

PERANCANGAN DAN KOMPARASI WEB SERVER NGINX DENGAN WEB SERVER APACHE SERTA PEMANFAATAN REVERSE PROXY SERVER PADA NGINX

Putu Satya Saputra^{*1}, Putu Aditya Pratama¹, Luh Putu Ary Sri Tjahyanti¹

¹Teknologi Informasi, Fakultas Teknik, Universitas Panji Sakti
Email: ¹satya@unipas.ac.id, ¹aditya@unipas.ac.id, ²ary.tjahyanti@unipas.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 6 Juni 2023, diterima untuk diterbitkan: 29 Oktober 2023)

Abstrak

Server web adalah suatu *software* atau perangkat lunak yang melayani halaman web melalui internet menggunakan Hypertext Transfer Protocol (HTTP). Fungsi utamanya adalah menerima permintaan dari klien, memprosesnya, dan mengirim kembali halaman web atau data yang diminta. Server web adalah komponen penting dari infrastruktur World Wide Web dan digunakan untuk menghosting situs web, aplikasi web, dan layanan online lainnya. Contoh server web yang paling populer digunakan yaitu Nginx dan Apache. Nginx dan Apache adalah platform *open-source* yang umum digunakan saat ini. Keduanya memiliki kelebihan dan kekurangan yang berbeda. Nginx terkenal memiliki performa yang lebih baik dalam mengatasi permintaan dari banyak klien secara bersamaan, sementara Apache dikenal fleksibel dan mendukung banyak modul maupun *plugin* tambahan yang dapat meningkatkan fungsionalitasnya serta lebih mudah untuk dikonfigurasi oleh pengguna yang baru menggunakan platform *web server*. Penelitian ini akan melakukan uji coba pada kedua platform dan memberikan rekomendasi mengenai platform yang lebih sesuai digunakan dalam skenario tertentu. Hasil penelitian akan menganalisis kelebihan dan kekurangan masing-masing platform serta penerapan Nginx sebagai *reverse proxy server* untuk meningkatkan keamanan *web server*.

Kata kunci: *server web, nginx, apache, reverse proxy server*

DESIGN AND COMPARISON OF NGINX WEB SERVER WITH APACHE WEB SERVER AND USE OF REVERSE PROXY SERVER ON NGINX

Abstract

A web server is a software application that serves web pages over the internet using the Hypertext Transfer Protocol (HTTP). Its primary function is to receive requests from clients, process them, and send back the requested web pages or data. Web servers are critical components of the World Wide Web infrastructure and are used to host websites, web applications, and other online services. The most popular examples of web servers used are Nginx and Apache. Nginx and Apache are the most commonly used open-source platforms today. Both have different advantages and disadvantages. Nginx is known for having better performance in handling requests from multiple clients simultaneously, while Apache is known to be flexible and supports many additional modules and plugins that can increase its functionality and be easier to configure for users who are new to using the web server platform. This research will configure both platforms and provide recommendations regarding which platform is more suitable for use in certain scenarios. The results of the research will analyze the advantages and disadvantages of each platform and the application of Nginx as a reverse proxy server to improve web server security.

Keywords: *web server, nginx, apache, reverse proxy server*

1. PENDAHULUAN

Web server merupakan perangkat lunak yang menyajikan konten ke perangkat klien, seperti browser web, melalui internet atau jaringan lokal. Ketika klien meminta halaman web atau konten lain, *web server* memproses permintaan tersebut dan

mengembalikan informasi yang diminta ke klien. Web Server menggunakan *Hypertext Transfer Protocol* (HTTP) untuk berkomunikasi dengan klien serta dapat mendukung protokol lain seperti HTTPS (HTTP over SSL/TLS) untuk koneksi yang aman (A. Tedyyana and O. Ghazali, 2021). Server web dapat menyajikan konten statis, seperti halaman HTML,

gambar, dan video, serta konten dinamis yang dihasilkan oleh skrip atau aplikasi sisi server, seperti PHP, Java, atau Python (Riswandi, Kasim, and M. F. Raharjo, 2020). Contoh server web termasuk Apache, Nginx, Microsoft IIS, dan Google Web Server (GWS). Nginx dan Apache adalah dua platform *web server* yang paling populer digunakan saat ini di berbagai negara.

