PERBANDINGAN PERFORMA REVERSE PROXY DALAM MEMBAGI DOMAIN PADA LAYANAN SERVER WEB MENGGUNAKAN NGINX DAN APACHE DI VIRTUALBOX

Fadrul Arhan[1]

[1] *Program Studi Informatika, Fakultas Teknologi Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer El Rahma Yogyakarta*
e-mail: [1]fadrul.arhan@gmail.com,

Abstrak

Reverse proxy digunakan untuk memungkinkan satu alamat IP publik melayani beberapa domain atau situs web yang berbeda. Penelitian ini menggunakan Nginx sebagai reverse proxy dan melakukan uji coba pada server web yang memiliki dua domain. Penelitian ini bertujuan untuk menguji penggunaan reverse proxy dalam membagi domain pada layanan server web menggunakan Nginx dan apache serta membandingkan performa Nginx dan apache dalam membagi domain pada layanan server web. Hasil penelitian menunjukkan bahwa penggunaan reverse proxy dengan Nginx dapat membagi domain dengan efektif, meningkatkan kinerja server web, dan meningkatkan keamanan dengan menyembunyikan alamat IP asli server. Namun, penelitian juga menemukan beberapa masalah teknis yang perlu diperhatikan saat menggunakan reverse proxy. Oleh karena itu, penelitian ini menyimpulkan bahwa penggunaan reverse proxy dengan Nginx dapat membantu membagi domain pada layanan server web dengan efektif, namun perlu diperhatikan beberapa masalah teknis yang muncul dalam penggunaannya. Oleh karena itu, reverse proxy dalam penelitian ini adalah solusi yang digunakan untuk membagi domain pada layanan server web. Salah satu server web yang dapat berperan sebagai reverse proxy adalah nginx. Dalam konteks ini, penggunaan reverse proxy dengan Nginx memungkinkan pengguna untuk mengarahkan permintaan yang masuk ke beberapa domain atau alamat IP yang berbeda ke backend server yang sesuai.Secara keseluruhan, penelitian ini memberikan gambaran yang jelas tentang penggunaan reverse proxy dengan Nginx untuk membagi domain pada layanan server web. Hasil penelitian ini dapat membantu para pengembang web untuk meningkatkan kinerja dan keamanan layanan web mereka dengan memanfaatkan teknologi reverse proxy dengan Nginx.

Kata kunci *— Reverse proxy, Nginx, Domain, Server*

Abstrack

Reverse proxies are used to allow one public IP address to serve multiple different domains or websites. This research uses Nginx as a reverse proxy and tests it on a web server that has two domains. This study aims to test the use of reverse proxies in dividing domains on web server services using Nginx and Apache and to compare the performance of Nginx and Apache in dividing domains on web server services. The results show that using a reverse proxy with Nginx can divide domains effectively, improve web server performance, and improve security by hiding the server's real IP address. However, the research also found some technical issues to be aware of when using a reverse proxy. Therefore, this study concludes that using a reverse proxy with Nginx can help effectively divide domains on web server services, but it is necessary to pay attention to some technical problems that arise in its use. Therefore, the reverse proxy in this study is a solution used to divide domains on web server services. One web server that can act as a reverse proxy is nginx. In this context, the use of a reverse proxy with Nginx allows users to redirect incoming requests to multiple domains or different IP addresses to the appropriate backend server. Overall, this study provides a clear picture of the use of a reverse proxy with Nginx to divide domains on services. server network. The results of this research can help web developers improve the performance and security of their web services by utilizing reverse proxy technology with Nginx.

Keywords *— Reverse proxy, Nginx, Domain, Server*

1. PENDAHULUAN

Website yang lambat dapat menjadi masalah besar bagi pengguna, terutama dalam hal pengalaman pengguna dan tingkat konversi. Ada beberapa faktor yang dapat menyebabkan website menjadi lambat, salah satunya adalah penggunaan server web yang tidak efisien. Dalam penggunaan reverse proxy, NGINX dapat digunakan sebagai server proxy untuk memecah domain. Fungsi lain dari reverse proxy bagi web server adalah memisahkan request terhadap konten statis dan dinamis dimana request konten statis diproses oleh reverse proxy dan request terhadap konten dinamis akan diteruskan ke web server.Layanan web menjadi salah satu elemen penting dalam operasional perusahaan dan organisasi. Dalam menghadapi lalu lintas web yang tinggi dan kompleksitas infrastruktur, penggunaan reverse proxy telah menjadi solusi yang umum digunakan untuk membagi dan mengarahkan lalu lintas web antara beberapa server backend. Reverse proxy bertindak sebagai perantara antara klien dan server backend, meningkatkan keamanan, mempercepat waktu respons, dan memastikan ketersediaan layanan web.[1]

Dalam lingkungan server web yang kompleks, sering kali diperlukan penggunaan reverse proxy untuk membagi domain atau mengalihkan lalu lintas web ke server backend yang tepat. Dua solusi reverse proxy yang sangat populer adalah Nginx dan Apache. Nginx dikenal karena kinerja tinggi, kecilnya penggunaan sumber daya, dan kemampuan skala yang baik, sementara Apache dikenal karena fleksibilitas dan dukungan yang luas untuk modul tambahan.

