

Evaluasi Ketahanan Server Website Sistem Pendukung Keputusan pada Investasi Lahan di Jawa Barat Menggunakan JMeter

1st Adifa Syahira

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

adifasyahira@student.telkomuniversity.
ac.id

2nd Casi Setianinsih

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Ashridini Maharawati

Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ashridini@telkomuniversity.ac.id

Abstrak — Di era digital, Sistem Pendukung Keputusan (DSS) berbasis web sangat penting bagi investasi properti, khususnya untuk evaluasi lahan kosong di Jawa Barat. Studi ini mengkaji ketahanan server DSS yang dirancang untuk menganalisis faktor-faktor yang mempengaruhi nilai tanah, termasuk lokasi dan infrastruktur. Dengan menggunakan Apache JMeter, stress test dilakukan untuk mensimulasikan lalu lintas pengguna yang tinggi dan menilai kinerja server di bawah beban yang bervariasi. Hasil menunjukkan bahwa sistem mempertahankan kinerja yang stabil hingga 500 pengguna secara bersamaan namun mengalami peningkatan variabilitas waktu respons dan beberapa koneksi gagal pada 1.500 pengguna. Temuan ini menunjukkan bahwa meskipun sistem dapat diandalkan pada beban yang lebih ringan, optimalisasi lebih lanjut diperlukan untuk kinerja yang konsisten pada beban yang lebih berat.

Kata Kunci — Web-based Decision Support System (DSS), Server Resilience, Stress Testing, Apache JMeter, Performance Evaluation

I. PENDAHULUAN

Dalam era digital yang terus berkembang, penggunaan sistem pendukung keputusan (SPK) berbasis web telah menjadi sangat penting, terutama dalam sektor investasi properti[1]. Di Jawa Barat, investasi lahan kosong merupakan salah satu bentuk investasi yang membutuhkan analisis mendalam dan data yang akurat untuk mendukung pengambilan keputusan yang efektif. Sistem pendukung keputusan ini dirancang untuk membantu investor dalam menilai berbagai faktor yang mempengaruhi nilai lahan, seperti lokasi, infrastruktur, aksesibilitas, dan potensi pengembangan[2].

Namun, dengan semakin tingginya ketergantungan pada sistem berbasis web, kinerja dan ketahanan server yang mengelola aplikasi ini menjadi faktor krusial. Ketahanan server mengacu pada kemampuan sistem untuk beroperasi dengan baik di bawah berbagai kondisi beban, tanpa mengalami kegagalan atau penurunan performa yang signifikan[3]. Sebuah server yang tidak mampu menangani lalu lintas tinggi dapat menyebabkan downtime, keterlambatan respons, atau bahkan kegagalan sistem, yang pada akhirnya dapat merugikan pengguna dan menurunkan kepercayaan terhadap aplikasi tersebut[4].

Untuk mengatasi tantangan ini, diperlukan evaluasi ketahanan server menggunakan alat yang dapat mensimulasikan

berbagai skenario beban. JMeter adalah salah satu alat yang sering digunakan untuk tujuan ini. Alat ini memungkinkan pengujian performa dengan mengirimkan sejumlah besar permintaan HTTP ke server, sehingga dapat mensimulasikan kondisi penggunaan yang intensif[5].

Penelitian ini bertujuan untuk mengevaluasi ketahanan server dari sistem pendukung keputusan pada investasi lahan di Jawa Barat. Dengan menggunakan JMeter, penelitian ini akan mengidentifikasi bagaimana server merespons berbagai tingkat beban dan menentukan batas kemampuan server sebelum terjadi penurunan performa yang signifikan[5]. Hasil dari evaluasi ini diharapkan dapat memberikan wawasan yang berguna bagi pengelolaan infrastruktur server dan pengembangan lebih lanjut dari sistem yang ada.