Nginx adalah salah satu platform *web server open-source* yang populer digunakan oleh organisasi dan perusahaan untuk menjalankan situs web dan aplikasi web. Nginx memiliki fungsi, keunggulan, dan kelemahan yaitu sebagai berikut. Fungsi *web server* Nginx yaitu menjalankan situs web dan aplikasi web, menerima permintaan HTTP dan HTTPS, menangani satu atau lebih klien secara bersamaan, meningkatkan keamanan situs web dengan fitur pengaturan izin akses dan pencegahan serangan DDoS, mempercepat waktu muat situs web dengan *caching* dan pengaturan konfigurasi yang tepat, mendukung protokol HTTP/2 dan TLS (Y. W. Luthfi Muhammad, Data Mahendra, 2018). Keunggulan dari *web server* Nginx adalah memiliki performa yang cepat dan efisien dalam menangani permintaan dari banyak klien secara bersamaan, konfigurasi yang simpel dan fleksibel dalam mengatur sumber daya server, memiliki kemampuan untuk berfungsi sebagai *reverse proxy server* dan *load balancer*, membutuhkan sumber daya yang lebih sedikit dibandingkan dengan platform web server lainnya, dapat diintegrasikan dengan teknologi terkini seperti Docker, Kubernetes, dan *cloud computing* (M. Rafli, 2022). *Reverse proxy server* adalah sebuah server yang berfungsi sebagai perantara antara klien dan server (R. A. Muzaki dkk., 2020). Dalam hal ini, *server proxy* menerima permintaan dari klien dan meneruskannya ke server yang sesuai. Berbeda dengan *proxy server* tradisional yang berada di antara klien dan internet, *reverse proxy server* berada di antara klien dan satu atau lebih server, dan mengelola semua permintaan masuk atas nama server. Ketika seorang klien mengirim permintaan ke server *reverse proxy*, server tersebut meneruskannya ke server yang sesuai kemudian mengirimkan respons kembali ke klien. Klien tidak pernah berkomunikasi langsung dengan server sehingga dapat meningkatkan keamanan dan performa. Kelemahan dari web server Nginx diantaranya yakni kurang mendukung fungsi yang berkaitan dengan konten dinamis dan membutuhkan *plugin* atau modul tambahan untuk mengatasi hal ini, tidak sepenuhnya kompatibel dengan seluruh modul Apache, kurangnya dokumentasi dibandingkan dengan platform web server lainnya.

Apache adalah platform *web server open-source* yang sudah populer dan digunakan secara luas di seluruh dunia. Berikut merupakan beberapa fungsi, keunggulan, dan kelemahan dari *web server* Apache. Fungsi *web server* Apache yaitu memungkinkan pengaturan izin akses dan autentikasi, mempercepat

waktu muat situs web dengan *caching* dan pengaturan konfigurasi yang tepat, dapat diintegrasikan dengan teknologi terkini seperti PHP, MySQL, dan Perl (M. F. Kurniawan dkk., 2021). Keunggulan dari *web server* Apache diantaranya yakni mendukung banyak modul dan *plugin* tambahan yang dapat meningkatkan fungsionalitasnya serta lebih mudah untuk dikonfigurasi oleh pengguna yang baru menggunakan platform *web server* (A. M. Komaruddin dkk., 2019). Kelemahan dari *web server* Apache adalah kurang efisien dalam menangani permintaan dari banyak klien secara bersamaan dibandingkan dengan platform *web server* lainnya, lebih banyak memakan sumber daya dibandingkan dengan Nginx, dapat menjadi lebih rentan terhadap serangan DDoS jika tidak diatur dengan benar.

