

PERFORMANCE TESTING OF ACADEMIC WEBSITE USING LOAD TESTING METHOD SUPPORTED BY APACHE JMETERTM AT XYZ UNIVERSITY

Soni Sampari Raweyai^{*1}, Indrastanti Ratna Widiarsari²

^{1,2}Informatics Engineering Study Program, Universitas Kristen Satya Wacana, Indonesia
Email: ¹672018349@student.uksw.edu, ²indrastanti@uksw.edu

(Article received: January 27, 2024; Revision: February 28, 2024; published: May 27, 2024)

Abstract

This study aims to evaluate the performance quality of the academic website of University XYZ through load testing using the Load Testing Method supported by Apache JMeterTM. The main issue addressed is how the website's performance can be measured and assessed in the context of normal, peak, and stress usage. The research methodology involves a qualitative approach to understand the meaning, interpretation, and context of the phenomenon, coupled with a quantitative approach to measure, analyze, and organize data in numerical or statistical forms. The research findings indicate that in the basic testing scenario, the website successfully met the test criteria with an average response time of approximately 0.855 seconds for GET requests, below the established maximum limit. POST requests required an average time of around 0.273 seconds with no response failures. In the peak testing scenario simulating high traffic conditions, the website remained optimal with average response times for both GET and POST requests below the 3-second limit, without response failures. Stress testing scenarios demonstrated the efficient operation of the website, even though the average response time for GET requests increased to approximately 2.564 seconds. The test results affirm that the University XYZ website functions well under various service usage conditions, including heavy loads. The overall average response time for GET requests across all scenarios is approximately 1.558 seconds, while POST requests have an average response time of around 0.355 seconds. Special attention is given to the impact of the number of threads or users and the number of students on the website's performance.

Keywords: Academic Website, Apache JMeterTM, Performance Quality, Load Testing Method..

TESTING KUALITAS KINERJA WEBSITE AKADEMIK MENGGUNAKAN METODE LOAD TESTING DIDUKUNG DENGAN APACHE JMETERTM PADA UNIVERSITAS XYZ

Abstrak

Penelitian ini bertujuan untuk mengevaluasi kualitas kinerja *website* akademik Universitas XYZ melalui uji beban menggunakan Metode *Load Testing* yang didukung oleh *Apache JMeterTM*. Abstrak ini disusun dengan menekankan permasalahan, tujuan, metode, dan hasil penelitian. Permasalahan utama yang diangkat adalah bagaimana kinerja *website* dapat diukur dan dievaluasi dalam konteks penggunaan normal, puncak, dan stres. Metode penelitian melibatkan pendekatan kualitatif untuk memahami makna, interpretasi, dan konteks fenomena, serta pendekatan kuantitatif untuk mengukur, menganalisis, dan menyusun data dalam bentuk angka atau statistik. Hasil penelitian menunjukkan bahwa pada skenario pengujian dasar, *website* berhasil memenuhi target uji dengan waktu rata-rata permintaan GET sekitar 0,855 detik, di bawah batas maksimal yang ditetapkan. Permintaan POST membutuhkan waktu rata-rata sekitar 0,273 detik tanpa kegagalan respons. Pada skenario pengujian puncak, *website* tetap optimal dengan waktu rata-rata permintaan GET dan POST di bawah batas 3 detik, tanpa kegagalan respons. Skenario pengujian stres menunjukkan bahwa *website* beroperasi dengan efisien, meskipun waktu rata-rata permintaan GET meningkat menjadi sekitar 2,564 detik. Hasil pengujian menegaskan bahwa *website* Universitas XYZ berfungsi baik dalam berbagai kondisi penggunaan layanan, termasuk beban berat. Rata-rata waktu respons untuk permintaan GET pada semua skenario adalah sekitar 1,558 detik, sementara permintaan POST memiliki rata-rata waktu respons sekitar 0,355 detik. Perhatian khusus diberikan pada dampak jumlah thread atau pengguna serta jumlah mahasiswa pada kinerja *website*.

Kata kunci: Apache JMeterTM, Kualitas Kinerja, Metode Pengujian Beban, Website Akademis.

1. PENDAHULUAN

Dalam era digital saat ini, penggunaan *website* akademik telah menjadi hal umum di lingkungan perguruan tinggi [1]. *Website* ini menjadi sarana vital dalam menyediakan informasi, layanan, dan interaksi di antara mahasiswa, dosen, dan pihak terkait lainnya [2]. Dengan meningkatnya kompleksitas sistem dan jumlah pengguna, kualitas kinerja *website* akademik menjadi aspek krusial yang mempengaruhi pengalaman pengguna [3]. Keterlambatan respons, lambatnya memuat halaman, atau bahkan kegagalan saat dihadapkan pada beban kerja tinggi dapat menyebabkan ketidakpuasan bagi pengguna, terutama mahasiswa dan dosen yang mengandalkan akses cepat ke informasi akademik. [4].

Kualitas kinerja *website* akademik sangat berpengaruh terhadap pengalaman pengguna. Jika *website* mengalami keterlambatan respons, lambat dalam memuat halaman, atau bahkan mengalami kegagalan saat dihadapkan pada beban kerja yang tinggi, hal ini dapat menimbulkan ketidakpuasan bagi pengguna [5]. Mahasiswa dan dosen yang membutuhkan akses cepat dan mudah ke informasi akademik dapat mengalami kendala yang menghambat kegiatan akademik mereka. Oleh karena itu, penting untuk melakukan pengujian kualitas kinerja *website* akademik secara menyeluruh. Salah satu metode yang dapat digunakan adalah *Load Testing*, di mana *website* diuji dengan memberikan beban kerja yang tinggi secara bersamaan [6].