II. METODE PENELITIAN

A. Server

Server adalah perangkat keras atau perangkat lunak yang menyediakan layanan atau sumber daya kepada perangkat lain dalam jaringan. Jenis-jenis server meliputi web server untuk menyediakan halaman web, file server untuk menyimpan dan mengelola file, database server untuk mengelola basis data, mail server untuk menangani email, dan DNS server untuk mengonversi nama domain menjadi alamat IP. Fungsi utama server mencakup pengelolaan sumber daya jaringan, keamanan data, ketersediaan layanan, serta backup dan pemulihan data[6]. Tren terbaru dalam server termasuk penggunaan cloud computing untuk fleksibilitas dan skalabilitas yang lebih besar.

B. REST

REST adalah arsitektur API yang banyak digunakan dan beroperasi di atas protokol HTTP. Dengan menggunakan arsitektur ini, klien HTTP, seperti aplikasi Android, dapat menggunakan metode HTTP seperti GET, POST, PUT, dan DELETE untuk berinteraksi dengan sumber daya di server. Arsitektur REST menggunakan JSON (JavaScript Object Notation) sebagai format komunikasi standar antara klien dan server. API yang mengikuti prinsip REST disebut sebagai "RESTful API"[7].

C. Stress Test (Uji Stres)

Uji stres adalah jenis pengujian kinerja yang dirancang untuk mengevaluasi batas-batas ketahanan dan kapasitas maksimum sistem atau aplikasi. Tujuan utamanya adalah untuk mengidentifikasi titik kegagalan sistem dengan memberikan beban yang jauh melebihi kapasitas normal yang dirancang untuk ditangani oleh sistem tersebut[8]. Ini merupakan langkah penting dalam memahami bagaimana sistem berperilaku dalam kondisi ekstrem dan untuk menilai kemampuannya dalam menghadapi situasi yang tidak biasa atau beban yang sangat berat.

D. Apache JMeter

Apache JMeter adalah alat open-source yang digunakan untuk menguji dan menganalisis kinerja aplikasi web dengan mensimulasikan beban pengguna untuk mengevaluasi bagaimana server atau aplikasi beroperasi di bawah tekanan[9]. JMeter memungkinkan pengguna untuk melakukan berbagai jenis pengujian kinerja, termasuk uji beban, stres, dan skalabilitas, dengan mengirimkan sejumlah besar permintaan simultan ke sistem.

Selama pengujian, JMeter mengumpulkan metrik kinerja seperti waktu respons dan tingkat kesalahan, serta menyediakan laporan dan grafik untuk analisis. Dengan dukungan untuk berbagai protokol dan kemampuan untuk menjalankan pengujian terdistribusi, JMeter membantu dalam mengevaluasi performa aplikasi, mengidentifikasi masalah, dan mempersiapkan sistem untuk menghadapi beban tinggi[10].

III. METODE PENELITIAN

A. Lingkup Penelitian

Lingkup penelitian mencakup server yang digunakan untuk mengoperasikan SPK berbasis web di Jawa Barat. Penelitian ini difokuskan pada pengujian beban untuk menilai kinerja server dalam skenario penggunaan intensif, yang mencerminkan kondisi nyata saat terjadi lonjakan akses pengguna.

B. Persiapan Lingkungan Pengujian

1. Pengaturan Server: Server yang akan diuji harus dipersiapkan dengan konfigurasi yang sesuai, mencakup spesifikasi perangkat keras, sistem operasi, dan perangkat lunak yang diperlukan untuk menjalankan aplikasi web.
2. Konfigurasi Apache JMeter: Menyiapkan JMeter untuk pengujian, termasuk instalasi dan konfigurasi sesuai dengan skenario pengujian yang direncanakan. Ini meliputi pembuatan test plan, yang mencakup[11]:
 - a. Thread Group : Menentukan jumlah pengguna virtual (threads) dan ramp-up period.
 - b. HTTP Request : Mengonfigurasi permintaan HTTP yang akan dikirim ke server, termasuk jenis metode (GET, POST, PUT, DELETE) dan URL endpoint.
 - c. Listeners : Menambahkan komponen untuk menangkap dan mencatat data selama pengujian, seperti waktu respons, throughput, dan tingkat kesalahan.
3. Pelaksanaan Pengujian
 - a. Load Testing: Melakukan simulasi beban pengguna dari rendah hingga tinggi untuk menilai performa server dalam kondisi normal hingga intensif.

