JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

Analisis Kinerja Sistem Informasi Pesantren pada Shared Hosting Menggunakan Apache JMeter

Yori Adi Atma*[1], Raemon Syaljumairi[2]
[1,2] Jurusan Teknologi Informasi, Politeknik Negeri Padang, Padang
e-mail: ***[1]yori@pnp.ac.id**, [2]raemon@pnp.ac.id

Abstrak

Sistem Informasi Manajemen Pesantren (SIP) rentan terhadap lonjakan akses pengguna secara bersamaan, terutama pada periode kritis seperti pendaftaran santri baru dan pengumuman nilai. Namun, kajian empiris terkait kinerja SIP pada lingkungan shared hosting dengan keterbatasan sumber daya masih terbatas. Penelitian ini bertujuan mengevaluasi kinerja dan skalabilitas SIP berbasis web yang diimplementasikan pada layanan shared hosting dengan spesifikasi Single Core CPU dan 1 GB RAM. Metode yang digunakan adalah pendekatan eksperimental kuantitatif melalui load testing dan stress testing menggunakan Apache JMeter. Skenario pengujian dibagi menjadi tiga tingkat beban, yaitu 20, 50, dan 100 pengguna simultan yang mengakses lima halaman utama sistem. Hasil pengujian menunjukkan bahwa pada beban 20 pengguna sistem berjalan stabil dengan rata-rata waktu respons sebesar 3.452 ms, throughput 5,23 request/second, dan tingkat kesalahan 0%. Pada beban 50 pengguna terjadi penurunan performa dengan waktu respons meningkat menjadi 6.763 ms, throughput menurun menjadi 3,62 request/second, dan error rate sebesar 0,457%. Pada skenario 100 pengguna, sistem mengalami kegagalan signifikan dengan error rate mencapai 43,367% serta throughput sebesar 4,77 request/second, yang mengindikasikan keterbatasan sumber daya server. Penelitian ini menyimpulkan bahwa shared hosting hanya optimal untuk penggunaan skala kecil, serta merekomendasikan penggunaan Virtual Private Server (VPS) untuk menangani lonjakan akses secara simultan.

Kata kunci *— Load testing, stress testing, JMeter, kinerja web*

1. PENDAHULUAN

Pondok pesantren sebagai institusi pendidikan berbasis Islam yang memiliki peran strategis di Indonesia saat ini tengah bertransformasi menuju ekosistem digital. Implementasi Sistem Informasi Manajemen Pesantren (SIP) menjadi komponen penting dalam mendukung pengelolaan administrasi, mulai dari pendaftaran santri, pengelolaan data akademik, hingga sistem keuangan [1]. Pemanfaatan *framework* modern seperti Laravel yang dikombinasikan dengan basis data MariaDB mampu meningkatkan efisiensi pengembangan serta keamanan data [2]. Namun demikian, tantangan utama muncul ketika sistem dihadapkan pada kondisi nyata, yaitu lonjakan akses pengguna secara bersamaan dalam satu waktu [3].

Permasalahan utama dalam penelitian ini berkaitan dengan penggunaan infrastruktur *shared hosting* sebagai media implementasi SIP. Kondisi ini secara nyata dialami oleh sistem yang menjadi objek penelitian, yaitu SIP yang telah beroperasi secara live dan dapat diakses publik melalui https://bpn.ponpes.id, sehingga seluruh proses administrasi mulai dari pendaftaran santri hingga pengumuman nilai bergantung sepenuhnya pada ketersediaan infrastruktur shared

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

hosting tersebut. Meskipun *shared hosting* merupakan solusi yang ekonomis, layanan ini memiliki keterbatasan sumber daya yang ketat, terutama pada penggunaan *Central Processing Unit* (CPU) dan *Random Access Memory* (RAM) [4]. Keterbatasan ini seringkali tidak diimbangi dengan pemahaman yang memadai mengenai kapasitas maksimum sistem, sehingga berpotensi menyebabkan kegagalan operasional pada momen krusial seperti Penerimaan Santri Baru (PSB) maupun pengolahan nilai akademik secara serentak.

