

Optimalisasi Performa *Website* LSP SMKN 2 Kraksaan melalui Pengujian Beban dengan Apache JMeter

Regina Juniantika^a, Nugroho Setyo Wibowo^b, Vigo Dewangga^c, Dian Rizky^d

^aDepartment of Information Technology, Politeknik Negeri Jember, Indonesia

^bDepartment of Information Technology, Politeknik Negeri Jember, Indonesia

^cDepartment of Language, Communication, and Tourism, Politeknik Negeri Jember, Indonesia

^dDepartment of Agribusiness Management, Politeknik Negeri Jember, Indonesia

ARTICLE INFO

Article history:

Received 2 March 2025

Received in revised form

10 April 2025

Accepted 30 April 2025

Available online 19 May 2025

Keywords:

Performance Testing

Load Testing

Website

Apache JMeter

ABSTRACT

The professional certification process at SMKN 2 Kraksaan is carried out using a web-based information system. Therefore, this research examines the number of users who can access the website and the common issues that often occur, such as server access failures and slow page load times, which affect the website's performance. This study aims to determine whether the website can serve a large number of users efficiently. By conducting load testing, the study measures the average response time required for users to access the website. Based on the load testing conducted on the LSP SMKN 2 Kraksaan website, specifically in the assessment section, the website can handle simultaneous user access, accommodating up to 500 users within a 1-second access duration. However, when the number of users reaches 1000, test results indicate that some users experience failures due to the website exceeding its maximum capacity in the assessment section.

1 Pendahuluan

Kualitas pendidikan di Indonesia sering disorot, karena dianggap mampu menjadi tenaga kerja yang kompetitif di dalam dan luar negeri. Namun, banyak lulusan sekolah terutama lulusan SMK, masih tidak memiliki pekerjaan. Dengan demikian, UU nomor 9 tahun 2016 menetapkan bahwa kualitas sumber daya manusia (SDM) harus ditingkatkan, terutama untuk lulusan SMK [1].

Salah satu cara untuk melakukan ini adalah dengan mengembangkan kurikulum SMK yang sesuai dengan kompetensi keahlian pengguna lulusan. Sehingga kualitas lulusan diharapkan dapat ditingkatkan, akreditasi menjadi lebih baik, dan akses ke sertifikasi profesi menjadi lebih mudah. Maka LSP (Lembaga Sertifikasi Profesi) yang bertanggung jawab untuk mengembangkan sertifikasi akreditasi, standar kompetensi, dan unit Tempat Uji Kompetensi (TUK) untuk bidang keahlian tertentu [2]. Dengan begitu, tingkat pengangguran tamatan SMK di Indonesia akan secara bertahap menurun.

Proses sertifikasi profesi yang terdapat di SMKN 2 Kraksaan dilaksanakan menggunakan sebuah sistem informasi berbasis *website*. Dengan adanya sistem informasi ini, maka mempermudah warga SMKN 2 Kraksaan untuk menggunakannya. Dari banyaknya pengguna yang dapat mengakses situs *website* ini, masalah yang biasanya sering terjadi yaitu kegagalan *server* untuk diakses, lambatnya waktu muat halaman *website*, sehingga menyebabkan terganggunya performa *website*. Jadi apabila terlalu banyak pengguna yang mengakses, maka beban *website* semakin meningkat dan akan melambat.

Sebuah alat bernama JMeter akan digunakan untuk meneliti beban pada *website* LSP SMKN 2 Kraksaan. *Tools* ini berfungsi sebagai simulasi dengan memberikan banyaknya pengguna yang mengakses *website* secara bersamaan [3]. Kemudian akan dievaluasi apakah situs *website* mampu melayani banyak pengguna dengan cepat. Setelah mengetahui hasil pengujian beban, kemudian menghitung hasil kinerja dengan menunjukkan kapasitas kinerja sistem terhadap beban. Apabila

kinerja sistem menjadi tidak stabil dan memerlukan waktu yang cukup lama, maka itu menunjukkan bahwa sistem telah mencapai pada kapasitas maksimumnya.

Untuk menyelesaikan masalah tersebut, penelitian akan dilakukan dengan mengukur rata-rata waktu (*response time*) yang dibutuhkan dalam mengakses *website*. Pengukuran mencakup waktu akses keseluruhan halaman serta waktu akses setiap URL (*Uniform Resource Locator*) pada berbagai skenario beban pengguna. Setelah data diperoleh, hasil analisis akan disajikan dalam bentuk tabel untuk memvisualisasikan performa *website* berdasarkan parameter yang diuji serta membuat rekomendasi dari hasil pengujian sebelumnya.

