Analisis Perbandingan Performa Web Server Apache, Nginx, Dan Litespeed Studi Kasus Implementasi Di VPS

Ganda Putra Perdana [1], Agus Prihanto[2]
[1,2] Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya
[1]ganda.18031@mhs.unesa.ac.id
[2]agusprihanto@unesa.ac.id

Abstrak - Performa web server memegang peranan penting dalam menjamin kecepatan akses dan kenyamanan pengguna dalam era digital yang didominasi oleh aplikasi berbasis web. Penelitian ini bertujuan untuk membandingkan performa tiga web server populer—Apache, Nginx, dan LiteSpeed—dalam lingkungan Virtual Private Server (VPS). Pengujian dilakukan menggunakan situs web sederhana berbasis HTML, CSS, dan JavaScript, tanpa platform CMS, pada VPS dengan spesifikasi identik dan sistem operasi Linux. Parameter yang diuji mencakup response time, throughput, error rate, latensi, load time, serta penggunaan sumber daya CPU dan RAM. Pengujian dilakukan menggunakan alat seperti Apache Benchmark (ab) dan JMeter untuk mensimulasikan berbagai tingkat beban lalu lintas. Hasil penelitian ini diharapkan dapat memberikan panduan yang berguna bagi pengembang web, administrator sistem, dan penyedia layanan hosting dalam memilih web server yang paling optimal untuk kebutuhan mereka, berdasarkan efisiensi sumber daya dan kinerja dalam kondisi lalu lintas tinggi.

Kata kunci - Apache, Nginx, LiteSpeed, Virtual Private Server (VPS), response time, throughput, latensi, Apache Benchmark, JMeter.

I. PENDAHULUAN

Penelitian sebelumnya membahas berbagai perbandingan performa web server menggunakan berbagai metode pengujian. Cahyadi et al. (2023) melakukan analisis perbandingan kinerja web server Nginx dan LiteSpeed menggunakan Httperf pada sistem operasi Debian. Hasilnya menunjukkan bahwa Nginx memiliki kinerja lebih baik dengan rata-rata waktu respons 0,23 ms pada website berkapasitas 4 MB, unggul dalam throughput, koneksi, dan balasan yang memberikan pengalaman pengguna yang optimal. Prihaditama dan Agung (2024) membandingkan performa Apache dan Lighttpd menggunakan metode stresstest pada VPS, di mana Apache unggul dalam throughput dan response time, tetapi kurang optimal dalam menangani aset gambar pada WordPress, sementara Lighttpd lebih baik dalam penerimaan dan pengiriman data meski memiliki tingkat error lebih tinggi.

Alfatah (2022) menunjukkan bahwa pengujian performa web server dengan Httperf efektif untuk mengevaluasi parameter seperti request, throughput, connection, dan reply, baik pada website statis maupun dinamis dengan berbagai beban. Chandra (2019) membandingkan Apache dan Nginx menggunakan Apache Bench, menemukan bahwa Nginx memiliki waktu penyelesaian request lebih cepat dibandingkan Apache, khususnya dengan jumlah request besar hingga 1.000.000. Penelitian ini juga menyarankan pengembangan lebih lanjut untuk menguji aspek concurrency level dan jumlah URL yang diakses bersamaan.

Selain itu, Maharani et al. (2023) mengkaji kualitas perangkat lunak pada website perguruan tinggi menggunakan metode Webqual, Apache J-Meter, dan Web Server Stress Tool. Penelitian mereka menemukan bahwa Universitas Teknokrat Indonesia memiliki tingkat error 0,744%, memerlukan peningkatan kapasitas server. Universitas Malahayati mencatat waktu akses tertinggi, 280.675,882 ms, yang mengindikasikan perlunya optimasi. Dalam aspek throughput dan received, Institut Teknologi Sumatera, Universitas Teknokrat Indonesia, dan Universitas Lampung menunjukkan performa terbaik, sedangkan website UIN memiliki waktu loading rata-rata terlama, 710,376 ms. Penelitian ini memberikan wawasan mengenai performa server web dan kualitas layanan yang dapat menjadi dasar dalam optimasi server web.

Penelitian-penelitian ini memberikan landasan penting dalam analisis perbandingan performa web server seperti Apache, Nginx, dan LiteSpeed, khususnya pada VPS, serta menyoroti berbagai aspek yang memengaruhi performa server web dalam implementasi dan optimasi.

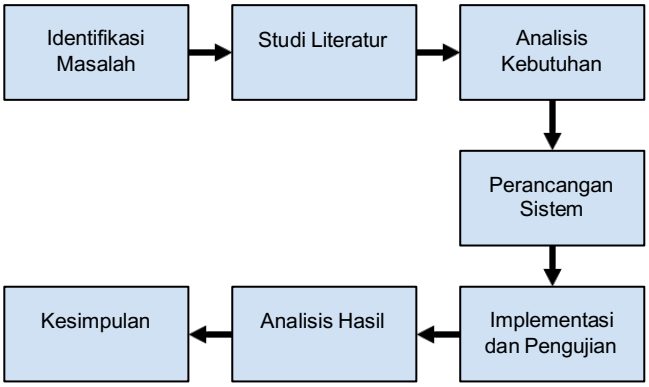
II. METODOLOGI PENELITIAN

Metode penelitian adalah serangkaian prosedur atau cara yang digunakan untuk mengumpulkan, menganalisis, dan menginterpretasikan data atau informasi guna menjawab pertanyaan penelitian atau menguji hipotesis. Tujuan utama

metode penelitian adalah menghasilkan pengetahuan yang valid, akurat, dan dapat dipercaya dalam bidang yang sedang diteliti.

Metodologi eksperimental design digunakan untuk analisis perbandingan performa web server Apache, Nginx, dan LiteSpeed yang diimplementasikan pada VPS. Eksperimen dilakukan dengan menginstal masing-masing web server pada VPS yang memiliki spesifikasi perangkat keras dan konfigurasi sistem operasi yang serupa, memastikan kesetaraan lingkungan pengujian. Setiap web server dikonfigurasi untuk menjalankan situs web sederhana berbasis HTML, CSS, atau JavaScript.

Pengujian performa dilakukan menggunakan alat seperti Apache Benchmark (ab) atau JMeter untuk mengevaluasi parameter teknis seperti waktu respons, throughput, latensi, dan penggunaan sumber daya (CPU, dan RAM). Eksperimen dilakukan dengan berbagai skenario beban, seperti jumlah permintaan yang meningkat secara bertahap atau simulasi lalu lintas tinggi, untuk mengamati kemampuan setiap web server dalam menangani peningkatan jumlah pengguna.



