

Analisis Performansi Web Server Saat Menangani Permintaan *Client* Menggunakan Metode *Reverse Proxy Caching Nginx* dan *Varnish*

Analysis of Web Server Performance in Handling Client Requests Using the Reverse Proxy Caching Nginx and Varnish Method

Muhammad Naufal Ammar Azi¹, Bongga Arifwidodo^{*,2}, Eka Wahyudi³

^{1,2}Prodi S1 Teknik Telekomunikasi, ³Prodi D3 Teknik Telekomunikasi
Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto
Jl. D.I. Panjaitan No. 128 Purwokerto, Jawa Tengah, Indonesia

*²Penulis Korespondensi: bongga@ittelkom-pwt.ac.id
¹18101022@ittelkom-pwt.ac.id, ³ekawahyudi@ittelkom-pwt.ac.id

Received 23-09-2022, accepted 17-01-2023, published 31-01-2023

Abstrak

Web Technology Surveys melaporkan peningkatan pengguna internet yang mengakses website terus meningkat setiap tahunnya terutama web server berbasis Apache dan Nginx. Kondisi tersebut berdampak pada performa dari sebuah web server menjadi berat bahkan jika melebihi kapasitas bisa terjadi *downtime* server. Sehingga diperlukannya *reverse proxy*. Metode ini mampu menyeimbangkan beban pada beberapa server dan menyediakan *caching* untuk server *back-end* yang kurang optimal khususnya pada web server berbasis Apache. Penelitian menggunakan *reverse proxy caching Nginx* dan *Varnish* pada web server Apache untuk meningkatkan kinerja web server saat melayani permintaan *client*. Pengujian performa web server dilakukan dengan mengirim 3 macam uji yaitu beban koneksi dengan jumlah *concurrency* yang sama kemudian diukur dengan *apache benchmark Tools*. Berdasarkan hasil pengujian bahwa Apache web server yang telah dikonfigurasi *reverse proxy caching Nginx* memberikan kinerja lebih baik dari *reverse proxy caching Varnish*. Parameter *request per second* sebesar 1207 req/s (39,6%), *Time taken for test* sebesar 16,65 detik (28,3%), *Time per request* sebesar 166,4 ms (28,4%), dan *Transfer rate* sebesar 13,15 MB/sec (38,4%).

Kata kunci: Nginx, Reverse Proxy, Varnish, Web server

Abstract

Web Technology Surveys report increased internet users accessing websites yearly, especially web servers based on Apache and Nginx. These conditions impact the performance of a web server to be heavy. Even if it exceeds capacity, server downtime can occur. Therefore, it would help if a reverse proxy is used. This method can balance the load on multiple servers and provide caching for back-end servers that could be more optimal, especially on Apache-based web servers. This research uses reverse proxy caching Nginx and Varnish on the Apache webserver to improve web server performance when serving client requests. Web server performance testing is carried out by sending three tests, namely the connection load with the same number of concurrencies and then being measured with the Apache Benchmark Tools. Based on the test results, the Apache web server configured for Nginx caching reverse proxy performs better than the Varnish caching reverse proxy. The request parameter per second is 1207 req/seconds (39.6%), the Time taken for the test is 16.65 seconds (28.3%), the Time per request is 166.4 ms (28.4%), and the Transfer rate of 13.15 MB/sec (38.4%).

Keywords: Nginx, Reserve Proxy, Varnish, Web server

I. PENDAHULUAN

Web Server memberikan kemampuan layanan terhadap penerimaan dan pengiriman akses dari HTTP dan HTTPS melalui aplikasi web browser [1]. Seiring waktu banyaknya trafik menyebabkan halaman web “populer” memiliki trafik yang tinggi dan sayangnya jika melebihi beban kapasitas server akan berakibat pada downtime server [2]. Perlunya system yang mampu menyeimbangkan beban (*load balancing*) trafik dan salah satu jenisnya merupakan *proxy*, baik *forward proxy* dan *reverse proxy* [3]. Penelitian [4] melaporkan ketika web server dikonfigurasi dengan *reverse proxy* terjadi peningkatan performa pada web server dibandingkan tanpa menggunakan *reverse proxy*. Dalam menjalankan fungsinya, terdapat beberapa *reverse proxy caching* seperti *Nginx*, *Squid*, *Varnish*, *aiCatch*, *Lighttpd*, dan lain sebagainya. Banyak *reverse proxy caching* yang tersedia, namun penggunaan *Nginx* dan *Varnish* paling banyak digunakan [5].