Namun, Nginx dan Apache memiliki popularitas yang tinggi, performa dan keamanan keduanya dapat bervariasi. Performa yang optimal sangat penting dalam menghadapi lalu lintas yang tinggi dan memastikan pengalaman pengguna yang baik. Selain itu, keamanan juga menjadi faktor kunci dalam melindungi infrastruktur web dari serangan yang berpotensi merugikan.

Oleh karena itu, diperlukan penelitian yang komprehensif untuk membandingkan performa dan keamanan Nginx dan Apache sebagai reverse proxy dalam konteks membagi domain pada layanan server web di VirtualBox. Dengan pemahaman yang lebih baik tentang perbedaan kinerja dan keamanan antara kedua solusi ini, pengembang dan organisasi dapat membuat keputusan yang tepat dalam memilih reverse proxy yang sesuai dengan kebutuhan mereka.Melalui penelitian ini, dapat memberikan pemahaman yang lebih mendalam tentang kemampuan, kelebihan, dan kekurangan Nginx dan Apache sebagai reverse proxy dalam membagi domain pada layanan server web. Hasil penelitian ini akan memberikan kontribusi pada peningkatan performa dan keamanan infrastruktur web serta membantu pengembang dalam mengoptimalkan penggunaan solusi reverse proxy yang efektif dan efisien. Oleh karena itu, penelitian terkait penggunaan reverse proxy untuk memperbaiki performa website dengan memecah domain pada server NGINX menjadi sangat penting untuk meningkatkan performa dan keamanan website pada lingkungan IT modern.[6]

2. METODE PENELITIAN

a. Metode Pengumpulan Data

Untuk mendapatkan data yang benar dan hasil yang relevan, maka digunakan metode pengumpulan data sebagai berikut :

1) Studi Pustaka
   Pengambilan data yang bersumber dari buku-buku yang mendukung konsep teori yang berkaitan dengan reverse proxy menggunakan nginx.
2) Studi Literatur
   Pengumpulan data dengan mempelajari penelitian sebelumnya yang memiliki karakteristik sama dengan penelitian yang dilakukan, baik dari segi teknologi maupun pembahasan.

3) Studi Lapangan
Pengumpulan data dengan pengamatan langsung atau observasi terhadap objek yang diteliti secara cermat dan sistematik.

b. Metode Analisis Dan Pengembangan Sistem

Metodologi penelitian menggunakan SPDLC (Security Policy Development Life Cycle) yang memiliki tahapan sebagai berikut :

1) Identifikasi
2) Analisis
3) Desain
4) Implementasi
5) Audit
6) Evaluasi

c. Pengacuan Pustaka

Penelitian yang dilakukan oleh Naufal (2022), dengan tema AnalisisPerformansi Reverse proxy Caching NGINX Dan Varnish Pada Web server Apache dalam Menangani Client Request Menggunakan Ubuntu 20.04. Pada penelitian kali ini adalah reverse proxy caching NGINX dan varnish. Uji performa pada web server dilakukan dengan beberapa macam uji yaitu dengan cara mengirimkan beban 200, 2.000 dan 20.000 koneksi dengan 200 Conccurency yang diukur dengan apache benchmark Tool. Berdasarkan hasil pengujian reverse proxy caching NGINX memberikan hasil benchmarking lebih baik dari reverse proxy caching Varnish dengan parameter request per detik dengan hasil 1207 req/seconds (39,6% Lebih baik dibandingkan Varnish), Time per request sebesar 166,4 millisecond (28,4% Lebih baik dibandingkan Varnish), dan Transfer rate sebesar 13,15 MB/sec (38,4% Lebih baik dibandingkan Varnish). Time taken for test sebesar 16,65 seconds (28,3% Lebih baik dibandingkan Varnish). Berdasarkan hasil di atas dapat disimpulkan bahwa Apache web server yang sudah dikonfigurasikan dengan reverse proxy caching NGINX menunjukkan performa lebih baik dibandingkan dengan reverse proxy caching Varnish.[1]

Selanjutnya, penelitian yang dilakukan oleh Inas (2018) membuat Optimasi Web server NGINX Menggunakan Metode Reverse proxy Di PT. Beon Intermedia, Penelitian ini bertujuan untuk meningkatkan kinerja akses website yang lambat di PT. Beon Intermedia. Solusinya adalah dengan mengoptimalkan Web server NGINX menggunakan metode Reverse proxy. Penelitian ini menggunakan teknik reverse proxy yang disematkan bersamaan dengan system Cache oleh NGINX, gzip dan pagespeed. Website yang diuji adalah berbasis CMS Wordpress dengan total 4 nama domain. Hasil pengujian mencakup perbandingan 3 jenis konfigurasi terbaik dan data sebelum dan sesudah diimplementasikan. Alat pengujian yang digunakan adalah 2 situs website monitoring, yaitu Gtmetrix dan Pingdom. Indikator yang digunakan dalam pengujian meliputi Pagespeed Grade, Yslow Grade, Load Time, Total Page Size, dan Number of Requests. Hasil penelitian menunjukkan bahwa optimasi yang dilakukan dapat meningkatkan kecepatan akses website secara signifikan. Pengujian menggunakan 3 jenis konfigurasi menunjukkan bahwa website akan lebih cepat diakses jika nilai load time lebih kecil. Hasil pengujian di situs Gtmetrix menunjukkan bahwa konfigurasi pertama (ke-1) adalah yang terbaik dengan load time tercepat, yaitu 8,8 detik dibandingkan dengan 2 konfigurasi lainnya dengan selisih nilai 0,5 detik (9,3 detik) dan 0,4 detik (9,2 detik). Sedangkan pada situs Pingdom, konfigurasi terbaik adalah konfigurasi kedua (ke-2) yang menghasilkan load time 7,14 detik lebih cepat dibandingkan dengan 2 konfigurasi lainnya, dengan selisih 1,1 detik (8,23 detik) dan 1,2 detik (8,37 detik).[2]

