



ISSN : 2339 - 1871

## BETRIK BESEMAH TEKNOLOGI INFORMASI & KOMPUTER

Editor Office : Pusat Penelitian & Pengabdian Pada Masyarakat  
(PPPM) ITPA

Phone : 0857-9716-9578

email : [betriktpa@itpa.ac.id](mailto:betriktpa@itpa.ac.id)

### Meningkatkan Kinerja WordPress Melalui Optimasi Arsitektur Memori: Analisis Kuantitatif

Guntoro Barovich<sup>1</sup>, Molavi Arman<sup>2</sup>

Teknologi Informasi, Universitas Internasional Jakarta, Bekasi, Indonesia<sup>1</sup>

Manajemen Informatika, Fakultas Ilmu Komputer dan Rekayasa (FIKR),

Universitas Multi Data Palembang, Palembang, Indonesia<sup>2</sup>

Sur-el : [guntoro@jiu.ac.id](mailto:guntoro@jiu.ac.id)<sup>1</sup>, [\\*molavi.arman@mdp.ac.id](mailto:molavi.arman@mdp.ac.id)<sup>2</sup>

Penulis Korespondensi: Molavi Arman, [molavi.arman@mdp.ac.id](mailto:molavi.arman@mdp.ac.id)

**Abstrak:** Kinerja halaman *web* sangat penting untuk mempertahankan pengalaman pengguna dan mendukung keberhasilan layanan digital. Fokus penelitian ini adalah untuk mengetahui bagaimana kinerja layanan *web* menangani berbagai beban permintaan serta untuk menemukan komponen teknis yang paling signifikan yang memengaruhi kualitas layanan. Tujuan utama penelitian ini adalah untuk melihat waktu respons, stabilitas sistem, dan penggunaan sumber daya server ketika peningkatan permintaan secara bertahap meningkat, baik dalam kondisi puncak maupun lambat. Metode eksperimental berbasis simulasi beban (*load testing*) digunakan untuk melakukan pengujian. Skenario bertingkat melibatkan pengiriman permintaan HTTP dalam jumlah tertentu dan pengukuran otomatis pada setiap tahap. Selain penggunaan CPU dan memori, variabel yang diamati termasuk latency (min, rata-rata, maksimum, dan standar deviasi), respons waktu, waktu *round trip* (RTT), dan HTTP *request per second* (rata-rata RPS dan peak RPS). Selain itu, dua metrik pengalaman pengguna, *Largest Contentful Paint* (LCP) dan *Page Load Time*, dipelajari terkait dengan persepsi kecepatan dan kemungkinan peningkatan *bounce rate*. Hasil pengujian TTFB menunjukkan peningkatan yang signifikan pada server pengolahan dari 300–399 ms menjadi 239–247 ms setelah optimalisasi. RTT untuk TCP bolak balik di jaringan juga meningkat signifikan dari 64 ms menjadi 6–40 ms. Selanjutnya, *latency* turun hingga 6-7% dari 757-759ms sebelum optimalisasi. Walaupun pengujian dengan 20VUs dan 1200 request dilakukan, penggunaan memori dan CPU tetap stabil. Peak load turun menjadi 49-50% dari hasil pengujian p95 dan p99. Hasilnya menunjukkan bahwa penerapan dan optimalisasi memcache dan MySQLTuner memiliki dampak yang signifikan terhadap responsivitas aplikasi dan persepsi pengguna yang lebih baik tentang sistem yang diakses.

**Kata kunci :** *Web Performance, Latency Analysis, Load Testing, Round Trip Time (RTT), WordPress Optimization.*

**Abstract:** *The performance of a web page plays a crucial role in ensuring a smooth user experience and directly influences the success of digital services. This study aims to understand how a web service responds when subjected to varying traffic loads and to identify the technical factors that most significantly shape*

Received: 21-11-2025 | Accepted: 28-11-2025 | Published Online: 30-12-2025

All author: Guntoro Barovich, Molavi Arman

*service quality. The main focus is to evaluate response time, system stability, and server resource utilization as demand gradually increases, both during peak traffic and lighter load conditions. The research applies an experimental approach using load-testing techniques with a cascading scenario, where a predefined number of HTTP requests is sent and measured at each stage. The observed variables include CPU and memory consumption, latency values (minimum, average, maximum, and standard deviation), response time, round trip time (RTT), and throughput measured through average and peak HTTP requests per second (RPS). Additionally, two user-experience indicators—Largest Contentful Paint (LCP) and Page Load Time—are analyzed to understand their relationship with perceived speed and potential changes in bounce rate. The findings show notable improvements after optimization was applied. The server's TTFB decreased significantly from 300–399 ms to 239–247 ms. Network RTT also improved, dropping from 64 ms to between 6 and 40 ms. Overall latency declined by about 6–7% from the initial range of 757–759 ms. Even under a test scenario involving 20 virtual users and 1,200 total requests, CPU and memory usage remained stable, while peak load decreased to about 49–50% based on p95 and p99 metrics. These results indicate that implementing Memcache and MySQLTuner contributes substantially to improving application responsiveness and enhancing users' perception of system performance.*