Gambar 1 Metode Eksperimental

Berikut merupakan alur tahapan yang dilakukan dalam penelitian ini. Tahapan penelitian yang dilakukan mencakup beberapa langkah penting, yang akan dijelaskan sebagai berikut:

A. Identifikasi Masalah

Langkah ini bertujuan untuk mengenali permasalahan utama, yaitu menentukan web server mana yang memiliki performa terbaik dalam menangani berbagai skenario beban di lingkungan VPS. Fokus masalah diarahkan pada parameter seperti waktu respons, throughput, latensi, dan penggunaan sumber daya.

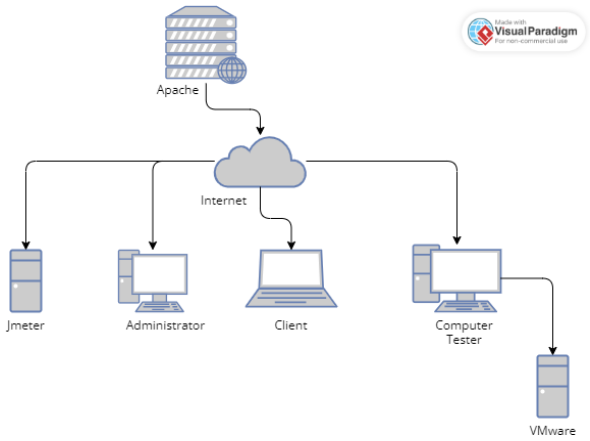
B. Studi Literatur

Mengumpulkan informasi dari berbagai sumber seperti jurnal, buku, dan dokumentasi resmi untuk memahami karakteristik teknis dan keunggulan masing-masing web server (Apache, Nginx, dan LiteSpeed). Selain itu, studi literatur juga mencakup alat pengujian performa seperti Apache Benchmark (ab) atau JMeter, serta metode statistik untuk analisis data.Analisis Kebutuhan

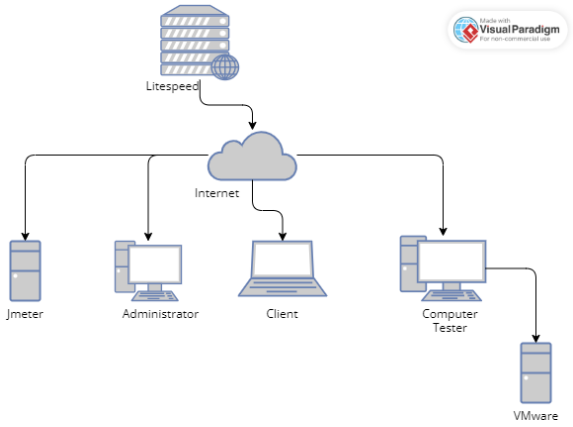
Penelitian ini memerlukan beberapa elemen penting untuk memastikan eksperimen dapat dilakukan dengan baik. Kebutuhan perangkat keras termasuk penyediaan VPS dengan spesifikasi yang cukup (misalnya CPU minimal 2 vCore, RAM 4GB, dan SSD 50GB). Kebutuhan perangkat lunak meliputi sistem operasi berbasis Linux (Debian/Ubuntu), Webmin, serta alat pengujian DNS seperti dnsperf dan queryperf. Alat pemantau kinerja (seperti htop, netstat, atau iftop) juga akan digunakan untuk mengamati penggunaan sumber daya selama pengujian. Semua alat ini akan digunakan untuk mengukur metrik seperti waktu respon DNS, throughput, dan penggunaan CPU/RAM selama eksperimen.

C. Perancangan Sistem

Desain sistem pengujian web server ini melibatkan penggunaan server VPS sebagai pusat pengujian, tempat di mana tiga web server—Apache, Nginx, dan LiteSpeed—dijalankan pada lingkungan Linux dengan spesifikasi identik. Masing-masing server dikonfigurasi menggunakan pengaturan dasar untuk mengoptimalkan performa. Pengujian dilakukan melalui node pengguna (client) dan alat uji JMeter untuk mensimulasikan permintaan secara simultan. Komponen utama dalam topologi ini meliputi web server yang diuji, JMeter sebagai alat uji beban, administrator pengelola sistem, client sebagai simulasi pengguna, komputer penguji, VMWare untuk simulasi latensi, serta koneksi internet sebagai penghubung seluruh elemen. Topologi ini dirancang untuk mencerminkan kondisi pengujian nyata, dengan fokus pada evaluasi performa, stabilitas, dan skalabilitas web server.



Gambar 1 Perancangan Sistem Apache



Gambar 2 Perancangan Sistem

D. Implementasi dan Pengujian

Bertujuan untuk merancang eksperimen yang dapat menghasilkan data yang valid dan terukur dalam perbandingan performa antara web server Apache, Nginx, dan LiteSpeed. Proses pengujian ini melibatkan beberapa langkah mulai dari persiapan lingkungan, konfigurasi server, hingga pengumpulan dan analisis data. data yang terkumpul akan dianalisis menggunakan teknik statistik untuk mengevaluasi performa setiap web server. Perbandingan dilakukan berdasarkan tabel berikut

No.	Parameter	Tujuan	Parameter	Tools
1.	Response Time	Mengukur kecepatan server dalam memproses permintaan HTTP.	Rata-rata waktu respon (ms)	Jmeter
2.	Throughput	Mengukur jumlah permintaan yang dapat diproses per detik oleh server.	Jumlah permintaan per detik (requests/second)	JMeter
3.	Error Rate	Mengukur persentase permintaan yang gagal dibandingkan dengan total permintaan.	(Jumlah error / Total request) * 100%	JMeter
4.	Latensi	Mengukur waktu yang dibutuhkan untuk pengiriman data antara server dan klien.	Waktu latensi (ms)	- Ping - curl dengan opsi -w untuk mengukur waktu koneksi
5.	Penggunaan Sumber Daya	Mengevaluasi efisiensi penggunaan CPU dan RAM oleh setiap web server.	- Persentase penggunaan CPU (%) - Persentase penggunaan RAM (%)	- htop - top - vmstat
6	Load Time	Pengujian Load Time (atau waktu muat) adalah	= Presentasi (Waktu) Load	- jmeter
		mengukur kecepatan "pengiriman" konten dari server ke browser pengguna.		

1. Struktur Skenario Pengujian

Pengujian ini dilakukan untuk mengukur Response Time dari tiga web server berbeda, yaitu Apache, Nginx, dan LiteSpeed. Response Time adalah waktu yang dibutuhkan server untuk merespons permintaan HTTP dari klien. Pengujian ini dilakukan menggunakan Apache JMeter dengan tiga skenario berbeda:

1.1. Baseline Load Test

Simulasi beban normal dengan 50 pengguna aktif Konfigurasi: Threads: 50, Ramp-up: 10s, Duration: 300s Tujuan: Menguji performa server dalam kondisi beban normal (pengguna harian).