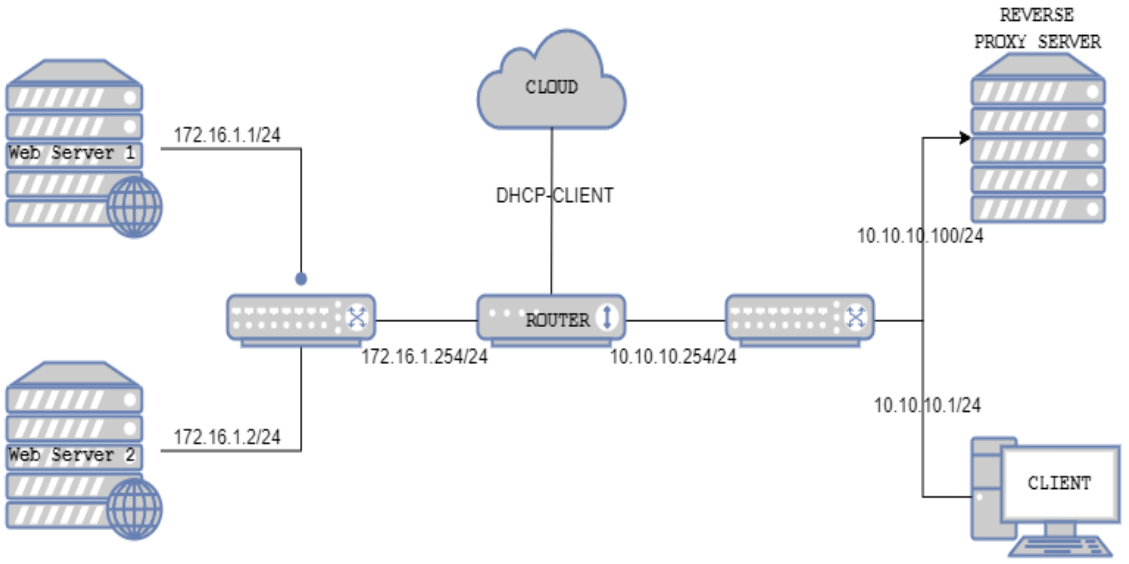
Seterusnya, penelitian yang dilakukan oleh Anindya (2020), tentang Optimasi apache web server menggunakan varnish web cache dan reverse proxy NGINX, Penelitian ini mengfokuskan kepada bagaimana cara optimasi kinerja web server menggunakan reverse proxy dan web cache

varnish. Pada penelitian ini web server di uji dengan 3 macam uji salah satunya dengan menggunakan tool apache benchmark dengan cara mengirimkan beban 100 request, 500 request dan 1000 request. Dari hasil pengujian yang dilakukan memberikan hasil benchmarking bahwa dari sisi respon time meningkat sebanyak 96%. Selanjutnya dilakukan pengujian dengan uji coba full traffic dengan 1000 request menunjukkan hasil respon time dari yang sebelumnya 31,201 ms menjadi 1,144 ms. [3]

3. HASIL DAN PEMBAHASAN

a. Desain Topologi Jaringan

Topologi jaringan dalam penggunaan reverse proxy untuk membagi domain pada layanan web server menggunakan Nginx berfungsi untuk memastikan bahwa permintaan dari klien (client) yang datang ke server dapat ditangani dengan efisien dan aman. Penggunaan topologi jaringan ini juga dapat meningkatkan keamanan layanan web, karena penggunaan reverse proxy dapat bertindak sebagai lapisan pertahanan (defense layer) yang dapat melindungi server backend dari serangan yang dilakukan oleh klien. Dengan menggunakan topologi jaringan yang tepat, penggunaan reverse proxy dengan Nginx dapat meningkatkan ketersediaan, kinerja, dan keamanan layanan web secara signifikan.