Hasil penelitian ini melihat kinerja dari web server dapat diketahui dengan menggunakan *tools* yaitu *Apache Benchmark tools* dan uji koneksi. Seperti halnya dengan penelitian [6] menggunakan *tools Apache Benchmarking Tool* untuk pengujian. Hasil menunjukkan penggunaan memory tertinggi adalah pada server *Nginx* dibanding server *Apache* yakni 256 MB (memory maksimal) terjadi saat 10.000 koneksi. Parameter penggunaan beban server, server *Apache* mampu menerima beban maksimal daripada server *NGINX* yaitu melayani 141,88 request tiap menitnya ketika 50.000 request. Penelitian [7] meneliti perbandingan performa *reverse proxy caching* *Nginx* dan *Varnish* Pada Web server *Apache* pada sumberdaya *hardware*. Berdasarkan hasil Kedua *reverse proxy caching* yang di uji terbukti memiliki manajemen sumberdaya perangkat keras CPU yang lebih baik dirapa web server *Apache* tanpa *reverse proxy caching*. Tetapi pada penggunaan *memory* server *Apache* tanpa *reverse proxy caching* memiliki tingkat penggunaan *memory* yang lebih rendah daripada server *Apache* dengan *reverse proxy caching*. Hasil dari manajemen sumberdaya *Varnish* lebih baik daripada *Nginx* pada sisi penggunaan CPU dan bebeda sangat tipis pada penggunaan *memory*. Sejalan dengan penelitina [8] membutuhkan hasil kinerja agar dapat dianalisa, untuk mengetahui kinerja dari server *Apache* maupun server *Nginx*. Hasil menunjukkan saat implementasi dan percobaan yang telah dilakukan, bahwa perhitungan rata-rata parameter response time maupun throughput, web server *Apache* lebih unggul dibandingkan dengan web server *Nginx*, tetapi kemampuan saat melayani transaksi web server *Nginx* lebih banyak daripada web server *Apache*.

Penelitian ini menganalisis performansi webserver *Apache* saat melayani permintaan user dengan menerapkan *Reserve Proxy Caching* dengan konfigurasi aplikasi *Nginx* dan *Varnish*. Sehingga didapat hasil yang lebih baik diantara *Nginx* dan *Varnis* saat melayani permintaan user dari sisi *request per second*, *Time per request* dan *Transfer rate*.

II. KAJIAN PUSTAKA

A. Web Server

Web server adalah sistem yang memberi layanan yang diminta (request) kepada pengguna melalui internet. Web server terdiri dari server fisik, sistem operasi, dan aplikasi yang memfasilitasi komunikasi HTTP. Web server juga dikenal sebagai internet server karena cara melayani para penggunanya adalah melalui internet. Pekerjaan web server pada dasarnya adalah menerima permintaan dari klien dan mengirim tanggapan terhadap permintaan tersebut. Sebuah web server menerima direktori dokumen yang diminta klien, kemudian merujuk ke nama file (untuk permintaan statis), lalu mengirimkan file yang ada dalam penyimpanan lokal kepada klien. Jika permintaan halaman dinamis maka web server akan menerjemahkannya ke sebuah program kemudian mengeksekusinya, dan mengirimkan output dari program tersebut kepada pengakses internet [1].

