

PENGARUH CLIENT REQUEST PADA WEB SERVER APACHE DAN NGINX DENGAN IPV6 MENGGUNAKAN APACHE BENCHMARK

Yulina Nur Khamidah¹, Bongga Arifwidodo², Muntaqo Alfin Amanaf³

Program Studi D3 Teknik Telekomunikasi, Institut Teknologi Telkom Purwokerto^{1,3}

Program Studi S1 Teknik Telekomunikasi, Institut Teknologi Telkom Purwokerto²

Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto

Informasi Makalah

Dikirim, 27 Juni 2022

Direvisi, 11 Oktober 2023

Diterima, 3 Januari 2024

Kata Kunci:

Web Server

Apache

Nginx

Apache Benchmark

INTISARI

Hypertext Transfer Protocol (HTTP) mendominasi jalur data yang melalui jaringan internet sebesar lebih dari 75%. Survey netcraft web server, menunjukkan bahwa Web server berbasis open source yang sering digunakan adalah Apache dan Nginx. Semakin pesatnya perkembangan internet ketersediaan IPv4 kian terbatas, untuk mengatasi hal tersebut digunakan alokasi IPv6 pada web server karena memiliki jumlah alamat yang lebih banyak dan keamanan yang lebih baik. Pada penelitian ini pengujian kinerja web server antara Nginx dan Apache menggunakan IPv6 dengan parameter pengujiannya yaitu Time Taken for Tests, Request Per Second, Transfer Rate (Kb/s), Time per Request (ms), dan Memory Usage. Skenario pengujian dilakukan dengan memberikan client request dengan jumlah yang bervariasi, dengan jumlah permintaan dari 500 sampai 500000 pada Apache Bench. Hasil penelitian ini webserver Nginx lebih baik dibandingkan Apache, sebesar 4% lebih cepat menyelesaikan request client terhadap parameter pengujian dan penggunaan memori 20% lebih sedikit dari alokasi maksimal.

ABSTRACT

Hypertext Transfer Protocol (HTTP) dominates the data path through the internet network by more than 75%. The Netcraft web server survey shows that the open source web servers that are often used are Apache and Nginx. With the rapid development of the internet, the availability of IPv4 is increasingly limited, to overcome this, IPv6 allocation is used on web servers because it has a larger number of addresses and better security. In this research, web server performance testing between Nginx and Apache uses IPv6 with test parameters namely Time Taken for Tests, Request Per Second, Transfer Rate (Kb/s), Time per Request (ms), and Memory Usage. The test scenario is carried out by giving a varying number of client requests, with the number of requests ranging from 500 to 500,000 on Apache Bench. The results of this research are that the Nginx webserver is better than Apache, 4% faster in completing client requests for test parameters and using 20% less memory than the maximum allocation..

Korespondensi Penulis:

Bongga Arifwidodo

Program Studi S1 Teknik Telekomunikasi

Institut Teknologi Telkom Purwokerto

JL. D.I Panjaitan No.128 Purwokerto

Email: bongga@ittelkom-pwt.ac.id

1. PENDAHULUAN

Kemudahan dan kelebihan yang terdapat pada web menyebabkan pesatnya perkembangan web sendiri. Selama masih terhubung dengan internet, aplikasi web dapat diakses dikarenakan aplikasi web memiliki kelebihan yaitu bersifat daring. Selain itu aplikasi web juga dapat dijalankan hanya dengan *server* dan *client* [1]. *Server* berfungsi untuk menerima permintaan dari *client* dengan menampilkan beberapa permintaan yang diminta oleh *client*. Sebuah server membutuhkan adanya Web Server untuk mengakses web dan memenuhi permintaan *client*. Kualitas hubungan *client* dan *server* dipengaruhi dengan performansi *web server*. *Web server* berbasis *opensource* yang sering digunakan adalah *Apache* dan *Nginx* [2].

Contoh: Pengujian menggunakan web server *Apache* dan web server *Nginx* telah dilakukan untuk mengetahui kinerja web server yang lebih unggul [3].