Beberapa faktor teknis turut memperparah kondisi tersebut. Pertama, *framework* Laravel yang menggunakan banyak dependensi serta Eloquent ORM cenderung memiliki konsumsi memori yang tinggi [5]. Kedua, pada lingkungan shared hosting, mekanisme pembatasan sumber daya seperti CloudLinux LVE (*Lightweight Virtual Environment*) dapat secara otomatis membatasi atau menghentikan proses ketika penggunaan CPU dan RAM melampaui ambang batas, yang umumnya ditandai dengan munculnya galat 508 *Resource Limit Reached* [4]. Ketiga, minimnya pengujian performa yang terukur menyebabkan tidak tersedianya data empiris yang dapat digunakan sebagai dasar dalam perencanaan kapasitas sistem.

Sejumlah penelitian sebelumnya telah membahas pengujian performa aplikasi web menggunakan Apache JMeter dengan berbagai pendekatan dan lingkungan infrastruktur. Penelitian oleh Noetzold et al. [6] menunjukkan bahwa optimasi konfigurasi Java Virtual Machine (JVM) mampu meningkatkan efisiensi CPU hingga 20% dan menurunkan penggunaan memori sebesar 15% pada aplikasi web dengan beban permintaan tinggi. Studi tersebut menegaskan bahwa pengelolaan sumber daya server memiliki pengaruh signifikan terhadap stabilitas dan responsivitas aplikasi web.

Penelitian lain yang dilakukan oleh Sunardi dan Suharjito [7] membandingkan performa framework Laravel dan Slim menggunakan metode *load testing* dan *stress testing* berbasis Apache JMeter. Hasil penelitian menunjukkan bahwa *framework* Laravel cenderung membutuhkan sumber daya yang lebih besar dibandingkan Slim Framework, terutama pada skenario beban tinggi. Temuan ini mengindikasikan bahwa karakteristik framework dapat memengaruhi performa sistem pada lingkungan server dengan keterbatasan sumber daya.

Selain itu, penelitian oleh Djaja dan Kurniawati [8] mengungkap bahwa penerapan infrastruktur berbasis Kubernetes mampu meningkatkan kapasitas pengguna simultan hingga 60% dibandingkan lingkungan *virtual machine* konvensional. Hasil tersebut menunjukkan bahwa pemilihan arsitektur infrastruktur memiliki peran penting dalam meningkatkan skalabilitas dan kestabilan layanan web pada kondisi trafik tinggi. Penelitian lain oleh Huang et al. [9] juga menunjukkan bahwa pengujian menggunakan Apache JMeter efektif dalam mengevaluasi performa sistem terdistribusi, khususnya dalam mengukur tingkat konkurensi, stabilitas layanan, dan ketahanan sistem terhadap beban akses tinggi.

Meskipun demikian, penelitian yang secara spesifik mengkaji kinerja Sistem Informasi Manajemen Pesantren berbasis Laravel pada lingkungan shared hosting dengan keterbatasan sumber daya yang terdefinisi yaitu *Single* Core CPU dan 1 GB RAM masih terbatas. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan melakukan evaluasi kinerja sistem melalui pendekatan eksperimental menggunakan Apache JMeter. Parameter yang dianalisis meliputi *average response time*, *throughput*, dan *error rate* berdasarkan skenario beban 20, 50, dan 100 pengguna simultan. Ketiga tingkat beban tersebut dipilih untuk merepresentasikan kondisi operasional yang berbeda, yaitu aktivitas administrasi harian, kondisi jam sibuk, hingga lonjakan akses ekstrem yang berpotensi terjadi pada momen krusial seperti Penerimaan Santri Baru (PSB), sebagaimana diuraikan lebih lanjut pada bagian Metode Penelitian. Hasil penelitian ini diharapkan dapat memberikan kontribusi berupa rekomendasi teknis dalam menentukan kapasitas infrastruktur yang optimal bagi pengembangan website di lingkungan pesantren.

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental kuantitatif dengan metode black-box testing untuk mengevaluasi kapasitas infrastruktur server tanpa melakukan modifikasi pada kode sumber aplikasi. Penelitian melibatkan tiga variabel, yaitu variabel bebas berupa jumlah pengguna simultan yang divariasikan pada tiga tingkat beban (20, 50, dan 100 pengguna), variabel terikat berupa average response time, throughput, dan error rate, serta variabel kontrol berupa alokasi sumber daya CPU dan RAM pada lingkungan shared hosting yang dijaga tetap selama pengujian berlangsung.