B. Apache

Apache adalah Open source yang dibangun dan dikelola oleh Apache.org. Apache terdiri dari dua blok bangunan utama dengan bangunan akhir yang terdiri dari banyak blok bangunan kecil lainnya. Blok Bangunan adalah Apache Core dan kemudian Modul Apache yang dalam arti memperluas inti Apache [9].

C. Nginx

Nginx atau yang biasa disebut “Engine-x” adalah software open-source yang memiliki kinerja tinggi sebagai server HTTP maupun reverse proxy. Nginx dapat memberikan konten statis dengan penggunaan efisien sumber daya sistem sehingga berdampak pada proses yang lebih cepat. Arsitektur Nginx dibangun secara modular sehingga mampu mendukung berbagai fitur seperti Load Balancing dan Reverse Proxying, Virtual hosts berbasis *name* dan IP. Nginx berjalan pada platform Unix, Linux, varian dari BSD, MacOS X, Solaris, dan Microsoft Windows [10].

D. Reverse Proxy Caching Nginx

Nginx dipilih karena memiliki kemampuan kinerjanya yang tinggi, baik dari stabilitas, fasilitas fitur, serta mudah dikonfigurasi dan yang paling utama adalah hemat dari sisi sumber daya [11]. Kondisi ini berbeda dengan server tradisional, Nginx berdiri sendiri artinya tidak bergantung pada ‘thread’ saat melayani permintaan (*request*). Sisi lain dari Nginx adalah *scalable asynchronous event-driven architecture* yaitu menggunakan teknik arsitektur yang digerakkan berdasarkan kondisi yang lebih skalabel. Kelebihan dari arsitektur ini, saat menerima beban trafik dengan menggunakan memori yang sedikit dengan jumlah yang bisa diprediksikan.

E. Varnish

Varnish cache merupakan aplikasi *Web accelerator* yang dikenal juga sebagai *Hyper Text Transfer Protocol (HTTP) reverse proxy*. Prinsip kerja Varnish yakni pada *front end* dari sebuah server HTTP dan memiliki kemampuan mengkonfigurasi cache tiap konten Website. Kemudian menyimpan data halaman Website di dalam memory, dengan tujuan mengurangi beban Web server saat memuat halaman yang sama. Kemampuan *Varnish* bukan hanya sebagai reverse proxy dimana mampu menyimpan content (caching), namun berfungsi sebagai pembagi beban pada server backend yang tersedia. Penyeimbang beban pada sistem varnish biasa disebut *back-end director* [12].

F. Transfer Rate

Transfer Rate adalah metrik standar yang digunakan untuk mengukur kecepatan di mana data atau informasi berpindah dari satu lokasi ke lokasi lain. Kecepatan transfer sering kali bergantung pada teknologi yang digunakan untuk transfer [13]. Transfer Rate dapat dikatakan bagus apabila nilai transfer yang dihasilkan cukup tinggi, karena nilai transfer yang dihasilkan tinggi artinya proses transfer data atau files yang dilakukan pada web server lebih cepat, Nilai transfer *rate* sebesar 660 Kbytes/s sudah bisa dikatakan bagus [14].

G. Time per Request

Time per Request adalah Rata-rata waktu yang dihabiskan per permintaan. Nilai pada Time per Request dapat dikatakan bagus apabila nilai yang dihasilkan semakin kecil, karena itu menunjukkan jika web server mampu menangani request lebih cepat [6].