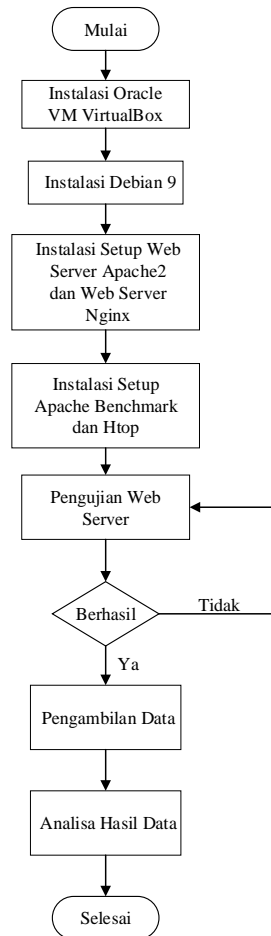
Penelitian sebelumnya melakukan pengujian dengan menggunakan web server *Apache* dan web server *Nginx* yang bertujuan untuk mengetahui kinerja web server manakah yang lebih unggul antara web server *Apache* dan web server *Nginx*. Hasil dari penelitian ini menyimpulkan bahwa web server *Apache* memiliki kinerja lebih unggul dibandingkan web server *Nginx* dalam kecepatan waktu [3]. Selanjutnya penelitian ini menggunakan web server *Apache*. Kinerja web server diuji dengan memberikan beban traffic yang tinggi ketika diakses. Pengujian dilakukan dengan menggunakan tools *httperf* dan *Apache Benchmark* untuk uji beban, dan *Independent Sample Test* digunakan sebagai teknik analisis terhadap dua data. Berdasarkan data yang didapatkan dari hasil pengujian dapat disimpulkan kinerja web server lebih baik dan mengalami peningkatan setelah dilakukan konfigurasi [4]. Penelitian dengan menggunakan dua *web server* yaitu *Apache* dan *Nginx* yang bertujuan untuk menentukan *web server* yang terbaik antara *web server Apache* dan *Nginx* dalam melayani *client request*. Hasil dari penelitian dari parameter *Time per Request* ini menyimpulkan bahwa performansi *web server Nginx* lebih baik dalam menyelesaikan *request client* dengan rata-rata waktu yang dibutuhkan lebih cepat dibandingkan *web server Apache* [5].

Penelitian menggunakan dua *web server* yaitu *Apache* dan *Nginx* yang bertujuan untuk menentukan *web server* yang terbaik antara *web server Apache* dan *Nginx* untuk aplikasi penjualan online, instalasi *web server* dipasang dalam sistem operasi *ubuntu* [6]. Hasil dari penelitian ini menyimpulkan bahwa performansi *web server Nginx* lebih baik dalam menyelesaikan *request client* dengan rata-rata waktu yang dibutuhkan lebih cepat dibandingkan *web server Apache*. Namun seiring penggunaan alamat IPv4 semakin pesat dengan pertumbuhan teknologi menyebabkan semakin terbatasnya ketersediaan alamat IPv4. Upaya mengantisipasinya dengan melakukan pengembangan standar baru yaitu IPv6 yang kedepannya dapat mengakomodasi pengalamatan IP dengan jumlah yang lebih banyak dibandingkan IPv4. Penerapan IPv6 untuk saat ini bisa diintegrasikan dengan IPv4 sehingga konfigurasi infrastruktur eksistingnya tidak berubah [7]. Pada penelitian ini menggunakan dua web server yaitu *Apache web server* dan *Nginx web server* dengan menggunakan pengalamatan IPv6. Pengujian performansi pada web server dengan parameter pengujiannya yaitu *Time Taken for Tests*, *Request Per Second*, *Transfer Rate (Kb/s)*, *Time per Request (ms)*, dan *Memory Usage*. Pengujian dengan *Apache Bench* dilakukan dengan memberikan client request dengan jumlah yang bervariasi, dengan jumlah permintaan koneksi dari 500 sampai 500000 request.