- b. Stress Testing: Menguji server di bawah kondisi ekstrem dengan beban yang melebihi kapasitas normal untuk mengidentifikasi titik kegagalan.

IV. SKENARIO STRESS TEST

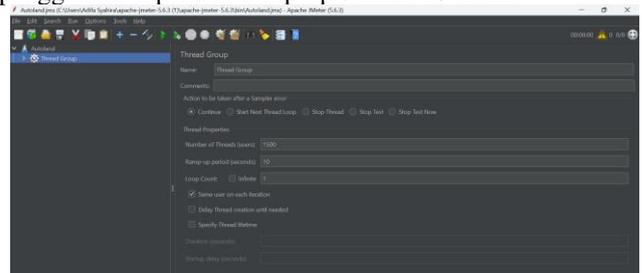
Bagian ini menjelaskan berbagai skenario uji beban serta rincian spesifikasi sistem. Berikut skenarionya :

A. Sistem Spesifikasi

Apache JMeter yang digunakan dalam penelitian ini adalah versi 5.4.1, dikonfigurasi dengan skenario uji yang mencakup 100, 500, dan 1500 pengguna virtual (threads) untuk mensimulasikan beban pengguna yang signifikan, dengan ramp-up time selama 10 detik untuk mencapai beban puncak. Konfigurasi test plan mencakup HTTP Request yang merepresentasikan aktivitas utama pada Sistem Pendukung Keputusan (SPK), serta penggunaan komponen Listener seperti Summary Report, Aggregate Report, dan Graph Results untuk mengumpulkan dan menganalisis data performa server secara real-time[12].

B. Skenario Stress Test

Penggunaan aplikasi ini dapat dimulai dengan mengkonfigurasi grup thread. Bagian properti thread dapat diisi sesuai dengan skenario yang telah disiapkan. Misalnya, skenario dengan 100 pengguna, 500 pengguna hingga 1500 pengguna dan periode ramp-up selama 10 detik.



GAMBAR 1 Pengaturan Konfigurasi Load Testing JMeter

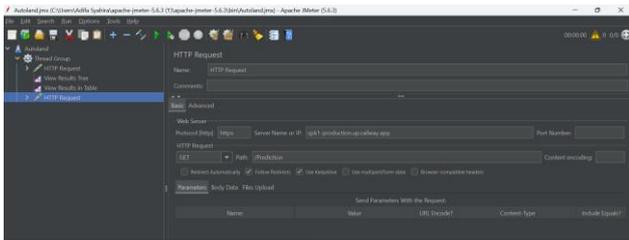
Setiap data akan digunakan secara acak dalam skenario uji tekanan. Skenario uji tekanan adalah sebagai berikut:

TABEL 1 Jumlah Pengguna yang Akan Diminta

No.	Jumlah Pengguna	Total Permintaan	Halaman Percobaan
1	100	100	https://spk1-production.up.railway.app/Prediction
2	500	500	https://spk1-production.up.railway.app/Prediction

3	1500	1500	https://spk1-production.up.railway.app/Prediction
---	------	------	---

Hasil pengujian diharapkan menunjukkan jumlah permintaan HTTP yang gagal kurang dari 10%, dan waktu rata-rata yang dibutuhkan untuk menyelesaikan semua permintaan HTTP harus kurang dari atau sama dengan 1 detik. Selanjutnya melakukan konfigurasi pengujian HTTP Request di Apache Jmeter.



GAMBAR 2 Konfigurasi Pengujian pada HTTP Request

Skenario ini akan menguji endpoint /Prediction di server spk1-production.up.railway.app menggunakan metode HTTP GET. Uji akan dilakukan dengan sejumlah pengguna yang mengakses endpoint tersebut secara bersamaan (jumlah pengguna dan ramp-up period ditentukan di Thread Group). Pengujian akan memeriksa seberapa cepat server merespons dan apakah ada kegagalan dalam merespons permintaan.