H. Time Taken for Tests

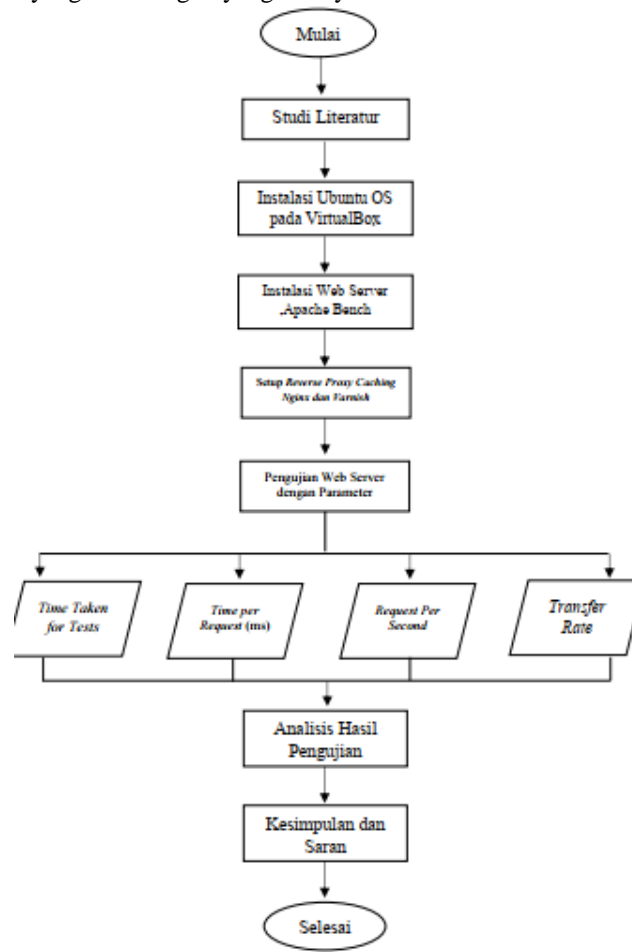
Time Taken for Tests merupakan waktu yang dibutuhkan dari saat koneksi socket pertama dibuat hingga saat respons terakhir diterima [15]. Nilai yang dihasilkan time taken for tests dapat dikatakan bagus apabila nilainya lebih kecil, artinya itu menunjukkan jika waktu yang dibutuhkan dalam menangani semua request lebih cepat.

I. Request Per Second

Request Per Second adalah jumlah permintaan per detik. Nilai dapat dikatakan bagus pada Request Per Second apabila nilai yang dihasilkan semakin besar, karena itu menunjukkan bahwa dalam 1 detik web server tersebut dapat menyelesaikan banyak request [16].

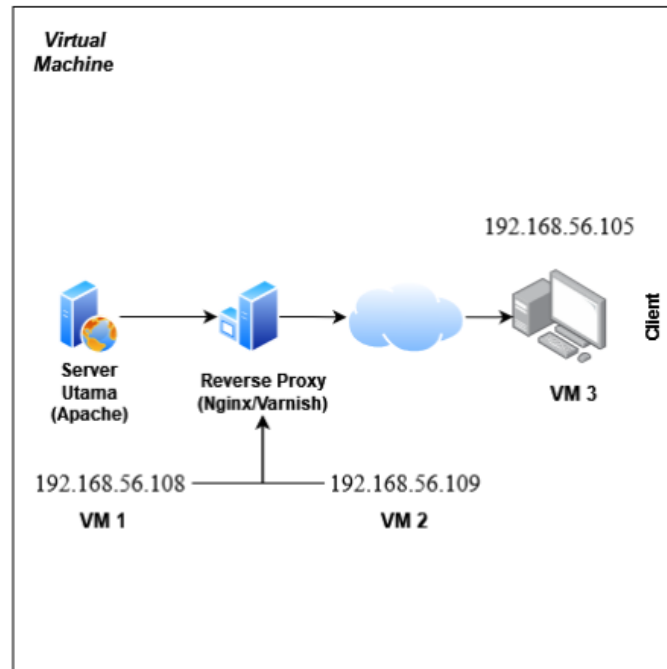
III. METODE PENELITIAN

Pada penelitian ini penulis meneliti mengenai Analisis Performansi Reverse Proxy Nginx dalam menangani Client Request menggunakan Ubuntu 20.4 dengan metode Eksperimen. Di mana metode ini bertujuan untuk menguji pengaruh suatu variable terhadap variable lain, atau menguji hubungan sebab akibat antara variable yang satu dengan yang lainnya.



