

Perbandingan Kinerja Nginx dan Caddy sebagai *Web Server* untuk Aplikasi PHP

Musa Amin

Program Studi Akuntansi Syariah, IAIN Pontianak
E-mail: musaiaintk@gmail.com

Abstrak

Penelitian ini membandingkan kinerja Nginx dan Caddy sebagai web server untuk aplikasi PHP melalui pengujian latensi, waktu respon, throughput, dan penggunaan sumber daya. Eksperimen dilakukan menggunakan alat load-testing K6 dengan tiga skenario beban kerja: ringan (50 virtual user/VU), sedang (500 VU), dan berat (2000 VU). Hasil penelitian menunjukkan bahwa Nginx memiliki keunggulan dalam efisiensi kinerja, mencatat latensi dan waktu respon yang lebih rendah serta throughput yang lebih tinggi pada beban kerja ringan hingga sedang. Namun, pada beban berat, Caddy menunjukkan stabilitas yang lebih baik dengan tingkat kegagalan permintaan 0%, meskipun konsumsi RAM meningkat signifikan hingga 28,72%. Dalam semua skenario, Nginx lebih efisien dalam penggunaan CPU dan RAM dibandingkan Caddy, kecuali pada beban berat di mana stabilitas menjadi keunggulan utama Caddy. Penelitian ini menyimpulkan bahwa Nginx lebih cocok untuk aplikasi dengan kebutuhan efisiensi tinggi, sementara Caddy lebih sesuai untuk aplikasi yang membutuhkan stabilitas koneksi tanpa kegagalan permintaan

KATA KUNCI — *Web Server, Nginx, Caddy, Php, Load-Testing*

1. PENDAHULUAN

Web server memegang peranan penting dalam infrastruktur aplikasi berbasis web. Sebagai penghubung antara pengguna dan aplikasi, performa *web server* menentukan seberapa cepat permintaan pengguna dapat diproses dan direspons. Dalam konteks aplikasi PHP, yang merupakan salah satu bahasa pemrograman *server-side* terpopuler, keberhasilan implementasi *web server* menjadi semakin kritis. PHP mendukung lebih dari 75% situs web secara global [1], termasuk *platform* populer seperti WordPress, Joomla, dan Laravel [2]. Dengan dominasi ini, efisiensi *web server* dalam menjalankan aplikasi PHP menjadi aspek penting yang patut diteliti.

Keunggulan PHP terletak pada fleksibilitas dan kemudahan penggunaannya. PHP mendukung pengembangan *Content Management System* (CMS) seperti WordPress, Drupal, dan Joomla [3], serta berbagai *web framework* seperti Laravel, CodeIgniter, dan Symfony. CMS dan *framework* ini digunakan secara luas untuk berbagai kebutuhan, mulai dari blog sederhana hingga aplikasi web skala besar. Namun, performa aplikasi PHP tidak hanya bergantung pada kode program, tetapi juga pada kemampuan *web server* yang digunakan untuk menanganinya [4].

Nginx dan Caddy adalah dua *web server* modern dapat digunakan untuk menjalankan aplikasi berbasis PHP. Nginx, diperkenalkan pada tahun 2004, dikenal sebagai salah satu *web server* tercepat dan paling efisien dalam menangani koneksi *concurrent* [5]. Nginx mengusung arsitektur *event-driven* yang memungkinkannya untuk mengelola ribuan koneksi secara bersamaan dengan konsumsi sumber daya yang minimal [6]. Di sisi lain, Caddy, meskipun masih terbilang baru, rilis pertama kali pada tahun 2015, menawarkan pendekatan modern dengan konfigurasi yang lebih sederhana dan fitur *automatic HTTPS*, yang dapat mempercepat proses pengembangan dan penerapan aplikasi [7].

Permasalahan yang muncul adalah, meskipun menjalankan aplikasi PHP yang sama, apakah setiap *web server* memiliki kinerja yang berbeda? Aplikasi PHP pada dasarnya berkomunikasi dengan *web server* untuk menangani permintaan HTTP, dan perbedaan arsitektur serta fitur *web server* dapat memengaruhi hasil akhirnya. Sebagai contoh, Nginx memiliki keunggulan dalam skenario dengan lalu lintas tinggi, sementara Caddy menawarkan kemudahan dalam konfigurasi yang dapat mengurangi waktu pengelolaan *server*. Namun, bagaimana perbandingan kedua *web server* ini dalam skenario nyata, khususnya untuk aplikasi PHP.