Gambar 1 Topologi Jaringan

b. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelititan perancangan external radius server sebagai berikut :

Tabel 1 Kebutuhan Perangkat Keras

Device	Spesifikasi
PC SERVER	*Processor Intel Core I3 2.9 Ghz*
	HDD Seagate 500 GB
	4 GB RAM
	2 LAN Ports

Device	Spesifikasi
Router Mikrotik RB 915G-2HnD	*CPU AR9344*
	RAM 128 *Mb*
	5 *LAN Ports*
	Main Storage 128 *MB*
Laptop	*Processor Intel Celeron 2.4 Ghz*
	2 GB RAM
	Wireless Atheros

c. Kebutuhan Perangkat Lunak

Perangkat keras yang dibutuhkan dalam penelititan perancangan external radius server sebagai berikut :

Tabel 2 Kebutuhan Perangkat Lunak

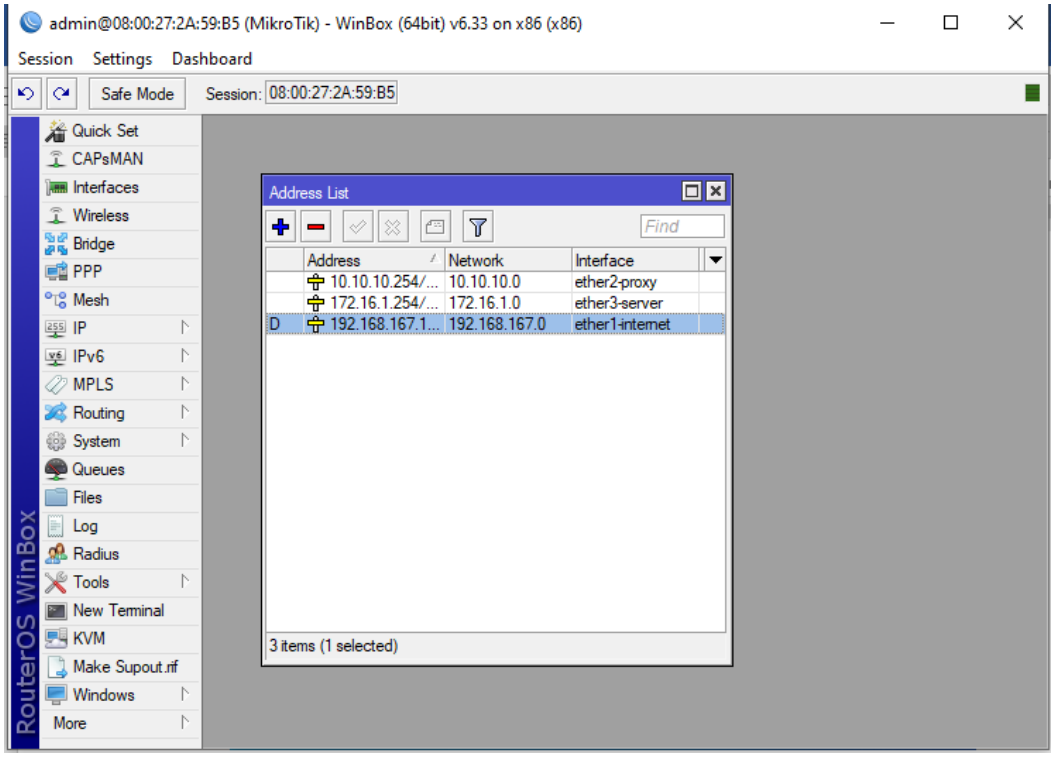
Device	**Versi**
Linux Debian	11.5.0-amd64
Mikrotik OS	Level 6
Winbox	3.6
Apache	2.4
NGINX	1.20
Virtualbox	7.1
Windows 10	Pro 64 bit
Putty	0.70

d. **Setup Router Mikrotik**

Langkah awal untuk mengkonfigurasi router mikrotik yaitu mengubah interface agar mudah dikenali, pada konfigurasi ini, ether1 digunakan untuk internet, ether2 digunakan untuk proxy dan ether3 digunakan untuk server.

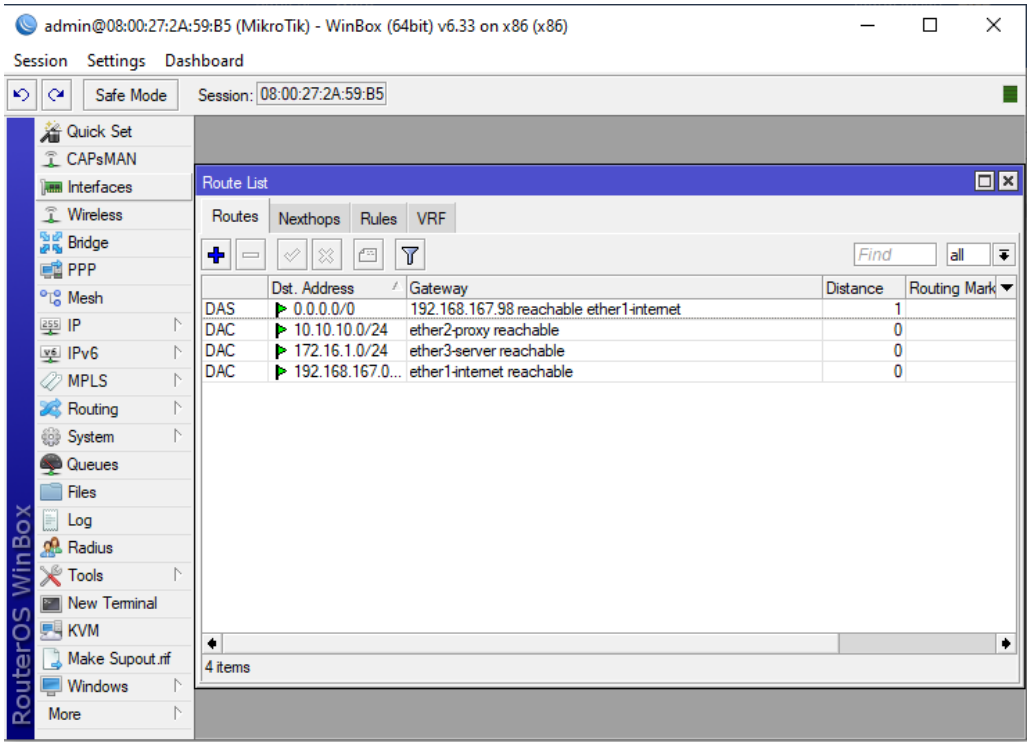
```
admin@router]> interface add interface=ether1 name=ether1-
internet

admin@router]> interface add interface=ether2 name=ether2-
server

admin@router]> interface add interface=ether1 name=ether3-
proxy
```