Gambar 1. Alur Penelitian

Gambar 1. Menjelaskan alur penulis melakukan studi literatur, di mana menelaah pada berbagai literatur yang berhubungan dengan tema penelitian studi terhadap sistem sejenis yang pernah dibangun agar mendapatkan referensi secara menyeluruh terhadap sistem yang akan dibangun. Selanjutnya melakukan instalasi Ubuntu OS pada VirtualBox sebagai virtual environment pada laptop. Selanjutnya, jika sudah berhasil maka dilanjutkan instalasi web server, Apache benchmark sebagai Tool pengujian pada penelitian ini. kemudian melakukan pengujian pada web server dengan mengamati parameter *Time Taken for Tests*, *Time per Request (ms)*, *Transfer Rate* dan *Request per Second* dengan menggunakan Apache Benchmark Tool, lalu setiap parameter yang telah dikumpulkan selanjutnya dilakukan analisis pada hasil pengujian kemudian diakhiri dengan kesimpulan dan saran penulis pada penelitian.



Gambar 2. Topologi Penelitian

Gambar 2. Merupakan gambar topologi yang menunjukkan bahwa server proxy akan tampak seperti server utama untuk client penghubung manapun, artinya server proxy yang akan meneruskan permintaan dari client ke server utama dan meneruskan konten atau informasi dari server utama kembali ke client lagi. Aplikasi Apache Benchmarking dan Siege digunakan saat melakukan pengujian, dengan memberikan koneksi sesuai dengan skenario ke web server. Layaknya jumlah atau trafik pengunjung yang mencoba untuk mengakses ke web server.

Tabel 1. menunjukkan variable pengujian berupa concurrency, dan koneksi. Variabel Concurrency hanya sebanyak 200, berbeda pada masing masing web server. Pada koneksi yang dikirimkan secara bersamaan menggunakan Apache Benchmark tools, angka yang digunakan cukup besar agar dapat menghasilkan perbedaan pada hasil pengesanan.

Tabel 1. Skenario Pengujian

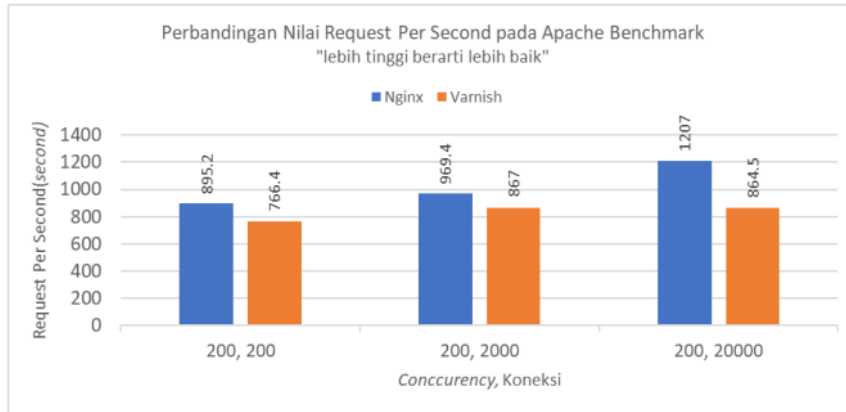
Variabel Pengujian	Web server	
	Reverse Proxy Caching	Reverse Proxy Caching
Concurrency	200	200
Koneksi	200, 2.000, 20.000	200, 2.000, 20.000

IV. HASIL DAN PEMBAHASAN

Pada bagian ini dipaparkan tentang hasil dan pembahasan, dengan pengujian Web Server. Parameter nya *Request per Second, Transfer Rate, Time taken for test, Time per Request.*