Nginx diketahui memiliki performa yang lebih baik dalam mengatasi permintaan dari banyak klien secara bersamaan (D. K. Hakim dkk., 2019), sedangkan Apache terkenal dengan fleksibilitas dan kemampuan untuk menangani banyak jenis tugas. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja Nginx dan Apache dalam situasi yang berbeda dan memberikan rekomendasi mengenai platform yang lebih cocok digunakan dalam skenario tertentu. Dalam penelitian ini, akan dilakukan uji coba pada kedua platform dan hasilnya akan dianalisis untuk memahami kelebihan dan kekurangan dari masing-masing platform. Penelitian perancangan dan komparasi *web server* Nginx dengan Apache penting karena pemilihan platform yang tepat dapat mempengaruhi kinerja dan keamanan situs web. Selain itu, penelitian ini juga dapat membantu pengembang web dalam memahami fitur dan keunggulan dari masing-masing platform *web server* dan bagaimana memanfaatkannya secara efektif.

Penelitian ini juga relevan dengan perkembangan teknologi web saat ini, dimana performa dan keamanan situs web sangat penting untuk keberhasilan bisnis online. Sebagai contoh, website *e-commerce* yang ramai pengunjung membutuhkan platform *web server* yang mampu menangani permintaan secara efisien dan cepat, sementara website yang lebih kecil dengan lalu lintas pengunjung yang lebih sedikit mungkin dapat menggunakan platform yang lebih fleksibel. Dalam penelitian ini, perlu juga dipertimbangkan faktor-faktor lain seperti biaya, keamanan, dan kemudahan penggunaan dalam memilih platform web server yang tepat. Keputusan pemilihan platform web server yang tepat akan penting untuk memastikan kinerja situs web yang optimal dan pengalaman pengguna yang baik.

Dalam melakukan penelitian perancangan dan komparasi *web server* Nginx dengan Apache, perlu dilakukan studi literatur terlebih dahulu untuk memahami konsep dan fitur masing-masing platform *web server*. Kemudian, dilakukan pengujian performa dan analisis fitur untuk mendapatkan

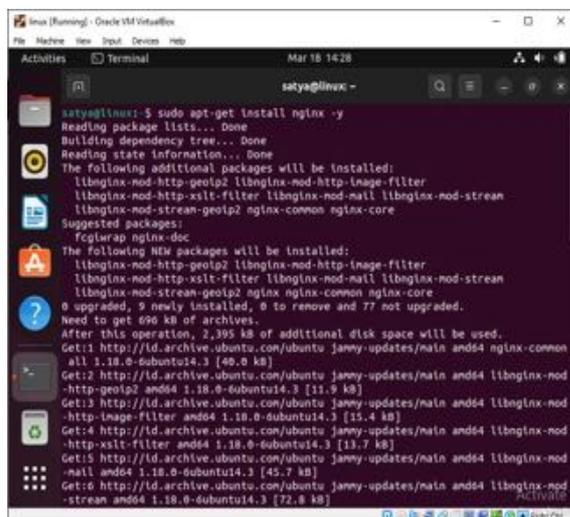
pemahaman yang lebih mendalam mengenai kedua platform *web server* tersebut. Secara keseluruhan, penelitian perancangan dan komparasi *web server* Nginx dengan Apache akan dilakukan uji coba dan analisis secara mendalam untuk memberikan rekomendasi yang tepat bagi pengguna dalam memilih platform yang sesuai untuk kebutuhan pengguna.