1.2. Spike Test

Simulasi lonjakan mendadak dalam jumlah pengguna hingga 100 pengguna Konfigurasi: Threads: 100, Ramp-up: 5s, Duration: 300s Tujuan: Mengukur kemampuan server menangani lonjakan lalu lintas mendadak.

1.3. Stress Test

Menguji batas maksimal server dengan 200 pengguna aktifKonfigurasi: Threads: 200, Ramp-up: 20s, Duration: 300s Tujuan: Mengetahui batas maksimum server sebelum mulai mengalami kegagalan atau degradasi performa. Ketiga skenario pengujian ini dilakukan selama 300 detik (5 menit) sesuai dengan jumlah pengguna (threads) yang telah ditentukan. Ketiga skenario pengujian ini dilakukan selama 300 detik (5 menit) sesuai dengan jumlah pengguna (threads) yang telah ditentukan.

2. Skema Pengujian

Skema pengujian ini dirancang untuk mengukur Response Time, Throughput, Error Rate, Latensi, dan Penggunaan Sumber Daya dari masing-masing web server.

2.1. Pengujian Response Time

Hasil dibandingkan dalam bentuk grafik garis untuk menunjukkan perubahan response time seiring peningkatan beban.Pengujian Throughput

2.2. Pengujian Error Rate

Hasil dibandingkan dalam bentuk grafik batang untuk memperlihatkan jumlah permintaan per detik yang diproses oleh masing-masing web server.

2.3. Pengujian Latensi

Hasil dibandingkan dalam bentuk grafik batang untuk menunjukkan stabilitas server dalam menangani permintaan dengan berbagai tingkat beban.

2.4. Pengujian Penggunaan Sumber Daya

Pengujian dilakukan sebanyak 4 kondisi yaitu idle, baseline load testing, spike testing, stress testing

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan dalam penelitian ini merupakan bagian yang menjelaskan data pengujian serta analisis mendalam terhadap performa web server Apache, Nginx, dan LiteSpeed yang diimplementasikan di VPS. Hasil penelitian mencakup data yang diperoleh dari serangkaian pengujian dengan parameter seperti response time, throughput, latency, dan penggunaan sumber daya.

A. Hasil Penelitian

1. Instalasi dan konfigurasi Webserver Apache

Pada tahap ini, dilakukan instalasi dan konfigurasi web server Apache di lingkungan VPS sebagai bagian dari pengujian performa. Proses instalasi dilakukan menggunakan perintah bawaan dari sistem operasi berbasis Ubuntu dengan menjalankan. Update repositori dan instalasi dependensi

```
sudo apt install -y apache2
```

Gambar 4 Perancangan Sistem

Setelah konfigurasi selesai, web server diuji untuk memastikan bahwa layanan berjalan dengan baik menggunakan perintah:

```
sudo systemctl status apache2
```

Gambar 5 Status Apache

Kemudian, dilakukan uji coba dengan mengakses halaman default Apache melalui browser atau menggunakan perintah curl:

```
curl -I http://ip_Server
```

Gambar 6 Check Server Apache

2. Konfigurasi webserver

Melakukan Update Sistem Operasi Langkah awal adalah memperbarui sistem agar paket-paket yang digunakan berada dalam versi terbaru. Jalankan perintah:

```
sudo apt update -y
```

Gambar 7 Update system

3. Membuat Direktori Web

Buatlah direktori tempat file website akan disimpan. Dalam kasus ini, direktori yang digunakan adalah /var/www/instagram-feed:

4. Membuat File Index.html

```
sudo nano /var/www/instagram-feed/index.html
```

Gambar 8 Membuat file index apache

5. Mengatur Hak Akses File

Agar Apache dapat mengakses dan menyajikan file dengan benar, atur kepemilikan file dan izinnya:

```
sudo chown -R www-data:www-data /var/www/instagram-feed
sudo chmod -R 755 /var/www/instagram-feed
```

Gambar 8 Mengatur hak akses apache

6. Membuat Konfigurasi Virtual Host

Untuk menambahkan situs baru pada Apache, perlu dibuat file konfigurasi virtual host. Buat file baru:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/instagram-feed
    <Directory /var/www/instagram-feed>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/instagram_feed_error.log
    CustomLog ${APACHE_LOG_DIR}/instagram_feed_access.log combined
</VirtualHost>
```

Gambar 9 Mengatur virtual host apache

7. Mengaktifkan Virtual Host Baru

Aktifkan situs baru dengan perintah:

```
sudo a2ensite instagram-feed.conf
sudo systemctl restart apache2
```

Gambar 10 Mengaktifkan virtual host apache

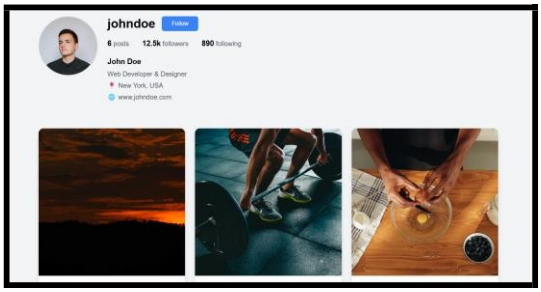
8. Merestart Apache

Setelah konfigurasi selesai, restart layanan Apache agar perubahan diterapkan:

```
sudo systemctl restart apache2
```

Gambar 11Merestart apache

Akses website melalui ip.



Gambar 12 Homepage Apache

B. Instalasi dan konfigurasi webserver Nginx

1. Instalasi webserver Nginx

Setelah menyelesaikan instalasi Apache, langkah berikutnya adalah menginstal dan mengonfigurasi web server Nginx di lingkungan VPS. Instalasi dilakukan menggunakan perintah bawaan sistem operasi berbasis Ubuntu dengan menjalankan perintah berikut.

```
sudo apt install -y nginx
```

Gambar 13 Install nginx

Setelah proses instalasi selesai, Nginx secara otomatis berjalan sebagai layanan. Untuk memastikan bahwa web server aktif dan berjalan dengan baik, digunakan perintah berikut.

```
sudo systemctl status nginx
```

Gambar 14 Status nginx

Jika layanan belum berjalan, dapat dimulai dengan perintah.

```
sudo systemctl restart nginx
```

Gambar 15 Resart nginx

Selanjutnya, dilakukan pengujian awal dengan mengakses alamat IP VPS melalui browser atau menggunakan perintah curl untuk memastikan halaman default Nginx dapat diakses.

