

# Analysis of Nginx Web Server Performance Using IPv6 with Load Balancing Method Based on Weighted Round Robin Algorithm Scheduling

Bongga Arifwidodo <sup>#1</sup>

<sup>1</sup>Telecommunication Engineering, Telkom University Purwokerto

\*Jl. DI. Panjaitan 128, Purwokerto, 53147, Central Java, Indonesia

[bonggaa@telkomuniversity.ac.id](mailto:bonggaa@telkomuniversity.ac.id)

Received on 18-07-2024, revised on 17-05-2025, accepted on 23-05-2025

## Abstract

The need for the internet affects the growth in the number of website visitors and increases the server's traffic load. The increasing number of visitors often causes the website to overload due to an excessive number of requests despite the website still using a single server. So, it is necessary to apply Load Balancing techniques. The implementation requires an algorithm, specifically the Load Balancing method, which is responsible for dividing traffic as a workload among multiple servers. This research utilizes the Weighted Round Robin (WRR) algorithm, which considers server load based on device specifications. The scenario tests optimal performance load sharing among the WRR 1:1:1, WRR 2:1:1, and WRR 3:1:1 configuration then measures Response time and CPU Utilization. Testing is performed 30 times in each test scenario, and then the average value is taken. Giving traffic loads of 1000, 2000, and 3000 Requests using H2load Benchmark. The results of the WRR 2:1:1 ratio show that it is the most optimal, as the Load is evenly distributed among the three web servers. Reading the average CPU usage for 1000-3000 Request traffic, it reaches 71%-79% on Server 1, 47%-56% on Server 2, and 48%-56% on Server 3. Then, the average Response time is 223.77ms at 1000 Requests, 233.13ms at 2000 Requests, and 235.37ms at 3000 Requests.

**Keywords:** Weighted Round Robin, Webserver, Load Balancing, IPv6

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



---

## Corresponding Author:

\*Bongga Arifwidodo

Telecommunication Engineering, Telkom University Purwokerto, Indonesia

Jl. DI. Panjaitan 128, Purwokerto, 53147, Central Java, Indonesia

Email: [bonggaa@telkomuniversity.ac.id](mailto:bonggaa@telkomuniversity.ac.id)

---

## I. INTRODUCTION

The results of the Indonesian Internet Service Providers Association (APJII) Survey announced that the number of internet users in Indonesia until the second quarter of 2020 rose to 73.7% of the population or the equivalent of 196.7 million users. Almost 200 million users from the R1 population of 266.9 million according to data from the Central Statistics Agency (BPS) [1]. The increasing amount of internet traffic and the development of the internet affect the traffic load, because the server is overloaded (congestion). Server overload causes the server to go down because it cannot fulfil requests [2]. As more and more people access web pages, it creates increased traffic loads on service providers or web servers. This becomes less than optimal. This condition requires Load-Balancing techniques to help manage the numerous requests that arise. Single-server configuration conditions result in failures caused by factors such as the level of Request traffic reaching thousands to millions at the same time (concurrent access) or overload. The impact of this can be detrimental to those who place their sites on a web server, as the site may not be accessible for a specific period. Many requests from users at the same time can cause a significant increase in the server's performance load, leading to server downtime. One approach to addressing the problem is the load-balancing method. Many algorithm options are available; this research

utilizes load-balancing techniques that can distribute traffic loads evenly across two or more connection lines, ensuring the server runs stably and optimally.

Over time, technological advancements have had a significant impact on load-balancing algorithms. There is an applied algorithm called the Weighted Round Robin algorithm, which is a variation of the Round Robin algorithm. It is a development of the Round Robin algorithm, with the concept of dividing higher traffic loads into servers or clusters that have larger resources. This means that it can calculate the difference in processing capability of each cluster member server. An administrator manually enters the load parameters processed by each server member in the cluster, and then the scheduling sequence is automatically performed based on the server load. Furthermore, requests are then directed to different servers [3]. The bigger the resource, the bigger the Load. With this scheduling algorithm, in addition to coping with the Load and increasing high availability, it can also minimize the response time when requesting the website. The Weighted Round Robin scheduling algorithm is one of the scheduling algorithms that allow the server workload to run in balance by assigning a weight to each cluster node [15]. This study aims to implement the Load Balancing method, which is responsible for analyzing and dividing traffic as a workload among several servers. This study also uses the Weighted Round Robin (WRR) algorithm, which calculates server load based on device specifications.