Penelitian sebelumnya sebagian besar berfokus pada perbandingan antara Nginx dan Apache, yang merupakan *web server* tradisional yang masih banyak digunakan. Studi menunjukkan bahwa Nginx umumnya memiliki performa yang lebih baik dalam hal *throughput* dan waktu respon dibandingkan Apache [5]. Pada penelitian yang lain, perbandingan kinerja Apache dan Nginx sebagai *reverse proxy* memberikan hasil bahwa Nginx lebih cepat dan efisien dalam menangani permintaan dari banyak klien secara bersamaan [8]. Pada penelitian yang lain, performa Nginx lebih baik dibandingkan Apache berdasarkan parameter *response time* dan *throughput* [9]. Namun, dengan hadirnya Caddy sebagai pesaing baru, dibutuhkan evaluasi lebih lanjut untuk memahami bagaimana ia bersaing dengan Nginx, terutama dalam konteks aplikasi PHP dinamis.

Selain itu, tantangan dalam pengujian kinerja *web server* terletak pada skenario uji yang realistis. Aplikasi PHP sering kali melibatkan berbagai komponen, termasuk *database*, *cache*, dan file statis. Oleh karena itu, pengujian harus mencakup berbagai metrik, seperti latensi, *throughput*, konsumsi CPU dan RAM, serta waktu respon di bawah beban kerja tinggi [10]. Studi ini juga perlu mempertimbangkan faktor lain, seperti kemudahan instalasi, fleksibilitas konfigurasi dan dukungan komunitas yang dapat memengaruhi adopsi teknologi oleh pengembang.

Dalam konteks penggunaan aplikasi PHP berbasis CMS atau *framework*, efisiensi *web server* menjadi semakin penting karena banyaknya permintaan yang harus ditangani, termasuk permintaan untuk memuat konten dinamis, menjalankan skrip PHP, dan mengakses *database*. Keberhasilan implementasi *web server* dapat berdampak langsung pada pengalaman pengguna akhir, seperti kecepatan akses dan stabilitas aplikasi [11].

Tujuan utama penelitian ini adalah untuk membandingkan kinerja Nginx dan Caddy sebagai *web server* untuk aplikasi PHP. Pengujian dilakukan dengan menggunakan aplikasi PHP yang sama, untuk memastikan bahwa perbedaan kinerja disebabkan oleh *web server*, bukan oleh aplikasi.

Dengan membandingkan kedua *web server* ini, penelitian ini diharapkan dapat memberikan wawasan yang berharga bagi pengembang dan administrator sistem dalam memilih *web server* yang paling sesuai untuk kebutuhan aplikasi mereka. Penelitian ini juga dapat menjadi landasan bagi penelitian lebih lanjut dalam mengeksplorasi teknologi *web server* modern lainnya yang mendukung pengembangan aplikasi web berbasis PHP.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen untuk membandingkan kinerja Nginx dan Caddy sebagai *web server* dalam menjalankan aplikasi PHP. Eksperimen dilakukan dalam lingkungan terkontrol dengan parameter yang distandarisasi untuk memastikan hasil yang valid dan dapat direproduksi [12]. Adapun metode penelitian ini mencakup desain eksperimen, konfigurasi sistem, aplikasi uji, pengumpulan data, serta analisis hasil.

Penelitian ini melibatkan dua lingkungan server yang identik, masing-masing menjalankan Nginx dan Caddy. Aplikasi PHP yang sama digunakan pada kedua server untuk memastikan bahwa perbedaan kinerja hanya disebabkan oleh perbedaan *web server*. Eksperimen

dilakukan dengan mengukur berbagai metrik kinerja, yaitu:

- 1) Latensi: Waktu yang diperlukan untuk merespons permintaan dari klien.
- 2) *Throughput*: Jumlah permintaan yang dapat ditangani per detik.
- 3) Konsumsi Sumber Daya: Penggunaan CPU dan memori selama eksperimen.
- 4) Waktu Respon: Rata-rata waktu yang dibutuhkan untuk memproses permintaan.