```
curl -I http://ip_Server
```

Gambar 16 Check server nginx

2. Konfigurasi Websserver

Langkah pertama adalah memperbarui sistem untuk memastikan seluruh paket berada pada versi terbaru:

3. Membuat Direktori Web

Buat direktori untuk menyimpan file website. Dalam konfigurasi ini digunakan direktori /var/www/instagram-feed:

4. Membuat File index.html

Selanjutnya, buat file index.html yang akan ditampilkan sebagai halaman utama. File ini berisi konten HTML statis bergaya Instagram dan menggunakan Tailwind CSS. Buat file tersebut dengan:

5. Mengajar Hak Akses Direktori

Agar Nginx dapat mengakses file dengan benar, atur kepemilikan dan izin file:

6. Membuat Konfigurasi Virtual Host Nginx

Konfigurasi virtual host dibutuhkan agar Nginx dapat menyajikan website dari direktori yang telah dibuat. Buat file konfigurasi baru:

```
server {
    listen 80;
    server_name [IP_SERVER];

    root /var/www/instagram-feed;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    error_log /var/log/nginx/instagram_feed_error.log;
    access_log /var/log/nginx/instagram_feed_access.log;
}
```

Gambar 17 Membuat virtual host nginx

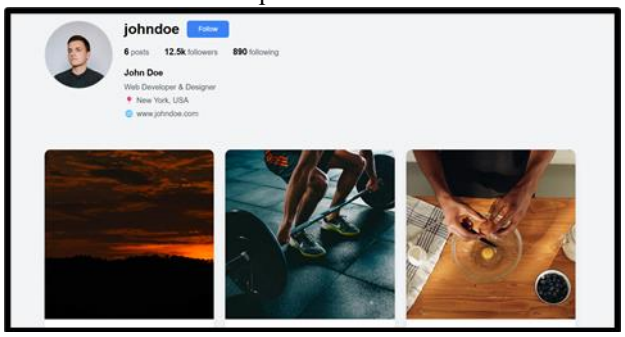
7. Mengaktifkan Konfigurasi Virtual Host

Untuk mengaktifkan konfigurasi baru, buat symbolic link ke direktori sites-enabled:

8. Merestart Layanan Nginx

Agar perubahan konfigurasi diterapkan, restart layanan Nginx:

Akses Website melalui ip.



Gambar 18 Homepage Litespeed

C. Instalasi dan konfigurasi webserver Litespeed

1. Instalasi webserver Litespeed

Setelah melakukan instalasi Apache dan Nginx, tahap berikutnya adalah menginstal dan mengonfigurasi web server LiteSpeed di lingkungan VPS. LiteSpeed memiliki dua versi utama, yaitu OpenLiteSpeed (versi open-source) dan LiteSpeed Enterprise (versi berbayar). Dalam penelitian ini,

digunakan OpenLiteSpeed karena bersifat gratis dan mendukung banyak fitur optimasi. Menambahkan Repository LiteSpeed untuk sistem berbasis Debian atau Ubuntu, pertama-tama tambahkan repository LiteSpeed.

Menginstal OpenLiteSpeed setelah menambahkan repository, jalankan perintah berikut untuk menginstal OpenLiteSpeed.

Memastikan LiteSpeed Berjalan setelah instalasi selesai, jalankan dan cek status layanan LiteSpeed dengan perintah.

Mengatur Kata Sandi Admin LiteSpeed WebAdmin LiteSpeed memiliki panel WebAdmin yang dapat diakses melalui browser. Untuk mengatur kata sandi admin, jalankan perintah berikut.

Mengakses Panel Admin LiteSpeed setelah konfigurasi awal selesai, WebAdmin LiteSpeed dapat diakses melalui browser dengan membuka.

Menguji Instalasi LiteSpeed untuk memastikan LiteSpeed berjalan dengan baik, akses halaman default menggunakan browser atau gunakan perintah curl.

2. Konfigurasi webserver Buat direktori untuk web Instagram Feed

Direktori ini digunakan sebagai tempat menyimpan file website (seperti file HTML, CSS, gambar, video). Lokasi /usr/local/lsws/instagram-feed dipilih agar sesuai dengan struktur direktori OpenLiteSpeed

3. Buat file index.html

File ini adalah halaman utama yang akan ditampilkan saat pengguna mengakses website. File HTML ini berisi struktur dan konten halaman, seperti profil Instagram yang ingin ditampilkan.

4. Konfigurasi Virtual Host OpenLiteSpeed

File ini berisi pengaturan seperti root direktori dokumen (docRoot), file index, konfigurasi log error dan akses, serta pengaturan rewrite untuk URL yang lebih fleksibel.

```
docRoot                   /usr/local/lsws/instagram-feed
indexFiles                index.html
autoIndex                 on

errorlog $VH_ROOT/logs/error.log {
  useServer               0
  logLevel                WARN
}

accesslog $VH_ROOT/logs/access.log {
  useServer               0
  logFormat               "%h %l %u %t \"%r\" %>s %b"
}