Tahapan penelitian yang dilakukan ditunjukkan pada Gambar 1, dimulai dari identifikasi arsitektur sistem dan lingkungan pengujian. Tahapan selanjutnya terdiri dari penentuan pendekatan evaluasi kinerja, perancangan skenario dan implementasi pengujian menggunakan Apache JMeter, pelaksanaan pengujian pada setiap skenario beban, pengumpulan data hasil pengujian, analisis parameter kinerja, hingga interpretasi hasil dan penyusunan kesimpulan.

Gambar 1. Diagram Alur Penelitian

2.1. Arsitektur Sistem dan Lingkungan Pengujian

Sistem yang dievaluasi dalam penelitian ini adalah website sistem informasi manajemen pesantren yang telah diimplementasikan secara live dan dapat diakses melalui URL publik https://bpn.ponpes.id. Aplikasi ini dibangun menggunakan *framework* PHP Laravel sebagai *back-*

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

end yang terintegrasi dengan sistem manajemen basis data relasional *MariaDB*. Dari sisi infrastruktur, sistem dihosting pada layanan *shared hosting* yang menerapkan mekanisme pembatasan sumber daya melalui CloudLinux LVE Manager. Lingkungan ini memberikan alokasi sumber daya yang terbatas, yaitu sebesar *Single Core Central Processing Unit* (CPU) dan 1 Gigabyte (GB) *Random Access Memory* (RAM), yang dalam penelitian ini diperlakukan sebagai variabel kontrol untuk mengevaluasi batas kapasitas sistem. Pengujian dilakukan dari sisi klien menggunakan perangkat komputer berbasis sistem operasi Windows yang menjalankan perangkat lunak Apache JMeter versi 5.6.3 sebagai alat simulasi beban [10].

Penggunaan sistem yang telah berjalan secara nyata bertujuan untuk memperoleh hasil pengujian yang merepresentasikan kondisi operasional sebenarnya, sehingga hasil evaluasi memiliki tingkat validitas yang lebih tinggi dibandingkan pengujian pada lingkungan simulasi. Selain itu, seluruh skenario pengujian dilakukan tanpa melakukan perubahan pada kode sumber aplikasi maupun konfigurasi server, sehingga hasil yang diperoleh mencerminkan performa sistem dalam kondisi eksisting. Dengan demikian, pendekatan ini memungkinkan analisis yang lebih akurat terhadap keterbatasan infrastruktur yang digunakan dalam menangani beban akses pengguna secara simultan [11].

2.2. Pendekatan Evaluasi Kinerja

Evaluasi kinerja pada penelitian ini dibagi menjadi dua tahapan utama, yaitu *load testing* dan *stress testing* [12]. Tahap *load testing* diimplementasikan untuk mengukur respons dan stabilitas server di bawah beban kerja yang wajar, mulai dari operasional normal harian hingga kondisi jam sibuk. Evaluasi ini bertujuan untuk memastikan sistem mampu memproses permintaan tanpa adanya kebocoran memori (*memory leak*). Selanjutnya, tahap *stress testing* dilakukan dengan memberikan beban akses ekstrem yang secara sengaja melampaui kapasitas maksimal sistem [13]. Tujuannya adalah untuk mendeteksi titik jenuh (*breaking point*) di mana server mulai menolak permintaan masuk atau menghasilkan pesan galat akibat kehabisan alokasi pemrosesan (CPU) dan memori (RAM) [14].

2.3. Skenario dan Implementasi Pengujian

Implementasi simulasi dilakukan dengan membangun *Test Plan* pada Apache JMeter. Elemen *HTTP Request Sampler* dikonfigurasi menggunakan metode transmisi data GET untuk mengakses titik-titik akhir (*endpoint*) utama yang mewakili modul fungsional dari aplikasi pesantren. Rincian titik akhir URL yang menjadi target pengujian dapat dilihat pada Tabel 1.