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah pengujian performa dan fungsionalitas *web server* apache dan nginx. Metode pengujian performa adalah proses mengukur dan membandingkan performa *web server* Nginx dan Apache (D. W. L. Pamungkas and S. Rochimah, 2019). Dalam konteks membandingkan performa *web server*, metode pengujian performa dapat membantu untuk mengidentifikasi server mana yang lebih baik dalam menangani beban dan menghasilkan respons yang lebih cepat dan stabil (D. I. Permatasari, 2020). Pengujian ini melibatkan server dengan beberapa permintaan (*request*) untuk mengukur berapa banyak permintaan yang dapat ditangani oleh server dalam satu waktu. Pengujian ini dapat memberikan informasi tentang seberapa baik server menangani beban tinggi dan apakah server mampu menangani permintaan secara efisien. Dalam melakukan pengujian performa, penting untuk mempertimbangkan faktor-faktor lain seperti lingkungan pengujian, konfigurasi server, dan parameter pengujian yang digunakan untuk memastikan bahwa hasil pengujian akurat dan dapat diandalkan.

3. HASIL DAN PEMBAHASAN

Fungsi dari *web server* Apache dan Nginx dapat digunakan secara bersamaan dan memanfaatkan masing – masing keunggulan setiap *web server*. Pengguna dapat menggunakan Nginx sebagai *reverse proxy server* dan Apache sebagai *backend server*.

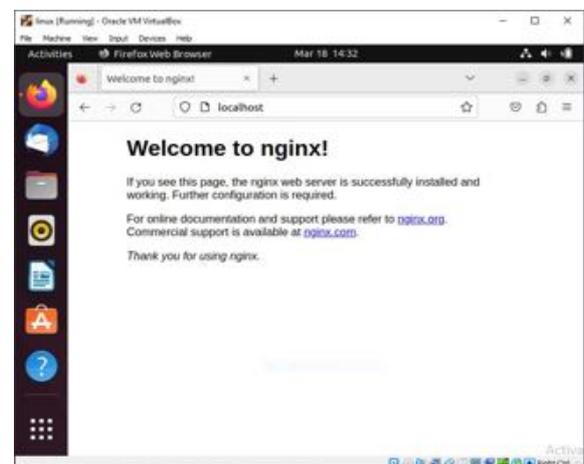


Gambar 1. Instalasi Web Server Nginx

Nginx akan menangani permintaan dari klien dan mengarahkan permintaan tersebut ke server backend Apache. Dalam skenario ini, Nginx bertindak sebagai pintu gerbang utama yang menerima permintaan HTTP dari *web client* dan menentukan server backend mana yang akan digunakan untuk menangani permintaan tersebut.

Langkah sederhana untuk menggunakan Nginx sebagai *reverse proxy server* dan Apache sebagai backend server adalah sebagai berikut. Melakukan instalasi dan konfigurasi Nginx sebagai *reverse proxy server* menggunakan Ubuntu Desktop. Instalasi *web server* Nginx memerlukan sistem operasi Ubuntu kemudian membuka terminal pada Ubuntu dan memastikan Ubuntu sudah terkoneksi dengan internet.

Gambar 1. menampilkan instalasi web server Nginx dengan mendownload pada terminal Ubuntu dengan perintah `sudo apt-get install nginx -y`.



Gambar 2. Testing Server Nginx

Selanjutnya pada Gambar 2. melakukan *testing* pada browser untuk memastikan *web server* Nginx sudah berjalan. Pada Ubuntu Desktop, testing dapat dilakukan dengan membuka *web browser* dan menulis *localhost* pada *address bar*, kemudian tampilan default Nginx akan tampil.

