web server plays a crucial role, receiving user requests, processing them, and then sending appropriate responses. Web server performance is one of the factors determining the quality of a website's service. If the web server is unable to handle a high volume of requests, users may experience access delays, decreased performance, and even service failures. Therefore, selecting the right web server is crucial when developing and managing a web-based information system.
Currently, there are various types of web servers available, both open source and commercial. The two most widely used open source web servers are Apache HTTP Server and Nginx. Both have different characteristics, advantages, and architectural approaches to handling user requests.
Apache HTTP Server is one of the oldest web servers still in use. Apache is known for its comprehensive documentation, extensive community support, and broad compatibility with various web applications. Furthermore, Apache provides numerous modules that allow server administrators to configure it according to their needs. This flexibility is what keeps Apache popular despite the emergence of more modern web server alternatives.
On the other hand, Nginx offers a different approach. Nginx is designed using an event-driven architecture that allows the server to handle multiple connections simultaneously with more efficient resource usage. Thanks to this architecture, Nginx is often used on websites with high traffic levels because it can provide good performance without requiring excessive server resources.
The architectural differences between Apache and Nginx result in different performance characteristics. Apache uses a process-based or thread-based approach to handling each incoming connection. Meanwhile, Nginx uses an asynchronous event-driven model that allows a single process to handle multiple connections simultaneously. These differences raise the question of which web server performs better when implemented in the same server environment.
One method that can be used to measure a web server's capabilities is stress testing. Stress testing is a testing method that involves applying a large workload to determine the limits of a system's capabilities. This method allows us to determine how a web server responds to increasing connections, changes in response time, server resource usage, and the system's ability to maintain performance under high load.

In this study, testing was conducted on the Debian operating system. Debian was chosen because it is a Linux distribution known for its stability, light weight, and widespread use as a server operating system in various production environments.
This study aims to analyze and compare the performance of Apache2 and Nginx web servers on the Debian operating system using stress testing. Testing was conducted by gradually applying access loads using benchmarking tools such as Apache Benchmark (ab), Autocannon, and JMeter. Parameters analyzed included throughput, latency, response time, requests per second (RPS), CPU usage, and memory usage. The results of this study are expected to provide an overview of the performance characteristics of each web server and serve as a reference for system administrators, application developers, and organizations looking to determine the most suitable web server for their infrastructure needs. In addition, this research can also be used as evaluation material in selecting a web server solution that is able to provide a balance between performance, stability, and efficient use of resources.

2. Methodology

2.1. Research Stages

The research was conducted through several systematically arranged stages to ensure that the implementation, testing, and analysis processes could be carried out in a structured manner. The research stages include:

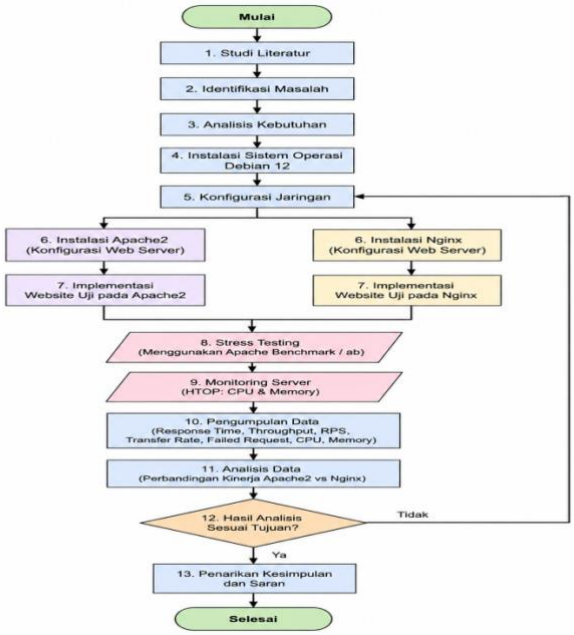

Figure 1: Flowchart of research stages.

This research was conducted through a series of interconnected stages, spanning from the preparatory phase to the formulation of the final conclusions. The process began with a literature review, involving the collection of references regarding the Apache2 and Nginx web servers, the Debian 12 operating system, and stress testing methodologies. These references—sourced from scientific journals, books, conference proceedings, and official documentation—served as the theoretical foundation and guided the selection of the research methodology.