Virtual Private Server (VPS) digunakan untuk menjalankan aplikasi PHP dan *web server* dalam penelitian ini memiliki spesifikasi sebagai berikut:

- 1) CPU: 2 vCPU
- 2) RAM: 2 GB
- 3) Disk: SSD 60 GB
- 4) Sistem operasi: Ubuntu 24.04 LTS
- 5) PHP *Runtime*: PHP versi 8.2 dengan FastCGI Process Manager (FPM)
- 6) *Database*: MariaDB versi 10.11

Web server dikonfigurasi dengan parameter spesifik untuk memastikan kondisi pengujian yang terukur. Nginx dijalankan dengan *worker_processes auto* (menyesuaikan ke 2 proses sesuai spesifikasi 2CPU) dan *worker_connections 768* pada blok *events*, sehingga mampu menangani hingga 1.536 koneksi konkuren (2×768). Sementara Caddy, yang berbasis bahasa Go, tidak memiliki konsep *worker_processes* atau *worker_connections* karena mengandalkan model konkurensi otomatis berbasis *goroutine*. Kedua *web server* diuji dengan batasan sumber daya yang sama (2 CPU, 2 GB RAM) untuk memastikan konsistensi.

Pengujian dilakukan menggunakan alat *load testing* k6 [13]. K6 merupakan alat uji dari Grafana Labs yang digunakan untuk menguji kinerja dan ketahanan API, microservices, dan situs web. Dengan menggunakan JavaScript, K6 mensimulasikan beban tinggi melalui pengguna virtual serta mendukung pengujian stres. Alat ini juga menyediakan metrik dan laporan untuk menganalisis perilaku aplikasi serta mengidentifikasi aspek yang perlu ditingkatkan [14].

Beban kerja yang digunakan dalam pengujian, yaitu:

- 1) Beban ringan: 50 *virtual user*/VU selama 1 menit.
- 2) Beban sedang: 500 VU selama 1 menit.
- 3) Beban berat: 2000 VU selama 1 menit.

Metrik kinerja hasil pengujian k6 disimpan ke dalam file *log* dan pemakaian sumber daya dipantau melalui *system monitoring* HetrixTools [15]. Data dikumpulkan untuk setiap sesi pengujian dan dianalisis lebih lanjut untuk mendapatkan wawasan tentang perbandingan kinerja kedua *web server*.

Berikut ini adalah struktur tabel, dan potongan skrip PHP serta skrip *load testing* k6 yang digunakan dalam pengujian:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
description	text	YES		NULL	
price	decimal(10,2)	NO		NULL	
created_at	timestamp	YES		current_timestamp()	

Gambar 1. Struktur tabel product

```

$name = "Product " . rand(1, 1000);
$description = "Description for " . $name;
$price = rand(100, 1000);

$stmt = $pdo->prepare("INSERT INTO product (name, description, price) VALUES
(:name, :description, :price)");
$stmt->execute(['name' => $name, 'description' => $description, 'price' => $price]);

```

Gambar 2. Skrip insert.php

```

$stmt = $pdo->query("SELECT * FROM product ORDER BY created_at DESC LIMIT 10");
$products = $stmt->fetchAll();

header('Content-Type: application/json');
echo json_encode($products);

```

Gambar 3. Skrip read.php

```

export let options = {
  stages: [
    { duration: '1m', target: 2000 },
  ],
};

export default function () {
  let insertRes = http.post('https://app.aminlabs.my.id/insert.php');
  if (insertRes.status !== 200) {
    console.error('Insert failed: ', insertRes.body);
  }

  let readRes = http.get('https://app.aminlabs.my.id/read.php');
  if (readRes.status !== 200) {
    console.error('Read failed: ', readRes.body);
  }

  sleep(1);
}