## II. RESEARCH METHOD

### A. Basic Theory

#### 1. Load Balancing

Load balancing is a computer network technique that distributes incoming requests to multiple computers or groups of computers to optimize the use of available resources. Load Balancing has a multifunctional role, including load balancing, traffic management, and traffic path diversion. Other functions can also serve as health measurement parameters on a server, both in terms of application and content, to improve available services and ensure ease of management [4].

#### 2. Weighted Round Robin

Load Balancing based on the weighted round-robin scheduling algorithm is an algorithm that shares the same working characteristics as the Round Robin algorithm. However, pay attention to a new condition that will be applied to the server, which will receive requests or workloads from the Load Balancer. By considering the resource capabilities of the server to be used and giving a larger number of tasks to servers that have higher resource capacity [5].

#### 3. Web server

A web server is a software that provides data services, functioning to receive HTTP or HTTPS requests from clients, such as Web browsers, and send back the results in the form of Web pages, which are generally in the form of HTML documents [6].

#### 4. IPV6

The Internet Protocol version 6 (IPv6) protocol is a type of network addressing used in the TCP/IP network protocol, based on the Internet Protocol version 6. It has a total length of 128-bits. The 128-bit address will be divided into 8 16-bit blocks, which can be converted into a 4-digit hexadecimal number each. Each block of hexadecimal numbers will be separated by a colon (::). Generally, the IPv6 format is often referred to as a colon-hexadecimal format, in contrast to IPv4, which uses a dotted-decimal format [7].

#### 5. Nginx

It is software that initially only functioned as a web serving. Nginx is now famous for its advanced features that enhance the display of web pages with improved performance. The features of Nginx are now able to process server activities such as proxy (POP3, SMTP, and IMAP), HTTP Chace, and Load Balancing [8].

#### 6. H2load Benchmark

Web server testing tool that supports HTTP/2 and HTTP/1.1 protocols and with support for SSL/TLS. H2load uses non-blocking I/O to process concurrent calls to the target GET/POST HTTP endpoint. To prevent excessive load on the running system, as a single thread can generate thousands of requests per second [9].

## 7. Oracle VM VirtualBox

Oracle VM VirtualBox is an open-source virtualization application. Virtualization means creating virtual machines that can run independently on top of the main operating system. All forms of hardware related to virtual machines are all simulated by the host pc. So that all hardware resources cannot exceed the original resources [10].

## 8. HTOP

HTOP is an interactive program for monitoring and managing system processes. It is an alternative to the top program on Unix. HTOP can display a frequently updated list of tasking processes on the computer. The sorting is usually based on CPU usage. Unlike TOP, HTOP provides a complete list of all running program processes, rather than just resource-consuming processes. HTOP utilizes colors to convey visual information, including processor, swaps, and memory status [11].

## 9. CPU Utilization

The resources used by a computer can be determined by examining the CPU Utilization parameter. The unit of measurement for CPU Utilization is calculated in percent (%) [12]. A brief spike in CPU utilization indicates that the virtual machine resources are being utilized optimally. However, if the CPU utilization value for a virtual machine exceeds 90% and the CPU ready value exceeds 20%, performance will be affected [13].

## 10. Response time

Time to First Byte (TTFB), also known as server response time, is the time it takes for a browser to receive the first byte of response information after making a request. [14]. Response time parameters describe the speed condition of a Web server when it can serve requests from a client. Units are calculated in millisecond (ms) units. The smaller the parameter value, the faster a Web server can serve Request traffic from a Client [12].