A. Request Per Second

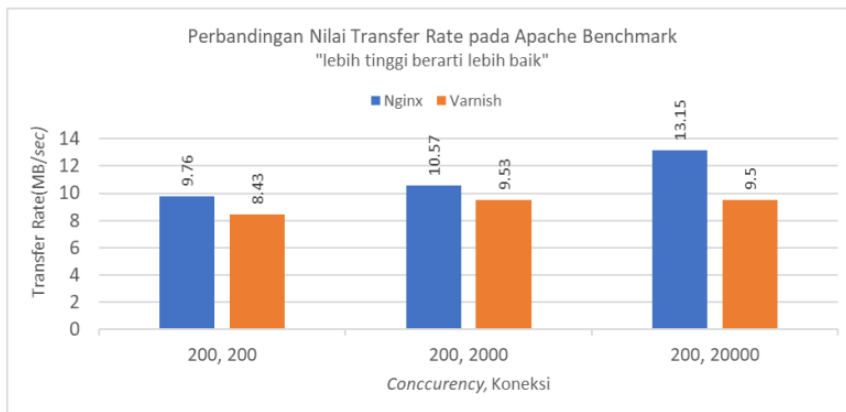
Request per Second berfungsi untuk mengetahui jumlah permintaan per detik, dapat dilihat berdasarkan Gambar 4 jumlah 200 concurrent user, dari nilai yang di dapatkan pada Request Per second reverse proxy caching nginx memiliki nilai yang cukup jauh dari reverse proxy caching varnish, pada reverse proxy caching nginx mengeksekusi dengan jumlah request yang lebih besar dari reverse proxy caching varnish, Karena nilai yang dihasilkan cukup tinggi dengan nilai sebesar 1207 req/sec maka reverse proxy caching nginx lebih baik dari reverse proxy caching varnish.



Gambar 3. Nilai Request per Second

B. Transfer Rate

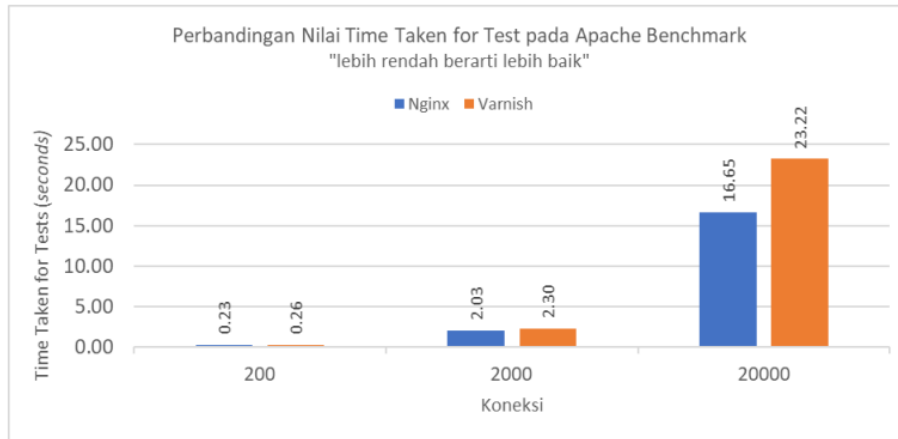
Transfer Rate adalah metrik standar yang digunakan untuk mengukur kecepatan di mana data atau informasi berpindah dari satu lokasi ke lokasi lain. Transfer Rate dapat dikatakan bagus apabila nilai transfer yang dihasilkan cukup tinggi, karena nilai transfer yang dihasilkan tinggi artinya proses transfer data atau files yang dilakukan pada web server lebih cepat. Pada Gambar 4. hasil pengujian dengan *concurrent user* 200 dan koneksi 200,2.000, dan 20.000 terdapat keunggulan pada reverse proxy caching Nginx dari reverse proxy caching Varnish pada setiap pengujian dalam transfer rate. Karena nilai yang dihasilkan pada reverse proxy caching nginx lebih besar maka kecepatan transfer yang dihasilkan lebih bagus performanya dibandingkan dengan reverse proxy caching varnish. Nilai yang dihasilkan dapat dipengaruhi karena koneksi internet yang tidak stabil tergantung dari kondisi internet peneliti saat pengambilan data.