Once a sufficient theoretical basis was established, the research problem was identified: comparing the performance of Apache2 and Nginx web servers in handling user requests under high-load conditions. Based on this identification, the research objectives and the parameters to be used for testing were defined.

The next stage was a requirements analysis, which aimed to determine the hardware, software, and network configuration requirements that would be used during the research. This analysis was conducted to ensure the testing environment had the appropriate specifications and optimal research results.

After all requirements were met, the Debian 12 operating system was installed on the server computer. This operating system was chosen because it is a Linux distribution with a high level of stability and is widely used as a server in production environments. Next, the network was configured by setting a static IP address, gateway, and network connectivity so that the server could be accessed by client computers via a Local Area Network (LAN).

The next stage was the installation and configuration of the Apache2 and Nginx web servers. The two web servers were not run simultaneously but were tested alternately using the same system configuration. After the installation process was complete, test websites were implemented on each web server using identical web pages. This ensured that differences in test results were influenced only by the capabilities of each web server, not by differences in website content.

Next, stress testing was conducted using the Apache Benchmark (ab) application. In this stage, the server is given a number of concurrent requests with varying numbers of users to determine the web server's ability to handle high access loads. During the testing process, server monitoring is performed using the htop application to observe CPU and memory usage on each web server.

All test results then proceed to the data collection stage. The collected data includes response time, throughput, requests per second (RPS), transfer rate, failed requests, CPU usage, and memory usage. This data is then analyzed by comparing the Apache2 and Nginx test results based on each predetermined parameter.

The next stage is data analysis, which evaluates all test results to determine which web server performs better on the Debian 12 operating system. Adjustments and retesting are required if the results align with the research objectives. However, if the analysis results align with the research objectives, the research proceeds to the conclusion and recommendation stage. At this stage, conclusions are drawn based on the analysis results and recommendations are provided that can serve as a reference for further research.

2.2. Stress Testing

Testing was carried out using the Apache Benchmark (ab) application with several load scenarios as follows:

Table 1: Apache

Request	Number of Requests	Concurrency
Light	1000	50
Medium	5000	100
Heavy	10000	200

2.3. Testing Parameters

Table 2: Testing Parameter

Parameter	Description
Requests per Second	The number of requests the server can process per second
Time per Request	The average time it takes the server to process one request
Transfer Rate	The data transfer rate during the testing process
Failed Requests	The number of requests that failed to be processed
CPU Usage	The percentage of processor usage during the testing process
Memory Usage	Memory usage during the stress testing process

3. Results and Discussion

3.1. Web server installation

The implementation stage was performed on a Debian 12 operating system running using Oracle VirtualBox. The testing environment was configured by installing two web servers, namely Apache2 and Nginx. Both web servers are run alternately to avoid HTTP port usage conflicts (port 80). Additionally, an installation of Apache Benchmark (AB) was carried out as a tool to perform stress testing as well as HTOP as a system resource usage monitoring application during the ongoing testing process.
The Apache2 deployment is initiated by ensuring that the server web service is successfully run using the apache2 status systemctl command. The testing results show that the Apache2 service is in active (running) status so that it is ready to be used as a test object.

Figure 2: Apache2 implementation

Next performed implementation of Nginx web server on Debian operating system. Once the installation process was completed, the Nginx service was run and ensured to work successfully using the HTTP port so that it could be used as a comparator web server in this study.