## B. Research Flow

In this research, several steps must be taken to obtain optimal results. This final project research is described in the following research flowchart:

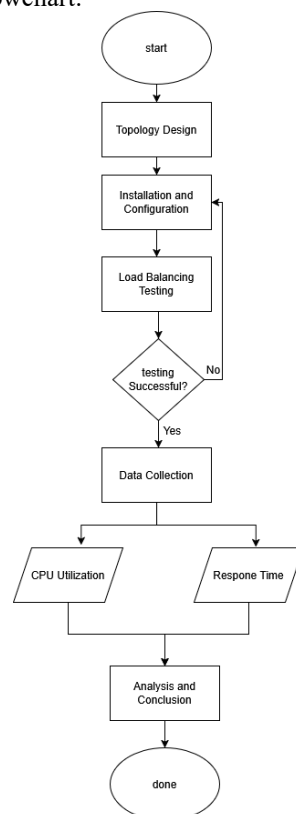


Fig. 1. Research Flowchart

Figure 1 shows the design flowchart in this research. Starting with the installation of Oracle VM VirtualBox on a laptop as a virtual machine container, which will be used to build the virtual machine. Furthermore, researchers create virtual machines using Debian 11 desktop virtual machines as clients, Debian 9 as three web servers with Nginx services, and one load balancer with Nginx services, where each virtual machine is installed separately. Furthermore, the Load Balancing configuration utilizes the Weighted Round Robin (WRR) algorithm, as it enables Load Balancing to distribute the Load according to the specifications on each Web server. Furthermore, the installation of the HTOP software tool on the three web servers is used, along with H2load Benchmark on the Client virtual machine.

HTOP is used as a CPU performance monitoring tool, which in this study is used to monitor CPU performance on the three web servers used. This allows researchers to see the value of CPU utilization through HTOP. While H2load Benchmark is a software tool that will be used in this study as a testing software or stress tool by generating traffic or providing traffic loads in the form of HTTP Requests to the Load Balancing system, H2load Benchmark will also display the value of Response time in each test.

The next stage is data collection and analysis of the data obtained from the tests that have been carried out by measuring CPU Utilization and Response time to find out which load ratio has the most optimal performance of the WRR 1: 1: 1, WRR 2: 1: 1 and WRR 3: 1: 1 test scenarios. Research conclusions can be drawn with consideration of the research objectives to obtain accurate results. Then, suggestions are made to encourage research on similar topics.

### C. Testing Topology

The testing topology used in this study consists of 1 Client, which is used to test by sending Requests directly through the browser and sending Requests to the Load Balancing IP address using H2load Benchmark. Load Balancing serves to divide the Load that will be distributed among three web servers, as shown in Figure 2.

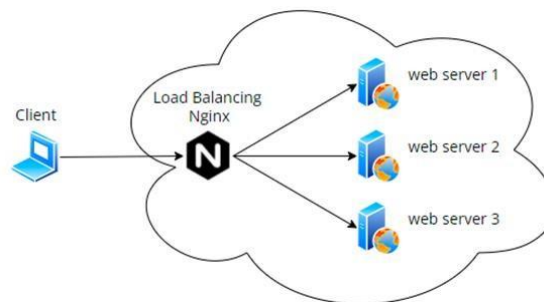


Fig. 2. Topology

TABLE I. IP ADDRESS

No.	Device	Ip Address
1	Load Balancing	2345::5/64
2	Web server 1	2345::2/64
3	Web server 2	2345::3/64
4	Web server 3	2345::4/64
5	Client	2345::9/64

### D. Load Balancer Configuration

In the Load Balancer configuration, the IP configuration is also carried out according to the information in Table I. Additionally, the Nginx service is installed as the Load Balancer service, and the algorithm used is the Weighted Round Robin algorithm. for configuration steps can be seen in the following steps:

- 1) Firstly, configure the IP on the Load Balancer using the :

```
#ip addr add [ip address] dev enp0s8
```

The IP address is customised as shown in TABLE I.