Gambar 4. Nilai Transfer Rate

C. Time Taken For Test

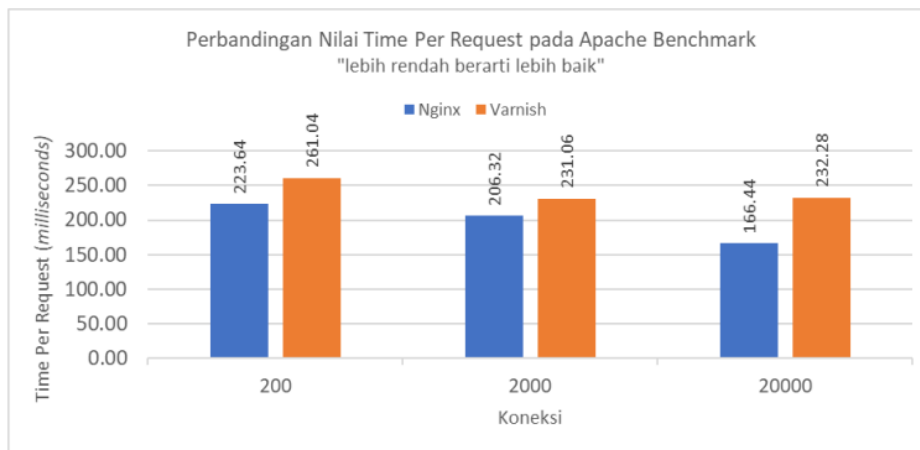
Pada Gambar 5. parameter Time taken for Test menunjukkan semakin banyak jumlah koneksi mengarah ke server, maka waktu yang dibutuhkan untuk menyelesaikan semua permintaan juga semakin meningkat. Dari semua pengujian yang dilakukan pada masing masing koneksi, Waktu yang dibutuhkan untuk melakukan satu kali pengujian oleh server 1 dengan menggunakan reverse proxy caching Nginx lebih cepat dibandingkan server 2 yang menggunakan reverse proxy caching Varnish.



Gambar 5. Nilai Time Taken For Test

D. Time Per Request

Pengujian ini berfungsi untuk mengetahui rerata waktu yang diperlukan ketika menangani 1 buah request. Hasil pengujian pada Gambar. 6 menunjukkan semakin banyaknya jumlah koneksi pada server maka nilai waktu setiap request akan semakin kecil. Kondisi pada server 2 waktu setiap request pada masing masing pengujian lebih besar dibandingkan dengan server 1 yang menggunakan reverse proxy caching nginx. Selain itu server 1 memiliki waktu rerata yang jauh lebih kecil jika dibandingkan server 2 yang menggunakan varnish. Hal ini menunjukkan bahwa server 1 dengan reverse proxy caching nginx sanggup melayani request lebih cepat dibandingkan dengan server 2 yang menggunakan reverse proxy caching Varnish.



Gambar 6. Nilai Time Per Request

V. KESIMPULAN

Hasil penelitian yang didapat dari pengujian menggunakan parameter uji *Transfer Rate*, *Time taken for Test*, *Request Per Second*, dan *Time Per Request* dengan beban koneksi sebanyak 200, 2.000 sampai 20.000 koneksi dan Concurrency sebanyak 200 pada Apache Benchmark. Apache web server menggunakan reverse proxy caching Nginx memiliki peningkatan performa dengan nilai lebih baik dari pada reverse proxy caching Varnish, dengan parameter request per second sebesar 1207 req/seconds (39,6% Lebih baik dibandingkan Varnish), Time taken for test sebesar 16,65 seconds (28,3% Lebih baik dibandingkan Varnish), Time per request sebesar 166,4 millisecond 232,277 (28,4% Lebih baik dibandingkan Varnish), dan Transfer rate sebesar 13,15 MB/sec (38,4% Lebih baik dibandingkan Varnish).