```

Gambar 4. Skrip load-test.js

3. HASIL DAN PEMBAHASAN

3.1. Latensi

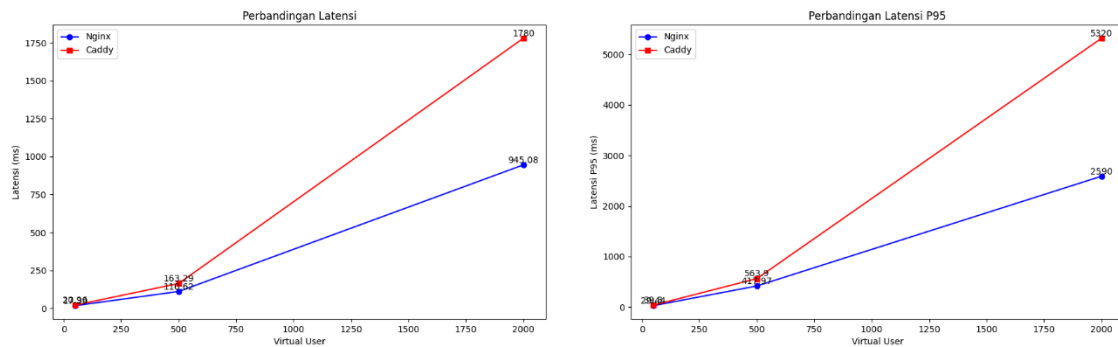
Hasil pengujian metrik latensi pada Tabel 1, menunjukkan bahwa Nginx secara konsisten memiliki latensi rata-rata dan latensi persentil ke-95 (P95) yang lebih rendah dibandingkan Caddy dalam semua skenario beban kerja. Pada beban ringan (50 VU), Nginx mencatat latensi rata-rata 17.39 ms dan P95 29.44 ms, sementara Caddy menunjukkan latensi rata-rata 20.96 ms dan P95 39.3 ms. Hal ini mengindikasikan bahwa Nginx lebih efisien dalam menangani permintaan kecil dengan respons yang lebih cepat.

Tabel 1. Metrik latensi

Virtual User	Web Server	Latensi (ms)	Latensi P95 (ms)
50	Nginx	17.39	29.44
	Caddy	20.96	39.3
500	Nginx	110.62	417.97
	Caddy	163.29	563.9
2000	Nginx	945.08	2590
	Caddy	1780	5320

Pada beban sedang (500 VU), perbedaan kinerja antara Nginx dan Caddy semakin terlihat. Nginx mencatat latensi rata-rata 110.62 ms dan P95 417.97 ms, sedangkan Caddy memiliki latensi rata-rata 163.29 ms dan P95 563.9 ms. Performa Nginx lebih stabil di bawah tekanan beban yang lebih tinggi, yang mencerminkan efisiensi arsitektur *event-driven*-nya. Sebaliknya, Caddy menunjukkan peningkatan latensi yang signifikan, terutama pada metrik P95, yang mengindikasikan degradasi performa.

Dalam skenario beban berat (2000 VU), Nginx tetap menunjukkan ketahanan yang lebih baik dibandingkan Caddy. Latensi rata-rata dan P95 Nginx tercatat masing-masing 945.08 ms dan 2590 ms, sedangkan Caddy mengalami lonjakan latensi hingga 1780 ms untuk rata-rata dan 5320 ms untuk P95. Hasil tersebut menunjukkan bahwa performa Caddy menurun drastis di bawah beban tinggi.