2. METODE

2.1. Alur Penelitian

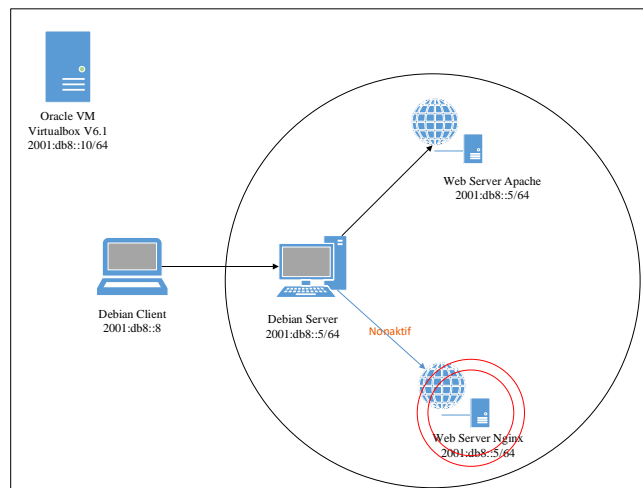
Penelitian ini dilakukan dengan beberapa tahapan dengan melakukan instalasi *Oracle VM Virtualbox* yang kemudian diinstal sistem operasi *Debian*, *Oracle VM Virtualbox* merupakan sebuah *software* virtualisasi yang fungsinya menjadi media pengoperasian sistem operasi tambahan atau tamu pada sistem operasi utama [8]. *Debian* merupakan sistem operasi komputer yang terdiri dari paket-paket perangkat lunak yang dirilis sebagai *software* yang bersifat *open source* [9], pada *debian* dipasang *setup web server Apache* dan *web server Nginx*, kemudian melakukan instalasi *Apache Benchmark* dan *htop* sebagai *software tool*. *Apache Benchmark* dirancang untuk menguji bagaimana kinerja instalasi *web server*, menunjukkan berapa banyak permintaan per detik yang dapat dilayani oleh instalasi *web server* [10]. *Htop* merupakan *tool* yang digunakan untuk memonitor program yang sedang berjalan dengan interaktif. *Htop* dapat menyampaikan informasi gambaran tentang prosesor, CPU dan memori yang digunakan [11]. Selanjutnya pengujian terhadap *web server Apache* dan *web server Nginx* dengan menggunakan *tool Apache Bench* dan *Htop*, selanjutnya yaitu pengambilan data hasil dari pengujian kedua *web server* tersebut. Kemudian dapat dianalisa dan disimpulkan dari hasil data yang didapatkan. Untuk lebih jelasnya seperti yang tertera pada gambar di bawah ini :



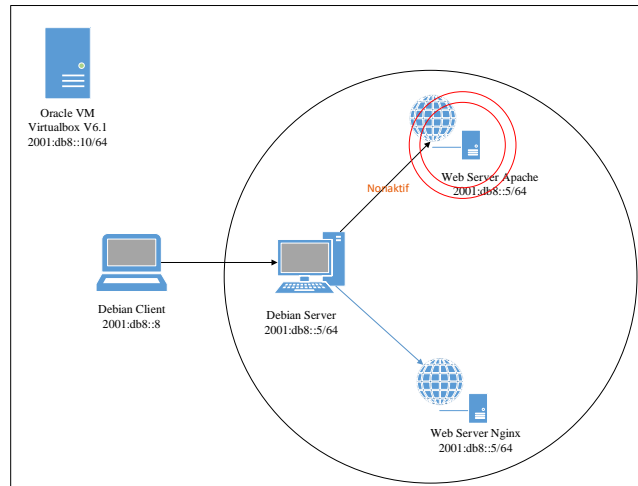
Gambar 1. Alur Penelitian

2.2. Topologi Pengujian

Untuk melakukan pengujian pada *web server Apache* pada sistem operasi *Debian 9*, maka *web server Nginx* dinonaktifkan terlebih dahulu karena kedua *web server* tidak dapat berjalan bersama-sama karena menggunakan *port* yang sama seperti yang ditunjukkan pada gambar 2.