Tabel 1. Daftar Titik Akhir (*Endpoint*) Target Pengujian

No	End Point (URL Path)	Keterangan Fungsionalitas Modul
1	/	Halaman utama sistem (*landing page*)
2	/berita	Modul pemuatan data berita secara dinamis
3	/hubungi-kami	Modul penyajian informasi kontak statis
4	/staf-pengajar	Modul penampilan agregasi data pengajar
5	/admin/login	Halaman autentikasi untuk administrator

Kelima *endpoint* pada Tabel 1 diakses secara berurutan oleh entitas pengguna virtual yang didefinisikan dalam modul *Thread Group* pada JMeter. Untuk mendapatkan komparasi data performa yang menyeluruh, skenario pengujian dibagi menjadi tiga tingkat beban lalu lintas akses. Setiap tingkat beban menerapkan parameter *ramp-up period* guna memastikan pengguna virtual masuk ke dalam sistem secara bertahap, bukan secara serentak dalam satu milidetik, sehingga menyerupai perilaku kunjungan manusia asli. Rincian skenario pembebanan disajikan pada Tabel 2.

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

Tabel 2. Skenario Simulasi Beban Akses (*Concurrent Users*)

Skenario Pengujian	Jumlah Pengguna Bersamaan	Representasi Kondisi Nyata
Skenario A (Beban Ringan)	20	Mewakili operasional administrasi harian
Skenario B (Beban Menengah)	50	Mewakili kondisi jam sibuk (*peak hours*)
Skenario C (*Stress Test*)	100	Mewakili lonjakan akses secara ekstrem

2.4. Parameter Analisis

Data hasil simulasi dari ketiga skenario tersebut kemudian ditangkap dan diekstraksi menggunakan fitur *listener* berupa *Summary Report* pada perangkat JMeter. Penilaian kelayakan sistem difokuskan pada tiga parameter metrik utama. Parameter pertama adalah *average response time*, yang mengukur rata-rata waktu pemrosesan dihitung dalam satuan milidetik yang dibutuhkan server sejak klien mengirim permintaan hingga umpan balik berhasil diterima seutuhnya. Parameter kedua adalah *throughput*, yang menunjukkan kapasitas aktual server dalam menyelesaikan jumlah permintaan per detik. Adapun parameter ketiga adalah persentase tingkat kegagalan (*error rate*), yang membandingkan jumlah permintaan yang gagal diproses, baik karena kehabisan waktu (*timeout*) maupun penolakan akibat batas *resource* tercapai atau istilah *error 508 Resource Limit Reached* terhadap total keseluruhan permintaan yang diterima oleh server.

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sekumpulan data kuantitatif yang merepresentasikan kinerja Sistem Informasi Manajemen Pesantren pada berbagai tingkat pembebanan akses. Analisis difokuskan pada perbandingan metrik kinerja antara kondisi operasional normal dan kondisi lonjakan trafik ekstrem yang dibatasi oleh infrastruktur *shared hosting* (*Single* Core CPU dan 1 GB RAM). Rekapitulasi hasil pengujian performa secara keseluruhan dari ketiga skenario disajikan pada Tabel 3.

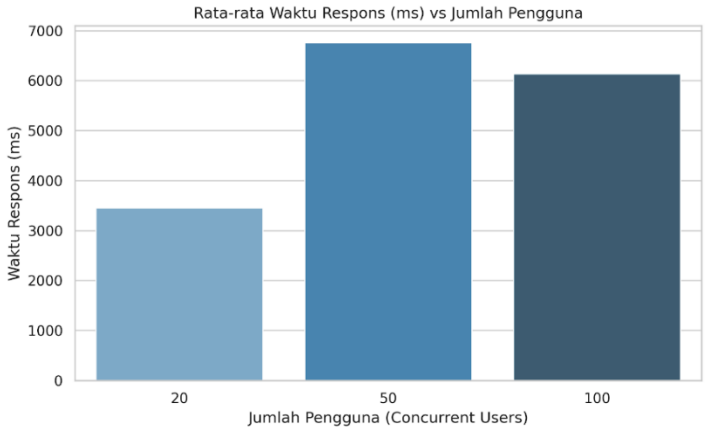
Tabel 3. Komparasi Hasil Pengujian Kinerja Sistem (*Summary Report*)

Skenario Pengujian	**Total Permintaan (*Samples*)**	**Rata-rata Waktu Respons (ms)**	***Throughput* (Req/sec)**	**Tingkat Kegagalan (*Error %*)**
Skenario A (20 Pengguna)	500	3.452	5,23	0,000
Skenario B (50 Pengguna)	1.750	6.763	3,62	0,457
Skenario C (100 Pengguna)	4.206	6.134	4,77	43,367