Selanjutnya setting ip address untuk masing masing interface yang nantinya mengarah ke internet, proxy dan juga server. Untuk ip address internet kita gunakan ip dhcp client yang diambil dari hotspot ataupun wifi yang terkonek ke pc real.



Gambar 1 setting ip address

Langkah selanjutnya adalah melakukan routing agar semua client dan server dapat terhubung, disini menggunakan routing dynamic dan static, routing dapat dilihat pada gambar di bawah



Gambar 2 setting routing pada router

e. **Setup *Reverse Proxy***

Pada penelitian kali ini akan mengonfigurasi penyiapan proxy terbalik yang sangat mendasar. Lingkungan aplikasi fiktif kita memiliki 2 instans signifikan: instans proxy balik dan instans target yang menjalankan sesuatu (mis. aplikasi) pada port 80. Kami ingin dapat mengakses aplikasi ini melalui proxy terbalik ketika kami menavigasi ke domain fiktif kami haha.com dan stmik.com. langkah pertama dalam konfigurasi reverse proxy adalah menginstall packet Nginx, pada penelitian kali ini menggunakan Nginx sebagai reverse proxy.

```
root@proxy:~# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6.1+deb11u3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@proxy:~#
```

Gambar 3 instalasi nginx pada server

selanjutnya kita akan mengonfigurasi file di /etc/Nginx/sites-available/defaultInstalasi package freeradius dengan perintah :

```
  GNU nano 5.4                    /etc/nginx/sites-available/default
server {
        server_name stmik.com www.stmik.com;
        listen 80;
        listen [::]:80;
        listen 443 ssl;
        ssl_certificate /etc/ssl/certs/haha.crt;
        ssl_certificate_key /etc/ssl/private/haha.key;

        location / {
                proxy_pass http://172.16.1.1:80;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $http_host;
                proxy_cache_bypass $http_upgrade;
      }
}

server {
        server_name haha.com www.haha.com;
        listen 80;
        listen [::]:80;
        listen 443 ssl;
        ssl_certificate /etc/ssl/certs/haha.crt;
        ssl_certificate_key /etc/ssl/private/haha.key;

        location / {
                proxy_pass http://172.16.1.2:80;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $http_host;
                proxy_cache_bypass $http_upgrade;
      }
}
```

Gambar 4 konfigurasi reverse proxy pada nginx

Ketika Nginx mem-proxy permintaan, ia secara otomatis mendefinisikan dua bidang header dalam permintaan yang diproxy dari klien, Host dan Connection dan menghapus header kosong. Host diatur ke variabel $proxy_host, dan Connection diatur untuk menutup.

Untuk menyesuaikan atau mengatur tajuk untuk koneksi yang diproxy, gunakan direktif proxy_set_header diikuti dengan header value. Dapat menemukan daftar semua Request Headers yang tersedia dan nilai yang diizinkan di sini. Jika ingin mencegah agar header tidak diteruskan ke server proxy, setel ke string kosong ""[4]. Melayani konten melalui HTTPS telah menjadi standar saat ini. Pada bagian ini, kami akan memberi contoh konfigurasi reverse proxy HTTPS Nginx termasuk parameter dan header proxy Nginx yang disarankan.

a) proxy_http_version 1.1 – Menentukan versi protokol HTTP untuk proxy, secara default ditetapkan ke 1.0. Untuk Websockets dan koneksi keepalive perlu menggunakan versi 1.1.
b) proxy_cache_bypass $http_upgrade – Menetapkan kondisi di mana respons tidak akan diambil dari cache.
c) Upgrade $http_upgrade and Connection "upgrade" – Field header ini diperlukan jika aplikasi menggunakan Websockets.
d) Host $host – Variabel $host dalam urutan prioritas berikut ini berisi: host name dari baris permintaan, atau host name dari bidang Host request header, atau nama server yang cocok dengan permintaan.
e) X-Real-IP $remote_addr – Meneruskan alamat IP jarak jauh pengunjung asli ke server yang diproxy.
f) X-Forwarded-For $proxy_add_x_forwarded_for – Daftar yang berisi alamat IP dari setiap server yang akan di proxy sebelum di kirim ke klien.
g) X-Forwarded-Proto $scheme – Ketika digunakan di dalam server block HTTPS, akan memblokir setiap respons dari HTTP dan akan merutekan ulang menjadi HTTPS.
h) X-Forwarded-Host $host – Menentukan host asli yang diminta oleh klien.
i) X-Forwarded-Port $server_port – Menentukan port asli yang diminta oleh klien.