Load testing adalah metode pengujian perangkat lunak yang dirancang untuk mengevaluasi kinerja dan daya tahan suatu sistem atau aplikasi di bawah kondisi beban kerja yang tinggi atau beban maksimal [7]. Tujuan utama dari *load testing* adalah untuk mengidentifikasi batas kemampuan sistem, menemukan titik kegagalan, dan menilai seberapa baik sistem dapat menjaga kinerjanya di bawah tekanan ekstrem. Dalam konteks penggunaan *website* akademik, *load testing* dapat memberikan pemahaman mendalam tentang bagaimana *website* tersebut berperilaku ketika diakses oleh banyak pengguna secara bersamaan. Beban kerja tinggi dapat disimulasikan untuk mengukur seberapa cepat *website* merespons permintaan, seberapa baik dapat menangani jumlah pengguna simultan, dan sejauh mana daya tampungnya [8].

Load Testing membantu mengidentifikasi batas kinerja *website*, mengungkapkan masalah yang mungkin muncul pada tingkat beban yang tinggi, serta memberikan wawasan tentang performa dan keandalan *website* akademik [9]. Dalam konteks Universitas XYZ, *website* akademiknya menjadi pusat informasi dan layanan bagi mahasiswa, dosen, dan staf universitas. Namun, dengan adanya lonjakan jumlah pengguna dan permintaan yang tinggi, kualitas kinerja *website* akademik mungkin menjadi perhatian. Mahasiswa yang mencoba mengakses jadwal kuliah, hasil ujian, atau pengumuman penting mungkin menghadapi kesulitan jika *website* tidak

mampu menangani beban kerja yang tinggi. Oleh karena itu, perlu dilakukan penelitian tentang kualitas kinerja *website* akademik Universitas XYZ menggunakan metode *Load Testing* yang didukung oleh *Apache JMeter* [10].

Studi kasus ini akan memberikan gambaran yang jelas tentang kemampuan *website* akademik dalam mengatasi beban kerja yang tinggi dan memberikan pengalaman pengguna yang baik. Hasil penelitian ini diharapkan dapat memberikan wawasan dan rekomendasi untuk meningkatkan kualitas kinerja *website* akademik Universitas XYZ serta dapat diterapkan pada institusi pendidikan lainnya [11].

Seperti pada penelitian yang telah dilakukan oleh Teresa Irmina dan Eroza Efriana, penulis mendapatkan bahwa penelitian ini mengevaluasi kualitas *website* resmi empat Pemerintah Daerah, yaitu Kabupaten Situbondo, Jember, Bondowoso, dan Banyuwangi, dalam konteks implementasi *e-Government* menggunakan metode *Automated Usability Testing Tools*, dengan *GTMetrix* sebagai alat uji. Evaluasi selama 10 hari kerja berfokus pada indikator seperti *Performance Score*, *Structured Score*, *Page Load Time*, *Largest Contentful Paint*, *Total Blocking Time*, dan *Cumulative Layout Shift*. Hasil menunjukkan Kabupaten Bondowoso unggul dalam optimalisasi *website*, terutama pada *Performance Score*, sedangkan Kabupaten Situbondo memimpin dalam *Structured Score*. Bondowoso memiliki *Page Load Time* tercepat dan *Largest Contentful Paint* paling cepat. *Total Blocking Time* menunjukkan kinerja baik untuk Situbondo dan Bondowoso. Pada *Cumulative Layout Shift*, Bondowoso dan Banyuwangi mendapat nilai baik. Evaluasi ini memberikan wawasan penting untuk meningkatkan layanan *e-Government* dan efisiensi layanan publik di daerah studi [12].

Selanjutnya penelitian dari Ikhwal, Naila, et al, penulis mendapatkan bahwa Pemanfaatan teknologi informasi menjadi kebutuhan mendasar dalam dunia bisnis saat ini, namun sulit menentukan kualitas perangkat lunak yang mendukung proses bisnis perusahaan. Kegagalan dalam pengembangan dan implementasi sistem informasi/perangkat lunak dapat berdampak besar bagi perusahaan. Oleh karena itu, pengukuran kualitas perangkat lunak sebelum peluncuran menjadi penting untuk memastikan kematangan aplikasi tersebut. Penelitian ini dilakukan di PT. Walden *Global Services* Bandung, perusahaan penjaminan mutu perangkat lunak, dengan fokus pengujian pada *website* aplikasi PT. Gramedia Pustaka Utama (www.sp.gramediadigital.com). Metode pengujian menggunakan pendekatan *blackbox*, mencakup Uji Fungsional, Uji Keamanan, Uji Kinerja Ujung Depan, Uji Kinerja Ujung Belakang, dan Uji Antarmuka Pengguna. Hasil pengujian ini bertujuan untuk meyakinkan pelanggan bahwa sistem layak digunakan dan memenuhi persyaratan kinerja [13].

Penelitian dari Randi, Ruuhwan dan Alam, penulis menemukan bahwa penelitian menggunakan *GTmetrix* untuk mengevaluasi kualitas website berdasarkan *PageSpeed* dan *YSlow*, dengan metode load view sebagai standar pengujian kesetabilan dan kecepatan akses. Hasil pengujian *GTmetrix* menunjukkan i dengan *Grade D* (65%), sedangkan *YSlow Score* lebih baik dengan *Grade C* (73%), *fully Load 7.2 s*, *total page 2.49 MB*, dan jumlah *request 26*. Penyebab kurang maksimalnya *PageSpeed* adalah *Serve Scaled Image*, *Optimize Image*, dan *Leverage Browser Cache*. Analisis load view menunjukkan bahwa gambar berukuran besar berformat png menjadi penyebab lambatnya load [3].