Hasil pengujian pada Tabel 3 menunjukkan adanya degradasi kinerja yang signifikan seiring dengan peningkatan jumlah pengguna simultan. Pola ini mengindikasikan bahwa kapasitas infrastruktur server memiliki batas toleransi tertentu dalam menangani beban akses [15]. Pada skenario beban rendah, sistem masih mampu mempertahankan stabilitas layanan dengan tingkat kegagalan yang sangat rendah. Namun, ketika jumlah pengguna meningkat, terjadi peningkatan waktu respons yang cukup tajam disertai penurunan kemampuan server dalam memproses permintaan secara efisien.

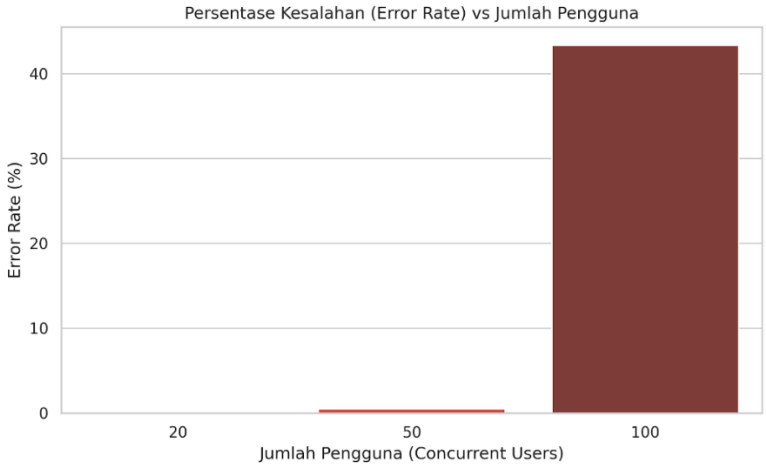
Kondisi tersebut menunjukkan bahwa keterbatasan sumber daya pada lingkungan *shared hosting*, khususnya pada aspek CPU dan RAM, berpengaruh langsung terhadap performa aplikasi berbasis Laravel. Selain itu, tingginya *error rate* pada skenario beban ekstrem mengindikasikan bahwa sistem telah mencapai batas kapasitas optimalnya sehingga tidak lagi mampu menangani permintaan secara konsisten [16]. Untuk memberikan gambaran yang lebih rinci mengenai karakteristik kinerja sistem pada setiap tingkat pembebanan, hasil pengujian tersebut selanjutnya dianalisis secara terpisah berdasarkan tiga parameter utama, yaitu rata-rata waktu respons, tingkat kegagalan, dan throughput, sebagaimana disajikan pada Gambar 2 hingga Gambar 4 berikut.

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790



Gambar 2. Grafik Rata-rata Waktu Respons terhadap Jumlah Pengguna

Evaluasi pertama ditinjau dari parameter rata-rata waktu respons (*average response time*). Seperti yang ditunjukkan pada Gambar 2, pada pembebanan ringan dengan 20 pengguna, sistem mencatatkan waktu respons sebesar 3.452 ms (sekitar 3,4 detik). Angka ini menunjukkan bahwa server masih dapat memproses setiap antrean HTTP GET secara stabil meskipun jeda pemuatan halaman sudah mulai terasa. Namun, ketika beban dinaikkan menjadi 50 pengguna, terjadi degradasi kinerja yang sangat tajam, di mana waktu respons melonjak hampir dua kali lipat menjadi 6.763 ms (6,7 detik). Secara teoretis, perlambatan ini terjadi karena keterbatasan *Single Core* CPU yang harus menangani *context switching* secara ekstrem untuk mengeksekusi banyak proses PHP (*FastCGI*) dan *query database* (MariaDB) secara bersamaan [17]. Menariknya, pada pengujian 100 pengguna, waktu respons rata-rata tampak sedikit turun menjadi 6.134 ms.



Gambar 3. Grafik Tingkat Kegagalan (*Error Rate*) terhadap Jumlah Pengguna

Evaluasi kedua difokuskan pada tingkat kegagalan, yang merupakan indikator krusial dalam uji stres karena secara langsung merepresentasikan kemampuan sistem dalam mempertahankan keandalan di bawah tekanan beban tinggi. Gambar 3 memvisualisasikan dengan jelas titik jatuh (breaking point) dari infrastruktur *shared hosting* yang diuji. Pada Skenario A dengan 20 pengguna, tingkat kegagalan tercatat 0%, yang menunjukkan bahwa sistem masih berada dalam kondisi operasional yang stabil dan seluruh permintaan dapat diproses dengan baik.