2 Studi Literatur

2.1 Kualitas Software

Kualitas perangkat lunak didefinisikan sebagai untuk memenuhi kebutuhan fungsional dan kinerja yang didokumentasikan, pengembangan standar yang didokumentasikan, dan karakteristik implisit yang diharapkan dari *software* yang dibangun secara profesional [4]. Berdasarkan definisi di atas terlihat bahwa sebuah perangkat lunak dikatakan berkualitas apabila memenuhi tiga ketentuan pokok:

1. Memenuhi kebutuhan pemakai, yang berarti bahwa jika perangkat lunak tidak dapat memenuhi kebutuhan pengguna perangkat lunak tersebut, maka yang bersangkutan dikatakan tidak atau kurang memiliki kualitas.
2. Memenuhi standar pengembangan perangkat lunak, yang berarti bahwa jika cara pengembangan perangkat lunak tidak mengikuti metodologi standar, maka hampir dapat dipastikan bahwa kualitas yang baik akan sulit atau tidak tercapai.
3. Memenuhi sejumlah kriteria implisit – yang berarti bahwa jika salah satu kriteria implisit tersebut tidak dapat dipenuhi, maka perangkat lunak yang bersangkutan tidak dapat dikatakan memiliki kualitas yang baik.

2.2 Pengujian Software

Pengujian *software* sangat penting untuk memastikan bahwa perangkat lunak yang telah dibuat berjalan dengan baik dan memenuhi syarat untuk fungsi yang diharapkan. Pengembang atau penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan dapat ditemukan dan diperbaiki segera [3].

Tujuan dari pengujian *software* sendiri adalah supaya perusahaan bisa mengidentifikasi kesalahan dan juga fitur yang tidak sesuai dengan persyaratan yang sebenarnya dan jika ada bug ataupun kesalahan dalam *software* maka perusahaan harus memenuhi kembali dalam proses *development*. Ada berbagai jenis pengujian perangkat lunak, pengujian dilakukan dengan tujuan dan strategi spesifik. Salah satunya yaitu pengujian performa, dimana pengujian ini menguji bagaimana perangkat lunak berjalan pada beban kerja yang berbeda.

Salah satu faktor yang akan membuat penggunaan layanan menjadi nyaman bagi pengguna adalah kemampuan situs web untuk dibuka. Performa ini akan dipengaruhi oleh ukuran file halaman yang dimaksud. Semakin kompleks fungsinya, semakin besar ukurannya. Waktu yang dibutuhkan untuk menampilkan halaman web adalah komponen yang berhubungan dengan kinerja *website* [5].

2.3 Performance Testing

Performance testing merupakan salah satu jenis pengujian kinerja yang menentukan atau memverifikasi fitur kecepatan, skalabilitas, dan stabilitas aplikasi dibawah beban kerja tertentu. [4]. Oleh karena itu, pengujian *performance* biasanya dilakukan untuk membantu menemukan kelemahan sistem yang menjadi acuan dasar untuk pengujian di masa mendatang.

Jenis pengujian *performance* digunakan untuk memastikan bahwa perangkat lunak menjalankan beban kerja yang diharapkan dengan baik. Tujuan utama pengujian ini bukan mencari bug, tetapi untuk menghilangkan *performance bottleneck* [3].

2.4 Apache JMeter

Apache JMeter adalah aplikasi Java 100% murni yang dimaksudkan untuk mengukur kinerja dan memuat tes perilaku fungsional. Awalnya dibuat untuk menguji aplikasi Web, tetapi sekarang telah diperluas untuk melakukan fungsi tes lainnya [3]. Untuk seorang QA Engineer, Apache Jmeter adalah aplikasi open source berbasis Java yang dapat digunakan untuk melakukan pengujian kinerja. Untuk melakukannya, JMeter dapat melakukan pengujian *load* /stress untuk aplikasi web, aplikasi FTP, dan pengujian *database server*.