Penelitian dari Wante, Syaiful dan Abdul, penulis menemukan bahwa penelitian ini menggunakan *Apache JMeter* untuk melakukan uji performa pada website *Invitees* dengan fokus pada waktu muat fungsi utama. Data dikumpulkan melalui studi dokumen yang mencakup statistik jumlah pengunjung dan user *Invitees* pada periode Januari-Juni 2022. Analisis deskriptif digunakan untuk menganalisis data. Hasil pengujian menunjukkan waktu muat rata-rata pada skenario pertama (2.7 detik) dan skenario kedua (1.7 detik) memenuhi target pengujian (<3 detik). Namun, pada skenario ketiga dengan beban dua kali lebih banyak, waktu muat rata-rata sebesar 4.2 detik, tidak memenuhi target pengujian. Kesimpulannya, website *Invitees* berfungsi optimal saat layanannya digunakan, tetapi perlu perbaikan jika diakses oleh lebih dari 10 user secara bersamaan [14].

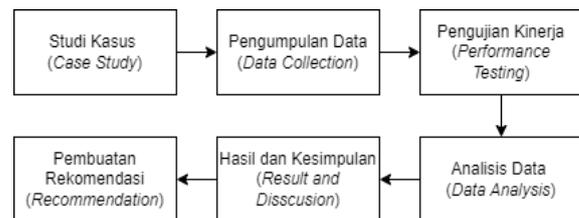
Penelitian dari Ade, Ahmadi et al, penulis menemukan bahwa Penelitian ini berfokus pada *performance testing*, sebuah proses pengujian perangkat lunak untuk menguji berbagai aspek seperti kecepatan, waktu *respons*, stabilitas, keandalan, skalabilitas, dan penggunaan resource di bawah beban kerja tertentu. Menggunakan *TCEExam* sebagai sistem ujian *online*, pengujian *load testing* dengan target menangani 1000 user dilakukan dengan pembatasan spesifikasi *server*. Hasil pengujian dengan *Apache JMeter* menunjukkan bahwa, meskipun menggunakan spesifikasi *server* tertinggi, masih terdapat *error* dalam *load testing* sehingga target 1000 user belum dapat terpenuhi. Virtualisasi mempermudah pengaturan spesifikasi server sesuai kebutuhan pengujian [15].

Berdasarkan beberapa penelitian yang didapat dan permasalahan yang dijelaskan sebelumnya serta latar belakang, penelitian ini bertujuan sebagai alat untuk Testing Kualitas Kinerja Website Akademik menggunakan Metode *Load Testing* didukung dengan *Apache JmeterTM* (Studi Kasus: Universitas XYZ) untuk mensimulasikan seberapa besar daya tamping dari website akademik yang dites.

2. METODE PENELITIAN

Metode penelitian yang digunakan untuk penelitian "*Testing* Kualitas Kinerja Website

Akademik menggunakan Metode *Load Testing* didukung dengan *Apache JMeterTM* (Studi Kasus: Universitas XYZ)" yaitu kualitatif dan kuantitatif [16], pendekatan kualitatif digunakan untuk memahami dan menjelaskan makna, interpretasi, dan konteks dari suatu fenomena. Fokusnya pada pemahaman mendalam, nuansa, dan kompleksitas [17], sedangkan untuk pendekatan kuantitatif digunakan untuk mengukur, menganalisis, dan menyusun data dalam bentuk angka atau statistik. Fokusnya pada generalisasi, pengujian hipotesis, dan pembuatan generalisasi yang dapat diterapkan secara lebih umum [18], gambaran alur metode penelitian secara jelas dapat dilihat pada Gambar 1:



Gambar 1. Alur Metode Penelitian

Gambar 1 menjelaskan mengenai tahapan proses alur metode penelitian yang digunakan penulis dalam penelitian ini. Tahap Studi Kasus (*Case Study*) berguna sebagai pendekatan yang memberikan pemahaman mendalam tentang kualitas kinerja website akademik di Universitas XYZ. Metode ini melibatkan pengumpulan data dari kasus nyata yang ada di lingkungan universitas tersebut. Data yang dikumpulkan meliputi hasil pengujian kinerja website akademik menggunakan *Apache JMeterTM*, parameter kinerja yang dievaluasi, serta masalah dan rekomendasi yang dihasilkan.

Tahap Pengumpulan Data (*Data Collection*) dilakukan melalui beberapa metode, seperti observasi langsung, wawancara dengan pihak terkait (misalnya, administrator website, dosen, dan mahasiswa), dan analisis dokumentasi yang terkait dengan website akademik Universitas XYZ. Data yang dikumpulkan mencakup informasi tentang infrastruktur website, karakteristik pengguna, dan kebutuhan kinerja yang diharapkan, pengumpulan data mengenai jumlah pengunjung yang mengakses website menggunakan pendekatan:

$$\text{Traffic Perhari} = \frac{\text{Bulan Ke1} - \text{Bulan Ke2}}{\text{Jumlah bulan} \times \text{Jumlah hari}} \quad (1)$$

Dari pendekatan diatas dijelaskan mengenai bulan ke1 yang merupakan bulan dengan jumlah *traffic* meningkat drastis dan akan dikurangi dengan jumlah *traffic* bulan ke2 selanjutnya akan dibagi dengan jumlah bulan yang dikalikan dengan jumlah hari akan didapatkan jumlah total pengunjung atau *traffic*.

Tahap Pengujian Kinerja (*Performance Testing*) Metode *Load Testing* menggunakan *Apache JMeterTM* akan digunakan sebagai metode pengujian kinerja dalam penelitian ini. *Apache JMeterTM* dapat

digunakan untuk menghasilkan beban kerja yang tinggi pada website akademik dan merekam kinerja sistem dalam kondisi tersebut. Data yang dikumpulkan dari pengujian ini akan memberikan wawasan tentang kualitas kinerja saat ini, batasan sistem, dan masalah yang mungkin muncul.