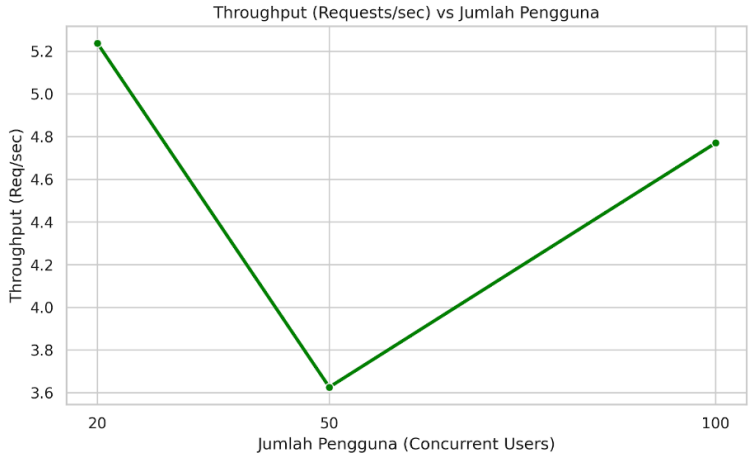
Pada Skenario B dengan 50 pengguna, mulai terlihat indikasi penurunan stabilitas sistem dengan munculnya error rate sebesar 0,457%. Meskipun persentasenya relatif kecil, kondisi ini mengindikasikan bahwa sistem telah mendekati batas kapasitas optimalnya. Pesan kesalahan

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

yang terjadi sebagian besar berasal dari halaman dinamis (/berita) yang memerlukan proses *query* basis data dan alokasi memori tambahan, sehingga meningkatkan beban kerja server.

Puncak degradasi kinerja terjadi pada Skenario C dengan jumlah 100 pengguna, di mana *error rate* meningkat secara signifikan hingga 43,367%. Lonjakan ini menunjukkan bahwa sistem tidak lagi mampu menangani permintaan secara konsisten, dan sebagian besar request mengalami kegagalan. Berdasarkan analisis log sistem, kegagalan tersebut disebabkan oleh mekanisme pembatasan sumber daya dari CloudLinux yang secara otomatis menghentikan proses ketika penggunaan RAM melebihi batas 1 GB, yang ditandai dengan munculnya Error 503 dan 508 *Resource Limit Reached*.

Kondisi ini mengindikasikan bahwa keterbatasan memori merupakan faktor dominan dalam terjadinya kegagalan sistem pada beban tinggi. Selain itu, tingginya tingkat kegagalan juga menunjukkan bahwa sistem tidak memiliki mekanisme penanganan beban atau istilah lainnya *load handling* yang memadai, seperti *queue management* atau *resource scaling*, sehingga tidak mampu mengakomodasi lonjakan permintaan secara simultan. Dengan demikian, error rate dapat digunakan sebagai indikator utama dalam menentukan batas kapasitas maksimum sistem pada lingkungan *shared hosting*.



Gambar 4. Grafik *Throughput* terhadap Jumlah Pengguna

Evaluasi ketiga meninjau parameter *throughput* (Gambar 4), yang mengukur kapasitas aktual server dalam menyelesaikan jumlah proses per detik. Pada Skenario A, sistem mencatatkan *throughput* tertinggi sebesar 5,23 *request/second*. Namun, pada Skenario B, kapasitas ini merosot ke titik terendah yaitu 3,62 *request/second*. Penurunan ini berkorelasi langsung dengan lonjakan waktu respons pada fase tersebut, di mana *server* memakan waktu lebih lama untuk memproses setiap antrean sehingga jumlah pekerjaan yang diselesaikan per detik menjadi berkurang. Pada Skenario C, *throughput* kembali naik ke angka 4,77 *request/second*, namun kenaikan ini terdistorsi oleh tingginya angka kegagalan, di mana *server* menyelesaikan pemrosesan dengan cara menolak (*drop*) *request* tersebut.