**Keywords:** *Web Performance, Latency Analysis, Load Testing, Round Trip Time (RTT), WordPress Optimization*

## 1. PENDAHULUAN

Beberapa tahun terakhir, pertumbuhan teknologi web telah menunjukkan peningkatan yang signifikan dalam jumlah pengguna dan kompleksitas layanan yang dibangun di atasnya. Website sekarang tidak hanya berfungsi sebagai media informasi; itu juga dapat digunakan sebagai platform untuk transaksi, sistem manajemen, pembelajaran, dan ruang publik digital di mana orang berinteraksi satu sama lain hampir setiap saat. Performa web menjadi bagian penting dari kualitas layanan digital karena jumlah pengguna yang meningkat dan berbagai aktivitas yang dilakukan di dalamnya. Di dunia digital yang serba cepat ini, WordPress muncul sebagai salah satu CMS yang paling populer. Lebih dari 40% website dunia menggunakan WordPress, mulai dari blog personal, portal berita, sampai website pemerintahan dan e-commerce[1], [2]. Popularitas WordPress memang tidak terlepas dari kemudahan penggunaannya, ketersediaan plugin yang melimpah, serta fleksibilitas untuk dikembangkan sesuai kebutuhan. Namun, di balik berbagai kelebihannya, WordPress tetap menyimpan tantangan klasik, yaitu masalah performa[3].

Ketika halaman *web* membutuhkan waktu lebih dari 300–500 ms untuk dimuat, sebagian pengguna mulai merasakan adanya keterlambatan. Jika waktu respon melebihi 1 detik, tingkat *bounce rate* meningkat hingga lebih dari 30% [4]. Dampaknya tidak hanya pada pengalaman pengguna, tetapi juga pada SEO, konversi bisnis, dan efisiensi server. Dalam banyak kasus, performa WordPress menjadi lambat bukan karena faktor kode saja, melainkan karena arsitektur WordPress yang sangat bergantung pada eksekusi PHP dan query ke database MySQL[5]. Pada skala kecil pun, WordPress sering kali menghasilkan banyak query, sehingga setiap permintaan halaman membutuhkan proses dinamis yang memadai[6], [7]. Untuk mengatasi persoalan tersebut, berbagai pendekatan optimasi telah dikembangkan. Dua pendekatan yang banyak direkomendasikan oleh praktisi sistem adalah penerapan caching, salah satunya melalui Memcached [8], serta optimasi database menggunakan alat seperti MySQLTuner [9]. [10] Memcached mengurangi akses

langsung ke *database* sehingga mempercepat *dynamic content rendering*, hal ini bisa disebabkan oleh banyaknya *autoloaded option* dan *plugin* yang diaktifkan [11]. *Caching* memungkinkan halaman atau objek tertentu disimpan sementara di memori, sehingga mengurangi kebutuhan server untuk menghasilkan halaman secara dinamis. Banyak cara untuk bisa meningkatkan respon dari suatu halaman web seperti CMS, salah satunya dengan mengaktifkan dan menggabungkan fitur redis dan *memcache* pada CMS wordpress [11], selain itu mampu menurunkan *Time to First Byte* (TTFB) dan database load secara signifikan. Selain penerapan *memcache*, *MySQLTuner* membantu mengidentifikasi *bottleneck* seperti *buffer size* yang terlalu kecil atau query yang tidak efisien, selain itu *bottleneck* utama ada pada *query database* dan *object rendering* [12]. *MySQLTuner* berfungsi menganalisis konfigurasi database dan memberikan rekomendasi agar MySQL bekerja secara lebih efisien.