Tahap Analisis Data (*Data Analysis*), Data yang dikumpulkan dari pengujian kinerja menggunakan *Apache JMeter* akan dianalisis secara kualitatif dan/atau kuantitatif. Metode analisis yang dapat digunakan termasuk analisis statistik, pemetaan beban kerja, perbandingan hasil pengujian dengan parameter kinerja yang ditetapkan, dan identifikasi masalah kinerja yang muncul.

Tahap Hasil dan Kesimpulan (*Result and Discussion*), tahap ini menjelaskan mengenai seluruh hasil yang didapatkan dari tahapan sebelumnya serta tahapan ini memberikan tanggapan dari penulis mengenai testing yang telah dilakukan.

Tahap Rekomendasi (*Recommendation*) Berdasarkan hasil analisis data, rekomendasi dapat

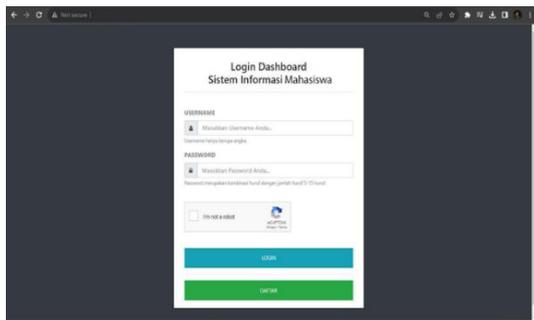
disusun untuk meningkatkan kualitas kinerja website akademik Universitas XYZ. Rekomendasi tersebut dapat berupa perbaikan infrastruktur, optimasi kode, peningkatan kapasitas server, atau penggunaan teknologi lainnya untuk meningkatkan respons dan keandalan *website*.

3. HASIL

Mendasari pada langkah metode penelitian, Sebelum melakukan penelitian mendalam, peneliti terlebih dahulu melakukan Pengumpulan Data (*Collection Data*) berupa survei, wawancara dan menyusun kuesioner yang terdiri dari beberapa pertanyaan, peneliti yang telah mendapatkan jawaban dari pihak-pihak terkait yang mengelola *website*, kemudian peneliti merangkum jawaban tersebut. Hasil dari pertanyaan yang telah dirangkum dapat dilihat pada Tabel 1.

Tabel 1. Rangkuman hasil Pengumpulan Data

No	Pertanyaan	Jawaban
1	Bagaimana Anda mengelola website tersebut?	Kami mengelola website melalui tim yang terdiri dari pengembang, desainer, dan manajer konten pada Universitas XYZ.
2	Apakah ada kendala dalam pengelolaan?	Ya, kami mengalami kendala terutama dalam pemeliharaan server dan jumlah user dalam mengakses <i>website</i> .
3	Seberapa efektif strategi yang digunakan?	Strategi kami dinilai cukup efektif dimana kami membagi perangkatan untuk dapat mengambil matakuliah setiap waktunya, akan tetapi terdapat beberapa permasalahan ketika user disetiap angkatan melebihi kapasitas dari server yang kami sediakan
4	Bagaimana tingkat kepuasan pengguna?	Mayoritas mahasiswa merasa sedikit kurang puas dikarenakan terjadi beberapa permasalahan ketika mengakses website ketika dalam kondisi <i>traffic</i> yang sedang ramai
5	Apakah Anda melakukan pembaruan secara rutin?	Ya, dalam perencanaan kedepan kami akan membangun dan menambahkan jumlah kapasitas server untuk seluruh program kami terutama dalam website akademik.
6	Apakah sebagai mahasiswa anda mempunyai masalah ketika mengambil matakuliah dalam <i>website</i> akademik XYZ ?	Ya, saya memiliki beberapa masalah ketika mengambil matakuliah, terutama website yang sering bad respon bahkan hingga kami dikeluarkan secara otomatis dari <i>website</i> akademik
7	Apakah sebagai mahasiswa persoalan ini sangat mempengaruhi proses kuliah anda ?	Ya, sangat mempengaruhi, terkadang ada beberapa dosen favorite yang telah kami pilih akan tetapi karena kualitas website yang lambat menanggapi permintaan sehingga kami tidak mendapatkan apa yang kami inginkan.

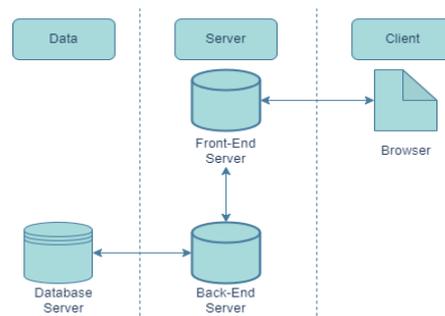


Gambar 2. Objek *website* yang akan diuji

Sebagai bahan yang telah menjadi objek penelitian, berupa sebuah *website*, peneliti memberikan gambaran mengenai antarmuka (*interface*) *website* tersebut. Gambaran ini, yang akan ditampilkan pada Gambar 2, mencakup elemen-elemen visualstruktur navigasi. Dengan menganalisis antarmuka *website*, peneliti bertujuan untuk mendapatkan pemahaman yang lebih mendalam tentang bagaimana pengguna berinteraksi dengan platform tersebut dan bagaimana elemen-elemen

tersebut mendukung tujuan dan fungsi keseluruhan dari *website* yang diteliti. Hasil temuan ini akan menjadi kontribusi penting dalam pemahaman aspek antarmuka dalam konteks penelitian ini.

Selain memperoleh jawaban dan informasi mengenai objek *website*, peneliti juga memperoleh gambaran tentang infrastruktur sistem *website* yang menggambarkan bagaimana *website* ini beroperasi dan berfungsi. Hasil temuan gambaran ini ditampilkan pada Gambar 3.