Selanjutnya dapat menjalankan pengujian ini dengan mengklik tombol play hijau di atas, dan hasilnya akan muncul "View Results in Table" untuk analisis lebih lanjut.

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
81	11:47:26.952	Thread Group...	HTTP Request	377	✓	6201	142	377	0
82	11:47:27.109	Thread Group...	HTTP Request	357	✓	6201	142	335	0
83	11:47:27.180	Thread Group...	HTTP Request	357	✓	6201	142	348	0
84	11:47:27.244	Thread Group...	HTTP Request	378	✓	6201	142	377	0
85	11:47:27.239	Thread Group...	HTTP Request	483	✓	6201	142	483	0
86	11:47:27.453	Thread Group...	HTTP Request	364	✓	6201	142	364	0
87	11:47:27.538	Thread Group...	HTTP Request	380	✓	6201	142	379	0
88	11:47:27.768	Thread Group...	HTTP Request	401	✓	6201	142	376	0
89	11:47:27.833	Thread Group...	HTTP Request	420	✓	6201	142	420	0
90	11:47:27.919	Thread Group...	HTTP Request	350	✓	6201	142	347	0
91	11:47:27.917	Thread Group...	HTTP Request	352	✓	6201	142	349	0
92	11:47:28.169	Thread Group...	HTTP Request	408	✓	6201	142	387	0
93	11:47:28.306	Thread Group...	HTTP Request	354	✓	6201	142	345	0
94	11:47:28.376	Thread Group...	HTTP Request	365	✓	6201	142	355	0
95	11:47:28.447	Thread Group...	HTTP Request	368	✓	6201	142	368	0
96	11:47:28.446	Thread Group...	HTTP Request	371	✓	6201	142	370	0
97	11:47:28.579	Thread Group...	HTTP Request	345	✓	6201	142	342	0
98	11:47:28.669	Thread Group...	HTTP Request	356	✓	6201	142	347	0
99	11:47:28.757	Thread Group...	HTTP Request	364	✓	6201	142	348	0
100	11:47:28.820	Thread Group...	HTTP Request	370	✓	6201	142	351	0

Gambar 3 Tampilan dari Hasil "View Results in Table"

Berikut merupakan hasil dari "View Results in Table" dari 100 pengguna yang diusulkan.

V. HASIL DAN ANALISA STRESS TEST

A. Pengujian 100 Pengguna

Dimulai dari pengujian 100 pengguna yang mana pengujian ini akan memberikan wawasan mengenai kemampuan situs web dalam menangani beban yang tinggi, serta membantu mengidentifikasi potensi bottleneck atau masalah kinerja lainnya yang mungkin muncul saat lalu lintas pengguna meningkat secara signifikan. Berikut hasil yang didapatkan :

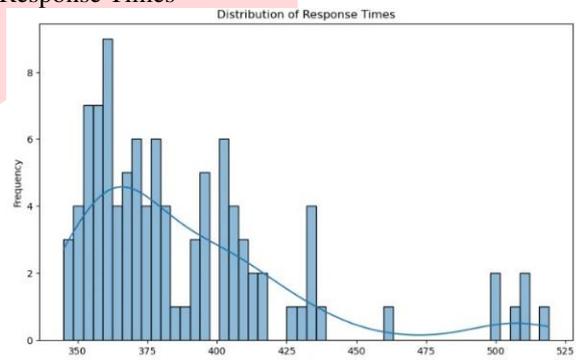
- Success Rate

```
# Calculate the success rate
success_rate = df['success'].mean() * 100
print(f"Success Rate: {success_rate:.2f}%")

Success Rate: 100.00%
```

Gambar 4 Rate Sukses 100 Pengguna

- Response Times



GAMBAR 5 Grafik Response Time 100 Pengguna

- Response Message



GAMBAR 6 Grafik Response Message 100 Pengguna

Berdasarkan Stress Test yang dilakukan pada 100 pengguna, hasilnya menunjukkan bahwa pengujian dengan 100 pengguna dapat dianggap berhasil 100%. Berikut adalah analisisnya:

1. Distribusi Waktu Respon

Distribusinya tidak simetris dan cenderung miring ke kanan, menandakan bahwa sebagian besar waktu respons berada di kisaran

Histogram menunjukkan puncaknya di sekitar

350 ms hingga 375 ms, yang mengindikasikan bahwa rentang waktu respons ini paling sering terjadi.