Meski kedua pendekatan ini sering disebut-sebut efektif, performanya tidak bersifat universal. Efektifitasnya sangat bergantung pada konteks server, jenis konten, jumlah request, arsitektur WordPress, serta cara implementasinya. Oleh karena itu, penelitian eksperimental diperlukan untuk memvalidasi apakah kombinasi *Memcached* dan *MySQLTuner* benar-benar dapat meningkatkan performa WordPress dalam konteks implementasi riil. Penelitian ini sendiri difokuskan pada pengamatan performa WordPress sebelum optimasi (baseline) dan sesudah optimasi (base optimization). Fokus pada baseline ini penting untuk memastikan bahwa setiap perubahan performa setelah optimasi nantinya dapat dibandingkan secara terukur. Pengukuran baseline dilakukan menggunakan beberapa parameter yang lazim digunakan dalam uji performa web, seperti waktu Responsiveness Metrics (*Time to First Byte* (TTFB), *Round Trip Time*, *Avg Latency*, *Standar Deviation* (STDDEV)), *Stability Metrics* (P95, P99, *Request Failed*), *Throughput Metrics* (*HTTP Req/s* (Peak RPS)), *Avg RPS*), *Resource Utilization* (*CPU Usage*, *Memori Usage*), *Frontend Rendering Performance metrics* (*Large Contentful Paint* (LCP), *Page Load Time* (PLT)). Untuk melakukan evaluasi latensi atau mengukur performa dari suatu web, bisa menggunakan K6 dengan menggunakan ribuan request dalam satu aktifitas pengujian, dimana parameter yang digunakan dalam tools ini adalah AVG P95, P99 latencies[13], dan penggunaan P95/P99 lebih akurat untuk real-user experience dibanding sekadar average. *SpeedVitals* untuk memvalidasi performa dari perspektif eksternal dan *wireshark* untuk melihat pergerakan *Round tripe time* setiap paket yang dikirimkan dan *time rendering* di sisi pengguna.

Penelitian ini diharapkan dapat memberikan pemahaman yang lebih mendalam tentang komponen yang membentuk kinerja WordPress dan memperkuat penelitian tentang teknik pengoptimalan server dan web performance engineering. Dengan memahami kondisi dasar kinerja sebelum optimasi, penelitian ini diharapkan dapat menjadi pijakan yang kuat bagi analisis lebih lanjut dan memberikan kontribusi.

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental kuantitatif, dengan melakukan pengukuran *performa wordpress* pada dua kondisi utama, yaitu kondisi *baseline* (tanpa optimasi) dan kondisi *post-optimization* [14], [15]. Fokus utama pada tahapan ini adalah mengukur performansi dasar wordpress sebelum dan sesudah, untuk memastikan evaluasi tahapan berikutnya memiliki landasan yang data yang kuat dan *valid*. Model penelitian mengacu pada prinsip *controlled performance benchmarking* pada lingkungan server terisolasi (Ferguson, 2020). Difokuskan pada pengukuran waktu muat halaman dan *respons server* dalam kondisi beban terkontrol seperti pengukuran waktu *Responsiveness Metrics* (*Time to First Byte* (TTFB), *Round Trip Time*, *Avg Latency*, *Standar Deviation* (STDDEV)), *Stability Metrics* (P95, P99, *Request Failed*), *Throughput Metrics* (HTTP Req/s (*Peak RPS*)), *Avg RPS*), *Resource Utilization* (*CPU Usage*, *Memori Usage*), *Frontend Rendering Performance metrics* (*Large Contentful Paint* (LCP), *Page Load Time* (PLT)) dibawah skenario yang sudah ditentukan.

### 2.1 Lingkungan Penelitian (*Enviromnetal Setup*)

Skenario lingkungan penelitian ini di bagi menjadi 2 kriteria, yaitu spesifikasi server dan kondisi pengujian meliputi kondisi *baseline* dan kondisi *base-optimization*. Spesifikasi server terdiri dari Linux debian sebagai sistem operasi, *apache* sebagai *web server*, *php* versi 8.x, *mariaDB* sebagai *database* dan *wordpress* terbaru yang digunakan sebagai target penelitian ini. Sedangkan tindakan yang dilakukan pada kondisi *baseline*, diantaranya belum dilakukan implementasi dan konfigurasi baik *memcache*, *page caching*, *object caching*, *tuning database*, pengaturan *PHP-FPM* dan hanya menerapkan *plugin wordpress* standar. Hal ini dilakukan untuk mendapatkan *performa baseline* yang apada adanya. Sedangkan pada kondisi *base optimization*, dilakukan implemnetasi baik *memcache*, *MySQLTunner* dengan konfigurasi standar penyesuaian pada *page caching* dan *object caching*.