Apache Jmeter dipilih karena, selain merupakan aplikasi open source, ia memiliki antarmuka pengguna yang mudah digunakan, kemampuan untuk menguji beban dan performa pada berbagai jenis aplikasi, *server*, dan protokol, dan kemampuan untuk mengambil sampel secara bersamaan dari berbagai fungsi pengujian yang dilakukan [5]. Apache Jmeter juga dirancang khusus untuk melakukan pengujian pada aplikasi *website*.

Jmeter mengirimkan request ke *webserver*, yang kemudian akan merespon request tersebut. Untuk setiap request yang dikirim oleh jmeter, *webserver* menghasilkan hasil seperti *load time*, ukuran byte, ukuran header, ukuran body, dan kode tanggapan untuk pesan respons [7].

2.5 Website

Website berasal dari kata World Wide Web, yakni layanan yang didapati oleh pemakai komputer yang terhubung dengan jaringan internet. *Website* merupakan aplikasi tertentu yang berjalan di atas platform atau operation system browser. Oleh karena itu, *website* yang dimaksud adalah halaman web yang

tersedia secara *online* dan dapat diakses dari mana pun yang terhubung ke jaringan internet [7]

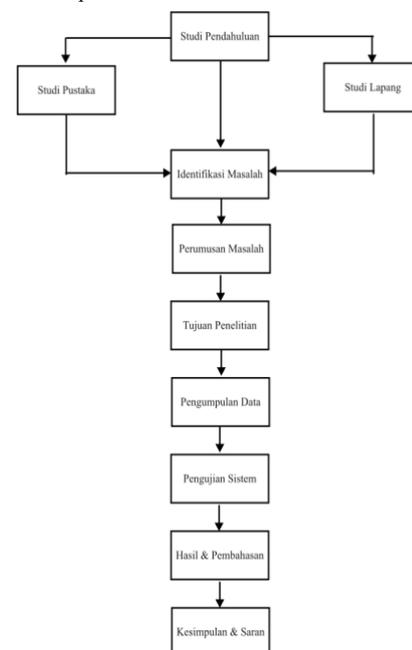
Salah satu istilah untuk "*website*" mengacu pada sekumpulan halaman yang terdiri dari beberapa halaman yang masing-masing berisi informasi digital, seperti teks, gambar, video, audio, dan animasi lainnya, yang disediakan melalui jalur internet. Dengan kata lain, *website* adalah halaman web yang berisi informasi yang ditampilkan oleh browser seperti Firefox, Chrome, dan Mozilla [8].

3 Metodologi

Metode yang digunakan dalam penelitian ini yaitu *performance testing* dengan jenis *load testing*. *Performance testing* merupakan salah satu jenis pengujian kinerja yang menentukan atau memverifikasi fitur kecepatan, skalabilitas, dan stabilitas aplikasi dibawah beban kerja tertentu [4].

Sedangkan, *load testing* merupakan pengujian yang bertujuan untuk menentukan target *load* dari sebuah sistem. Pengujian ini diperlukan saat adanya sebuah *event* musiman yang menyebabkan lalu lintas (*traffic*) sistem bertambah seiring berjalannya waktu. Pengujian akan dilakukan dengan memberikan sebuah respon dan data dari semua transaksi yang dilakukan sesuai skenario, sehingga dapat dengan mudah mencari kemacetan atau *bottleneck* s yang mempengaruhi performansi dari suatu sistem yang telah diuji [5].

Selanjutnya berikut adalah tahapan-tahapan pengujian dengan menggunakan metode *load testing*. Dalam kegiatan penelitian yang dilakukan ini ditunjukkan dalam Gambar 1 di bawah ini. Kegiatan penelitian ini diawali dari tahapan studi pendahuluan, melakukan identifikasi permasalahan, perumusan masalah, penentuan tujuan penelitian, pengumpulan data, selanjutnya pengujian sistem, hasil dan pembahasan, dan terakhir membuat kesimpulan.



Gambar 1. Tahapan Penelitian

4 Pembahasan

Berikut adalah penjelasan mengenai pengujian yang dilakukan pada website LSP SMKN 2 Kraksaan menggunakan metode *load testing* dengan tools apache jmeter.