Figure 3: Nginx Installation

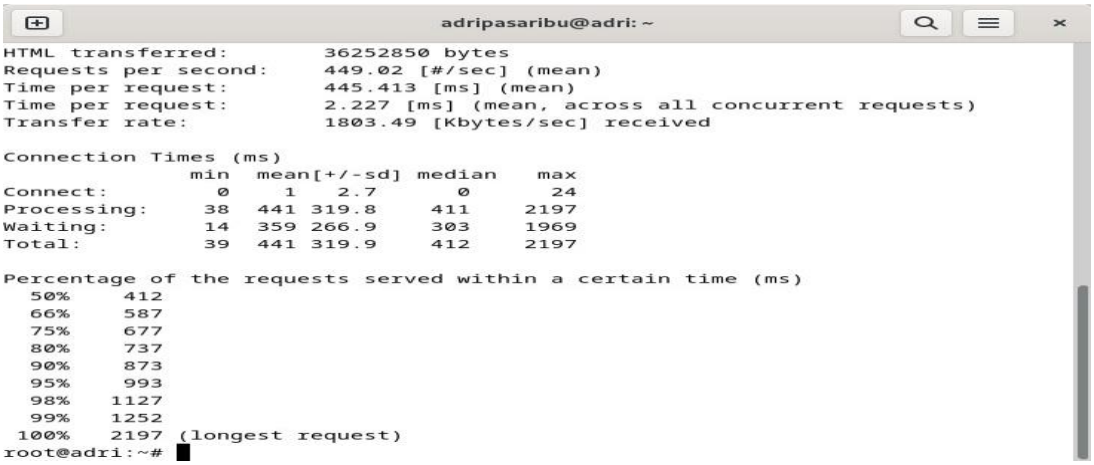

Figure 4: Nginx Implementation

3.2. Nginx Web Server Testing Results

The web performance testing of the Nginx server was conducted using Apache Benchmark (AB) with multiple testing scenarios, namely 1000 requests, 5000 requests, and 10000 requests. The purpose of this testing is to know the capability of Nginx in handling a number of requests simultaneously.

The first testing results were conducted using 1000 requests with a concurrency level of 50. Based on the benchmark results a Requests per Second value of 2772.29 requests/second was obtained, the average response time (Time per Request) was 18.036 ms, and the data transfer rate was 18.036 ms. 29609.87 KB/s. The entire request was successfully processed without a failure occurring (Failed Request = 0).

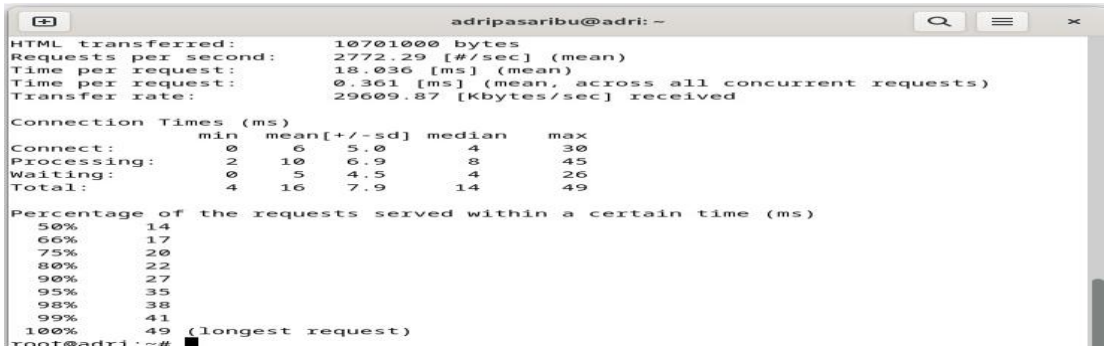

Figure 5: Nginx testing results

On the next testing the number of requests is increased to 5000 with a concurrency level of 100. The testing results show that the Nginx web server is able to maintain its performance with a Requests per Second value as high as 3970.94 requests/second, an average response time as ms0, a transfer as high as 566 42412.32 KB/s. The results show that the increase in the number of requests can still be handled well by Nginx without occurring a service failure.