- 2) Secondly, install the NGINX web server using the :

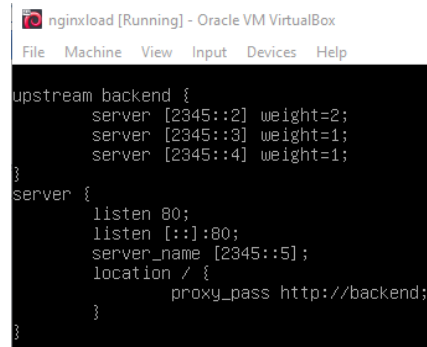
```
#Apt-get update
```

```
#Apt-get install Nginx -y
```

- 3) Then to make sure NGINX is installed and running properly, you can run the command :

```
#!/etc/init.d/Nginx status
```

- 4) Configure the default file located in the `/etc/Nginx/sites-available/` directory, using the :  
`#nano /etc/Nginx/sites-available/default.`
- 5) Next, restart after completing the configuration using the :  
`# /etc/init.d/Nginx restart`



```

upstream backend {
    server [2345::2] weight=2;
    server [2345::3] weight=1;
    server [2345::4] weight=1;
}
server {
    listen 80;
    listen [::]:80;
    server_name [2345::5];
    location / {
        proxy_pass http://backend;
    }
}

```

Fig. 3. Load Balancer Configuration Script

### E. Test Scenario

Researchers have configured all Virtual machines, including Load Balancing configuration, three web servers, and one Client. Load Balancing acts as a load divider for the 3 Web servers used. Testing in this study will be carried out by directly accessing the Load Balancing IP address through the Client browser 30 times and testing by sending traffic loads or HTTP Requests to Load Balancing using H2load Benchmark which will be carried out with 3 Weighted round robin (WRR) load ratio scenarios, namely WRR 1: 1: 1, WRR 2: 1: 1 and WRR 3: 1: 1. can be seen in TABLE II.

Load Ratio	Web server 1	Web server 2	Web server 3
WRR 1:1:1	1	1	1
WRR 2:1:1	2	1	1
WRR 3:1:1	3	1	1

Due to the largest specifications used, namely on Web server 1. To ensure that the provision of load ratios is also adjusted, giving greater weight to Web server 1. The WRR algorithm with a load ratio of 1:1:1 means that the Load Balancer will distribute one Request load to Web Server 1, Web Server 2, and Web Server 3. Furthermore, the 2:1:1 load ratio scenario means that the Load Balancer will assign two Request loads to Web Server 1, one Request to Web Server 2, and one request to Web Server 3. For the 3:1:1 load ratio scenario, it means that the Load Balancer will distribute a load of three Requests to Web Server 1 and one Request to Web Server 2 and Web Server 3. Testing HTTP requests to Load-Balancing IP addresses using the H2load Benchmark tool will also be carried out with three test scenarios based on the number of Requests sent, as shown in Table III.

Test n	Number of	Concurrent	Number of Tests
1	1000	100	30
2	2000	100	30
3	3000	100	30

Testing aims to determine the value of *CPU Utilization and Response time in each WRR load ratio scenario*, as shown in Table III. Each test is given 100 concurrent clients, where concurrent itself refers to the number of client requests made simultaneously. In the -1 test carried out with 1000 requests, 30 tests are conducted, and the average value is taken. The 2nd test is carried out with 2000 requests, and 30 tests will be conducted to determinethe average value. The 3rd test is performed with 3000 requests, repeated 30 times, and the average value will be taken. Testing is done through the *Client* using *H2load Benchmark*.

### III.RESULTS AND DISCUSSION

#### A. Browser Testing Results.

Testing is performed by sending a request through the Client browser 30 times per test, typing the IPv6 Load Balancer address. The Load Balancer will automatically distribute traffic according to the predetermined Weighted Round-Robin (WRR) ratio, namely WRR 1:1:1, WRR 2:1:1, and WRR 3:1:1.

##### 1. Browser Testing Results WRR 1:1:1

Testing is conducted to determine whether the Load Balancing system can distribute Request traffic according to the WRR 3:1:1 load ratio. By testing directly through the Client browser, as shown in Table IV. Where the yellow-colored column indicates the Web server that appears in the test.