Kemudian simpan saja file dan periksa Nginx untuk setiap kesalahan sintaks dengan perintah ini

```
root@proxy:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@proxy:~# _
```

Gambar 5 Proses pengecekan konfigurasi nginx

Selanjutnya, muat ulang layanan Nginx untuk menerapkan perubahan kami dengan melakukan perintah pada gambar 6 di bawah ini:

```
root@proxy:~# service nginx restart
root@proxy:~# _
```

Gambar 6 Restart server nginx

f. **Uji *Reverse Proxy***

Pengujian dilakukan dengan menggunakan aplikasi apache bencmarking. Aplikasi tersebut memberikan koneksi sesuai dengan yang di inginkan ke web server. Koneksi ini dapat dianalogikan jumlah pengunjung yang mengakses web server. Penelitian ini menggunakan variable pengujian berupa concurrency, dan koneksi. Variabel Concurrency hanya sebanyak 100,500 dan 1000, berbeda pada masing masing web server disebabkan karena keterbatasan dari perangkat yang ada, oleh karena itu angka yang ada disesuaikan dengan kemampuan dari perangkat/pc, pada koneksi yang dikirimkan secara bersamaan menggunakan Apache Benchmark tools, angka yang digunakan cukup besar agar dapat menghasilkan perbedaan pada hasil pengetesan. Dalam pengujian juga diperlukannya koneksi internet guna untuk melakukan test pada apache benchmarking tools. Pengujian dilakukan sebanyak 10 kali percobaan dan akan diambil rata rata dari total percobaan yang dilakukan, agar dapat mendapatkan nilai yang maksimal. Berdasarkan Tabel 5.1, berikut deskripsi skenario pengujian yang dilakukan.[5]

a) Jumlah request yang dikirimkan dan Perintah pada saat melakukan Benchmark menggunakan Apache Benchmark Tools

1) ab -n 100 -c 100 http://[Address]/
2) ab -n 500 -c 500 http://[Address]/
3) ab -n 1000 -c 1000 http://[Address]/

b) Berikut URL atau halaman web yang akan digunakan untuk melakukan pengujian pada web server.

1) http://172.16.1.1
2) http://172.16.1.2

Pengujian web server, virtual user akan mengakses beberapa URL. Pada URL dengan alamat http://172.16.1.1 dan http://172.16.1.2 merupakan URL dengan web server yang dikonfigurasikan dengan reverse proxy Nginx. Masing-masing URL akan diuji dengan Apache Benchmark Tool dengan request yang sudah ditentukan, Setelah itu akan dipantau dan diukur performansi Server dengan melihat parameter:

ime Taken for Tests

a) Jumlah request
b) Time per Request (ms)
c) Transfer Rate
d) Request Per Second

Berdasarkan scenario pengujian yang dibuat, akan dilakukan pengujian dengan menggunakan Apache Benchmark tools. Untuk mengukur performansi web server menggunakan Apache Benchmarking Tool dari sisi client dapat menggunakan perintah :Untuk melakukan uji coba harus memastikan terlebih dahulu *internet* sudah benar sesuai dengan topologi yang dibangun, berikut adalah informasi *internet* dari *client* :