Gambar 5. Perbandingan Latensi

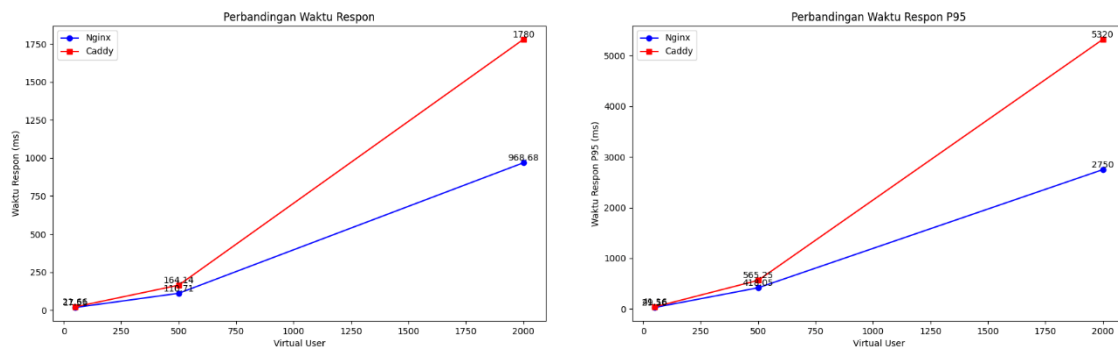
3.2. Waktu Respon

Hasil pengujian metrik waktu respon pada Tabel 2, menunjukkan bahwa Nginx memiliki waktu respon yang lebih rendah dibandingkan Caddy pada semua skenario, dengan perbedaan yang semakin besar seiring peningkatan beban kerja. Pada beban ringan (50 VU), selisih waktu respon rata-rata antara Nginx dan Caddy adalah 4.15 ms, sedangkan pada beban berat (2000 VU), selisih ini meningkat menjadi 811.32 ms.

Tabel 2. Metrik waktu respon

Virtual User	Web Server	Waktu Respon (ms)	Waktu Respon P95 (ms)
50	Nginx	17.51	29.56
	Caddy	21.66	41.16
500	Nginx	110.71	418.05
	Caddy	164.14	565.25
2000	Nginx	968.68	2750
	Caddy	1780	5320

Nginx juga menunjukkan keunggulan dalam waktu respon persentil 95 (P95), yang mencerminkan performa dalam skenario terburuk. Pada beban kerja berat, Nginx mencatat waktu respon P95 sebesar 2750 ms, jauh lebih rendah dibandingkan Caddy yang mencapai 5320 ms. Hal ini menunjukkan bahwa Nginx lebih efisien dalam menangani koneksi secara bersamaan, terutama di lingkungan dengan lalu lintas tinggi.



Gambar 6. Perbandingan Waktu Respon

3.3. Throughput

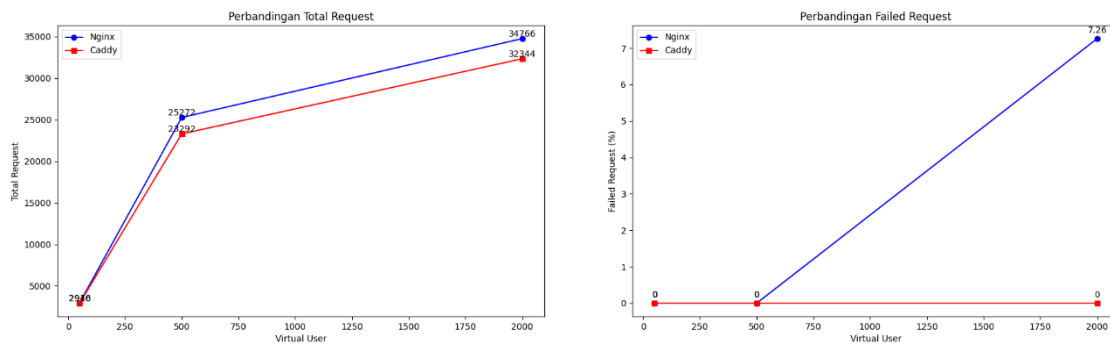
Hasil pengujian metrik *throughput* pada Tabel 3, menunjukkan bahwa Nginx memiliki keunggulan dalam hal *throughput* pada beban kerja ringan dan sedang. Pada beban kerja 50 VU, Nginx mencapai 48.27 rps dengan total 2946 permintaan, sedikit lebih tinggi dibandingkan Caddy yang mencapai 47.7 rps dengan total 2910 permintaan. Keduanya tidak mengalami kegagalan permintaan, menandakan bahwa kedua *web server* stabil dalam menangani beban ringan. Pada beban sedang dengan 500 VU, Nginx tetap unggul dengan *throughput* sebesar 409.19 rps dan total 25272 permintaan dibandingkan Caddy yang mencapai 376.93 rps dan total 23292 permintaan. Stabilitas kedua *web server* tetap terjaga pada tingkat ini dengan tingkat kegagalan permintaan 0%.