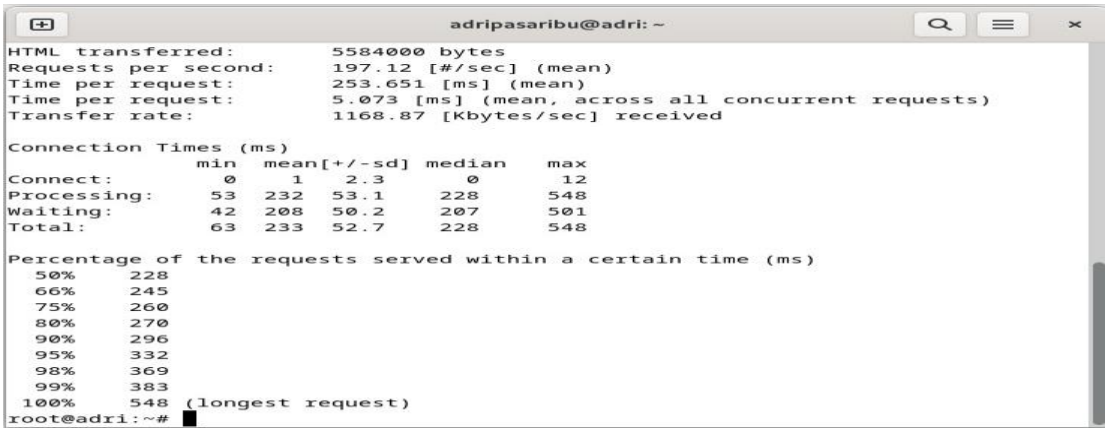

Figure 6: Nginx testing results

In general the testing results show that Nginx is capable of providing high performance on every testing scenario. This is influenced by the event-driven architecture that Nginx possesses so it is capable of handling many connections simultaneously with relatively efficient use of resources.

3.3. Apache2 Server Web Testing Results

Once the whole Nginx testing is done, testing continues on the Apache2 web server using the same testing scenario. The goal is to obtain objective comparative data on the performance of both web servers.
The first testing obtained a Requests per Second value of 197.12 requests/second, an average response time of 253.651 ms, and a transfer rate of 1168.87 KB/s

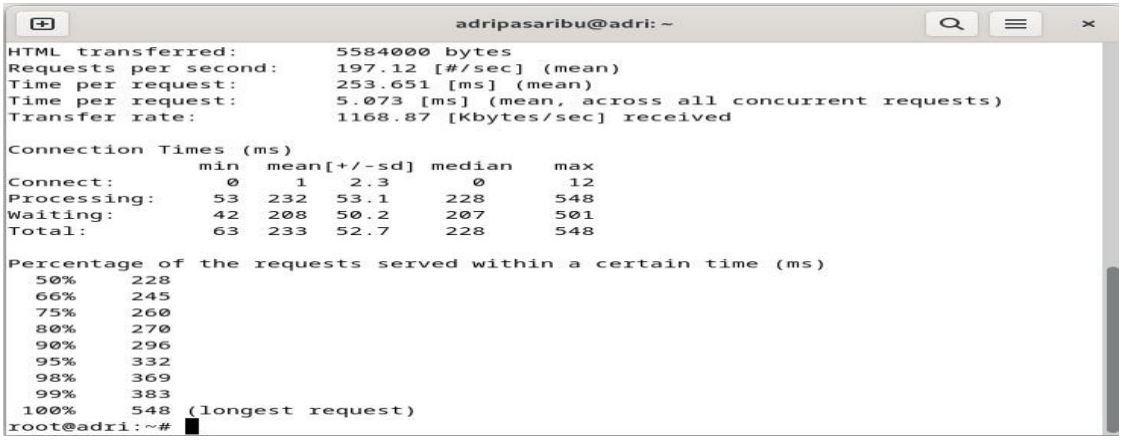

Figure 7: Nginx testing final results

Selanjutnya dilakukan pengujian menggunakan 5000 request dengan concurrency level 100. Berdasarkan hasil benchmark diperoleh nilai Requests per Second sebesar 283,42 request/detik, waktu respon rata-rata sebesar 352,838 ms, dan transfer rate sebesar 1642,75 KB/s.
Pengujian terakhir dilakukan menggunakan 10000 request dengan concurrency level 200. Hasil benchmark menunjukkan nilai Requests per Second sebesar 449,02 request/detik, waktu respon rata-rata sebesar 445,413 ms, dan transfer rate sebesar 1803,49 KB/s.
Berdasarkan hasil tersebut dapat diketahui bahwa Apache2 mampu menyelesaikan seluruh request tanpa mengalami kegagalan. Akan tetapi, nilai throughput yang diperoleh masih berada di bawah performa Nginx sehingga waktu respon yang dihasilkan menjadi lebih besar.

3.4. Monitoring System Resource Usage

In addition to conducting performance testing using Apache Benchmark, this study also monitored system resource usage using the HTOP application. This monitoring was performed concurrently with the benchmark process, allowing for real-time tracking of CPU and memory usage.