2. Bentuk Distribusi

Distribusinya tidak simetris dan cenderung miring ke kanan, menandakan bahwa sebagian besar waktu respons berada di kisaran yang lebih rendah, dengan ekor panjang mengarah ke waktu respons yang lebih tinggi.

Terdapat beberapa waktu respons yang cukup tinggi (sekitar 500 ms hingga 525 ms), tetapi frekuensinya sangat rendah.

3. Variabilitas Waktu Respons:

Terdapat variasi yang cukup besar dalam waktu respons, yang dapat menunjukkan adanya fluktuasi performa pada sistem yang diuji.

Beberapa waktu respons yang sangat tinggi mungkin mengindikasikan adanya outlier atau periode di mana sistem mengalami penurunan kinerja yang signifikan.

4. Keseragaman Pesan Respons:

Diagram batang menunjukkan bahwa terdapat 100 pesan respons yang semuanya memiliki nilai yang sama, ditampilkan sebagai satu batang tunggal pada sumbu x.

Ini menandakan bahwa tidak ada variasi dalam pesan respons yang diterima, semuanya seragam.

Secara keseluruhan, hasil dari analisis pada pengujian 100 pengguna ini menunjukkan bahwa sistem memenuhi ekspektasi dalam hal performa.

B. Pengujian 500 Pengguna

Setelah berhasil menguji sistem dengan 100 pengguna, pengujian dilanjutkan dengan menambah jumlah pengguna menjadi 500 untuk mengevaluasi kinerja sistem di bawah beban yang lebih besar. Pengujian ini bertujuan untuk mengidentifikasi apakah sistem masih mampu mempertahankan performa yang stabil dan konsisten saat jumlah pengguna meningkat secara signifikan. Berikut hasil yang didapatkan :

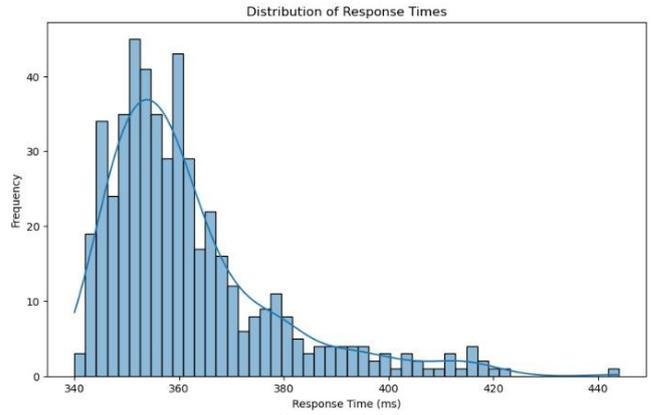
1. Success Rate

```
# Calculate the success rate
success_rate = df['success'].mean() * 100
print(f"Success Rate: {success_rate:.2f}%")

Success Rate: 100.00%
```

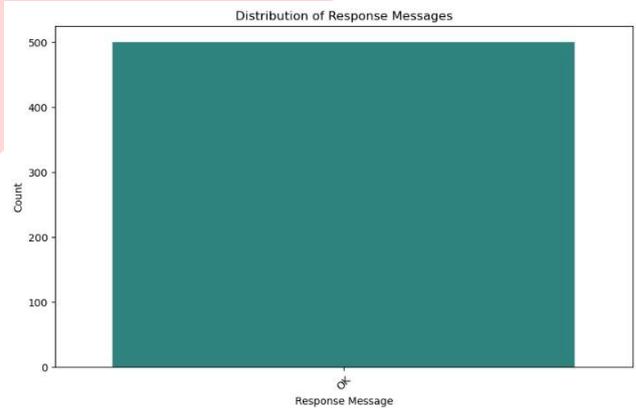
Gambar 7
Rate Sukses 500 Pengguna

2. Response Time



GAMBAR 8
Grafik Response Time 500 Pengguna

3. Response Message



Gambar 9
Grafik Response Message 500 Pengguna

Berdasarkan Stress Test yang dilakukan pada 500 pengguna, hasil pengujian menunjukkan bahwa pengujian dengan 500 pengguna berhasil 100%. Analisisnya sebagai berikut:

4. Distribusi Waktu Respons:

- a. Grafik memperlihatkan bahwa mayoritas permintaan memiliki waktu respons antara 340 ms hingga 380 ms.
- b. Puncak frekuensi terjadi di kisaran 350 ms hingga 360 ms, menunjukkan bahwa rentang waktu respons ini paling sering terjadi.

5. Bentuk Distribusi:

- a. Distribusinya miring ke kanan, yang berarti sebagian besar waktu respons berada di sisi yang lebih cepat, dengan ekor panjang yang mencapai waktu respons lebih tinggi.
- b. Ada beberapa waktu respons yang cukup tinggi hingga sekitar 440 ms, namun kemunculannya sangat jarang.

6. Variabilitas Waktu Respons:

Terdapat variasi dalam waktu respons, yang mungkin mengindikasikan adanya fluktuasi kinerja pada sistem yang diuji.

Beberapa waktu respons yang lebih tinggi dapat menunjukkan adanya outlier atau periode di mana sistem mengalami penurunan kinerja yang signifikan.

7. Keceragaman Pesan Respons:

- a. Grafik menunjukkan bahwa hanya satu jenis pesan respons yang diterima sebanyak 500 kali.
- b. Hal ini mungkin menunjukkan bahwa pengujian JMeter menghasilkan respons yang sangat konsisten tanpa adanya variasi, atau hanya satu tipe respons yang berhasil tercatat selama pengujian.
- c. Secara keseluruhan, hasil analisis dari pengujian pada 500 pengguna ini menunjukkan bahwa sistem berhasil memenuhi harapan dalam hal performa.

C. Pengujian 1500 Pengguna

Setelah memastikan keberhasilan pengujian dengan 500 pengguna, langkah selanjutnya adalah meningkatkan jumlah pengguna menjadi 1.500 untuk menilai kinerja sistem di bawah beban yang jauh lebih besar. Pengujian ini akan menguji batas kemampuan sistem dalam menangani lonjakan trafik yang signifikan, serta mengidentifikasi potensi masalah kinerja yang mungkin muncul saat jumlah pengguna meningkat secara drastis. Berikut hasil yang didapatkan :

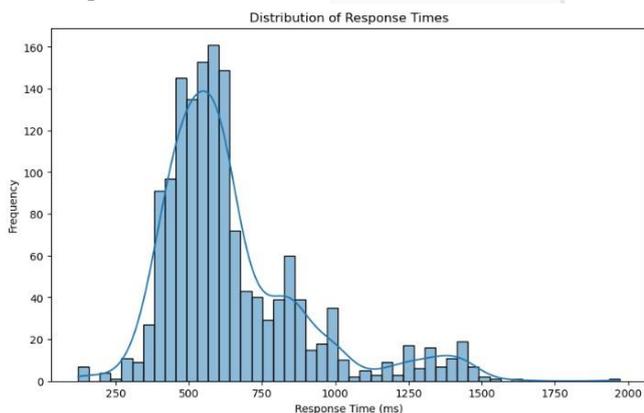
a. Success Rate

```
[5]: # Calculate the success rate
success_rate = df['success'].mean() * 100
print(f"Success Rate: {success_rate:.2f}%")

Success Rate: 92.60%
```