Gambar 2. Topologi pengujian pada *web server Apache*

Untuk melakukan pengujian pada *web server Nginx* pada operasi sistem *Debian 9*, maka *web server Apache* dinonaktifkan terlebih dahulu karena kedua *web server* tidak dapat berjalan bersama-sama karena menggunakan *port* yang sama seperti yang ditunjukkan pada gambar 3.



Gambar 3. Topologi pengujian pada web server Nginx

2.3. Konfigurasi Web Server Apache dan Web Server Nginx

Pada penelitian ini dalam satu sistem operasi diberikan dua layanan *web server Apache* dan *web server Nginx*. *Web server Apache* bertanggung jawab dalam menangani *request-response* pada HTTP dan melakukan *logging* dengan secara rinci [12]. *Nginx* merupakan *web server* yang bersifat *open source* yang memiliki fitur sebagai *reserve proxy multiple protocols*. *Nginx* banyak digunakan karena stabil, kinerja yang baik dan konsumsi sumber daya yang rendah [13]. Untuk pengalamatan pada *web server* digunakan pengalamatan IPv6 dengan satu alamat IP untuk kedua *web server*. Untuk menginstall instalasi *web server Apache* pada *debian* dilakukan dengan *command* pada *root apt install apache2*, untuk menginstall instalasi *web server Nginx* pada *debian* dilakukan dengan *command* *root apt install nginx*.

Untuk menambahkan alamat IPv6 yang akan digunakan pada *Oracle VirtualBox* dengan menambahkan pada pengaturan *Network Connections* pilih *Adaptor* yang digunakan lalu pada IPv6 ditambahkan alamat IPv6 yang akan digunakan, penulis menggunakan alamat IPv6 2001:db8::10/64 dengan *gateway* 2001:db8::1. Untuk menambahkan alamat IPv6 yang akan digunakan pada *debian server* dilakukan dengan mengetikkan *command* *ip address add 2001:db8::5/64 dev enp0s8*. Gambar 4. Menunjukkan bahwa alamat IPv6 2001:db8::05/64 berhasil ditambahkan pada *devstack enp0s8*.

```

root@yulinanur:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:74:da:49 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe74:da49/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d0:54:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 2001:db8::5/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed0:54f5/64 scope link
        valid_lft forever preferred_lft forever

```

Gambar 4. Pengalamatan IPv6 pada *debian server*

2.4. Konfigurasi dan Pengujian Apache Benchmark

Konfigurasi *Apache Bench* pada *web server Apache* telah terkonfigurasi dan terinstal secara otomatis saat menginstal *Apache* dalam sistem operasi. Pengujian menggunakan *Apache Bench* dilakukan untuk memberikan koneksi yang sesuai dengan permintaan client ke web server. Untuk mengukur performansi *web server* dengan *Apache Bench* dapat digunakan perintah *ab -c 100 -n 5000 http://[Address]/*

Tabel 1. Parameter *Apache Benchmark*

Parameter	Fungsi
<i>ab</i>	Command untuk menggunakan <i>Apache Benchmark</i>
<i>c</i>	<i>Request concurrent</i> (bersamaan) dalam satu waktu
<i>n</i> Address	Koneksi yang dibuat ke <i>server</i> tujuan Alamat IP atau halaman yang ingin diproses web <i>server</i> pada <i>Apache Benchmark</i>

Sesuaikan panduan penulisan tabel

Tabel 2. Parameter pengujian

Pengujian ke-	Jumlah permintaan per detik	Jumlah koneksi
1	100	500
2	100	1000
3	100	5000
4	100	8000
5	100	10000
6	100	15000
7	100	20000
8	100	25000
9	100	30000
10	100	35000
11	100	45000
12	100	50000
13	100	65000
14	100	75000
15	100	85000
16	100	100000
17	100	150000
18	100	200000
19	100	350000
20	100	500000