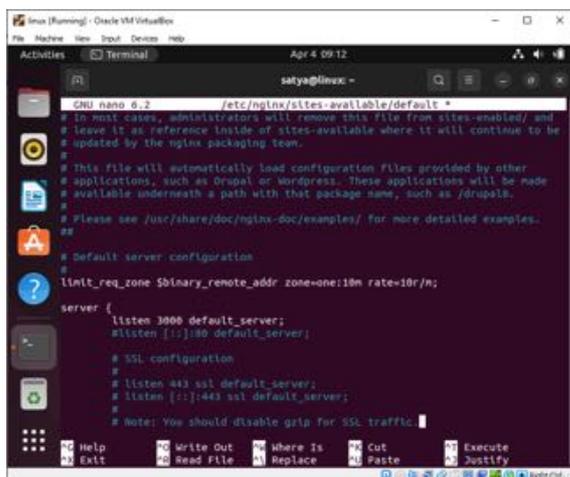


Gambar 3. Konfigurasi Nginx sebagai Reverse Proxy Server

Apabila Nginx sudah di install pada Ubuntu terminal kemudian membuat file konfigurasi di /etc/nginx/conf.d/ dengan nama misalnya "nginx.conf" sebagai *reverse proxy server*. *Reverse proxy server* adalah server yang bertindak sebagai perantara antara klien dan server di belakangnya.

Ketika klien meminta akses ke *web server*, permintaan tersebut akan diteruskan ke *reverse proxy server*, yang akan memproses permintaan tersebut dan mengirimkan kembali respons dari server yang sesuai ke klien. Seolah-olah klien berkomunikasi langsung dengan server, padahal sebenarnya klien berkomunikasi melalui *reverse proxy server* untuk menghindari serangan langsung ke server yang tidak diinginkan. Selanjutnya menambahkan konfigurasi seperti terlihat pada Gambar 3.

Gambar 3. Menampilkan konfigurasi web server Nginx sebagai *reverse proxy server*. *Reverse proxy server* dapat bertindak sebagai penghalang antara klien dan server asli, sehingga dapat membantu melindungi server asli dari serangan langsung seperti DDoS. Selain itu, *reverse proxy server* dapat digunakan untuk menerapkan kebijakan keamanan, seperti otentikasi dan otorisasi pengguna. *Reverse proxy server* dapat melakukan *caching*, yang berarti dapat menyimpan salinan sumber daya yang diminta oleh klien dan mengirimkan salinan tersebut langsung ke klien di masa depan, sehingga mempercepat waktu respon server dan mengurangi beban server asli. *Reverse proxy server* dapat digunakan untuk mengalihkan lalu lintas ke server yang berbeda-beda berdasarkan permintaan klien, seperti jenis permintaan atau lokasi geografis klien. Hal ini dapat membantu mengoptimalkan penggunaan sumber daya dan meningkatkan ketersediaan. *Reverse proxy server* dapat membantu mempermudah pengelolaan server asli dengan mengumpulkan permintaan klien ke satu titik masuk, sehingga memudahkan pengelolaan dan monitoring server.

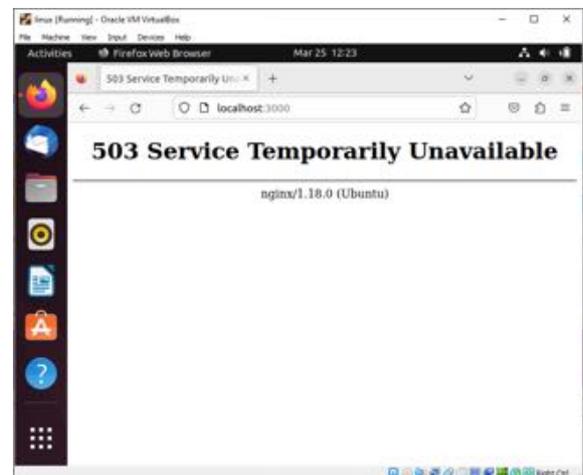


Gambar 4. Konfigurasi Nginx 10 Request per Menit

Gambar 4. menampilkan konfigurasi *web server* Nginx untuk membatasi permintaan klien 10 permintaan setiap menit. Membatasi permintaan