### 2.2 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini, meliputi variabel independen, variabel dependen. variabel independen terdiri dari

Tabel 1. Variabel Independen

Kode	Variabel Independen	Deskripsi
VI1	Konfigurasi server default	Server Berjalan tanpa <i>memcache</i> dan <i>MySQL Tuning</i>
VI2	Konfigurasi server <i>base optimization</i>	Server berjalan dengan penerpaan <i>memcache</i> dan <i>MySQL Tunnig</i>
VI3	Jumlah Request pada Load Test	Total Permintaan yang dikirimkan melalui K6
VI4	Jumlah Virtual User (Vus)	Jumlah Pengguna simulasi dalam pengujian

Sedangkan variabel pengukuran yang digunakan dalam penelitian ini, meliputi variabel pengukuran yang terdiri, anantara lain:

Tabel 2. Variabel Pengukuran

<b>Variabel Responsivitas (Responsiveness Metrics)</b>			
<b>Kode</b>	<b>Variabel Dependensi</b>	<b>Satuan</b>	<b>Deskripsi</b>
VP1	<i>Time to First Byte (TTFB)</i>	ms	Waktu dari request dikirim hingga byte pertama diterima
VP2	<i>Round Trip Time</i>	ms	Waktu perjalanan paket dari client ke server dan kembali
VP3	MIN <i>Latency</i>	ms	Waktu tercepat yang tercatat antara pengiriman permintaan dan penerimaan respons dari server.
VP4	<i>Avg Latency</i>	ms	Rata-rata waktu respon server untuk seluruh request yang berhasil diproses
VP5	<i>Max Latency</i>	ms	Waktu respon paling lambat
VP6	STDDEV	ms	Ukuran penyebaran atau variasi latensi terhadap nilai rata-rata.
<b>Variabel Stabilitas Sistem (Stability Metrics)</b>			
VP7	P95 Latency	ms	Latency yang dialami oleh 95% request tercepat
VP8	P99 Latency	ms	Latency terburuk yang masih termasuk 99% request
VP9	Request Failed	Req	Indikator kestabilan server
<b>Variabel Kapasitas Pemrosesan (Throughput Metrics)</b>			
VP10	HTTP Req/s (Peak RPS)	Req/s	Jumlah maksimum permintaan HTTP yang dapat diproses server dalam satu detik pada saat beban mencapai titik tertinggi.
VP11	Avg RPS	Req/s	Rata-rata jumlah permintaan HTTP yang diproses server setiap detik
<b>Variabel Pemanfaatan Sumber Daya (Resource Utilization)</b>			
VP12	CPU Usage	%	Persentase pemakaian kapasitas CPU yang digunakan server dalam memproses permintaan
VP13	Memori Usage	%	Persentase pemakaian kapasitas Memori yang digunakan server dalam memproses permintaan
<b>Variabel Frontend Rendering Performance Metrics</b>			
VP14	Large contentful Paint	sec	Waktu yang dibutuhkan untuk menampilkan content gambar dan lainnya
VP15	Page Load Time	ms	Waktu total halaman dimuat berdasarkan SpeedVitals

### 2.3 Skenario pengujian

Guna mendapatkan hasil pengujian yang baik, maka dilakukan skenario pengujian yang sama baik pada baseline ataupun pada *base-optimization*. Adapun skenario pengujian, antara lain:

- Jumlah *user* yang digunakan dalam pengujian adalah sebanyak 20 *Virtual Users*.
- Pengujian dilakukan 3 kali
- Total permintaan pengujian yang direncanakan sebanyak  $\pm 1200$  permintaan
- Durasi waktu pengujian selama 3 menit 30 detik
- Perangkat lunak yang digunakan untuk pengujian ini antara lain k6 load testing, digunakan untuk mengukur latency (P95, P99, MIN-MAX, AVG, RPS, STDDEV, CPU dan memory), speedvitals digunakan untuk mengukur TTFB, LCP, PLT. Sedangkan wireshark digunakan untuk mengukur RTT side-to-side

Tahapan dalam penelitian ini dapat dilihat pada gambar 1, berikut ini.



Gambar 1. Tahapan penelitian

### 3. HASIL DAN PEMBAHASAN

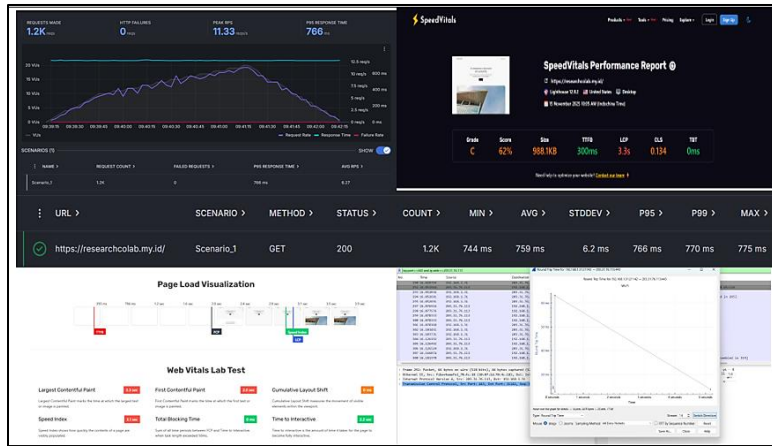
#### 3.1 Hasil

Setelah dilakukan pengujian baik pada *baseline* dan *base-optimization* dengan skenario yang sudah ditetapkan didapatkan data pengujian sebagai berikut.