Tabel 3. Metrik *throughput*

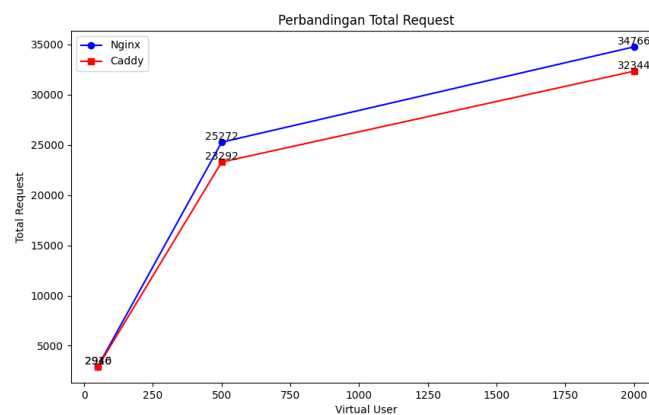
Virtual User	Web Server	Request per second (rps)	Total Request	Failed request
50	Nginx	48.27	2946	0%
	Caddy	47.7	2910	0%
500	Nginx	409.19	25272	0%
	Caddy	376.93	23292	0%
2000	Nginx	386.24	34766	7.26% (2526)
	Caddy	446.21	32344	0%

Namun, pada beban kerja berat dengan 2000 VU, pola kinerja kedua *web server* berubah. Nginx mencatat *throughput* sebesar 386.24 rps dengan total 34766 permintaan, tetapi terdapat tingkat kegagalan permintaan sebesar 7.26% (2526 permintaan gagal). Sebaliknya, Caddy mencatat *throughput* yang lebih tinggi sebesar 446.21 rps dengan total 32344 permintaan tanpa kegagalan permintaan. Ini menunjukkan bahwa Caddy lebih stabil di bawah beban sangat tinggi, meskipun jumlah total permintaan yang ditangani sedikit lebih rendah dibandingkan Nginx.

Dari segi stabilitas, Caddy unggul pada beban berat karena tidak ada permintaan yang gagal, menjadikannya pilihan yang lebih andal untuk aplikasi yang memerlukan keberhasilan koneksi tinggi di bawah tekanan besar. Sebaliknya, Nginx lebih unggul dalam menangani lebih banyak permintaan pada beban ringan hingga sedang, menunjukkan efisiensinya dalam situasi dengan tingkat permintaan tinggi tetapi stabilitas tidak menjadi perhatian utama.