4.1 Pengumpulan Data

Pengumpulan data dimulai dari pengambilan data berupa link URL halaman *website* LSP SMKN 2 Kraksaan bagian asesi. *Website* LSP SMKN 2 Kraksaan bagian asesi digunakan oleh peserta yang akan melakukan ujikom LSP. Dalam pengambilan data, didapatkan beberapa menu seperti berikut :

Tabel 1. Pengumpulan Data

No	Nama Page	Link/URL
1	Home	https://adminlsp.smkn2kraksaan.sch.id
2	Beranda	https://adminlsp.smkn2kraksaan.sch.id/beranda
3	Profil	https://adminlsp.smkn2kraksaan.sch.id/profil_peserta
4	Permohonan	https://adminlsp.smkn2kraksaan.sch.id/permohonan_peserta
5	Upload Bukti	https://adminlsp.smkn2kraksaan.sch.id/tampil Upload_bukti
6	Pengajuan	https://adminlsp.smkn2kraksaan.sch.id/status_pengajuan

4.2 Test Case

Langkah berikutnya adalah melakukan *Test Case*. Tujuan pembuatan *Test Case* adalah untuk memastikan bahwa suatu sistem dapat berfungsi dengan baik sesuai dengan kebutuhan awal dan mampu memberikan respons ketika ada masukan yang valid atau tidak valid [8].

Tabel 2. Test Case

No	Jumlah User	Keterangan
1	10	Setengah siswa dari 1 kelas
2	20	Jumlah siswa 1 kelas
3	50	Jumlah siswa 2 kelas
4	100	Jumlah siswa 1 jurusan
5	500	Jumlah siswa 2 jurusan
6	1000	Jumlah siswa langkatan

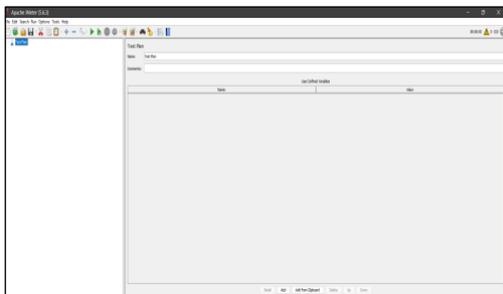
Tabel di atas menunjukkan jumlah user yang diperlukan untuk melakukan pengujian. *Test Case* ini menggunakan penilaian virtual, dimana dengan ini tidak ada orang masyarakat sekolah atau di luar sekolah yang terlibat secara langsung. Jumlah data untuk penilaian ini diambil dari angka terdekat atau perkiraan di lapangan.

4.3 Pengujian dengan JMeter

Berikut adalah langkah-langkah untuk melakukan pengujian beban menggunakan Apache JMeter :

a. Membuat Test Plan

Membuat rencana uji yang berisi urutan komponen uji yang akan menentukan bagaimana sebuah *server* akan disimulasikan.



Gambar 2. Test Plan JMeter

b. Membuat Thread Group

Rencana uji selanjutnya adalah *Thread Group*, yang digunakan untuk menggambarkan seorang pengguna aplikasi selama kegiatan pengujian. Untuk simulasi pengujian, penulis dapat menetapkan jumlah "pengguna" dalam pengaturan thread

group sesuai dengan *Test Case* yang sudah dibuat sebelumnya. Sebagai contoh, penulis mensimulasikan jumlah user sebanyak 20 dengan total waktu yang dibutuhkan selama 1 second dengan pengulangan sebanyak 1 kali pada bagian "beranda".

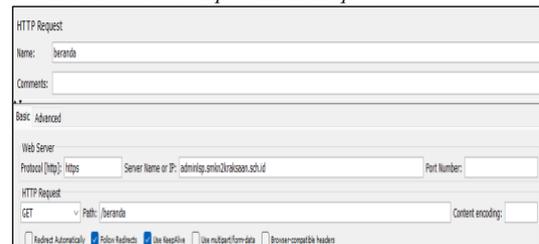


Gambar 3. Thread Group JMeter

Adapun bagian "Properties Thread" di menu Thread Group yang berfungsi sebagai berikut:

- *Number of Threads (users)*: mensimulasikan jumlah user.
- *Ramp-up period (seconds)*: total waktu yang dibutuhkan untuk jumlah user yang didefinisikan di Number of Thread.
- *Loop Count*: jumlah literasi atau perulangan test [9] [10][11].

c. Menambahkan sampler HTTP Request



Gambar 4. Sampler HTTP Request JMeter

Dalam HTTP Request Sampler, penulis dapat menuliskan beberapa bagian, seperti memasukkan protocol ke *website* objek penelitian melalui HTTP, memasukkan IP link atau URL menu "https://adminlsp.smkn2kraksaan.sch.id", dan Path diisi sesuai dengan pengujian di setiap menu.