Secara keseluruhan, hasil eksperimen menunjukkan bahwa sistem yang dibangun menggunakan Laravel dan MariaDB memiliki performa yang stabil pada kondisi beban rendah hingga menengah. Namun, keterbatasan infrastruktur shared hosting menjadi faktor utama yang membatasi skalabilitas sistem. Keterbatasan CPU berkontribusi terhadap peningkatan waktu respons, sedangkan keterbatasan RAM menyebabkan terjadinya kegagalan sistem secara masif pada beban tinggi. Temuan ini menegaskan pentingnya pemilihan infrastruktur yang sesuai dengan karakteristik beban aplikasi, terutama untuk sistem yang memiliki potensi lonjakan akses secara simultan.

JURTI, Vol.10 No.3, September 2026, ISSN: 2579-8790

4. KESIMPULAN

Berdasarkan hasil pengujian kinerja menggunakan metode *load testing* dan *stress testing* dengan *Apache JMeter*, dapat disimpulkan bahwa website sistem informasi manajemen pesantren yang diimplementasikan pada infrastruktur *shared hosting* dengan spesifikasi *Single Core* CPU dan 1 GB RAM memiliki keterbatasan dalam menangani beban akses simultan. Sistem menunjukkan performa yang stabil pada beban 20 pengguna dengan *error rate* 0%, namun mulai mengalami degradasi kinerja pada 50 pengguna yang ditandai dengan peningkatan waktu respons hingga 6763 ms. Pada skenario 100 pengguna, sistem mengalami kegagalan signifikan dengan *error rate* mencapai 43,367%, yang menunjukkan bahwa kapasitas server telah melampaui batas optimalnya.

Secara umum, arsitektur aplikasi berbasis Laravel dan MariaDB mampu memberikan kinerja yang baik pada kondisi beban rendah hingga menengah. Namun, keterbatasan sumber daya pada *shared hosting*, khususnya pada CPU dan RAM, menjadi faktor utama yang membatasi skalabilitas sistem. Oleh karena itu, penelitian ini memberikan kontribusi berupa bukti empiris bahwa pemilihan infrastruktur server memiliki peran krusial dalam menjaga stabilitas dan kinerja aplikasi web berbasis framework modern.

5. SARAN

Berdasarkan hasil penelitian, pengujian selanjutnya disarankan untuk mencakup metode transmisi POST guna mengevaluasi proses transaksional seperti autentikasi dan pengiriman formulir dengan dukungan penanganan *CSRF Token* pada *Apache JMeter*. Selain itu, perlu dilakukan studi komparatif pada berbagai jenis infrastruktur server, seperti *shared hosting*, VPS, dan *cloud computing*, untuk memperoleh analisis kapasitas yang lebih komprehensif. Penelitian lanjutan juga dapat difokuskan pada evaluasi performa sistem setelah penerapan teknik optimasi perangkat lunak, seperti *object caching* dan optimasi query basis data, guna mengurangi beban komputasi dan meminimalkan *error rate* pada server dengan sumber daya terbatas.

DAFTAR PUSTAKA

[1] U. Mawaddah, E. Dewi Wahyuni, and A. P. Kusuma, "Penerapan Metode Agile Dalam Sistem Informasi Manajemen Asrama Santri pada Yayasan Pondok Pesantren Darul Huda Blitar Berbasis Web," *J-INTECH (Journal of Information and Technology)*, vol. 11, no. 2, pp. 188–199, 2023.

[2] N. Jamilatul Huda and H. Humaedi Hidayat, "Sistem Informasi Manajemen Berbasis Edutren Pada Administrasi Kesantrian Pesantren," *MUDIR (Jurnal Manajemen Pendidikan)*, vol. 8, no. 1, Jan. 2026, doi: 10.55352/mudir.

[3] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, "Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API : Kajian Literatur terhadap Framework Laravel dan React," *Infomatek*, vol. 27, no. 1, pp. 165–178, Jun. 2025, doi: 10.23969/infomatek.v27i1.25122.

[4] A. C. Nurcahyo, H. Pranjoto, K. Widya, and M. Surabaya, "Implementasi Jaringan Cloud dan Server Open Data Analytic Room di Diskominfo Kabupaten Bengkayang," *JIFOTECH (JOURNAL OF INFORMATION TECHNOLOGY*, vol. 05, no. 01, pp. 244–251, 2025, doi: https://doi.org/10.46229/jifotech.v5i1.989.