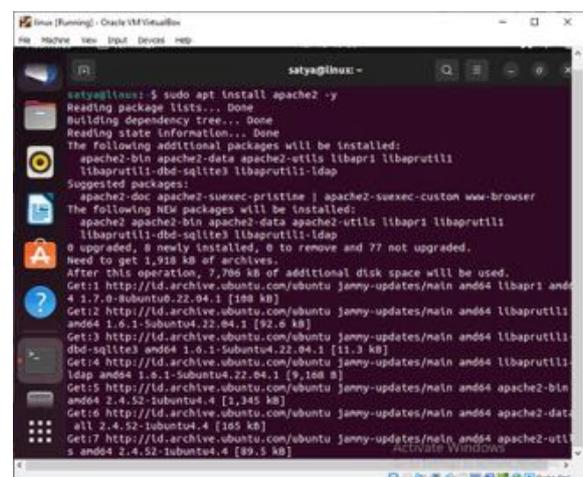
(*request*) per menit pada konfigurasi *web server* Nginx memiliki beberapa fungsi yaitu sebagai berikut. Dengan membatasi permintaan per menit, server dapat mengurangi beban yang ditangani oleh setiap permintaan, sehingga meningkatkan kinerja dan stabilitas server secara keseluruhan. Pembatasan permintaan per menit dapat membantu melindungi server dari serangan DDoS atau serangan *brute force*. Jika ada terlalu banyak permintaan yang datang dari alamat IP yang sama dalam waktu yang singkat, maka server dapat menganggapnya sebagai serangan dan memblokir akses tersebut.

Selain itu membatasi permintaan per menit server dapat memastikan bahwa setiap pengguna mendapatkan akses yang sama dan tidak diabaikan oleh permintaan lain yang memonopoli sumber daya server.



Gambar 5. Testing Server Nginx 10 Request per Menit

Instalasi dan konfigurasi dasar web server Nginx sudah selesai dilakukan yang ditampilkan pada Gambar 5. Selanjutnya melakukan instalasi Apache pada terminal Ubuntu.



Gambar 6. Instalasi Web Server Apache

Gambar 6. menunjukkan proses instalasi web server Apache pada terminal Ubuntu. Perintah untuk instalasi Apache yaitu `sudo apt install apache2 -y`.

Web server Apache secara otomatis berjalan pada port 80 untuk menjalankan web server.

Setelah melakukan instalasi Apache pada terminal Ubuntu, selanjutnya dilakukan testing pada web browser untuk memastikan web server Apache berjalan secara *default*. Pada Ubuntu Desktop, testing dapat dilakukan dengan membuka web browser dan menulis localhost pada *address bar*, kemudian tampilan *default* Apache.



Gambar 7. Testing Web Server Apache

Selanjutnya membuat konfigurasi Apache sebagai *backend server*. Konfigurasi ini akan mengarahkan permintaan dari web client ke server *backend* Apache yang berjalan di port 80. Konfigurasi Apache agar hanya menerima permintaan dari Nginx dapat dilakukan dengan menambahkan baris berikut pada file konfigurasi Apache.

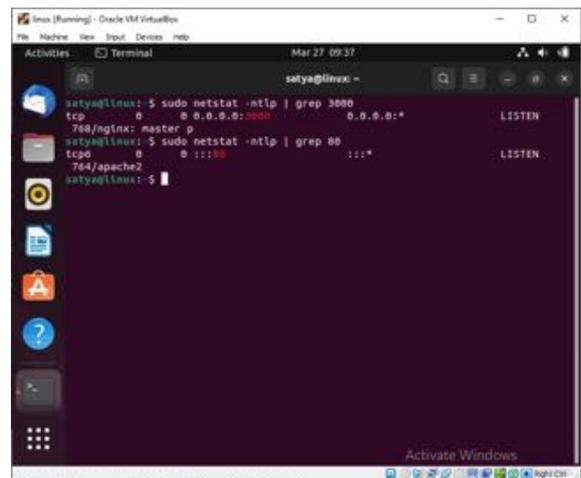
```
<Directory />
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
</Directory>
```

Dengan melakukan konfigurasi tersebut, Nginx akan menerima permintaan dari *web client* dan mengarahkan permintaan tersebut ke server *backend* Apache. Apache akan menangani permintaan tersebut dan mengirimkan respons ke Nginx yang kemudian diteruskan ke *web client*. Dalam skenario ini, pengguna dapat memanfaatkan keunggulan performa dan efisiensi Nginx untuk menangani permintaan dari *web client*, sementara Apache bertanggung jawab untuk menangani permintaan dari Nginx.