d. Hasil

Bagian komponen Listener Apache JMeter yang digunakan untuk melakukan penelitian pengujian beban berupa "View Results in Table" untuk menampilkan data berupa tabel dan untuk menghasilkan data berupa file lebih ringkas dan berformat CSV. Keberhasilan pengujian akan menampilkan "Success" dalam kolom dengan keterangan true bahwa pengujian tersebut berhasil, namun jika menampilkan "False" dalam kolom maka keterangan gagal untuk melakukan pengujian.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Send Bytes	Latency	Connect Time...
1	22:34:20.803	Beranda 20 1-1	beranda	391	✓	8964	141	390	117
2	22:34:20.905	Beranda 20 1-2	beranda	352	✓	8971	141	351	118
3	22:34:21.006	Beranda 20 1-4	beranda	288	✓	8971	141	287	149
4	22:34:21.106	Beranda 20 1-6	beranda	347	✓	8973	141	346	116
5	22:34:21.205	Beranda 20 1-7	beranda	332	✓	8971	141	332	114
6	22:34:21.305	Beranda 20 1-8	beranda	337	✓	8976	141	337	116
7	22:34:21.405	Beranda 20 1-9	beranda	325	✓	8971	141	324	116
8	22:34:21.507	Beranda 20 1-10	beranda	288	✓	8967	141	288	125
9	22:34:21.606	Beranda 20 1-10	beranda	694	✓	8968	141	694	416
10	22:34:21.657	Beranda 20 1-5	beranda	456	✓	8972	141	456	116
11	22:34:21.755	Beranda 20 1-11	beranda	365	✓	8969	141	365	119
12	22:34:21.855	Beranda 20 1-12	beranda	365	✓	8969	141	364	118
13	22:34:21.953	Beranda 20 1-13	beranda	394	✓	8969	141	393	119
14	22:34:21.954	Beranda 20 1-15	beranda	1446	✓	8972	141	1446	116
15	22:34:21.955	Beranda 20 1-17	beranda	413	✓	8967	141	412	119
16	22:34:21.754	Beranda 20 1-19	beranda	357	✓	8972	141	357	124
17	22:34:21.803	Beranda 20 1-20	beranda	361	✓	8972	141	361	124
18	22:34:21.944	Beranda 20 1-4	beranda	1446	✓	8972	141	1446	116
19	22:34:21.705	Beranda 20 1-16	beranda	1467	✓	8974	141	1466	117
20	22:34:21.805	Beranda 20 1-16	beranda	284	✓	8979	141	1790	148

Gambar 5. Hasil Pengujian Beban

Hasil pengujian beban gambar 5 diatas dengan menggunakan *Test Case* 20 user ditunjukkan berupa tabel GUI JMeter bahwa *website* LSP SMKN 2 Kraksaan pada menu beranda mampu untuk menampung beban yang diberikan.

Kesimpulan yang didapat dalam hasil *performance testing* dengan data dari *website* LSP SMKN 2 Kraksaan bagian asesi, berikut dibawah ini batas wajar pengguna yang dapat mengakses *website* ini :

Tabel 3. Kesimpulan Hasil Pengujian

No	Nama Page	Batas Performa
1	Home	N/A (tidak terbaca pada JMeter)
2	Beranda	500 User
3	Profil	500 User
4	Permohonan	500 User
5	Upload Bukti	500 User
6	Pengajuan	500 User

4.4 Hasil Analisa Pengujian

Selanjutnya menganalisis parameter performa tiap menu dengan beban yang digunakan sebanyak 10 dan 50 user sebagai perbandingan menggunakan tambahan listener aggregate graph pada JMeter. Parameter performa yang dicatat meliputi *response time*, *throughput*, dan *error rate* untuk setiap menu pada *website*.