Gambar 3. Arsitektur sistem *website* akademik XYZ

Peneliti juga mendapatkan data jenis dari pengguna dan kebutuhan kinerja website yang diharapkan dari universitas XYZ, hal ini kan menjadi perbandingan hasil pengujian dengan parameter kinerja yang ditetapkan, hasil dari data tersebut akan dirangkum dalam Tabel 2 dan Tabel 3.

Tabel 2. Karakteristik Pengguna

Jenis Pengguna	Jumlah Pengguna
Mahasiswa	> 3000 Pengguna
Dosen	> 100 Pengguna
Staff	> 30 Pengguna
Pengguna tak diketahui	> 50 Pengguna

Tabel 3. Parameter kinerja *website* yang diharapkan

Keadaan Traffic	Total Pengguna	Kinerja
Low Traffic	< 100 Pengguna	Optimal
Normal Traffic	< 300 Pengguna	Optimal
High Traffic	> 500 Pengguna	Cukup Optimal

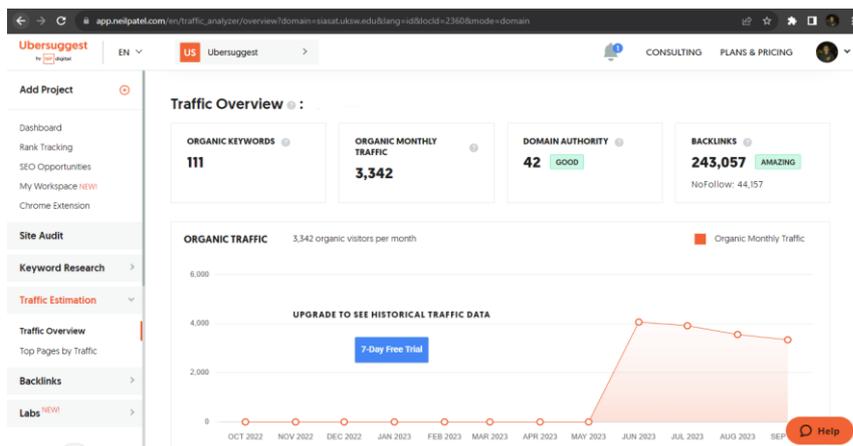
Berdasarkan data yang telah dikumpulkan melalui survey, wawancara dan kuesioner sebagai data untuk memulai pengujian, kemudian penulis melakukan langkah berikutnya yaitu Pengujian Kinerja (*Performance Testing*).

Dalam Pengujian Kinerja peneliti akan menentukan *Testing Planning* dan Skenario Pengujian, Menurut hasil penelitian dari *WebsiteBuilderExpert*, waktu pemuatan (*load time*) yang optimal untuk *website non-ecommerce* adalah 3 detik [19]. *Cami Bird* juga menyebutkan bahwa waktu pemuatan ideal untuk sebuah website yang juga mempengaruhi peringkat di mesin pencari adalah kurang dari 3 detik [20]. Oleh karena itu, dalam penelitian ini, target waktu pemuatan untuk website akademik adalah kurang dari 3 detik.

Rencana pengujian (*test plan*) dan sasaran pengujian adalah sebagai berikut: Waktu pemuatan tidak boleh melebihi 3 detik, Tidak boleh terjadi respon yang gagal (*error*) dari permintaan (*request*) yang dikirimkan selama pengujian berlangsung, dan Data pengujian harus berhasil masuk ke dalam fungsi yang dimaksud.

Pengujian akan dilakukan secara bertahap dan bervariasi, menggunakan teknik peningkatan jumlah pengguna secara bertahap yang dikenal sebagai teknik "*step load*". Skala yang akan digunakan adalah meningkatkan jumlah pengguna sebanyak dua kali lipat per skenario. Hal ini akan efektif untuk mengevaluasi kinerja sistem dalam berbagai kondisi pengguna yang beragam dan bertahap.

Menurut *Logicfy*, ketika melakukan pengujian *load testing*, disarankan untuk menentukan jumlah *thread* yang akan digunakan berdasarkan data asli pengunjung atau pengguna yang mengakses situs *web* [21]. Oleh karena itu, data pengunjung dari aplikasi yang ingin diuji sebagaimana terlihat dalam Gambar 3 menggunakan web <https://app.neilpatel.com/>, dapat dijadikan referensi. Dari Gambar 3 tersebut, terdapat total rata-rata 388 *traffic* per bulannya. Terdapat pula lonjakan jumlah pengunjung pada bulan Juni 2023 dikarenakan jumlah mahasiswa yang harus dan mendaftar Tugas Akhir serta bimbingan melalui aplikasi tersebut dikarenakan *testing* ini akan menggunakan data *traffic* tertinggi, maka berdasarkan hasil penulis menggunakan data pada bulan Juni 2023 yaitu 4063 data *traffic* dan Juli 2023 yaitu 3914 data *traffic*.

Gambar 4. Pencarian *traffic* pada web <https://app.neilpatel.com/>

Berdasarkan dari Gambar 4 yang merupakan jumlah *traffic*, diperlukan jumlah *traffic* perhari, maka data *traffic* perharinya sebagai berikut:

$$\text{Traffic Perhari} = \frac{\text{Juni 2023} - \text{Juli 2023}}{\text{Jumlah bulan} * \text{Jumlah hari}} \quad (2)$$

$$\text{Traffic Perhari} = \frac{7.977}{2 * 30} = 132.95 \quad (3)$$

Persamaan (2) merupakan persamaan untuk menentukan jumlah *traffic perhari* yang nantinya akan digunakan sebagai acuan untuk menentukan skenario pengujian.

Pada persamaan (2), *Traffic Perhari* merupakan variabel untuk menampung hasil, *Juni 2023 - Juli 2023* merupakan bulan yang diambil, bulan Juni 2023 dan Juli 2023 digunakan karena merupakan bulan dengan banyaknya *traffic* pada *website* yang diuji, *Jumlah bulan* merupakan

jumlah bulan yang digunakan yaitu terdapat dua bulan yang digunakan, *Jumlah hari* merupakan total hari dalam satu bulan.