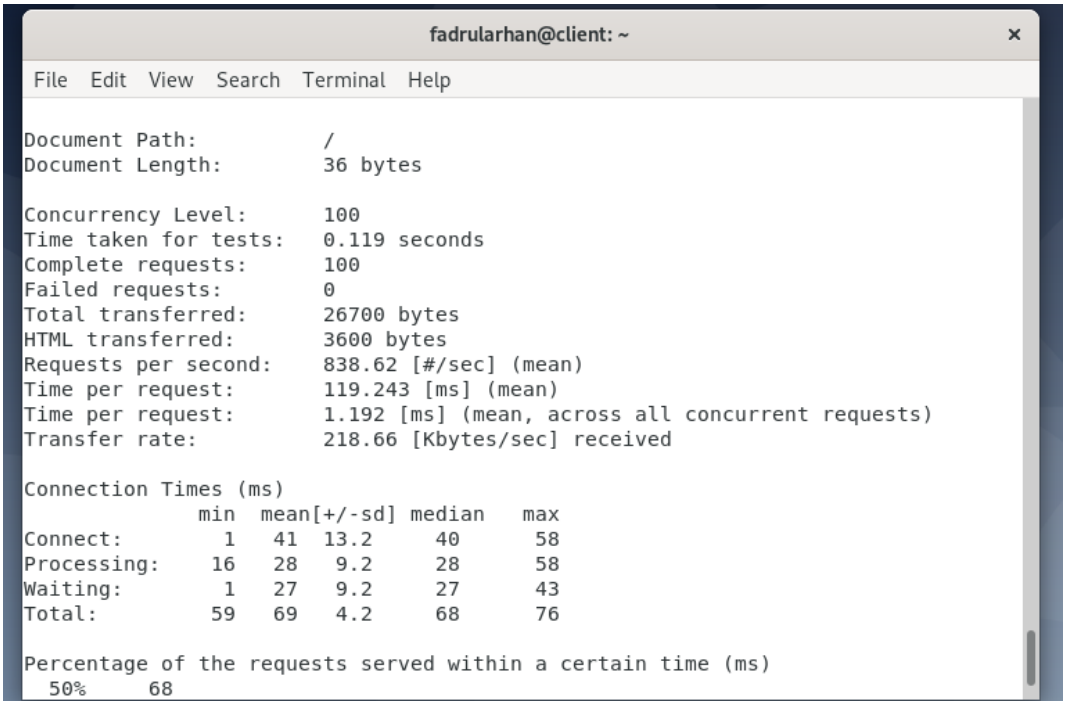
ab -n 200 -c 20 http://[Address]/

Berikut merupakan parameter yang digunakan oleh Apache Benchmarking tools:

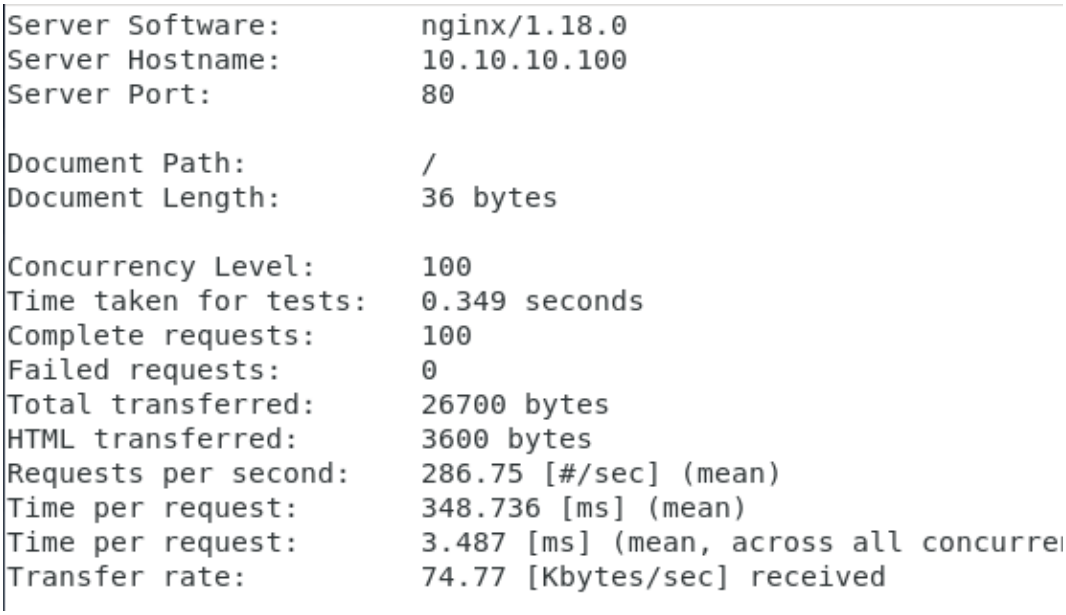
a) Parameter n adalah jumlah koneksi yang dibuat ke Server tujuan, dengan contoh diatas berarti koneksi yang dibuat adalah 200 koneksi.

b) Parameter c adalah jumlah request concurrent (bersama) yang dibuat, dengan contoh diatas berarti jumlah request yang dibuat adalah 20 request dalam satu waktu.

c) Parameter terakhir adalah Address. Address dapat berupa alamat IP atau halaman yang akan diproses oleh web server di benchmark.



Gambar 7 Apache benchmarking tools



Gambar 8 Hasil pengujian reverse proxy

Gambar 8 adalah hasil dari pengujian sesudah dilakukan nya konfigurasi reverse proxy, dari gambar di atas bisa dilihat transfer rate dan request nya lebih besar sesudah melakukan konfigurasi reverse proxy.

Selanjutnya, akan diuji dengan variabel pada tabel di atas, dan beberapa point yang akan diuji yaitu complete request, request periode per second dan transfer rate, nanti hasil nya akan di

bandingkan sebelum dan sesudah menggunakan reverse proxy, karena keterbatasan pada perangkat yang ada, maka concurrency yang di terapkan hanya sampai 1000.

Tabel 3 Hasil pengujian reverse proxy nginx

Concurrency	*Complete Request*	*Request Persecond/s*	*Transfer rate kb/s*
100	100	286.75/sec	178.21
500	500	608.26/sec	200.60
1000	1000	608.63/sec	117.83