rewrite {
  enable                  1
  autoLoadHtaccess        1
}
```

Gambar 18 Membuat virtual host

5. Restart dan Testing OpenLiteSpeed

Agar semua konfigurasi yang sudah diubah diterapkan, layanan OpenLiteSpeed perlu dimulai ulang. Restart memastikan server menggunakan konfigurasi terbaru.

D. Pembahasan

1. Struktur Skenario Pengujian

Pengujian ini dilakukan untuk mengukur Response Time dari tiga web server berbeda, yaitu Apache, Nginx, dan LiteSpeed. Response Time adalah waktu yang dibutuhkan server untuk merespons permintaan HTTP dari klien. Pengujian ini dilakukan menggunakan Apache JMeter dengan tiga skenario berbeda, yaitu:

1.1. Baseline Load Test → Simulasi beban normal dengan 50 pengguna aktif (Threads: 50, Ramp-up: 10s, Duration: 300s). Tujuan: Menguji performa server dalam kondisi beban normal (pengguna harian).

1.2. Spike Test → Simulasi lonjakan mendadak dalam jumlah pengguna hingga 100 pengguna (Threads: 100, Ramp-up: 5s, Duration: 300s). Tujuan: Mengukur kemampuan server menangani lonjakan lalu lintas mendadak.

1..3. Stress Test → Menguji batas maksimal server dengan 200 pengguna aktif (Threads: 200, Ramp-up: 20s, Duration: 300s). Tujuan: Mengetahui batas maksimum server sebelum mulai mengalami kegagalan atau degradasi performa.

Ketiga jenis request ini dilakukan dalam periode waktu 300 detik (5 menit), sesuai dengan jumlah pengguna (threads) yang ditentukan dalam setiap skenario.

2. Skema Pengujian

Skema pengujian yang dirancang melibatkan beberapa langkah untuk menguji performa web server dalam menangani permintaan HTTP. Pengujian dimulai dengan membuka halaman utama website untuk mensimulasikan akses pengguna ke beranda. Selanjutnya, dilakukan pengujian dengan membuka konten Website, yang mencerminkan aktivitas pengguna dalam membaca dan berinteraksi dengan diskusi. Skenario pengujian mencakup tiga jenis permintaan, yaitu request ke homepage (GET /) untuk mengakses halaman utama. Skema ini dirancang untuk mensimulasikan beban nyata pada server dan mengukur seberapa baik masing-masing web server menangani lalu lintas pengguna dalam berbagai skenario.

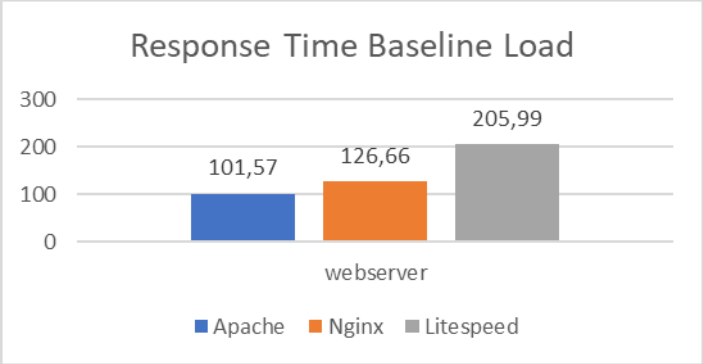
3. Analisis Hasil dari Response Time

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh Response Time dari masing-masing web server dalam skenario sebagai berikut.

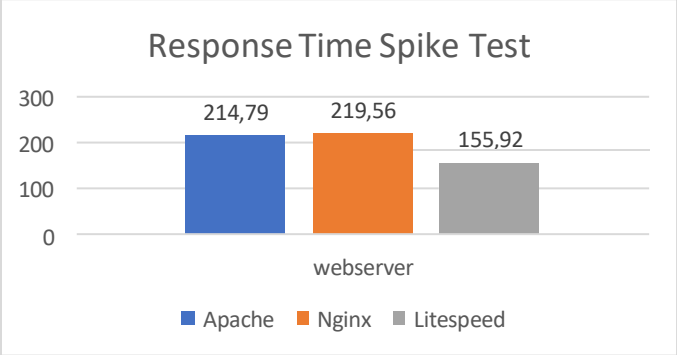
Tabel 2 Pengujian Response Time

No.	Webserver	Baseline Load	Spike	Stress
1.	Apache	101,67 ms	214,79 ms	127,68 ms
2.	Nginx	126,66 ms	219,56 ms	107,64 ms

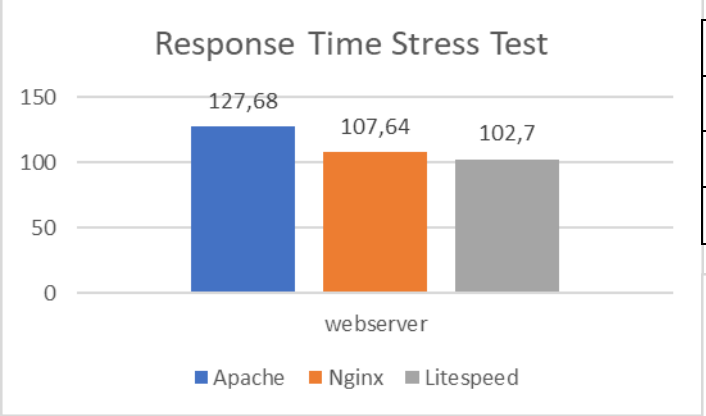
3.	Litespeed	205,99 ms	155,92 ms	102,70 ms



Gambar 19 Grafik hasil pengujian response time baseline load



Gambar 20 Grafik hasil pengujian response time spike test



Gambar 21 Grafik hasil pengujian response time stress test

Pengujian response time dilakukan untuk mengukur waktu yang dibutuhkan oleh web server dalam merespons permintaan dari pengguna. Pengujian ini dilakukan dalam tiga skenario, yaitu Baseline Load, Spike dan Stress. Berdasarkan hasil pengujian pada skenario Baseline Load, Apache memiliki waktu respon paling rendah sebesar 101,67 ms, sementara Nginx mencatat 126,66 ms dan LiteSpeed 205,99 ms. Saat pengujian dalam skenario Spike, LiteSpeed menunjukkan performa terbaik dengan response time sebesar 155,92 ms, sedangkan Apache dan Nginx masing-masing mencatat waktu respon sebesar 214,79 ms dan 219,56 ms. Dalam skenario Stress, LiteSpeed kembali unggul dengan response time sebesar 102,70 ms, disusul oleh Nginx sebesar 107,64 ms, dan Apache sebesar 127,68 ms. Berdasarkan hasil tersebut, dapat disimpulkan bahwa :

a. Apache unggul pada beban normal, namun performanya menurun saat terjadi spike dan stress.
b. Nginx stabil di semua kondisi, namun tidak pernah menjadi yang tercepat.
c. Litespeed kurang optimal pada beban normal, tetapi sangat baik dalam menangani spike dan stress, bahkan menjadi yang tercepat pada kedua kondisi tersebut.