Dari persamaan didapatkan jumlah *traffic* untuk perharinya yaitu $132.95 \dots = \sim 133$, sehingga pada skenario pertama didapatkan jumlah *user* atau *number of thread* adalah 133 *traffic*. Berdasarkan perasamaan diatas penulis membuat skenario pengujian terdapat tiga skenario yang digunakan berdasarkan jumlah *traffic* yang dikalikan dua skala sebelumnya, sehingga skenario yang dihasilkan didapatkan pada Tabel 4.

Tabel 4. Skenario Pengujian

Skenario	Number Of Treads (user)	Ramp-up periode (second)	Loop Count
Skenario 1	133	1	1
Skenario 2	266	1	1
Skenario 3	532	1	1

Dalam Tabel 4, terdapat tiga skenario yang menggambarkan berbagai situasi pengujian beban (*load testing*). Skenario pertama, yaitu skenario 1, mencoba mensimulasikan kinerja *website* saat berada dalam kondisi yang telah ditetapkan dalam perjanjian tingkat layanan (*Baseline Testing*). Hal ini didasarkan pada data jumlah pengguna per hari dari bulan Juni hingga Juli. Skenario kedua, atau skenario 2, berfokus pada simulasi kinerja *website* saat digunakan secara intens (*Peak Testing*). Pengujian ini didasarkan pada data pengguna per hari dari bulan Juni hingga Juli yang dikalikan dua sesuai dengan skala yang ditetapkan, yang mencerminkan situasi ketika layanan sedang sangat diminati. Skenario ketiga, yaitu skenario 3, mencoba mensimulasikan kinerja *website* dalam kondisi paling ekstrem, di mana layanan digunakan secara maksimal (*Stress Testing*). Dalam skenario ini, jumlah pengguna digandakan menjadi dua kali lipat dari hasil *Peak Testing* [22].

Selanjutnya dilakukan *Thread Group*, *Thread Group* yang digunakan pada testing terbagi menjadi dua yaitu konfigurasi *GET login* dan *POST login*, kedua *Thread Login* ini berguna untuk mengetahui performance dari aplikasi tersebut [23]. Mendasari pada Tabel 4 Skenario pengujian *Thread Group* akan dibagi menjadi 3 yaitu *Thread Group 1* dengan total 133 *Traffic*, *Thread Group 2* dengan total 266 *Traffic* dan *Thread Group 4* dengan total 532 *Traffic*. Setelah melakukan konfigurasi *Thread Group* selanjutnya adalah membuat konfigurasi *HTTP Request* untuk fungsi *login* dan juga akan disiapkan fungsi *Listener* untuk menampilkan hasil dari pengujian selain itu digunakan juga fungsi *reporting* dari *library Apache JMeter* untuk menampilkan hasil analisis menggunakan *web browser*. Dalam konfigurasi *Sampler HTTP Request*, metode permintaan yang digunakan adalah GET dan POST, alamat server atau alamat IP yang harus diisi adalah tautan ke web dari aplikasi yang ingin ditest dan alamat *path* yang digunakan adalah salah satu fitur yang akan ditest, pemilihan *listener* yang digunakan yaitu *View Result Tree*, *Summary Report* dan *View Result in Table*.

Sebagai contoh dari hasil konfigurasi *Thread Group* dan *Listener*.

Setelah semua fungsi telah dikonfigurasi untuk pengujian, tambahan beberapa *listener* pada *thread group* dan juga perintah untuk membuat *Dashboard Result* agar hasil dapat dilihat pada *dashboard* melalui *web browser*. Ini mencakup *View Result Tree* yang memperlihatkan detail hasil pengujian, *View Result in Table* untuk menampilkan hasil pengujian dalam format tabel, dan *Summary Report* untuk menyajikan ringkasan dari hasil pengujian. Hasil dari pengujian ini dalam skenario pertama terdokumentasi dalam tahap selanjutnya.

Selanjutnya akan dilakukan *testing* dengan konfigurasi yang telah diset, hasil setiap *thread group* GET dan POST.

Sample #	Start Time	Thread Name	Label	Sample Success	Status	Bytes	Sent Bytes	Latency	Connect Time
1	08:50:47.075	Performance T...	GET HTTP Req...	495	Success	2632	232	87	18
2	08:50:46.981	Performance T...	GET HTTP Req...	495	Success	2632	232	193	87
3	08:50:47.007	Performance T...	GET HTTP Req...	533	Success	2632	232	74	13
4	08:50:46.997	Performance T...	GET HTTP Req...	638	Success	2632	232	137	73
5	08:50:47.013	Performance T...	GET HTTP Req...	621	Success	2632	232	119	53
6	08:50:46.981	Performance T...	GET HTTP Req...	675	Success	2632	232	129	87
7	08:50:47.038	Performance T...	GET HTTP Req...	661	Success	2632	232	78	18
8	08:50:47.038	Performance T...	GET HTTP Req...	634	Success	2632	232	98	32
9	08:50:47.116	Performance T...	GET HTTP Req...	546	Success	2632	232	88	18
10	08:50:47.158	Performance T...	GET HTTP Req...	533	Success	2632	232	64	6
11	08:50:47.137	Performance T...	GET HTTP Req...	532	Success	2632	232	80	27
12	08:50:47.238	Performance T...	GET HTTP Req...	532	Success	2632	232	86	19
13	08:50:47.228	Performance T...	GET HTTP Req...	333	Success	2632	232	45	16
14	08:50:47.316	Performance T...	GET HTTP Req...	465	Success	2632	232	44	4
15	08:50:47.296	Performance T...	GET HTTP Req...	487	Success	2632	232	47	6
16	08:50:47.276	Performance T...	GET HTTP Req...	547	Success	2632	232	45	6
17	08:50:47.338	Performance T...	GET HTTP Req...	591	Success	2632	232	49	3
18	08:50:47.356	Performance T...	GET HTTP Req...	482	Success	2632	232	51	9
19	08:50:47.396	Performance T...	GET HTTP Req...	470	Success	2632	232	235	19
20	08:50:47.378	Performance T...	GET HTTP Req...	331	Success	2632	232	249	24
21	08:50:47.433	Performance T...	GET HTTP Req...	472	Success	2632	232	214	18
22	08:50:47.418	Performance T...	GET HTTP Req...	509	Success	2632	232	217	8