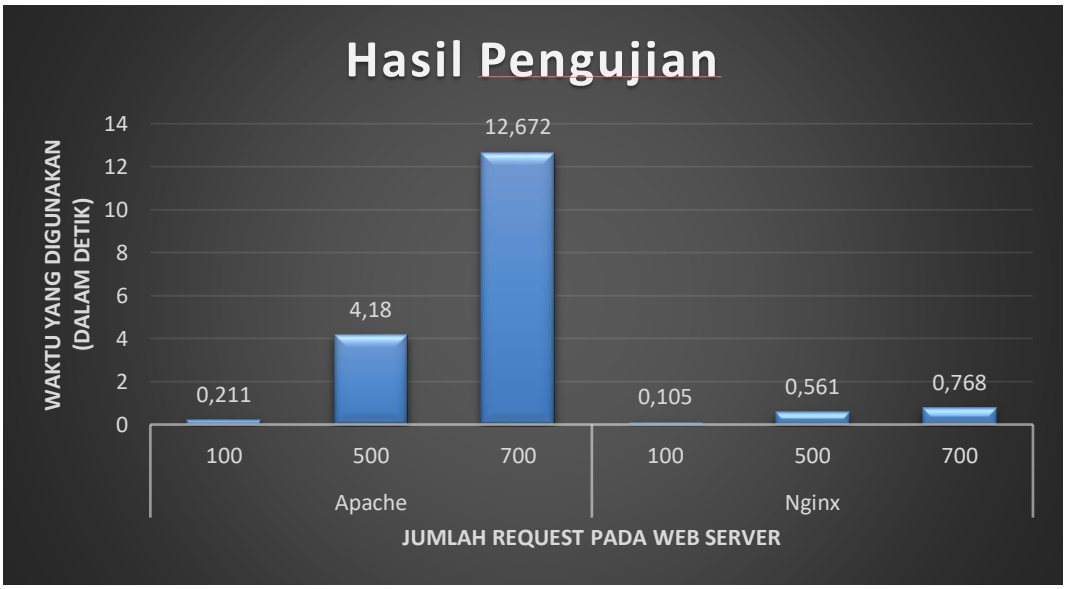
Berdasarkan Tabel 3 di atas terlihat perbedaan sebelum dan sesudah konfigurasi dimana terlihat sebelum konfigurasi request/sec rata-rata 150kb/sec sedangkan hasil sesudah konfigurasi rata rata sebesar 492kb/sec dan transfer rate sesudah konfigurasi menjadi 165kb/s. Hal ini berarti kinerja web server yang baik adalah setelah dikonfigurasi reverse proxy.

Selanjutnya akan dilakukan pengujian untuk waktu yang digunakan dalam mengakses web server, dapat dilihat pada tabel di bawah.

Tabel 4 Hasil pengujian waktu yang digunakan pada web server

Web Server	*Complete Request*	**Waktu yang digunakan**
Apache	100	0,211 Seconds
	500	4,180 Seconds
	1000	12,672 Seconds
Nginx	100	0,105 Seconds
	500	0,561 Seconds
	1000	0,768 Seconds

Dapat dilihat pada tabel di atas setelah dilakukan reverse proxy nginx bisa meningkatkan waktu respon terhadap client lebih cepat dibandingkan dengan apache dengan hasil 1000 request client nginx hanya membutuhkan waktu 0,768 detik, Hasil tersebut dapat dilihat pada gambar grapich di bawah



Gambar 9 Grafik Hasil pengujian

4. KESIMPULAN

Reverse proxy dengan Nginx berhasil membagi domain untuk beberapa server backend dan meningkatkan performa server web dengan meminimalkan downtime dan waktu respon yang lebih cepat dan juga reverse proxy nginx memberikan performa yang yang lebih baik daripada apache di setiap tahap pengujian. Sedangkan web server dengan reverse proxy nginx masih dapat menggunguli performa web server Apache tanpa reverse proxy pada beban 1000 request per menit.

5. SARAN

Berdasarkan hasil dari penelitian dan pengujian yang sudah dilakukan, maka dapat dikembangkan lagi dengan melakukan perbandingan dengan solusi reverse proxy lainnya, seperti HAProxy, untuk memberikan pembandingan yang lebih lengkap dan mendalam.

DAFTAR PUSTAKA

[1]Alharbi, F., Zhou, Y., Qian, F., Qian, Z., & Abu-Ghazaleh, N. (2022). DNS Poisoning of Operating System Caches: Attacks and Mitigations. IEEE Transactions on Dependable and Secure Computing, 19(4). https://doi.org/10.1109/TDSC.2022.3142331

[2]Chandra, A. Y. (2019). Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. Jurnal Sistem Dan Informatika (JSI), 14(1), 48–56. https://doi.org/10.30864/jsi.v14i1.248

[3]Dewi Wachyuni, & Joko Christian Candra. (2022). Implementasi Web Service Untuk Manajemen Akun Linux Pada Server Badan Litbang Perhubungan Menggunakan Php. KRESNA: Jurnal Riset Dan Pengabdian Masyarakat, 2(2). https://doi.org/10.36080/jk.v2i2.49

[4]Dimas Erlangga, M., & Prihanto, A. (2022). Analisis Reliabilitas Multiserver Menggunakan Load Balancing Dengan Metode Denial Of Service. Journal of Informatics and Computer Science, 03.

[5]Hasibuan, M., & Elhanafi, A. M. (2022). Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box. Sudo Jurnal Teknik Informatika, 1(4). https://doi.org/10.56211/sudo.v1i4.160

[6]Luthfi Muhammad, Data Mahendra, Y. W. (2018). Perbandingan performa reverse proxy caching Nginx dan varnish pada web server apache. Pengembangan Teknologi Informasi Dan Ilmu Komputer, 2(4).