Sesuaikan panduan penulisan tabel

```
ab -c 100 -n 1000 http://[2001:db8::5]/
```

Gambar 5. Pengujian *Apache Benchmark*

```
Server Software:      nginx/1.10.3
Server Hostname:     2001:db8::5
Server Port:         80

Document Path:       /
Document Length:     10701 bytes

Concurrency Level:    100
Time taken for tests: 1.041 seconds
Complete requests:    500
Failed requests:      0
Total transferred:   5468500 bytes
HTML transferred:    5350500 bytes
Requests per second: 480.00 [#]/sec (mean)
Time per request:    208.297 [ms] (mean)
Time per request:    2.083 [ms] (mean, across all concurrent requests)
Transfer rate:       5127.62 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0   19  21.1   11   98
Processing: 31  160  65.3  167  281
Waiting:    1  153  68.6  159  273
Total:      97  179  53.8  175  314

Percentage of the requests served within a certain time (ms)
 50%    175
 66%    189
 75%    214
```

Gambar 6. Hasil *Benchmarking web server*

2.5. Parameter Pengujian

Parameter performansi *web server* yang diukur sebagai berikut:

1. *Time Taken for Test*

Time taken for test merupakan durasi yang digunakan antara saat *apache bench* pertama kali terhubung ke *server* dan saat menerima *respons* akhir.

2. *Request per Second*

Request per Second merupakan jumlah permintaan layanan per satuan detik dari *client* yang dikirimkan ke *web server*.

3. *Transfer Rate*

Transfer Rate merupakan nilai dari transaksi perdetik yang didapatkan dari memproses permintaan *client*.

4. *Time per Request*

Time per Request merupakan jumlah waktu rata-rata yang diperlukan untuk memproses sejumlah permintaan secara bersamaan.

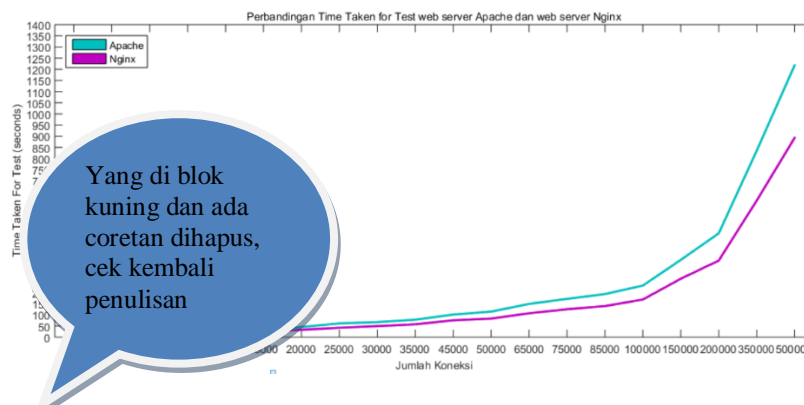
5. *Memory Usage*

Memory Usage merupakan jumlah penggunaan *memory server* pada saat *web server* sedang berjalan.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pengujian Parameter *Time Taken for Test*

Pengujian *Time Taken for Test* bertujuan untuk mengukur durasi waktu yang digunakan antara saat *apache bench* pertama kali terhubung ke *server* dan saat menerima *respons* akhir dalam satu kali percobaan.

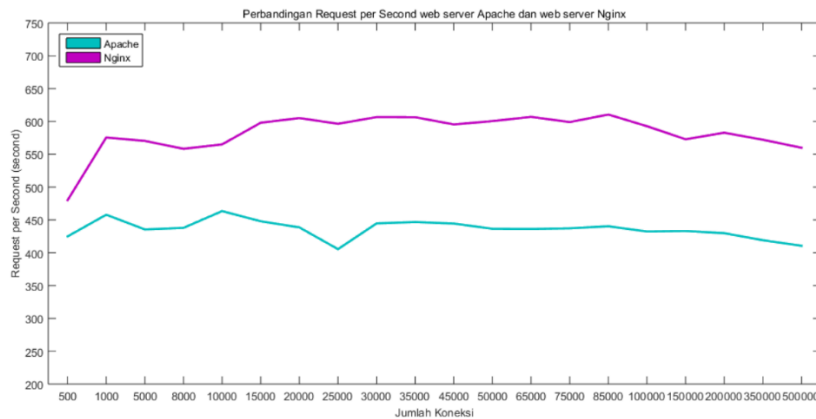


Gambar 7. Perbandingan *Time Taken for Test* web server Apache dan web server Nginx