TABLE IV. BROWSER TESTING RESULTS WRR 1:1:1

Testing(n)	Web server 1	Web server 2	Web server 3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
Total	10	10	10

From the test table in TABLE IV. it can be seen that the results of the 1st test to the 30th test are in accordance with the WRR 1:1:1 comparison ratio, which is 10 times for Web server 1, Web server 2, and Web server 3, which shows that the Load Balancer has been able to distribute traffic requests in accordance with the WRR 1:1:1 load ratio.

##### 2. Browser Testing Results WRR 2:1:1

Testing is conducted to determine whether the Load Balancing system can distribute Request traffic according to the WRR load ratio of 2:1:1. By testing directly through the Client browser, it can be observed in Table V. Where the yellow-colored column indicates the Web server that appears in the test, from the test table in TABLE V. it can be seen that the results of the 1st test to the 30th test are in accordance with the WRR 2:1:1 load ratio, namely 15 times for Web server 1, 8 times for Web server 2 and 7 times for Web server 3, which shows that the Load Balancer can distribute traffic requests in accordance with the WRR 2:1:1 load ratio.

TABLE V. BROWSER TESTING RESULTS WRR 2:1:1

Testing(n)	Web server 1	Web server 2	Web server 3
1			
2			
3			

Testing(n)	Web server 1	Web server 2	Web server 3
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
Total	15	8	7

3. Browser Testing Results WRR 3:1:1

Testing is conducted to determine whether the Load Balancing system can distribute Request traffic according to the WRR 3:1:1 load ratio. By testing directly through the Client browser, as shown in Table VI. Where the yellow-colored column indicates the Web server that appears in the test. From the test results in TABLE VI. it can be seen that the results of the 1st test to the 30th test are in accordance with the WRR 3:1:1 load ratio, namely 18 times for Web Server 1, 6 times for Web Server 2 and Web Server 3, which shows that the Load Balancer can distribute traffic requests in accordance with the WRR 3:1:1 load ratio.

TABLE VI. BROWSER TESTING RESULTS WRR 3:1:1

Testing(n)	Web server 1	Web server 2	Web server 3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			

Testing(n)	Web server 1	Web server 2	Web server 3
29			
30			
Total	18	6	6

### B. Response time test results

Testing is performed by sending HTTP Request traffic to the Load Balancer using the H2load Benchmark, which is repeated 30 times for each test scenario determined, and the average value is taken. The test results are presented in Table VII.

Experiment	1000 Req	2000 Req	3000 Req
WRR 1:1:1	243.03ms	254.83ms	255.07ms
WRR 2:1:1	223.77ms	233.13ms	235.37ms
WRR 3:1:1	234.4ms	236.33ms	237.37ms