Tabel 3. Rekapitulasi hasil pengujian baseline dan base-optimization

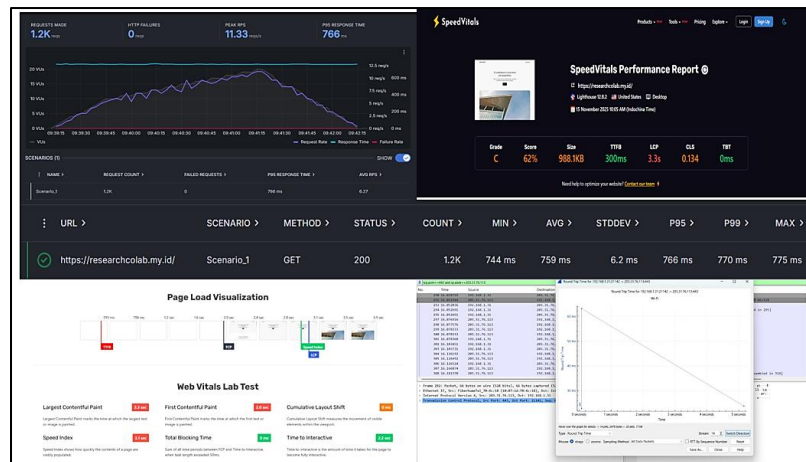
Kode	Variabel	Satuan	Baseline			Base Optimization		
			1	2	3	1	2	3
VP01	TTFB	ms	300	344	399	247	239	245
VP02	RTT	ms	64,5	64,5	64,5	6 (150ms)	6 (220ms)	40.85 (950ms)
VP03	MIN	ms	744	745	745	699	701	699
VP04	Avg Latency	ms	759	759	757	712	713	712
VP05	MAX	ms	775	781	838	719	858	719
VP06	STDDEV	ms	6,2	7,7	8	7,2	7,8	6,7
VP07	P95 Latency	ms	766	766	766	717	721	719
VP08	P99 Latency	ms	770	770	770	719	721	719
VP09	Request failed	reqs	0	0	0	0	0	0
VP10	HTTP Req/s (PEAK RPS)	reqs/s	11,3	11,6	11,7	11,6	11,6	11,6
VP11	AVG RPS	reqs/s	6,27	6,28	6,28	6,44	6,43	6,44
VP12	CPU	%	2,49	2,49	2,63	2	3	2,99
VP13	Memori	%	2,52	2,58	2,99	2,59	2,44	2,58
VP14	LCP	sec	3,3	3	3,2	3	3,1	2,6
VP15	PLT	ms	1,5	1,5	1,5	1,5	1,5	1,5

Gambar 2 memperlihatkan bukti pengujian menggunakan speedvitals, k6 dan wireshark pada saat melakukan pengujian baseline.



Gambar 2. Bukti hasil pengujian baseline menggunakan k6, speedvitals dan wireshark

Gambar 3 memperlihatkan bukti hasil pengujian menggunakan speedvitals, k6 dan wireshark pada saat melakukan pengujian base optimization.



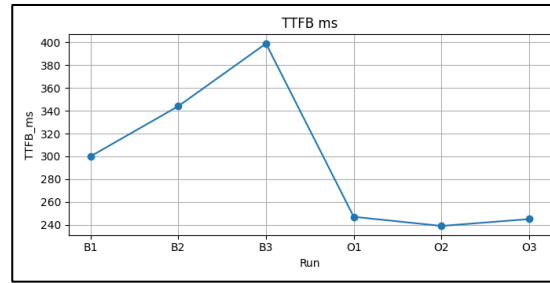
Gambar 3. Bukti hasil pengujian base optimization menggunakan k6, speedvitals dan wireshark

### 3.2 Pembahasan

Tabel 3 merupakan hasil rekapitulasi pengujian yang telah dilakukan sebanyak 3 kali baik pada pengujian *baseline* maupun pada *base optimization*. Metrik utama yang dianalisis mencakup TTFB (*Time To First Byte*), LCP (*Largest Contentful Paint*), PLT (*Page Load Time*), *latency* (rata-rata, p95, p99), *throughput* (RPS), RTT (*Round Trip Time*), serta utilisasi sumber daya (CPU dan memori). Dari hasil ini didapatkan beberapa informasi diantaranya:

#### 1. *Time to First Byte (TTFB)*

Pengujian pada kondisi *baseline*, TTFB berada di rentang 300-399 ms dan terjadi penurunan setelah dilakukan *optimization* dengan melakukan penerapan *memcache* dan *mysql tuner* data TTFB menurun menjadi 239-247 ms. Terjadi perbaikan yang signifikan pada proses *server processing* dan *network handshake*. Seperti tampak pada gambar 4.