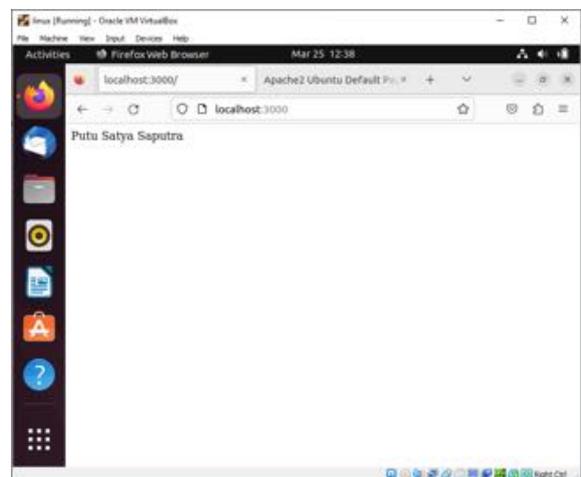
Selain itu, web server Apache dapat digunakan untuk menangani permintaan dinamis dan Nginx untuk menangani permintaan statis. Apache memiliki kemampuan untuk menangani permintaan dinamis seperti PHP, Perl, atau Python, sementara Nginx lebih efisien dalam menangani permintaan statis seperti file gambar, CSS, atau JavaScript. Dalam skenario ini, pengguna dapat mengkonfigurasi Apache untuk menangani permintaan dinamis dan Nginx untuk menangani permintaan statis. Dengan cara ini, pengguna dapat mengoptimalkan penggunaan masing-masing *web server* dan mempercepat waktu respon server.

Integrasi lainnya antar *web server* Apache dan Nginx dapat digunakan sebagai *load balancer*. *Load balancer* dapat digunakan untuk membagi beban trafik antara Apache dan Nginx, sehingga keduanya dapat bekerja secara bersama-sama untuk mengoptimalkan performa server. Dalam skenario ini, *load balancer* akan menerima permintaan dari *web client* dan membagi beban trafik ke server *backend* yang tersedia. Pengguna dapat menggunakan aplikasi *load balancer* seperti HAProxy atau memanfaatkan fitur *built-in* dari Nginx atau Apache.

Dengan memanfaatkan keunggulan masing-masing *web server* dan mengintegrasikannya dengan cara yang tepat, pengguna dapat meningkatkan performa dan efisiensi server secara signifikan. Namun, penting untuk diketahui bahwa konfigurasi *web server* sangat tergantung pada kebutuhan dan karakteristik aplikasi web yang akan dijalankan, sehingga perlu dilakukan evaluasi dan uji coba sebelum melakukan implementasi pada lingkungan produksi.



Gambar 8. Testing Server Nginx dan Apache pada Terminal



Gambar 9. Testing Server Nginx dan Apache pada Browser

Web server Nginx dan Apache dapat dijalankan secara bersamaan sehingga *developer* dapat memaksimalkan masing - masing keunggulan *web*

server tersebut. Namun sebelum dapat menjalankan kedua *web server* tersebut secara bersamaan harus menggunakan port yang berbeda. Gambar x menampilkan hasil testing menjalankan web server Nginx dan Apache secara bersamaan dengan port yang berbeda pada terminal Ubuntu. *Web server* Apache menggunakan port 80 sedangkan *web server* Nginx menggunakan port 3000.