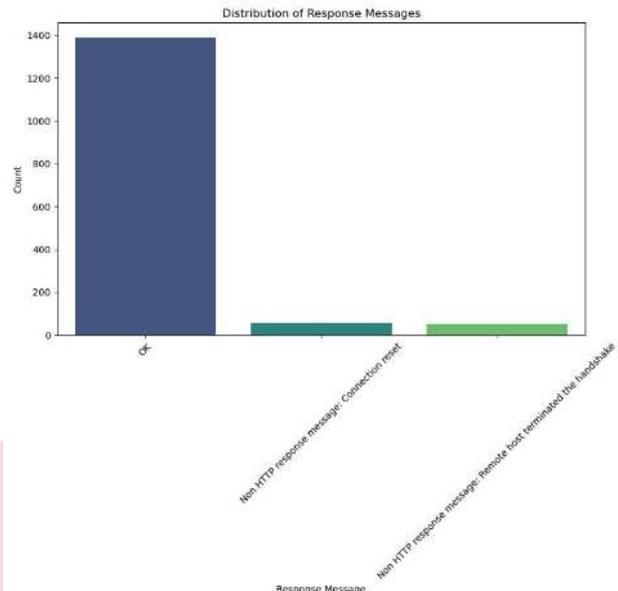
GAMBAR 10
Rate Sukses 1500 Pengguna

b. Response Time



GAMBAR 11
Grafik Response Time 1500 Pengguna

c. Responses Message



GAMBAR 12
Grafik Respon Message 1500 Pengguna

Berdasarkan Stress Test yang dilakukan pada 1.500 pengguna, hasil pengujian menunjukkan tingkat keberhasilan sebesar 92,60%. Analisisnya sebagai berikut:

1. Distribusi Waktu Respons
 - a. Bentuk Distribusi: Grafik menunjukkan distribusi yang menyerupai distribusi normal dengan kemiringan ke kanan. Ini berarti sebagian besar waktu respons terkonsentrasi di sisi kiri (lebih cepat), dengan ekor yang memanjang ke kanan, menunjukkan adanya beberapa waktu respons yang lebih lambat.
 - b. Puncak (Mode): Puncak distribusi terjadi di sekitar 500-600 ms, menunjukkan bahwa mayoritas waktu respons berada dalam rentang tersebut.
 - c. Rentang Waktu Respons: Sebagian besar waktu respons berada di antara 250 ms hingga 1000 ms, dengan penurunan tajam setelahnya. Beberapa outlier menunjukkan waktu respons di atas 1500 ms.
2. Variabilitas Waktu Respons:
 - a. Variabilitas waktu respons dalam pengujian ini cukup besar, dengan sebagian besar permintaan memiliki waktu respons di sekitar 500-600 ms, namun terdapat juga beberapa permintaan dengan waktu respons yang jauh lebih tinggi, mencapai hampir 2000 ms. Variasi ini menunjukkan adanya kebutuhan untuk mengoptimalkan permintaan dengan waktu respons yang lebih lambat guna meningkatkan konsistensi kinerja aplikasi.

3. Keseragaman Pesan Respons
 - a. Mayoritas Respons: Grafik menunjukkan bahwa mayoritas respons adalah "OK" dengan jumlah yang sangat besar (sekitar 1.400), menunjukkan bahwa sebagian besar permintaan berhasil mendapatkan status yang diharapkan.
 - b. Respons Minoritas: Kategori "Connection reset" dan "Remote host terminated the handshake" memiliki jumlah yang sangat kecil (masing-masing sekitar 25), menunjukkan hanya sedikit permintaan yang mengalami masalah dan gagal mendapatkan respons yang sukses.
 - c. Keseragaman respons sangat tinggi karena mayoritas permintaan menerima respons "OK". Meskipun ada beberapa variasi dalam kategori respons yang gagal, jumlahnya tidak cukup signifikan untuk mengurangi keseragaman keseluruhan.

Secara keseluruhan, distribusi respons menunjukkan tingkat keseragaman yang tinggi dengan dominasi respons "OK", yang menandakan bahwa sistem yang diuji cukup andal dan konsisten dalam merespons permintaan.