Gambar 7 menunjukkan bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* maka semakin meningkat waktu yang dibutuhkan untuk menyelesaikan *request*. Hasil perbandingan pada grafik menunjukkan durasi waktu yang dibutuhkan *server* saat menyelesaikan semua *request* dengan menggunakan *web server Nginx* lebih cepat dibandingkan *web server Apache*.

3.2. Hasil Pengujian Parameter *Request per Second*

Pengujian *Request per Second* bertujuan untuk mengukur jumlah permintaan layanan yang ditangani per detik dari *client* yang dikirimkan ke *web server*.

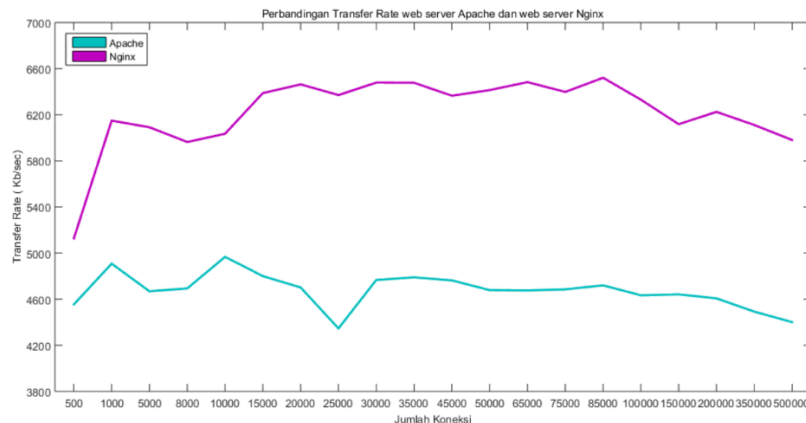


Gambar 8. Perbandingan *Request per Second* web server Apache dan web server Nginx

Gambar 8 menunjukkan bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* tidak berpengaruh pada peningkatan jumlah *request*nya. Pengujian *web server Apache* dengan 20000 koneksi lebih besar nilai *Request per Second*nya dibandingkan dengan 25000 koneksi. Hal ini bergantung dengan beban kerja atau kemampuan *web server*. Jika saat beban *server* meningkat maka banyaknya *request* yang dilayani dapat berkurang. Hasil perbandingan pada grafik menunjukkan bahwa *web server Nginx* mampu melayani *request* lebih banyak dibandingkan *web server Apache*.

3.3. Hasil Pengujian Parameter *Transfer Rate*

Pengujian *Transfer Rate* bertujuan untuk mengukur nilai dari transaksi perdetik yang didapatkan dari memproses permintaan *client*.