4. KESIMPULAN

Secara umum web server Nginx dan Apache masing – masing memiliki keunggulan, kelemahan serta perbedaan fungsi. Nginx diketahui lebih cepat dan efisien dalam menangani permintaan dari banyak klien secara bersamaan dibandingkan dengan Apache. Apache memiliki konfigurasi yang lebih kompleks dibandingkan dengan Nginx, tetapi lebih mudah dipahami oleh pengguna baru karena struktur konfigurasi yang lebih mirip dengan bahasa pemrograman. Nginx memiliki konfigurasi yang lebih sederhana dan lebih mudah diatur dan dikelola, tetapi bisa menjadi lebih sulit dipahami oleh pengguna yang baru menggunakan platform web server.

Web server Apache memiliki lebih banyak modul dan plugin tambahan yang tersedia di internet, yang memungkinkan pengguna untuk menyesuaikan fungsionalitas server webnya. Nginx juga memiliki modul dan plugin tambahan yang tersedia di internet, tetapi tidak sebanyak Apache. Apache lebih dikenal dan banyak digunakan di seluruh dunia, sehingga lebih mudah untuk menemukan sumber daya dan dokumentasi tentang Apache. Nginx telah menjadi semakin populer dalam beberapa tahun terakhir, tetapi masih tidak sepopuler Apache, yang dapat membuatnya lebih sulit untuk menemukan sumber daya dan dokumentasi yang dibutuhkan. Web server Apache dan Nginx memiliki kelebihan dan kekurangan masing-masing. Jika performa yang cepat dan efisien adalah prioritas utama, maka Nginx bisa menjadi pilihan yang lebih baik. Namun, jika pengguna ingin lebih banyak modul dan dukungan, serta lebih mudah diatur dan dipahami, maka Apache bisa menjadi pilihan yang lebih baik. Pada akhirnya, keputusan untuk menggunakan Apache atau Nginx tergantung pada kebutuhan dan preferensi pengguna.

5. DAFTAR PUSTAKA

- A. M. Komaruddin, D. M. Sipitorini, and P. Rispian, "Load Balancing dengan Metode Round Robin Untuk Pembagian Beban Kerja Web Server," *Siliwangi*, vol. 5, no. 2, pp. 47–50, 2019.
- A. Tedyyana and O. Ghazali, "Teler Real-Time Http Intrusion Detection At Website With Nginx Web Server," *Int. J. Informatics Vis.*, vol. 5, no. 3, pp. 327–332, 2021.
- D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter

- pada Sistem Informasi Pertanian," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020.
- D. K. Hakim, D. Y. Yulianto, and A. Fauzan, "Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 85, 2019.
- D. W. L. Pamungkas and S. Rochimah, "Pengujian Aplikasi Web - Tinjauan Pustaka Sistematis," *J. IPTEK*, vol. 23, no. 1, pp. 17–24, 2019.
- M. F. Kurniawan, W. Setianto, and A. Ilyas, "Implementasi Module Google Pagespeed Pada Apache Web Server Untuk Meningkatkan Performa Aplikasi Web," *IC-Tech*, vol. 16, no. 2, pp. 47–51, 2021.
- M. Rafli, "Jurnal Pengujian Kinerja Load Balancing Web Server menggunakan Nginx Reverse Proxy Berbasis OS Centos 7," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 1824–1840, 2022.
- R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," *2020 Int. Work. Big Data Inf. Secur. IWBIS 2020*, no. December, pp. 85–90, 2020.
- Riswandi, Kasim, and M. F. Raharjo, "Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2," *J. Eng. Technol. Appl. Sci.*, vol. 2, no. 1, pp. 19–31, 2020.
- Y. W. Luthfi Muhammad, Data Mahendra, "Perbandingan Performa Reverse Proxy Caching Nginx dan Varnish Pada Web Server Apache," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 4, pp. 1457–1463, 2018.