Gambar 7. Perbandingan Total Request dan Failed Request



Gambar 8. Perbandingan Total Request

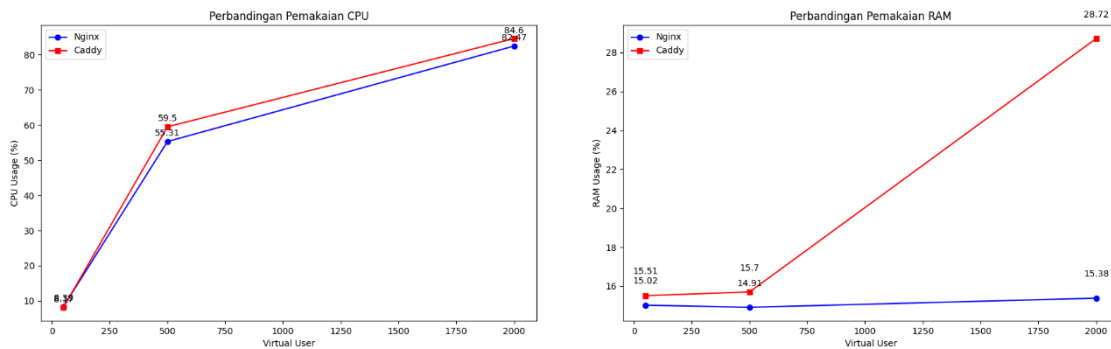
3.4. Sumber Daya

Hasil pengujian menggunakan sumber daya pada Tabel 4, menunjukkan bahwa Nginx lebih efisien dalam penggunaan CPU dibandingkan Caddy di semua skenario beban kerja. Pada kondisi awal saat aplikasi tidak aktif, tingkat pemakaian CPU yaitu 3.72%, dan pemakaian RAM sebesar 15.18%. Namun, pada beban kerja aktif, Nginx secara konsisten menggunakan CPU lebih sedikit dibandingkan Caddy. Pada beban kerja ringan (50 VU), perbedaan pemakaian CPU antara keduanya kecil dan tidak signifikan. Namun, pada beban kerja sedang (500 VU) dan berat (2000 VU), Nginx menggunakan CPU masing-masing sebesar 55.31% dan 82.47%, dibandingkan dengan Caddy yang mencapai 59.50% dan 84.60%. Hal ini menunjukkan keunggulan arsitektur event-driven Nginx dalam menangani beban kerja tinggi secara efisien.

Tabel 4. Metrik sumber daya

Virtual User	Web Server	CPU (%)	RAM (%)
50	Nginx	8.39	15.02
	Caddy	8.17	15.51
500	Nginx	55.31	14.91
	Caddy	59.50	15.70
2000	Nginx	82.47	15.38
	Caddy	84.60	28.72

Dalam hal penggunaan RAM, kedua web server menunjukkan pola yang berbeda. Saat aplikasi tidak aktif, konsumsi RAM awalnya sama, yaitu 15.18%. Pada beban kerja ringan dan sedang, Nginx menunjukkan efisiensi yang lebih baik dibandingkan Caddy, dengan pemakaian RAM Nginx cenderung lebih rendah. Pada beban berat (2000 VU), pemakaian RAM oleh Caddy meningkat signifikan hingga 28.72%, jauh lebih tinggi dibandingkan Nginx yang hanya menggunakan 15.38%.



Gambar 9. Perbandingan Pemakaian CPU dan RAM

4. KESIMPULAN

Penelitian ini menunjukkan bahwa Nginx memiliki keunggulan dalam efisiensi kinerja dibandingkan Caddy dalam menjalankan aplikasi PHP, terutama pada skenario beban kerja ringan hingga sedang. Nginx secara konsisten mencatat latensi dan waktu respon yang lebih rendah, serta *throughput* yang lebih tinggi dibandingkan Caddy, meskipun pada beban kerja berat Caddy menunjukkan keunggulan dalam stabilitas dengan tingkat keberhasilan permintaan yang lebih baik. Dari segi penggunaan sumber daya, Nginx lebih efisien dalam penggunaan CPU dan RAM di semua tingkat beban kerja, sementara Caddy menunjukkan peningkatan konsumsi RAM yang signifikan pada beban berat. Keseluruhan hasil ini menunjukkan bahwa Nginx lebih cocok untuk aplikasi yang membutuhkan efisiensi tinggi dan mampu menangani lalu lintas besar, sedangkan Caddy lebih sesuai untuk aplikasi yang memprioritaskan keberhasilan koneksi tanpa kegagalan permintaan, terutama dalam skenario beban kerja berat.

5. SARAN

Untuk penelitian selanjutnya, disarankan agar pengujian diperluas dengan skenario yang lebih kompleks, seperti integrasi dengan sistem *cache* dan penggunaan *database* dengan kueri yang lebih berat, sehingga lebih mencerminkan kondisi aplikasi web nyata. Selain itu, evaluasi juga dapat mencakup aspek keamanan, seperti kemampuan mitigasi serangan DDoS, untuk menilai sejauh mana kedua *web server* dapat memberikan perlindungan terhadap ancaman *cyber* dalam berbagai skenario.

DAFTAR PUSTAKA

- [1] W3Techs, "Usage Statistics And Market Share Of Server-Side Programming Languages For Websites, November 2024." Accessed: Nov. 30, 2024. [Online]. Available: https://W3techs.Com/Technologies/Overview/Programming_Language
- [2] P. Adidtya, N. A. Farhana, P. Harliana, And I. M. K. Karo, "Web-Based Application Development Using PHP-Native Framework On Agent Of Change Integrity Zone