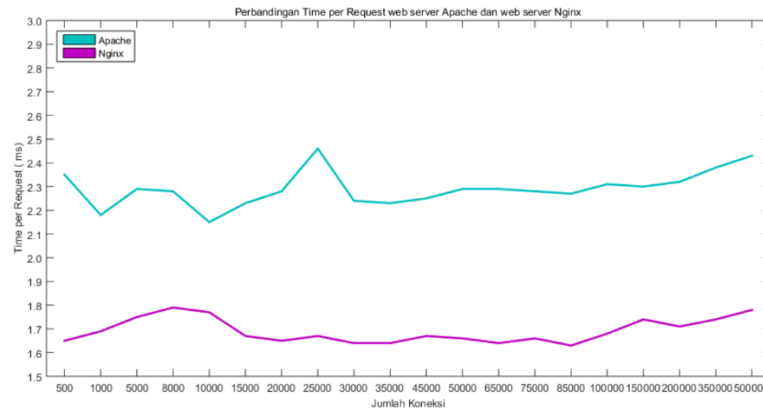


Gambar 9. Perbandingan *Transfer Rate* web server Apache dan web server Nginx

Gambar 9 menunjukkan bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* tidak berpengaruh pada peningkatan *Transfer Ratenya*. Pengujian pada *web server Apache* dengan 20000 koneksi lebih besar nilai *Transfer Ratenya* dibandingkan dengan 25000 koneksi. Hal ini bergantung dengan beban kerja atau kemampuan *web server*. Jika saat beban *server* meningkat maka kemampuan *web server* dalam melayani permintaan *client* berkurang perdetiknya. Hasil perbandingan pada grafik menunjukkan bahwa *web server Nginx* nilai *Transfer Ratenya* lebih tinggi dibandingkan *web server Apache*. Hal ini menunjukkan bahwa *web server Nginx* memiliki kecepatan lebih tinggi dalam melayani *request client*.

3.4. Hasil Pengujian Parameter *Time per Request*

Pengujian *Time per Request* bertujuan untuk mengukur jumlah waktu rata-rata yang diperlukan untuk memproses sejumlah permintaan secara bersamaan.

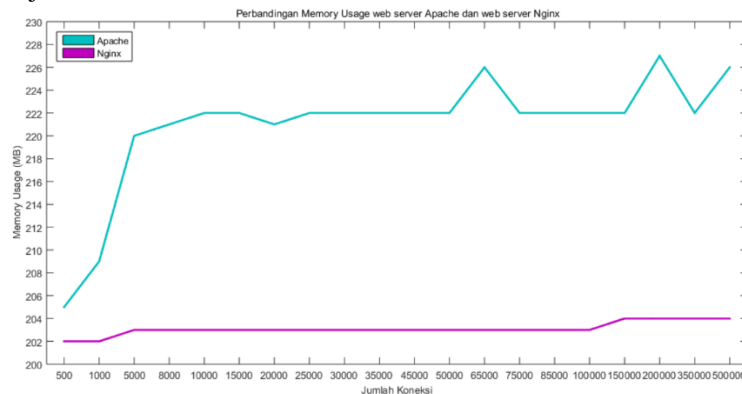


Gambar 10. Perbandingan *Time per Request* web server Apache dan web server Nginx

Gambar 10 menunjukkan pada *web server Apache* bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* maka nilai *Time per Request* nya semakin tinggi. Dan pada *web server Nginx* semakin banyak jumlah koneksi yang ditujukan ke *server* maka nilai *Time per Request* nya semakin rendah. Hal ini bergantung dengan beban kerja atau kemampuan *web server*. Hasil perbandingan pada grafik menunjukkan bahwa *web server Nginx* nilai *Transfer Ratenya* lebih rendah dibandingkan *web server Apache*. Hal ini menunjukkan bahwa *web server Nginx* memiliki kecepatan lebih tinggi dalam melayani *request client*.

3.5. Hasil Pengujian Parameter *Memory Usage*

Pengujian *Memory Usage* bertujuan untuk mengukur besarnya penggunaan *memory server* pada saat *web server* sedang berjalan.



Gambar 11. Perbandingan *Memory Usage* web server Apache dan web server Nginx

Perbandingan nilai *Memory Usage* yang ditunjukkan pada gambar 11 menunjukkan bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* maka semakin meningkat penggunaannya memorinya. Perbandingan nilai *Memory Usage* yang ditunjukkan pada gambar 4.10 dengan nilai *concurrent* tetap yaitu 100 dan jumlah koneksi 5000 sampai 50000 menunjukkan juga bahwa semakin banyak jumlah koneksi yang ditujukan ke *server* maka semakin meningkat penggunaannya memorinya. Hasil perbandingan pada grafik menunjukkan bahwa penggunaan memori pada *web server Nginx* lebih kecil penggunaannya dibandingkan *web server Apache*.