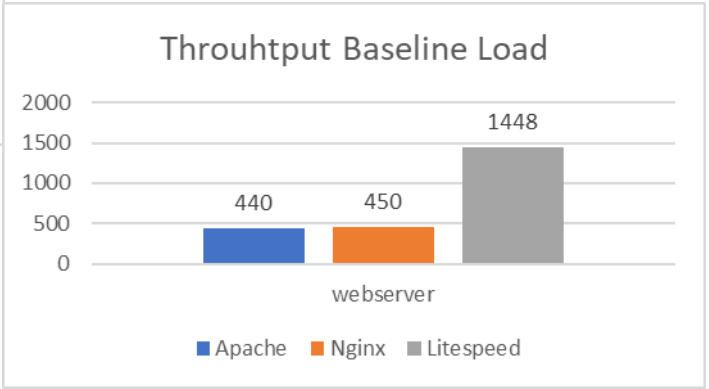
Hasil ini menunjukkan bahwa pemilihan webserver sebaiknya disesuaikan dengan pola trafik aplikasi. Untuk beban normal, Apache lebih efisien. Sedangkan untuk aplikasi dengan potensi lonjakan trafik atau beban berat, Litespeed menawarkan waktu respons yang lebih baik.

4. Analisis Hasil dari Throughput/Transfer Rate

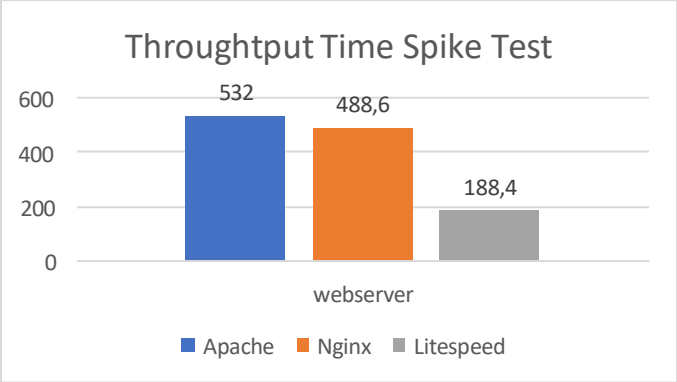
Throughput mengukur jumlah permintaan yang dapat diproses oleh server dalam satu detik. Ini menunjukkan seberapa banyak permintaan yang dapat dilayani dalam waktu tertentu, yang berkaitan erat dengan kemampuan web server untuk menangani beban tinggi.

Tabel 3 Pengujian Throughput

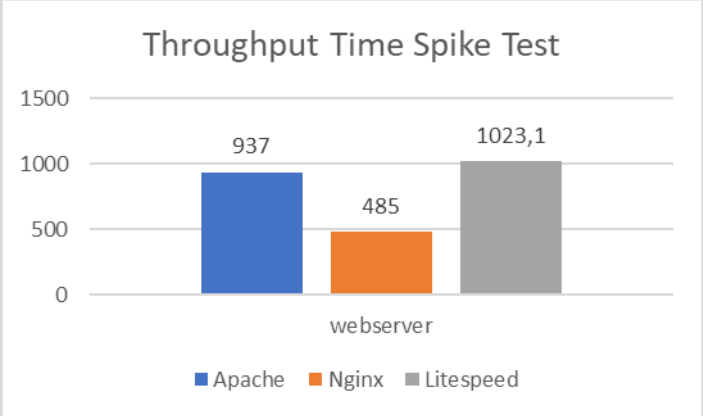
.	Webserver	Baseline Load	Spike	Stress
1.	Apache	440 req/s	532 req/s	937 req/s
2.	Nginx	450 req/s	488,6 req/s	485 req/s
3.	Litespeed	1448,71 req/s	188,4 req/s	1023,1 req/s



Gambar21 Grafik hasil pengujian throughput stress test



Gambar 22 Grafik hasil pengujian throughput stress test



Gambar 23 Grafik hasil pengujian throughput stress test

Pengujian throughput dilakukan untuk mengukur jumlah permintaan yang dapat diproses per detik oleh server. Pengujian ini dilakukan dalam tiga scenario, yaitu Baseline Load, Spike dan Stress. Berdasarkan hasil skenario pengujian pada Baseline Load, Litespeed memiliki hasil terbaik dengan 1448,71 req/s di ikuti oleh Nginx dengan 450 req/s dan Apache 440 req/s. lalu untuk scenario Spike Test Apache lebih unggul dengan 532 req/s, disusul Nginx dengan 488,6 req/s dan Litespeed menempati urutan paling akhir di 188,4 req/s. Sedangkan untuk scenario Stress Test, Litespeed lebih unggul dengan 1023,1 req/s, disusul Apache 937 req/s dan Nginx dengan 485 req/s. Dari seluruh hasil pengetesan ini dapat diartikan bahwa :

a. **Litespeed** sangat unggul pada kondisi beban normal dan stress, namun performanya menurun drastis saat terjadi spike/lonjakan trafik.
b. **Apache** menunjukkan performa yang stabil dan bahkan meningkat pada kondisi spike dan stress, menjadikannya pilihan yang andal untuk menangani lonjakan dan beban berat.
c. **Nginx** relatif stabil pada baseline dan spike, namun tidak mampu meningkatkan throughput pada kondisi stress, sehingga kurang optimal untuk beban sangat tinggi.

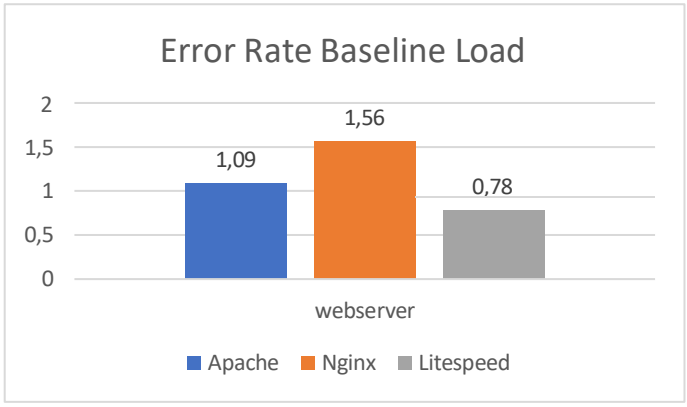
Untuk kebutuhan throughput tinggi pada beban normal dan stress, **Litespeed** adalah pilihan terbaik. Untuk kestabilan performa menghadapi lonjakan dan beban berat, **Apache** lebih unggul. Sedangkan **Nginx** cocok untuk beban sedang, namun kurang optimal pada stress test. Analisis ini menunjukkan pentingnya memilih webserver sesuai karakteristik trafik dan kebutuhan aplikasi.

5. Analisis Hasil Error Rate

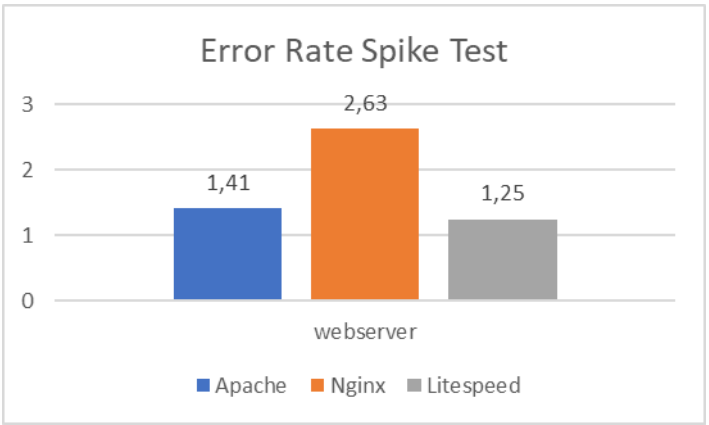
Error Rate merupakan persentase jumlah permintaan yang gagal diproses oleh server dibandingkan dengan total permintaan yang dikirimkan. Semakin tinggi nilai error rate, semakin banyak permintaan yang gagal dipenuhi oleh server, yang menunjukkan penurunan stabilitas dan keandalan server dalam menangani beban lalu lintas.

Tabel 1 Pengujian Error Rate

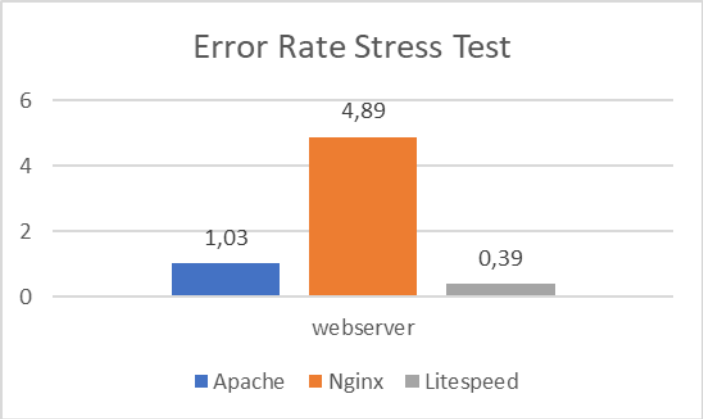
No.	Webserver	50 users	100 users	200 users
1.	Apache	1,09 %	1,41 %	1,03 %
2.	Nginx	1,56 %	2,63 %	4,89 %
3.	Litespeed	0,78 %	1,25 %	0,39 %



Gambar 23 Grafik hasil pengujian error rate baseline load



Gambar 24 Grafik hasil pengujian error rate stress test



Gambar 25 Grafik hasil pengujian error rate stress test

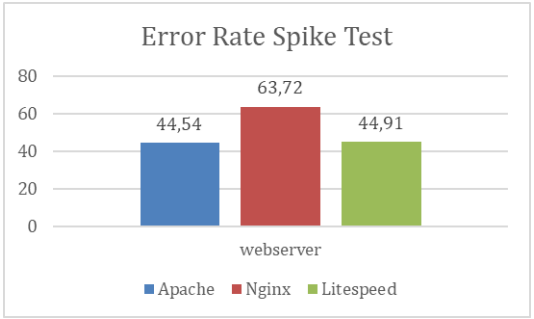
Berdasarkan Tabel 4.2, Apache menunjukkan error rate stabil: 1,09% (50 pengguna), naik sedikit menjadi 1,41% (100 pengguna), lalu turun ke 1,03% (200 pengguna). Nginx mengalami peningkatan error rate signifikan seiring bertambahnya beban, dari 1,56% menjadi 4,89%, menandakan penurunan performa. Sebaliknya, LiteSpeed mencatat performa paling stabil dan andal, dengan error rate terendah: 0,78% (50 pengguna), 1,25% (100 pengguna), dan turun ke 0,39% (200 pengguna). Dapat disimpulkan bahwa LiteSpeed adalah web server paling andal dalam pengujian ini, Apache cukup stabil untuk beban sedang hingga tinggi, sedangkan Nginx membutuhkan optimasi lebih lanjut untuk menangani lalu lintas berat secara efektif.