VI. KESIMPULAN

Dari hasil pengujian dengan 100 hingga 1.500 pengguna, dapat disimpulkan bahwa sistem secara umum mampu menangani beban dengan baik dan konsisten. Pada pengujian dengan 100 dan 500 pengguna, waktu respons sebagian besar permintaan berada dalam rentang yang wajar dan sistem mampu mempertahankan keseragaman respons yang tinggi, menunjukkan bahwa performa tetap stabil meskipun jumlah pengguna meningkat. Namun, saat beban meningkat menjadi 1.500 pengguna, terlihat adanya peningkatan variabilitas dalam waktu respons, dengan beberapa permintaan mengalami waktu respons yang jauh lebih lama dan beberapa kegagalan koneksi. Hal ini mengindikasikan bahwa meskipun sistem cukup andal dalam menangani beban pengguna yang lebih ringan, diperlukan optimasi lebih lanjut untuk memastikan stabilitas dan konsistensi performa ketika menghadapi beban yang lebih berat.

REFERENSI

- [1] E. Nurelasari and E. Purwaningsih, "Sistem Pendukung Keputusan Pemilihan Perumahan Terbaik Dengan Metode TOPSIS," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 8, no. 4, p. 317, Oct. 2020, doi: 10.26418/justin.v8i4.41036.
- [2] S. Rosyida, "SISTEM PENDUKUNG KEPUTUSAN UNTUK PEMILIHAN JENIS INVESTASI DENGAN METODE FUZZY MULTI CRITERIA DECISION MAKING," p. 78839421.
- [3] M. I. Perkasa and E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token," *Jurnal ULTIMA Computing*, vol. 10, no. 1, pp. 19–26, Jul. 2018, doi: 10.31937/sk.v10i1.838.
- [4] D. Darmadi, R. Susandi, D. Komunikasi, D. Informatika, and K. Payakumbuh, "Testing the Effectiveness of Cluster Systems by Implementing High Availability on Virtual Servers (Case Study at BPS-Statistics of Sumatera Utara Province)," 2022.
- [5] N. Luh, A. S. Ginasari, K. S. Wibawa, N. Kadek, and A. Wirdiani, "Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter," 2021.
- [6] A. M. Fanggidae, H. Hermawan, and H. I. Pratiwi, "Sistem Monitoring Server Dengan Menggunakan SNMP," *WIDYAKALA JOURNAL*, vol. 6, no. 2, p. 163, Sep. 2019, doi: 10.36262/widyakala.v6i2.218.
- [7] Y. K. Kurniawan, Y. Oslan, and H. K. Abstrak, "IMPLEMENTASI REST-API UNTUK PORTAL AKADEMIK UKDW BERBASIS ANDROID." 2021.
- [8] D. Andriansyah, "PERFORMANCE DAN STRESS TESTING DALAM MENGOPTIMASI WEBSITE," *Computer Based Information System Journal*, vol. 7, no. 1, pp. 23–28, Mar. 2019, doi: 10.33884/cbis.v7i1.995.
- [9] K. A. Prasetya, S. Rizqika Akbar, and R. Primananda, "Implementasi Lingkungan Test pada Moodle dengan Apache JMeter," 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [10] Mangapul Siahaan and R. W. Wijaya, "Performance Comparison Between Laravel and ExpressJs Framework Using Apache JMeter," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 7, no. 2, pp. 545–554, Jan. 2024, doi: 10.31289/jite.v7i2.10571.
- [11] L. W. Suwarsono, A. N. Aisha, and F. N. Nugraha, "The Role of E-Learning Readiness on Workload: Perspective Engineering and non-Engineering Students," *International Journal of Innovation in Enterprise System*, vol. 6, no. 01, pp. 85–94, Jan. 2022, doi: 10.25124/ijies.v6i01.165.
- [12] Ade Ismail, Ahmadi Yuli Ananta, Sofyan Noor Arief, and Elok Nur Hamdana, "Performance Testing Sistem Ujian Online Menggunakan Jmeter Pada Lingkungan Virtual," *Jurnal Informatika Polinema*, vol. 9, no. 2, pp. 159–164, Feb. 2023, doi: 10.33795/jip.v9i2.1190.