DAFTAR PUSTAKA

- [1]. Muliana Utama, R., & Pratama Putri, F. (2018). Analisis Perbandingan Kinerja Web server Apache dan Nginx pada VPS dengan Menggunakan HTTPERF untuk Sistem Operasi CentOS. In *Jurnal Web Informatika Teknologi* (Vol. 3, Issue 1).
- [2]. D. Perancangan, L. Pengujian, D. Balancing, Menggunakan, R. Dani, and F. Suryawan, "khazanah informatika Jurnal Ilmu Komputer dan Informatika," *Perancangan Dan Pengujian Load Balancing Dan Failover Menggunakan Nginx*, vol. 3, no. 1, pp. 2477–698, 2017.
- [3]. I. Gusti, L. Putra, and E. Prisma, "Implementasi Load Balancing Pada Web Server Dengan Menggunakan Apache," Dec. 2016.
- [4]. J. Jurnal and Vol, "Analisis Dan Implementasi Reverse Proxy Sebagai Media Komunikasi Client Server Menggunakan Apache (Studi Kasus Pada Lab. Jaringan Komputer IST Akprind Yogyakarta)," vol. 5, no. 2, 2017.
- [5]. Luthfi, M., Data, M., & Yahya, W. (2018). Perbandingan Performa Reverse proxy caching Nginx dan Varnish Pada Web server Apache (Vol.2, Issue 4).
- [6]. I. K. S. Satwika and K. N. Semadi, "Perbandingan Performansi Web Server Apache Dan Nginx Dengan Menggunakan Ipv6," *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, vol. 15, no. 1, Feb. 2020.
- [7]. Luthfi, M., Data, M., & Yahya, W. (2018). Perbandingan Performa Reverse proxy caching Nginx dan Varnish Pada Web server Apache (Vol. 2, Issue 4). <http://j-ptiik.ub.ac.id>
- [8]. Christopher, H., Unsong, Y., & Andjarwirawan, J. (2021.). Analisa Kinerja Apache dan Nginx dalam Arsitektur Microservice Menggunakan Siege.
- [9]. Vokasional, J., & Ferina Irza, I. (2017). VOTEKNIKA Analisis Perbandingan Kinerja Web server Apache dan Nginx Menggunakan Httpperf Pada Portal Berita (Studi Kasus beritalinux.com). *Teknik Elektronika & Informatika*, 5.
- [10]. Aziz, A., & Tampati, T. (2015). Analisis Web server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web server Apache dengan Nginx. In *Analisis Web server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web server Apache dengan Nginx JURNAL MULTINETICS* (Vol. 1, Issue 2).
- [11]. Z. Lei, H. Zhou, S. Ye, W. Hu, and G.-P. Liu, "Cost-Effective Server-side Re-deployment for Web-based Online Laboratories Using NGINX Reverse Proxy," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17204–17209, 2020, doi: 10.1016/j.ifacol.2020.12.1748.
- [12]. I. Fakultas, Komputer, T. Erlinda, M. Data, and R. Siregar, "Perancangan Klaster Server Web Dengan Availabilitas Tinggi Menggunakan Teknologi Failover, Load Balancing dan Distributed File System," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 2548–964, 2018.
- [13]. N. Agatha, "Cara Konfigurasi Nginx Reverse Proxy," *Hostinger Tutorial*, Apr. 14, 2020. <https://www.hostinger.co.id/tutorial/nginx-reverse-proxy> (accessed Jan. 20, 2022).
- [14]. Aziz, A., & Tampati, T. (2015). Analisis Web server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web server Apache dengan Nginx. In *Analisis Web server untuk Pengembangan Hosting Server Institusi: Perbandingan Kinerja Web server Apache dengan Nginx JURNAL MULTINETICS* (Vol. 1, Issue 2).
- [15]. "ab - Apache HTTP server benchmarking tool - Apache HTTP Server Version 2.4," *Apache.org*, 2022. <https://httpd.apache.org/docs/2.4/programs/ab.html> (accessed May 29, 2022).
- [16]. M. Arman, "Analisa Kinerja Web Server E-learning Menggunakan Apache Benchmark dan Httpperf," *Jurnal Integrasi* |, vol. 93, no. 2, pp. 2085–3858, 2016.