Gambar 5. Result in Table Thread Group 1 GET Login (133 Traffic)

Sample #	Start Time	Thread Name	Label	Sample Success	Status	Bytes	Sent Bytes	Latency	Connect Time
1	08:53:02.442	Performance T...	POST HTTP Req...	305	Success	8125	491	110	67
2	08:53:02.463	Performance T...	POST HTTP Req...	318	Success	8125	491	82	40
3	08:53:02.534	Performance T...	POST HTTP Req...	238	Success	8125	491	83	6
4	08:53:02.462	Performance T...	POST HTTP Req...	388	Success	8125	491	82	11
5	08:53:02.504	Performance T...	POST HTTP Req...	388	Success	8125	491	83	6
6	08:53:02.555	Performance T...	POST HTTP Req...	381	Success	8125	491	65	4
7	08:53:02.564	Performance T...	POST HTTP Req...	388	Success	8125	491	67	5
8	08:53:02.603	Performance T...	POST HTTP Req...	279	Success	8125	491	83	5
9	08:53:02.616	Performance T...	POST HTTP Req...	313	Success	8125	491	83	5
10	08:53:02.684	Performance T...	POST HTTP Req...	279	Success	8125	491	44	5
11	08:53:02.643	Performance T...	POST HTTP Req...	389	Success	8125	491	65	10
12	08:53:02.664	Performance T...	POST HTTP Req...	272	Success	8125	491	53	5
13	08:53:02.703	Performance T...	POST HTTP Req...	333	Success	8125	491	57	5
14	08:53:02.623	Performance T...	POST HTTP Req...	314	Success	8125	491	64	4
15	08:53:02.724	Performance T...	POST HTTP Req...	388	Success	8125	491	57	4
16	08:53:02.742	Performance T...	POST HTTP Req...	389	Success	8125	491	61	12
17	08:53:02.763	Performance T...	POST HTTP Req...	388	Success	8125	491	53	5
18	08:53:02.823	Performance T...	POST HTTP Req...	348	Success	8125	491	54	4
19	08:53:02.784	Performance T...	POST HTTP Req...	388	Success	8125	491	52	4
20	08:53:02.843	Performance T...	POST HTTP Req...	389	Success	8125	491	49	20
21	08:53:02.882	Performance T...	POST HTTP Req...	359	Success	8125	491	64	14
22	08:53:02.883	Performance T...	POST HTTP Req...	279	Success	8125	491	63	4

Gambar 6. Result in Table Thread Group 1 POST Login (133 Traffic)

Berdasarkan pada Gambar 5 dan Gambar 6 mengenai *Result in Table* menunjukkan bahwa *request* GET dan POST berjalan normal dengan status sukses serta tidak ada *error* yang ditunjukkan pada saat proses dan *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 133 user, dengan 1 request baru setiap 1 detik menghasilkan rata-rata 855 *milisecond* dengan nilai *error* 0%, serta nilai *Throughput* 11.5/sec data ini menunjukkan bahwa *request* GET pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* yang rendah, mengenai

Result in Table untuk *request* POST berjalan normal dengan status sukses serta tidak ada *error* yang ditunjukkan pada saat proses dan mengenai *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 133 user, dengan 1 request baru setiap 1 detik menghasilkan rata-rata 273 *milisecond* dengan nilai *error* 0%, serta nilai

Throughput 38.6/sec data ini menunjukkan bahwa *request* POST pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* yang rendah.

Sample #	Start Time	Thread Name	Label	Sample Timings	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	08:54:57.24	Performance Te...	GET HTTP Req...	264	🟢	7721	286	79	28
2	08:54:57.63	Performance Te...	GET HTTP Req...	445	🟢	7721	286	166	113
3	08:54:57.724	Performance Te...	GET HTTP Req...	375	🟢	7721	286	93	42
4	08:54:57.682	Performance Te...	GET HTTP Req...	408	🟢	7721	286	181	101
5	08:54:57.743	Performance Te...	GET HTTP Req...	388	🟢	7721	286	79	19
6	08:54:57.773	Performance Te...	GET HTTP Req...	447	🟢	7721	286	133	52
7	08:54:57.672	Performance Te...	GET HTTP Req...	447	🟢	7721	286	144	54
8	08:54:57.628	Performance Te...	GET HTTP Req...	499	🟢	7721	286	182	142
9	08:54:57.725	Performance Te...	GET HTTP Req...	372	🟢	7721	286	67	8
10	08:54:57.613	Performance Te...	GET HTTP Req...	523	🟢	7721	286	216	152
11	08:54:57.633	Performance Te...	GET HTTP Req...	357	🟢	7721	286	196	132
12	08:54:57.698	Performance Te...	GET HTTP Req...	429	🟢	7721	286	53	8
13	08:54:57.685	Performance Te...	GET HTTP Req...	429	🟢	7721	286	143	81
14	08:54:57.764	Performance Te...	GET HTTP Req...	381	🟢	7721	286	67	10
15	08:54:57.692	Performance Te...	GET HTTP Req...	463	🟢	7721	286	136	74
16	08:54:57.725	Performance Te...	GET HTTP Req...	383	🟢	7721	286	57	6
17	08:54:57.793	Performance Te...	GET HTTP Req...	386	🟢	7721	286	57	7
18	08:54:57.679	Performance Te...	GET HTTP Req...	347	🟢	7721	286	89	8
19	08:54:57.623	Performance Te...	GET HTTP Req...	340	🟢	7721	286	133	7
20	08:54:57.643	Performance Te...	GET HTTP Req...	335	🟢	7721	286	122	122
21	08:54:57.636	Performance Te...	GET HTTP Req...	350	🟢	7721	286	142	7
22	08:54:57.644	Performance Te...	GET HTTP Req...	348	🟢	7721	286	117	8