Tabel 4. Hasil Uji 10 User

Label	Response time	Error %	Throughput
Home	572 ms	100.00%	3.03122 ms
Beranda	542 ms	0.00%	2.96736 ms
Profil	592 ms	0.00%	3.21543 ms
Permohonan	586 ms	0.00%	3.0175 ms
Upload bukti	521 ms	0.00%	3.31675 ms
Pengajuan	450 ms	0.00%	4.53309 ms
TOTAL	544 ms	16.6 7%	12.84522 ms

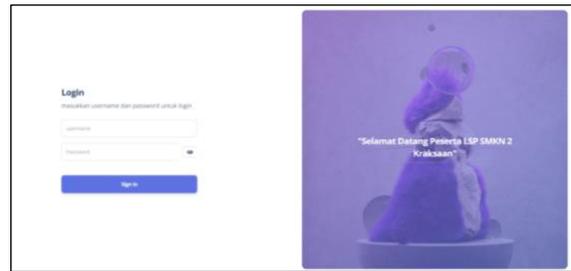
Tabel 5. Hasil Uji 50 User

Label	Response time	Error %	Throughput
Home	1823 ms	100.00%	9.32836 ms
Beranda	2160 ms	0.00%	6.20424 ms
Profil	2544 ms	0.00%	5.10777 ms
Permohonan	3240 ms	0.00%	4.17885 ms
Upload bukti	3064 ms	0.00%	4.10105 ms
Pengajuan	2522 ms	0.00%	5.06637 ms
TOTAL	2559 ms	16.67%	17.19296 ms

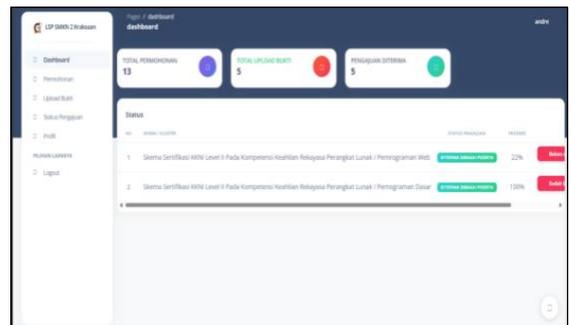
Kesimpulan yang didapatkan dari hasil pengujian dengan JMeter pada skenario 10 dan 50 virtual user, terlihat bahwa adanya *peningkatan respons time* seiring bertambahnya jumlah pengguna. Hal ini menunjukkan bahwa peningkatan beban pada *server* berdampak langsung terhadap performa *website*. Secara keseluruhan, hasil pengujian ini menunjukkan bahwa *website* mengalami penurunan kualitas performa ketika beban user meningkat, dengan indikasi utama berupa kenaikan waktu *respons*, *error* yang konsisten pada menu "Home", dan potensi *bottleneck* pada beberapa menu dengan lonjakan waktu *respons* tinggi.

4.5 Rekomendasi Hasil

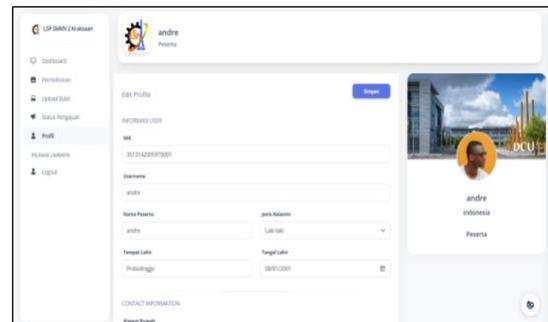
Untuk mengoptimalkan hasil dari pengujian beban pada endpoint "Home" yang sebelumnya mengalami tingkat kegagalan tinggi, maka penulis membuat rekomendasi *website* dengan tautan baru yang kemudian akan dilakukan pengujian beban ulang menggunakan JMeter. Adapun *website* yang direkomendasikan penulis memuat menu-menu seperti berikut ini :