4. KESIMPULAN

Hasil penelitian ini dengan melakukan pengujian kinerja web server antara Nginx dan Apache menggunakan IPv6, Pengujian dengan Apache Bench dilakukan dengan memberikan client request dengan jumlah yang bervariasi, dengan jumlah permintaan dari 500 sampai 500000 request. Hasil penelitian ini dengan pengujian Apache Benchmark terhadap variasi client request bahwa Nginx lebih unggul dibandingkan Apache, dalam sisi waktu 4 detik lebih cepat menyelesaikan request client dan penggunaan memori lebih sedikit, hanya 20% dari alokasi maksimal.

DAFTAR PUSTAKA

- [1] I. Cartealy, "Linux Networking : Ubuntu, Kubuntu, Debian," in *Jaringan Komputer,linux*, Jasakom, 2013.
- [2] A. Aziz and T. Tampati, "Analisis Web Server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web Server Apache dengan Nginx," *JURNAL MULTINETICS*, vol. 1, no. 2, pp. 12-20, 2015.
- [3] E. Khoif, "ANALISA IMPLEMENTASI LOAD BALANCING ROUND ROBIN DAN LEAST CONNECTION PADA WEB SERVER (STUDI KASUS PT UCC)," *JTIKA*, vol. 3, no. 2, pp. 244-252, 2021.
- [4] D. P. Ardian and Hariyanto, "Performansi Web Server Apache dan Nginx Pada Aplikasi penjualan Online," *Indonesian Journal on Networking and Security*, vol. 9, no. 3, pp. 1-6, 2020.
- [5] I. F. Irza, Z. Zuhendra and E. Efrizon, "Analisis Perbandingan Kinerja Web Server Apache dan Nginx Menggunakan Httperf Pada Portal Berita (Studi Kasus beritalinux.com)," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 5, no. 2, pp. 75-82, 2017.
- [6] K. S. S. I and N. S. Ketut, "PERBANDINGAN PERFORMANSI WEB SERVER APACHE DAN NGINX DENGAN MENGGUNAKAN IPV6," *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, vol. 15, no. 1, pp. 10-15, 2020.
- [7] Y. C. Albert, "Analisis Performansi Antara Apache & Nginx Web Server dalam Menangani Client Request," *JURNAL SISTEM DAN INFORMATIKA (JSI)*, vol. 14, no. 1, pp. 48-56, 2019.
- [8] Basorudin and dkk, "Perancangan dan Implementasi Sistem Operasi Linux Debian untuk Konfigurasi Content Management System (CMS) Wordpress Dengan Winscp," *BULLETIN OF COMPUTER SCIENCE RESEARCH*, vol. 3, no. 1, pp. 21-29, 2022.
- [9] Apache, "Apache Http Server Project," Apache, 8 6 2020. [Online]. Available: <https://httpd.apache.org/docs/2.4/programs/ab.html>. [Accessed 3 3 2023].
- [10] J. Haritsah and dkk, "Analisis Proses Sistem Untuk Implementasi Interplanetary File System (Ipfs) Pada Smart Contract Ethereum," in *Smart Contract Ethereum, e-Proceeding of Engineering*, 2019.
- [11] A. Molavi, "Analisa Kinerja Web Server E-learning Menggunakan Apache Benchmark dan Httperf," *Jurnal Integrasi*, vol. 8, no. 2, pp. 93-100, 2016.
- [12] H. C. Y. Unsong and J. Andjarwirawan, "Analisa Kinerja Apache dan Nginx dalam Arsitektur Microservice Menggunakan Siege," *Jurnal Infra*, vol. 9, no. 2, 2021.
- [13] M. A. A. Putra, I. Fitri and A. Iskandar, "Implementasi High Availability Cluster Web Server Menggunakan Virtualisasi Container Docker," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 1, pp. 9-13, 2020.
- [14] A. Molavi, "Metode Pertahanan Web Server Terhadap Distributed Slow HTTP DoS Attack," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 56-70, 2020.