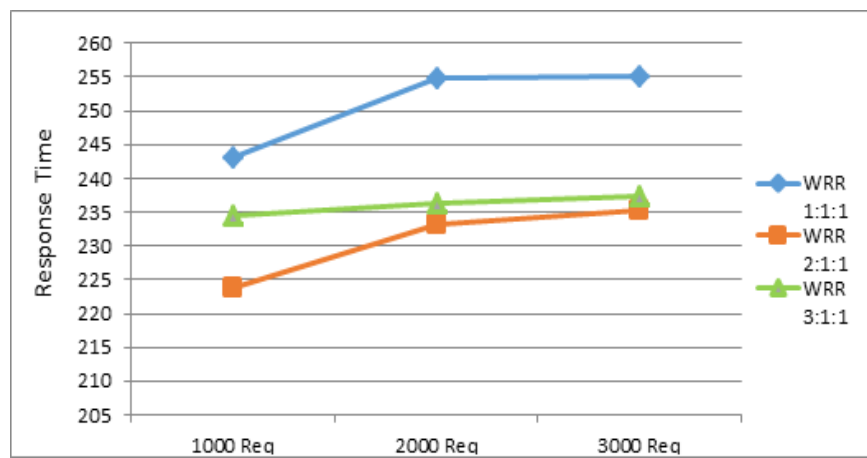


Fig. 4. Response time

At the number of connections or Request 1000, the average Response time values for the WRR 1:1:1 algorithm, WRR 2:1:1, and WRR 3:1:1 are 243.03 ms, 223.77 ms, and 234.4 ms, respectively. At the number of connections or Request 2000 in the WRR algorithm, both WRR 1:1:1, WRR 2:1:1, and WRR 3:1:1 experience an increase in the average Response time value as the number of traffic requests increases. At WRR 1:1:1 of 254.83 ms, WRR 2:1:1 of 233.13 ms, and at WRR 3:1:1 of 236.33 ms. Likewise, the number of connections or Request 3000 in the WRR algorithm, both WRR 1:1:1, WRR 2:1:1, and WRR 3:1:1, experienced an increase in the average Response time value as the number of traffic requests given increased. In WRR 1:1:1 of 255.07 ms, WRR 2:1:1 of 235.37 ms, and WRR 3:1:1 of 237.37 ms. From the measurement results of the average Response time value, it can be seen that the WRR 2:1:1 algorithm has the lowest Response time value compared to WRR 1:1:1 and WRR 3:1:1 from each measurement based on the number of requests given, so it can be concluded that Load Balancing using the Weighted round robin algorithm with a load ratio of WRR 2:1:1 according to the server specifications in this study, can serve connections or requests from clients faster than the load ratios WRR 1:1:1 and WRR 3:1:1.

### C. CPU Utilization Testing Results

CPU Utilization data is collected by monitoring the CPU of each Web server through the HTOP software, with up to 30 data points used and the average value calculated. The test results are shown in Fig. 5.



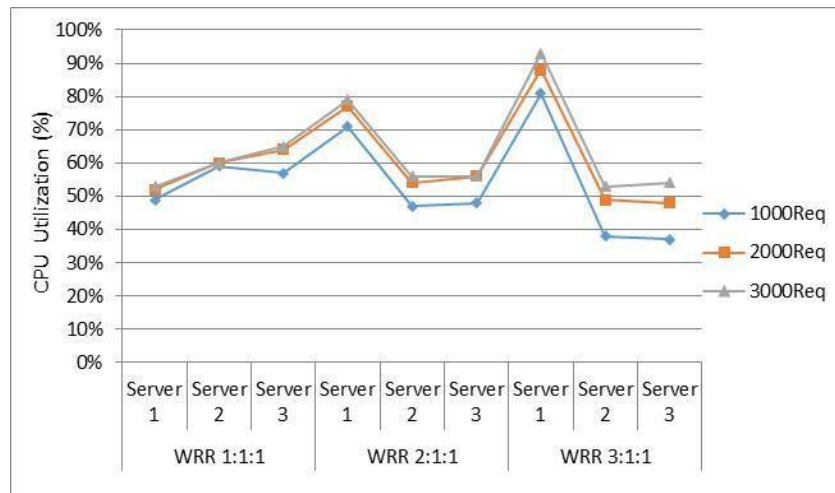


Fig. 5. Fig 5. CPU Utilization

Figure 5. shows that as the number of requests increases, the CPU performance process also increases to handle incoming requests. Furthermore, the higher the WRR load ratio, the greater the difference in CPU usage among the three Web servers. In the condition where the highest number of requests is given, namely 3000, each server in each load ratio scenario can still receive requests with a CPU usage value of 53% to 93%. From this exposure, the server can still work optimally when receiving between 1000 and 3000 Requests.

To determine the most optimal load ratio, it is necessary to know the differences in specifications of each server used so that this knowledge can provide a more appropriate load ratio and result in the most optimal load distribution. Judging from the specifications and test results carried out on each server used in this study, the optimal load ratio is to use a 2:1:1 WRR load ratio. The WRR 2:1:1 load ratio enables the Load Balancing system to distribute the Load equally among the three servers despite their differing specifications, as demonstrated by the test results. The Response time for the WRR 2:1:1 load ratio has the lowest value compared to the WRR 1:1:1 and WRR 3:1:1 load ratio.

#### IV.CONCLUSION

Based on the results of the research that has been done, there are several conclusions that the use of Load Balancing Web server Nginx with IPv6 using the Weighted round robin algorithm shows that server load sharing with the WRR ratio scenario 2: 1: 1 is the most optimal because it can divide the Load evenly to the three Web servers used, with average CPU usage on 1000-3000 Request traffic reaching 71%-79% on Server 1, 47%-56% on Server 2 and 48%-56% on Server 3. Thus, resulting in an average Response time of 223.77 ms at 1000 Requests, 233.13 ms at 2000 Requests, and 235.37 ms at 3000 Requests. The Response time value is lower than the WRR 1:1:1 and WRR 3:1:1 ratio.