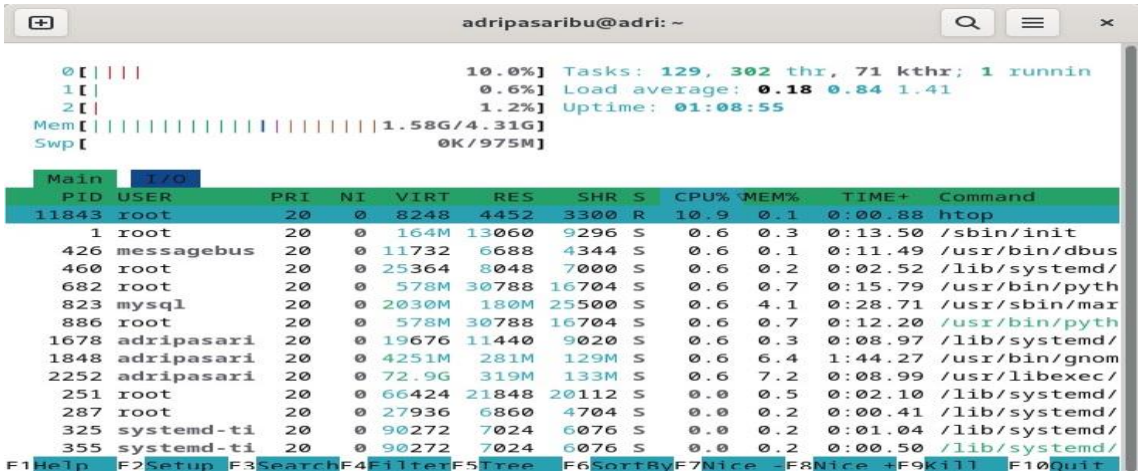

Figure 8: Monitoring using htop.

Monitoring results indicate that system memory usage reached approximately 1.82 GB out of a total of 4.31 GB, while CPU usage increased during the stress testing process. This demonstrates that the web server utilized system resources in accordance with the applied load. This monitoring serves as an indicator for assessing resource usage efficiency during the testing process.

3.4.1. Analysis of Test Results

The performance difference stems from the distinct architectures of the two web servers. Nginx employs an event-driven, asynchronous approach, enabling it to handle numerous simultaneous connections with greater resource efficiency. In contrast, Apache2 utilizes a process- or thread-based approach; consequently, each request requires an additional process or thread, leading to increased resource consumption as the number of connections rises.Monitoring results obtained via HTOP further indicate that CPU and memory usage remained stable throughout the testing period. This demonstrates the Debian operating system's capability to effectively run both web servers during stress testing.Overall, it can be concluded that Nginx delivers superior performance compared to Apache2 in the testing environment used for this study, particularly regarding throughput, response time, and system resource efficiency.

4. Conclusion

Based on the implementation and testing results, it can be concluded that both Apache2 and Nginx operate successfully on the Debian Linux operating system. Both were able to handle all requests without failure throughout the testing process.Stress testing results indicate that Nginx outperforms Apache2. Nginx is capable of handling higher request volumes with faster response times and more efficient system resource utilization. Although CPU and memory usage increased during testing, the system remained stable, allowing the testing process to proceed smoothly.Given these results, Nginx is recommended for servers handling high data traffic due to its superior performance. For future research, testing could be conducted with a larger number of users or by employing alternative testing methods to obtain more comprehensive and accurate results.

Acknowledgement

We extend our gratitude to everyone who provided support throughout this research process. Although this study did not receive funding from any specific sponsors or institutions, the various forms of support provided were invaluable in ensuring its successful completion. We hope that the findings of this research will prove beneficial and serve as a reference for future studies.

References

[1] Januhandini, R., & Oktiawati, U. Y. (2025). *Analisis Kinerja Web Server Apache, Nginx, dan Caddy dengan Metode Stress Testing Menggunakan Autocannon*. Journal of Internet and Software Engineering, 6(2), 96–100.