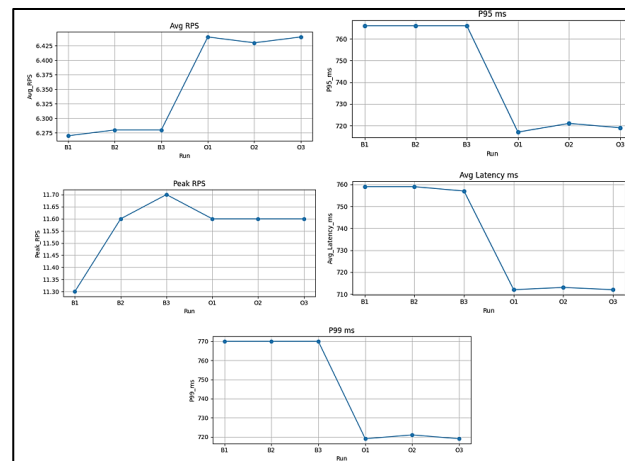


Gambar 4. Hasil pengujian TTFB baseline dan setelah dioptimasi

## 2. Round Trip Time

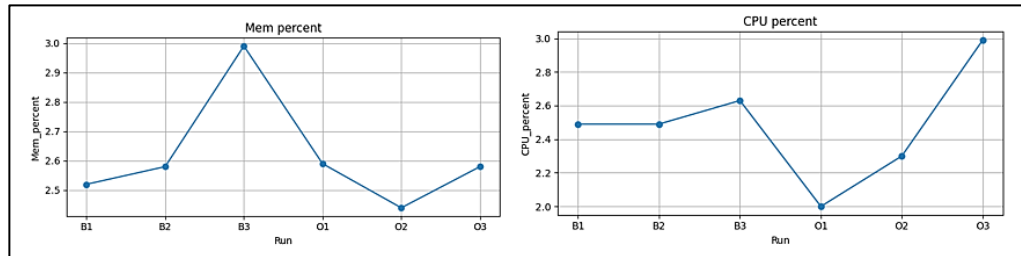
RTT rendah setelah dilakukan optimization dari 64.5 ms menjadi 6-40 ms waktu yang diperlukan segmen TCP untuk melakukan bolak-balik di jaringan dan waktu tiba paket antara 950 ms – 1.5 s (end-to-end arrival time) mulai dari traversing di jaringan publik dan proses render awal disisi klient

- Latency rata-rata** : terjadi penurunan latency dimana pada saat pengukuran baseline didapatkan 757-759 ms dan setelah optimization antara 712-713 ms penurunan 6-7% dan mengindikasikan efisiensi pada penerapan memcache. Tanpa pada gambar 5.
- Stabilitas sistem (peak load)**: pada pengujian baseline di parameter p95 dan p99 didapatkan data di angka 766-770 ms dan pada saat dilakukan optimization terjadi penurunan di angka 717-721 ms penurunan yang konsisten  $\pm$  49-50 %. Yang menandakan tail latency membaik, server stabil pada saat menerima lonjakan permintaan dan tidak terdapatnya request yan failure. Sehingga bisa dikatakan sistem memiliki benan toleransi yang baik. Tanpa pada gambar 5.
- RPS konsisten meningkat** : rata-rata RPS menunjukkan perbaikan kecil tetapi stabil jika dilihat dari pengujian baseline dimana rata-rata didapatkan 6.27-6.28 req/s dan pada saat telah dioptimasi 6.43-644 req/s dan peak RPS di kisaran 11.6-11.7 req/s, ini menunjukkan perbaikan lebih pada sisi delay dibandingkan troughput. Tanpa pada gambar 5.



Gambar 5. Hasil Avg RPS, P95, P99, Avg Latency dan Peak RPS

6. **Pemanfaatan sumber daya:** penggunaan CPU pada baseline berada pada rentang nilai 2.49-2.63% sedangkan sesudah optimisasi turun menjadi 2.0-2.99% dan ini tergantung dari kondisi. Penurunan ini mencerminkan efisiensinya pada proses rendering konten dan pengelolaan cache dan tidak terjadi bottleneck jika dilihat dari stabilnya penggunaan memory di 2.52-2.99%. Tampak pada gambar 6.