#### REFERENCES

- [1] B. APJII, "Apjii", Asosiasi Penyelenggara Jasa Internet Indonesia, vol. 74. p. 1, 2020, [Online]. Available: <https://apjii.or.id/content/read/104/503/BULETIN-APJII-EDISI-74--November-2020>.
- [2] C. Ma, and Y. Chi, "Evaluation Test and Improvement of Load Balancing Algorithms of Nginx", IEEE Access., vol. 10, 2022, doi: 10.1109/ACCESS.2022.3146422.
- [3] J. L. Ruoyu, L. Yunchun and L. Wen, n Integrated Load-balancing Scheduling Algorithm for Nginx-Based Web Application Clusters, vol. 1060. Journal of Physics: Conference Series, 2018.
- [4] J. T. Bezboruah and A. Bora, Some Aspects of Implementation of Web Services in Load Balancing Cluster-Based Web Server, vol. 10. International Journal of Information Retrieval Research (IJIRR), 2020.
- [5] S. Sree Priya and T. Rajendran, " Load balancing using improved weighted round robin algorithm in cloud computing environment ", International Journal of Cloud Computing, vol. 13, no. 05, 2024.
- [6] T. T. Kwan, R. E. McGrath and D. A. Reed, "NCSA's World Wide Web Server: Design and Performance", IEEE Computer, vol. 28, no. 11, pp. 68-74, November 1995.
- [7] ITBU, "Alamat IP versi 6", [https://p2k.itbu.ac.id/id3/3070-2950/Alamat-Ip-Versi-6\\_27376\\_itbu\\_ensiklopedia-dunia-q-itbu.html](https://p2k.itbu.ac.id/id3/3070-2950/Alamat-Ip-Versi-6_27376_itbu_ensiklopedia-dunia-q-itbu.html) (accessed Feb. 10, 2022).

- [8] Jamain, R. Y., Periyadi & Ismail, S. J. I., 2015. IMPLEMENTATION OF WEB APPLICATION SECURITY WITH WEB APPLICATION FIREWALL. E-Proceeding of Applied Science, Vol 1, p. 2191.
- [9] Upasana, "H2load for REST API Benchmarkmarking", 2020. <https://www.javacodemonk.com/H2load-for-rest-api-Benchmarkmarking-a04b11a3> (accessed Mar. 10, 2022).
- [10] N. Huda, "Apa itu VirtualBox?", jagongoding.com, 2020. <https://jagongoding.com/others/apa-itu-virtual-box/> (accessed Feb. 02, 2022).
- [11] Linuxsec, "HTOP - Monitoring Memory, CPU, and Running Process Usage in Linux", linuxsec.org, 2019. <https://www.linuxsec.org/2019/05/perintah-HTOP.html> (accessed Feb. 02, 2022).
- [12] I. R. Wijaya, R. Munadi, Hafidudin, " Analysis of Load Balancing Performance Using Dynamic Ratio Algorithm on Three Web Server Loads ", e-Proceeding Eng., vol. 6, no. 1, pp. 1–8, 2019.
- [13] VMware, "CPU (%)", 2019. <https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.monitoring.doc/GUID-FC93B6FD-DCA7-4513-A45E-660ECAC54817.html> (accessed Jun. 14, 2022).
- [14] Gtmetrix, "Lighthouse : Reduce initial server response time." <https://gtmetrix.com/reduce-initial-server-response-time.html> (accessed Mar. 10, 2022).
- [15] A. Hanafiah and R. Wandri, " Implementation of Load Balancing with Weighted Round Robin Scheduling Algorithm in Overcoming Web Server Load," IT J. Res. Dev., vol. 5, no. 2, pp. 226–233, 2021, doi: 10.25299/itjrd.2021.vol5(2).5795.