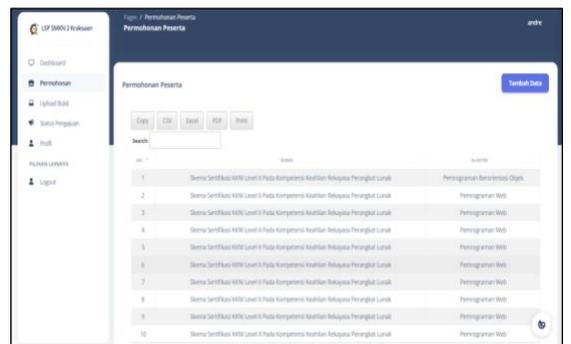
Gambar 6. Tampilan Rekomendasi Home



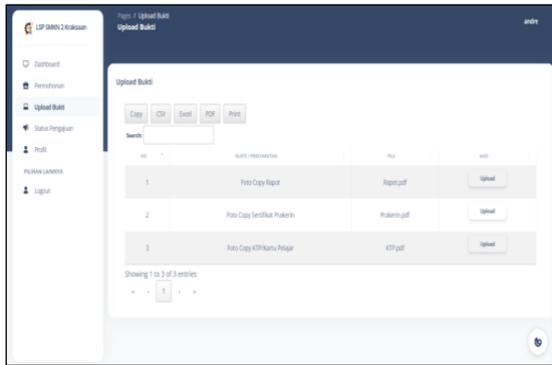
Gambar 7. Tampilan Rekomendasi Beranda



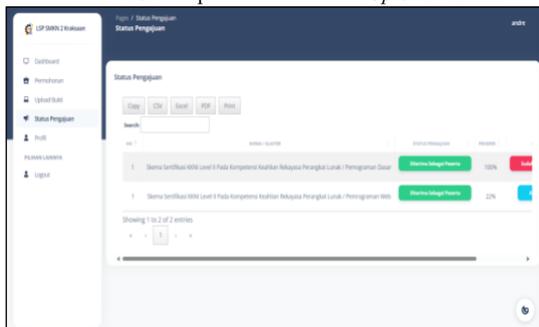
Gambar 8. Tampilan Rekomendasi Profil



Gambar 9. Tampilan Rekomendasi Permohonan



Gambar 10. Tampilan Rekomendasi Upload Bukti



Gambar 11. Tampilan Rekomendasi Pengajuan

Selanjutnya penulis melakukan pengujian terhadap *website* yang penulis rekomendasikan dengan menggunakan JMeter. Pengujian dilakukan dengan menggunakan *Test Case* yang sebelumnya sudah disusun. Dibawah ini adalah hasil *performance testing* dari rekomendasi *website* yang sudah dibuat, dimana batas performa yang dapat mengakses rekomendasi *website* sebagai berikut :

Tabel 6. Hasil Kesimpulan Pengujian Setelah Rekomendasi

No	Nama Page	Batas Performa
1	Home	500 User
2	Beranda	500 User
3	Profil	500 User
4	Permohonan	500 User
5	Upload Bukti	500 User
6	Pengajuan	500 User

Setelah dibuatnya rekomendasi *website*, hasil pengujian beban yang diperoleh menunjukkan peningkatan dibandingkan pengujian beban sebelumnya.

Tabel 7. Hasil Uji 10 User Setelah Rekomendasi

Label	Response time	Error %	Throughput
Home	223 ms	0.00%	8.74126 ms
Beranda	400 ms	0.00%	7.01262 ms
Profil	309 ms	0.00%	8.30565 ms
Permohonan	319 ms	0.00%	6.59631 ms
Upload bukti	276 ms	0.00%	6.61813 ms
Pengajuan	201 ms	0.00%	6.9979 ms
TOTAL	288 ms	0.00%	20.30457 ms

Tabel 8. Hasil Uji 50 User Setelah Rekomendasi

Label	Response time	Error %	Throughput
Home	1970 ms	0.00%	13.24854 ms
Beranda	2833 ms	0.00%	5.23286 ms
Profil	2664 ms	0.00%	6.37267 ms
Permohonan	1849 ms	0.00%	5.43715 ms

Upload bukti	1905 ms	0.00%	5.3775 ms
Pengajuan	1328 ms	0.00%	8.3949 ms
TOTAL	2092 ms	0.00%	21.39037 ms

Secara keseluruhan, *website* yang direkomendasikan penulis berhasil meningkatkan performa dengan menurunnya *response time*, menghilangkan *error*, dan meningkatkan *throughput*. Sehingga *website* menjadi lebih stabil, cepat, dan mampu menangani lebih banyak permintaan pengguna.

5 Kesimpulan

Diperoleh kesimpulan bahwa *website* tersebut mampu menangani beban akses pengguna secara bersamaan hingga 500 pengguna dalam durasi 1 detik, namun mengalami kegagalan pada saat diuji dengan 1000 pengguna karena melebihi batas maksimal kapasitasnya, dengan tingkat kegagalan mencapai 100% khususnya pada menu "home". Pengujian juga berhasil menampilkan hasil secara visual dalam bentuk tabel yang mencerminkan rata-rata waktu respons, tingkat *error*, dan *throughput* pada setiap menu di *website* tersebut. Setelah diberikan rekomendasi perbaikan, performa *website* menunjukkan peningkatan yang signifikan dengan waktu respons yang lebih cepat, *error* 0%, dan *throughput* yang lebih tinggi, sehingga *website* menjadi lebih stabil dan efisien dalam menangani beban pengguna.