Gambar 6. Pengukuran CPU dan memori

Temuan utama dari seluruh dataset tersebut menyatakan bahwa terjadi responsivitas yang meningkat secara signifikan jika dilihat dari TTFB dan latency. Selain itu stabilitas sistem menjadi lebih baik, ini bisa dilihat dari terjadinya penurunan P95 dan P99 serta tidak disertai adanya request failure pada saat pengujian yang menandakan bahwa sistem stabil di bawah beban konstan 20 Vus dan 1200 permintaan. RTT sangat rendah, tetapi arrival time disisi pengguna lebih tinggi, ini bisa disebabkan oleh faktor eksternal seperti jaringan public, routing dan jitter di sisi pengguna. Dan penggunaan CPU menurun, yang menunjukkan optimasi yang dilakukan memiliki dampak positif pada efisiensi server. Jika dilihat dari perspektif reliabilitas, seluruh pengujian tidak menghasilkan request failure (tabel 3), dan utilisasi CPU/memori server tetap rendah (CPU  $\approx$ 2-3%, memori <3%) pada gambar 4, yang lebih lanjut menunjukkan bahwa server tidak CPU-bound atau memory-bound pada skala beban yang diuji. Secara implikatif, bottleneck dan peluang perbaikan besar justru terletak pada: (1) optimasi lapisan transport/HTTP (mis. keep-alive, TLS session reuse, pengaturan TCP), (2) pengurangan berat front-end (gambar, blok JS/CSS) untuk memperbaiki LCP/PLT, dan (3) stabilisasi rute distribusi (CDN/edge) atau peningkatan kualitas koneksi klien bila target pengguna tersebar di geografis yang berbeda.

## 4. KESIMPULAN DAN SARAN

### 4.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan pada dua kondisi pengujian yaitu pengujian pada kondisi baseline dan kondisi base optimization menggunakan tools yang sudah dipersiapkan yaitu k6, *speedvitals* dan *wireshark*, bisa disimpulkan bahwa:

1. Optimasi memberikan peningkatan performa yang konsisten, berdasarkan hasil pengujian yang disajikan dari parameter TTFB, Rata-rata Latency, P55 dan P99.

2. Server mampu mempertahankan stabilitas beban pada saat dilakukan pengujian menggunakan 20 VUs dimana total permintaan yang dilakukan sebanyak  $\pm 1200$  request tanpa terjadi failure sama sekali.
3. RTT server berada pada kondisi yang ideal yaitu (6-40 ms) sehingga faktor keterlambatan sebesar 950ms (dari hasil sample) dan bahkan bisa lebih. Ini dipengaruhi oleh kondisi jaringan di sisi pengguna bukan dari sisi performa server.
4. Efisiensi Penggunaan sumber daya meningkat, terjadi penurunan konsumsi CPU setelah dilakukan optimasi yaitu penerapan dan penyesuaian/tuning di sisi memcache dan MySQLTuner .
5. Secara keseluruhan. Optimasi yang dilakukan memberikan dampak positif terhadap responsifitas dan pengalaman pengguna.

#### 4.2 Saran

Beberapa bagian yang bisa menjadi pengembangan kedepan agar performa server menjadi lebih baik maka perlu memperhatikan beberapa hal berikut ini:

1. Integrasi agresif CDN dan cache diharapkan dapat menurunkan LCP dan TTFB karena keterlambatan jaringan publik.
2. Penggunaan skenario pengguna antara 50-200 VUs diperlukan untuk menentukan batas kapasitas sistem.
3. Untuk melacak distribusi delay di berbagai wilayah, maka perlu integrasikan sistem pengukuran langsung dari pengguna melalui tools monitoring yang terkontrol.
4. Rekonfigurasi sisi server dan indeksasi database untuk menekan latency, karena dalam pengujian ini Latency masih lebih dari 700 milidetik sehingga masih dapat menyebabkan bottleneck pada query database atau plugin WordPress.
5. Menetapkan SLO seperti TTFB < 250 ms dan P95 < 700 ms agar pengujian memiliki konteks operasional.

#### 5. UCAPAN TERIMA KASIH

Terima kasih kepada Bapak Molavi Arman, Universitas Multi Data Palembang yang telah bersedia dalam menyediakan *space server* sebagai objek penelitian sehingga penelitian ini bisa dilaksanakan dan diselesaikan dengan baik.