6. Analisis Hasil Latensi

pengiriman data antara server dan klien. Semakin rendah latensi, semakin cepat data dapat diterima oleh klien setelah dikirim oleh server.

Tabel 5 Pengujian Latensi

No.	Webserver	Latensi
1.	Apache	44,54 ms
2.	Nginx	63,72 ms
3.	Litespeed	44,91 ms



Gambar 26 Grafik hasil pengujian latensi

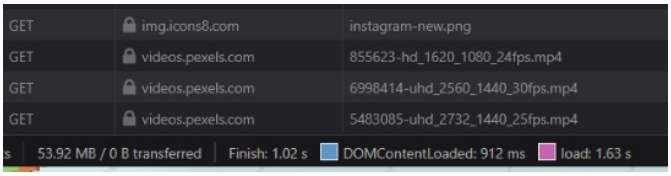
Hasil pengujian latensi pada ketiga web server menunjukkan bahwa Apache memiliki latensi terendah sebesar 44,54 ms, diikuti LiteSpeed dengan 44,91 ms, sedangkan Nginx mencatat latensi tertinggi sebesar 63,72 ms. Ini mengindikasikan bahwa Apache dan LiteSpeed lebih cepat dalam merespons permintaan dibanding Nginx. Perbedaan ini kemungkinan dipengaruhi oleh arsitektur internal dan cara masing-masing server menangani koneksi. Meskipun selisihnya tidak besar, dalam skenario trafik tinggi, latensi yang lebih rendah dapat memberikan dampak signifikan terhadap performa layanan. Dengan demikian, Apache dan LiteSpeed lebih cocok digunakan untuk aplikasi yang membutuhkan respons awal yang cepat, sementara Nginx memerlukan penyesuaian jika digunakan dalam konteks yang sensitif terhadap latensi.

7. Analisis Hasil dari Penggunaan CPU dan RAM

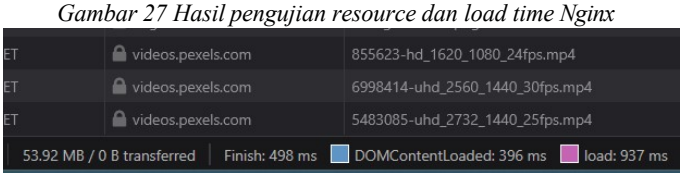
Pengujian penggunaan CPU dan RAM terhadap Apache, Nginx, dan LiteSpeed menunjukkan bahwa LiteSpeed adalah web server paling efisien dan stabil dalam berbagai skenario beban (Idle, Baseline, Spike, Stress). LiteSpeed secara konsisten menggunakan CPU dan RAM secara optimal, bahkan saat beban tinggi, dengan pemakaian CPU hingga 99,79% dan RAM tetap rendah di kisaran 20%. Nginx juga efisien, khususnya dalam penggunaan RAM, meskipun penggunaan CPU-nya sangat tinggi untuk menjaga performa. Sebaliknya, Apache menunjukkan penggunaan CPU yang tidak stabil, terutama saat lonjakan trafik, dan penggunaan RAM yang cenderung lebih tinggi dari dua server lainnya. Secara keseluruhan, LiteSpeed paling unggul dalam efisiensi sumber daya, diikuti Nginx, sementara Apache kurang cocok untuk lingkungan dengan beban tinggi atau tidak menentu.

8. Analisis Hasil dari Pengujian Load Time dan Resource Size

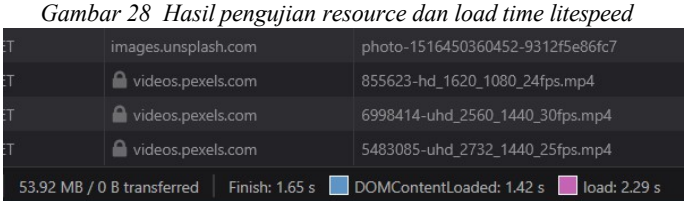
Ketiga server web yang diuji menunjukkan ukuran resource size yang sama, yakni sebesar 53,92 MB untuk total resource yang di muat. Hal ini menandakan bahwa perbedaan performa yang terjadi bukan disebabkan oleh besarnya data yang harus diunduh, melainkan faktor lain seperti efisiensi penanganan request dan proses render halaman. Dari segi Load time, Litespeed menjadi yang tercepat dengan hanya memperlukan waktu selama 1,51s, unggul tipis dari Nginx yang memperlukan Load Time selama 1,52. selanjutnya diikuti oleh Apache dengan Load Time paling lama, yaitu di 1,62s. Dengan ukuran data yang sama, performa Litespeed yang lebih baik ini mengindikasikan efisiensi yang lebih tinggi dalam mengelola proses loading halaman dibanding kedua server lainnya.



Gambar 27 Hasil pengujian resource dan load time Nginx



Gambar 28 Hasil pengujian resource dan load time litespeed



Gambar 29 Hasil pengujian resource dan load time litespeed

IV. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil pengujian performa terhadap tiga web server, yaitu Apache, Nginx, dan LiteSpeed, dapat disimpulkan bahwa masing-masing memiliki keunggulan tersendiri tergantung pada skenario penggunaan. Pada pengujian response time, LiteSpeed menunjukkan performa terbaik dalam menghadapi beban lonjakan (Spike) dan tekanan tinggi (Stress), sementara Apache unggul pada kondisi beban seimbang (Baseline Load). Dalam pengujian throughput, LiteSpeed mampu memproses permintaan lebih banyak per detik dalam skenario Baseline Load dan Stress Test, sedangkan