[2] Khamidah, Y. N., dkk. (2024). *Pengaruh Client Request Pada Web Server Apache dan Nginx*. Techno, 25(1)

[3] Faisal, F., dkk. (2025). *Analisis Kinerja Web Server Apache, Nginx, dan Caddy*. Journal of Informatics and Computer Science. [4] Azi, M., Arifwidodo, B., & Wahyudi, E. (2023). Analisis Performansi Web Server Saat Menangani Permintaan Client Menggunakan Metode Reserve Proxy Caching dan Varnish. *Journal of Telecommunication Electronics and Control Engineering (JTECE)*, *5*(1), 14-21. https://doi.org/10.20895/jtece.v5i1.843

[4] Herman, A. M., Wicoksono, W., Panchadria, P. A., & Laela, N. L. B. (2025). *Performance Comparison of NGINX, Apache, and Lighttpd Using WRK on a Debian*. Bit-Tech, 8(1).

[5] Shao, P., Huang, L., Weng, L., & Liu, Z. (2024). Technical Support System for High Concurrent Power Trading Platforms Based on Microservice Load Balancing. Processes, 12(6). [6] R. Januhandini and U. Y. Oktiawati, "Analisis Kinerja Web Server Apache, Nginx, dan Caddy dengan Metode Stress Testing Menggunakan Autocannon", *JISE*, vol. 6, no. 2, pp. 96 - 100, Nov. 2025

[6] irwan, I. I., Avrillaila Akbar Harahap, Dadang Heksaputra, & Tri Rochmadi. (2026). Perancangan Sistem Informasi Pelayanan Servis Elektronik Berbasis Website pada Sahabat Phone Metode Prototype. *Jurnal SINTA: Sistem Informasi Dan Teknologi Komputasi*, *3*(1), 61–70. https://doi.org/10.61124/sinta.v3i1.161

[7] Azi, M., Arifwidodo, B., & Wahyudi, E. (2023). Analisis Performansi Web Server Saat Menangani Permintaan Client Menggunakan Metode Reserve Proxy Caching dan Varnish. *Journal of Telecommunication Electronics and Control Engineering (JTECE)*, *5*(1), 14-21. https://doi.org/10.20895/jtece.v5i1.843

[8] Azis, A., Sugiarti, Y., Kumaladewi, N., & Huda, M. Q. (2019). Designing and Building an Information System of Career Development and Alumni Based on Android (Case Study: Information Systems Department, Syarif Hidayatullah State Islamic University Jakarta). 2018 6th International Conference on Cyber and IT Service Management, CITSM 2018, Citsm, 15

[9] A. Annisa, M. Ardiana, P. N. Rahayu, T. Brian, and M. A. Jami'in, "Performance Evaluation of Web Applications Using JMeter Load Testing for Server Capacity and Response Efficiency", *JAIC*, vol. 10, no. 1, pp. 585–590, Feb. 2026.

[10] W. Wicoksono, Herman Azzra Mustaqhim, Panchadria Pramana Anwas, and Laela Neng Linda Badratul, "Performance Comparison of NGINX, Apache, and Lighttpd Using WRK on a Debian", *bit-Tech*, vol. 8, no. 1, pp. 670–680, Aug. 2025.

[11] R. Januhandini and U. Y. Oktiawati, "Analisis Kinerja Web Server Apache, Nginx, dan Caddy dengan Metode Stress Testing Menggunakan Autocannon", *JISE*, vol. 6, no. 2, pp. 96 - 100, Nov. 2025

[12] Nur Hidayat, A. M., Rizaldy, A., Hartono, N., & Harwalis. (2024). Pengujian Kinerja Web Server Elastic Cloud Compute (EC2) Free Tier pada Amazon Web Service Menggunakan JMeter. Jurnal Sistem Informasi dan Informatika, 7(1).

[13] Perdana, G. P., dkk. (2025). Analisis Perbandingan Performa Web Server Apache, Nginx, OpenLiteSpeed, dan Caddy. JINACS.

[14] Nur Hidayat, A. M., Rizaldy, A., Hartono, N., & Harwalis. (2024). Pengujian Kinerja Web Server Elastic Cloud Compute (EC2) Free Tier pada Amazon Web Service Menggunakan JMeter. Jurnal Sistem Informasi dan Informatika, 7(1).

[15] Herman et al. (2025), yang menggunakan konfigurasi lingkungan pengujian yang seragam untuk membandingkan performa web server secara objektif.