#### DAFTAR RUJUKAN

- [1] A. Farisi, A. Aryo P, and Mulyati, “Pelatihan Pembuatan Website Tanpa Coding Dengan Memanfaatkan Content Management System Wordpress Untuk Umum Dan Mahasiswa Kampus MDP,” *Jurnal PkM Pemberdayaan Masyarakat*, vol. 2, no. 3, pp. 74–83, Jul. 2021, doi: 10.56327/jurnalpkm.v2i3.29.

- [2] D. T. Murphy, M. F. Zibrán, and F. Z. Eishita, "Plugins to Detect Vulnerable Plugins: An Empirical Assessment of the Security Scanner Plugins for WordPress," in *2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA)*, IEEE, Jun. 2021, pp. 39–44. doi: 10.1109/SERA51205.2021.9509274.
- [3] G. Sharma, A. Kute, N. Jadhav, N. Kolhe, and A. Tiwari, "Review of Web Content Management Systems and their Increasing Demand in Market," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 1231–1235, Dec. 2022, doi: 10.22214/ijraset.2022.48167.
- [4] D. Kamerer, "Reconsidering bounce rate in web analytics," *Journal of Digital & Social Media Marketing*, vol. 8, no. 1, p. 58, Jun. 2020, doi: 10.69554/EJGO6858.
- [5] I. T.; Sitohang and D. A. A. Mubarak, "Analisis Strategi SEO dalam Pengelolaan Konten WordPress: Kasus Impactlabs Indonesia," *MAMEN: Jurnal Manajemen*, vol. 4, no. 3, pp. 358–371, Jul. 2025, doi: 10.55123/mamen.v4i3.5781.
- [6] N. K. Hulu, T. Tommy, and D. Handoko, "Optimalisasi Kinerja Cms Wordpress Melalui Penerapan Replikasi Database Mysql," *JUDIS: Jurnal Multidisiplin dan Sains*, vol. 1, no. 2, pp. 160–172, Mar. 2025, doi: 10.63854/jms.v1i2.46.
- [7] M. Sholeh, T. S. Iman Saputra, and S. Suraya, "Penerapan Laravel untuk Mengatasi Kelemahan Keamanan WordPress pada Website Badan Layanan Umum Daerah," *Jurnal SAINTEKOM*, vol. 15, no. 2, pp. 141–152, Sep. 2025, doi: 10.33020/saintekom.v15i2.938.
- [8] F. Dipraja and A. Rahman, "Penerapan Redis Cluster Meningkatkan Efisiensi Caching Arsitektur Microservices," *Intellect : Indonesian Journal of Innovation Learning and Technology*, vol. 04, no. 01, pp. 171–179, Jun. 2025, doi: 10.57255/intellect.v4i1.1445.
- [9] E. Budiman, M. Fadli, David Kurniawan, and E. R. Susanto, "Tinjauan Literatur Dengan Pendekatan Systematic Literature Review Untuk Optimasi Kueri Dalam Basis Data," *Storage: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 4, no. 2, pp. 82–91, May 2025, doi: 10.55123/storage.v4i2.5182.
- [10] I. Malavolta, K. Chinnappan, L. Jasmontas, S. Gupta, and K. A. K. Soltany, "Evaluating the impact of caching on the energy consumption and performance of progressive web apps," in *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems*, 2020, pp. 109–119.
- [11] S. S. Raweyai and I. R. Widiyari, "Performance Testing Of Academic Website Using Load Testing Method Supported By Apache Jmeter At Xyz University," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 3, pp. 721–730, May 2024, doi: 10.52436/1.jutif.2024.5.3.1796.
- [12] C. A. Győrödi, D. V. Dumșe-Burescu, R. Ș. Győrödi, D. R. Zmaranda, L. Bandici, and D. E. Popescu, "Performance Impact of Optimization Methods on MySQL Document-Based and Relational Databases," *Applied Sciences*, vol. 11, no. 15, p. 6794, Jul. 2021, doi: 10.3390/app11156794.
- [13] A. R. Khamdani, A. R. Muslikh, and A. S. Affandi, "Comparative Analysis Of Performance And Efficiency Of Load Balancing Algorithms On Ingress Controller," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1, pp. 453–468, Mar. 2025, doi: 10.52436/1.jutif.2025.6.1.4040.
- [14] R. Zulfikar *et al.*, *Metode Penelitian Kuantitatif (Teori, Metode Dan Praktik)*. Bandung: Cv Widina Media Utama, 2024.
- [15] R. Akbar, R. A. Siroj, M. Win Afgani, and U. Islam Negeri Raden Fatah Palembang Abstract, "Experimental Research Dalam Metodologi Pendidikan," *Jurnal Ilmiah Wahana Pendidikan, Januari*, vol. 2023, no. 2, pp. 465–474, doi: 10.5281/zenodo.7579001.