Apache unggul pada Spike Test. Pada parameter error rate, LiteSpeed mencatat nilai terendah dan paling stabil, menunjukkan keandalannya dalam menghadapi peningkatan jumlah pengguna, sedangkan Nginx mengalami peningkatan error rate yang signifikan. Dalam hal latensi, Nginx menunjukkan waktu paling rendah, menjadikannya pilihan yang baik untuk aplikasi yang membutuhkan respons cepat. Dari sisi penggunaan sumber daya, LiteSpeed terbukti paling efisien dalam pemanfaatan CPU dan RAM, terutama dalam kondisi beban tinggi, sementara Apache menunjukkan pola penggunaan sumber daya yang tidak stabil dan cenderung boros RAM. Terakhir, pada pengujian load time dan resource size, Litespeed menunjukkan performa terbaik dengan waktu pemuatan halaman paling cepat, meskipun data yang dimuat sama di ketiga server. Secara keseluruhan Litespeed lebih unggul dalam lima parameter pengujian, disusul dengan Apache dan Nginx.

B. Saran

Berdasarkan hasil penelitian, terdapat beberapa saran yang dapat diterapkan. Pertama, optimalisasi konfigurasi sangat disarankan agar setiap web server dapat bekerja lebih efisien sesuai kebutuhan aplikasi, seperti melalui penyesuaian buffer size, jumlah worker process, dan pengaturan caching. Kedua, penelitian lanjutan perlu dilakukan dengan menambahkan parameter pengujian lain seperti aspek keamanan dan ketahanan terhadap serangan DDoS, serta memperluas cakupan server yang diuji dengan memasukkan alternatif modern seperti Caddy atau Traefik. Ketiga, dalam penerapan di lingkungan produksi, LiteSpeed dan Nginx direkomendasikan untuk situs dengan lalu lintas tinggi karena performa dan efisiensinya. Sementara itu, Apache masih relevan digunakan pada proyek yang membutuhkan modul spesifik atau konfigurasi kompleks yang tidak mudah diterapkan pada server lain.

UCAPAN TERIMA KASIH

Penulis menyampaikan rasa syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan sebaik-baiknya. Ucapan terima kasih juga penulis sampaikan kepada Bapak Agus selaku dosen pembimbing, yang telah meluangkan waktunya untuk memberikan arahan dan masukan yang sangat berarti. Tak lupa, penulis juga mengucapkan terima kasih yang tulus kepada kedua orang tua tercinta atas doa dan dukungan yang tiada henti, serta kepada teman-teman yang telah memberikan semangat dan bantuan selama proses penyusunan penelitian ini.

REFERENSI

[1] Al Harits, S., & Turmudi, A. (n.d.). NOVICE RESEARCH EXPLORATION (NRE) ρ Securing Servers From DDoS Attacks Using NGINX.

[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] Leo Cahyadi, M., Setiawan, H., & Romegar Mair, Z. (2023). Analisis Perbandingan Kinerja Web Server Nginx dan Litespeed Menggunakan Httperf Dengan Sistem Operasi Debian.

[4] Nanda Prihaditama, D., & Wiseto Prasetyo Agung, I. (2024). Analisis Perbandingan Performa Web Server Apache Dengan Lighttpd Metode Stresstest Pada VPS. 5(2).

[5] Nimas Maharani, C., Darwis, D., Penulis, N., Dedi, K. :, & Submitted, D. (2023). Analisis Perbandingan Kualitas Perangkat Lunak Pada Website Perguruan Tinggi Menggunakan Metode Webqual, Apache J-Meter, Dan Web Server Stress Tool. 4(1), 34–41. https://doi.org/10.33365/jtsi.v4i1.2436

[6] Oki Alfatah, R. (2022). ANALISIS PERBANDINGAN PERFORMA WEB SERVER APACHE DAN LIGHTTPD MENGGUNAKAN

HTTPERF PADA VPS DENGAN SISTEM OPERASI UBUNTU SERVER. www.uir.ac.id

[7] Tang, H. (2024). EVALUATION OF HTTP DDOS CYBER ATTACK ON WEB SERVERS APACHE AND NGINX.

[8] Wahyudi, R. M., Sholihah, A. N., Pambudi, G. R., Muttaqin, R. N., Qurrota A'yun, A., Nuraeni, Y. S., Akhdyan, M. R., & Pinem, A. E. (2024). Mengimplementasikan SSL/TLS pada Web Server Apache di dalam Jaringan Internal Praktikum untuk Pengembangan Web Server. In Prodi Sistem Informasi (Vol. 3, Issue 1). http://jurnalilmiah.org/journal/index.php/majemuk

[9] Munyaradzi, Z., Maxmillan, G., & Amanda, M. N. (2013). Effects of Web Page Contents on Load Time over the Internet. International Journal of Science and Research (IJSR), 2(9), 2319-7064.