[5] A. Iskandar, M. I. Sarif, M. F. Hafiz, D. R. Harahap, and S. S. T. Purba, "Analisis Komparatif Kinerja Laravel Octane dan Webman pada Layanan API Berbasis Worker

Model PHP," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 3, pp. 1583–1590, Dec. 2025, doi: 10.62712/juktisi.v4i3.717.

[6] D. Noetzold, A. G. de Moraes Rossetto, L. A. Silva, P. Crocker, and V. R. Q. Leithardt, "JVM optimization: An empirical analysis of JVM configurations for enhanced web application performance," *SoftwareX*, vol. 28, p. 101933, Dec. 2024, doi: 10.1016/J.SOFTX.2024.101933.

[7] A. Sunardi and Suharjito, "MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based," *Procedia Comput. Sci.*, vol. 157, pp. 134–141, Jan. 2019, doi: 10.1016/J.PROCS.2019.08.150.

[8] L. T. Djaja and Y. E. Kurniawati, "Enhancing Concurrent User Capacity Through Kubernetes Implementation: A Case Study at PT. XYZ," *Procedia Comput. Sci.*, vol. 269, pp. 1033–1042, Jan. 2025, doi: 10.1016/J.PROCS.2025.09.045.

[9] T. Huang *et al.*, "DTAIS: Distributed trusted active identity resolution systems for the Industrial Internet," *Digital Communications and Networks*, vol. 10, no. 4, pp. 853–862, Aug. 2024, doi: 10.1016/J.DCAN.2023.06.006.

[10] Q. Musyafa'ur Rosul and E. Sulistiyani, "Analisis Perbandingan Kinerja Alat Pengujian Beban Perangkat Lunak: Apache JMeter, Gatling, dan K6," *JIP (Jurnal Informatika Polinema)*, vol. 12, no. 2, pp. 393–399, 2026.

[11] I. Hamidah *et al.*, "Evaluasi Pengujian Kinerja Menggunakan JMeter Untuk Menunjang Stabilitas Aplikasi Layanan Perbankan Pada PT Bank Rakyat Indonesia Tbk," *Journal of Digital Business and Technology Innovation (DBESTI)*, vol. 2, no. 1, pp. 114–126, 2025. Available: https://journal.nurulfikri.ac.id/index.php/DBESTI

[12] J. Ramaprabha *et al.*, "An efficient hybrid scheduling and virtual machine framework for load balancing in cloud computing architectures," *Ain Shams Engineering Journal*, vol. 17, no. 6, p. 104119, 2026, doi: https://doi.org/10.1016/j.asej.2026.104119.

[13] A. Jyoti, A. Yadav, D. Kumar, P. Mamoria, and V. Yadav, "Classification & Technological Analysis of Load Balancing in Cloud Computing," *Recent Patents on Engineering*, vol. 20, no. 1, pp. 22–35, 2026, doi: https://doi.org/10.2174/0118722121358630241220044608.

[14] M. Azzahari, A. Putra, and R. Ridha, "Implementation of Server Up-Scaling to Improve the Performance and Reliability of the ZIS Information System," *Journal of Artificial Intelligence and Software Engineering*, vol. 5, no. 4, pp. 1458–1465, 2025, doi: 10.30811/jaise.v5i4.8725.

[15] K. S. Pradipta, G. A. J. Saskara, and B. G. K. Yudistira, "Implementasi Kubernetes Cluster untuk High Availability dan Load Balancing SIAK Undiksha," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 4, pp. 12831–12840, Jan. 2026, doi: 10.31004/riggs.v4i4.5846.

[16] H. Setiawan and E. Depandi, "Pengujian Load Testing Website Jurusan Teknik Informatika Politeknik Negeri Bengkalis," *JEKIN - Jurnal Teknik Informatika*, vol. 5, no. 1, pp. 1–12, Jan. 2025, doi: 10.58794/jekin.v5i1.820.

[17] E. Herdianto, "Analisis Kinerja dan Skalabilitas Website Berbasis Google Cloud Platform Pada Infrastruktur Cloud Computing," *JURTIKA: Journal of Digital Innovation, Systems and Informatics*, vol. 1, no. 1, 2026.