-
- Information System | Sinkron : Jurnal Dan Penelitian Teknik Informatika.” Accessed: Nov. 30, 2024. [Online]. Available: <https://jurnal.polgan.ac.id/index.php/sinkron/article/view/14118>
- [3] R. Palupi And A. E. Prasetya, “Pengaruh Implementasi Content Management System Terhadap Kecepatan Kinerja Menggunakan One Way Anova,” *J. Ilm. Inform.*, Vol. 10, No. 01, Art. No. 01, Mar. 2022, Doi: 10.33884/Jif.V10i01.4445.
- [4] Augustine And S. Seimahuira, “Penerapan Metode SAW Dalam Analisa Perbandingan Performa Web Server (Apache, Nginx, Lighttpd, Iis) Pada Bahasa Pemrograman PHP | REMIK: Riset Dan E-Jurnal Manajemen Informatika Komputer.” Accessed: Nov. 30, 2024. [Online]. Available: <https://jurnal.polgan.ac.id/index.php/remik/article/view/12075>
- [5] A. Y. Chandra, “Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request,” *J. Sist. Dan Inform. JSI*, Vol. 14, No. 1, Art. No. 1, Nov. 2019, Doi: 10.30864/Jsi.V14i1.248.
- [6] P. Dymora, M. Mazurek, And B. Sudek, “Comparative Analysis Of Selected Open-Source Solutions For Traffic Balancing In Server Infrastructures Providing WWW Service.” Accessed: Nov. 30, 2024. [Online]. Available: <https://www.mdpi.com/1996-1073/14/22/7719>
- [7] Caddy Web Server, “Automatic HTTPS - Caddy Documentation.” Accessed: Dec. 01, 2024. [Online]. Available: <https://caddyserver.com/>
- [8] P. S. Saputra, P. A. Pratama, And L. P. A. S. Tjahyanti, “Perancangan Dan Komparasi Web Server Nginx Dengan Web Server Apache Serta Pemanfaatan Reverse Proxy Server Pada Nginx,” *KOMTEKS*, Vol. 2, No. 1, Art. No. 1, Oct. 2023, Accessed: Mar. 25, 2025. [Online]. Available: <https://ejournal.unipas.ac.id/index.php/komteks/article/view/1307>
- [9] Busran And A. Ridwan, “Analisis Perbandingan Performa Apache Web Server Dan Nginx Menggunakan Apache Jmeter,” *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, Vol. 8, No. 2, Art. No. 2, Oct. 2020, Doi: 10.21063/Jtif.2020.V8.2.87-92.
- [10] D. Pradana, T. A. Cahyanto, R. Umilasari, H. Oktavianto, And A. M. Zakiyyah, “Perbandingan Kinerja Web Server Terhadap Performa Integrasi Multi Basis Data,” *BIOS J. Teknol. Inf. Dan Rekayasa Komput.*, Vol. 3, No. 2, Art. No. 2, Sep. 2022, Doi: 10.37148/Bios.V3i2.41.
- [11] M. D. Adila And T. Y. Hadiwandura, “Peningkatan Kinerja Dan Skalabilitas Website E-Commerce Menggunakan Load Balancing,” *J. Teknol. Inform. Dan Komput.*, Vol. 10, No. 2, Art. No. 2, Sep. 2024, Doi: 10.37012/Jtik.V10i2.2183.
- [12] H. H. Maila, D. Indra, And R. Satra, “Analisis Perbandingan Layanan Data Server Menggunakan Failover Cluster Pada Platform Nginx Dan Apache,” *Bul. Sist. Inf. Dan Teknol. Islam*, Vol. 1, No. 2, Art. No. 2, May 2020, Doi: 10.33096/Busiti.V1i2.829.
- [13] Grafana, “K6 Documentation,” Grafana Labs. Accessed: Dec. 01, 2024. [Online]. Available: <https://grafana.com/docs/k6/latest/>
- [14] F. Risqulla, C. Setianingsih, And A. L. Prasasti, “Evaluating The Performance Of Restful Apis Under Large HTTP Requests With K6,” *Eproceedings Eng.*, Vol. 11, No. 6, Art. No. 6, Dec. 2024.
- [15] Hetrixtools, “Server Monitor.” Accessed: Dec. 01, 2024. [Online]. Available: <https://hetrixtools.com/feature/server-monitor/>
-