Kontribusi Penulis

Kontribusi masing-masing penulis dalam penelitian ini adalah sebagai berikut: Konseptualisasi, Penulis 1 dan Penulis 2; metodologi, Penulis 1; perangkat lunak, Penulis 1; validasi, Penulis 2, Penulis 3, dan Penulis 4; analisis formal, Penulis 1; investigasi, Penulis 3; sumber daya, Penulis 4; kurasi data, Penulis 5; penulisan—persiapan draf asli, Penulis 1; menulis—meninjau dan mengedit, Penulis 2; visualisasi, Penulis 3; pengawasan, Penulis 4; administrasi, Penulis 4; perolehan pendanaan, Penulis 2. Setiap penulis telah memberikan kontribusi signifikan dalam penelitian ini, memastikan keberhasilan dan kualitas hasil yang dilaporkan.

Ucapan Terima Kasih

Kami mengucapkan terima kasih yang tulus dan sebesar-besarnya kepada LSP SMKN 2 Kraksaan Probolinggo, Jawa Timur atas dukungan dan layanan yang luar biasa dalam penyediaan data-data di lapangan yang akurat dan tepat waktu. Semoga hasil penelitian ini dapat bermanfaat dan kerjasama ini terus berlanjut dan membawa manfaat bagi masyarakat serta kemajuan.

Referensi

- [1] M. R. D. F. Aly, *Analisis Implementasi Revitalisasi Sumber Daya Manusia Dan Kurikulum Di SMK*. Universitas Islam Negeri Jakarta, 2022.
- [2] A. Maulana and C. Juliane, "Implementasi Manajemen Pengetahuan dalam Sistem Layanan Sertifikasi Profesi (Studi Kasus SMKN 1 Garut)," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 1897–1904, 2022, doi: 10.35957/jatisi.v9i3.2343.
- [3] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.
- [4] Rahadi, N. W., & Vikasari, C. Pengujian *Software* Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions. *Infotekmesin*, 11(1), 57–61. <https://doi.org/10.35970/infotekmesin.v11i1.124>. 2020

-
- [5] Suprpto, A., & Sasongko, D. Evaluasi Performa *Website* Berdasarkan Pengujian Beban Dan Stress Menggunakan *Load impact* (Studi Kasus *Website* Iain Salatiga). *Network Engineering Research Operation*, 6(1), 31. <https://doi.org/10.21107/nero.v6i1.198>. 2021
- [6] Tejaya, W., Rahman, S., Munir, A., Web, S., Jmeter, A., & Muat, W. Pengujian *Website* Invitees Menggunakan Metode *Load*. *Jurnal Kharisma Tech*, 01, 99–112. 2023
- [7] Busran, & Ridwan, A. Analisis Perbandingan Performa Apache Web *Server* dan Nginx Menggunakan Apache Jmeter. *Jurnal Teknoif*, 8(2), 87–92. 2020
- [8] D. 'Madhani, E. 'Darwiyanto, and A. 'Ghandi, "Performance Testing Menggunakan Metode *Load Testing* dan *Stress Testing* pada Sistem Core Banking PT. XYZ," e-Proceeding Eng., vol. 10, no. 6, pp. 1–11, 2023
- [9] A. Suprpto and D. Sasongko, "Evaluasi Performa *Website* Berdasarkan Pengujian Beban Dan Stress Menggunakan *Load impact* (Studi Kasus *Website* Iain Salatiga)," *Netw. Eng. Res. Oper.*, vol. 6, no. 1, p. 31, 2021, doi: 10.21107/nero.v6i1.198.
- [10] Arafat, M. Rancang Bangun Sistem Informasi Pemesanan *Online* Percetakan Sriwijaya Multi Grafika Berbasis *Website*. *Intech*, 3(2), 6–11. <https://doi.org/10.54895/intech.v3i2.1691>. 2022.
- [11] Hasibuan, A. N., & Dirgahayu, T. Pengujian dengan Unit *Testing* dan *Test Case* pada Proyek Pengembangan Modul Manajemen Pengguna. *Jurnal Informatika Universitas Islam Indonesia*, 2(1), 103–109. 2020