Gambar 7. Result in Table Thread Group 2 GET Login (226 Traffic)

Sample #	Start Time	Thread Name	Label	Sample Timings	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	08:55:38.471	Performance Te...	POST HTTP Req...	321	🟢	7721	404	133	122
2	08:55:38.440	Performance Te...	POST HTTP Req...	294	🟢	7721	404	99	49
3	08:55:38.523	Performance Te...	POST HTTP Req...	355	🟢	7721	404	134	68
4	08:55:38.426	Performance Te...	POST HTTP Req...	337	🟢	7721	404	188	137
5	08:55:38.533	Performance Te...	POST HTTP Req...	387	🟢	7721	404	111	60
6	08:55:38.543	Performance Te...	POST HTTP Req...	382	🟢	7721	404	81	30
7	08:55:38.501	Performance Te...	POST HTTP Req...	323	🟢	7721	404	133	91
8	08:55:38.617	Performance Te...	POST HTTP Req...	315	🟢	7721	404	67	9
9	08:55:38.481	Performance Te...	POST HTTP Req...	348	🟢	7721	404	123	113
10	08:55:38.548	Performance Te...	POST HTTP Req...	290	🟢	7721	404	78	28
11	08:55:38.593	Performance Te...	POST HTTP Req...	249	🟢	7721	404	73	24
12	08:55:38.624	Performance Te...	POST HTTP Req...	242	🟢	7721	404	62	13
13	08:55:38.574	Performance Te...	POST HTTP Req...	274	🟢	7721	404	72	19
14	08:55:38.604	Performance Te...	POST HTTP Req...	373	🟢	7721	404	189	64
15	08:55:38.512	Performance Te...	POST HTTP Req...	248	🟢	7721	404	142	89
16	08:55:38.488	Performance Te...	POST HTTP Req...	388	🟢	7721	404	184	132
17	08:55:38.633	Performance Te...	POST HTTP Req...	226	🟢	7721	404	54	4
18	08:55:38.623	Performance Te...	POST HTTP Req...	244	🟢	7721	404	49	5
19	08:55:38.627	Performance Te...	POST HTTP Req...	244	🟢	7721	404	52	6
20	08:55:38.604	Performance Te...	POST HTTP Req...	235	🟢	7721	404	48	6
21	08:55:38.646	Performance Te...	POST HTTP Req...	257	🟢	7721	404	59	3
22	08:55:38.693	Performance Te...	POST HTTP Req...	222	🟢	7721	404	63	6

Gambar 8. Result in Table Thread Group 2 POST Login (226 Traffic)

Berdasarkan pada Gambar 7 dan Gambar 8 mengenai *Result in Table* menunjukkan bahwa *request* GET dan POST berjalan normal dengan status sukses serta tidak ada *error* yang ditunjukkan pada saat proses, mengenai *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 226 user, dengan 1 *request* baru setiap 1 detik menghasilkan rata-rata 1256 *milisecond* dengan nilai *error* 0%, serta nilai *Throughput* 26.2/sec data ini menunjukkan bahwa *request* GET pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* cukup sibuk, mengenai *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 226 user, dengan 1 *request* baru setiap 1 detik menghasilkan rata-rata 315 *milisecond* dengan nilai *error* 0%, serta nilai *Throughput* 79.1/sec data ini menunjukkan bahwa *request* POST pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* cukup sibuk.

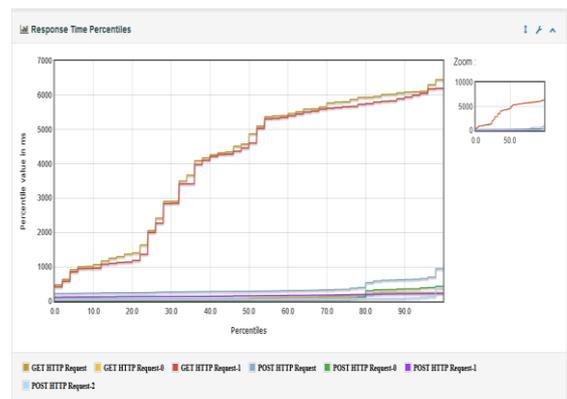
Sample #	Start Time	Thread Name	Label	Sample Timings	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	08:12:14.007	Performance Te...	GET HTTP Req...	287	🟢	8123	377	183	47
2	08:12:14.003	Performance Te...	GET HTTP Req...	332	🟢	8123	377	137	88
3	08:12:14.002	Performance Te...	GET HTTP Req...	256	🟢	8123	377	63	24
4	08:12:14.002	Performance Te...	GET HTTP Req...	270	🟢	8123	377	74	33
5	08:12:14.005	Performance Te...	GET HTTP Req...	281	🟢	8123	377	74	33
6	08:12:14.008	Performance Te...	GET HTTP Req...	337	🟢	8123	377	132	63
7	08:12:14.001	Performance Te...	GET HTTP Req...	295	🟢	8123	377	82	33
8	08:12:14.003	Performance Te...	GET HTTP Req...	323	🟢	8123	377	139	69
9	08:12:14.003	Performance Te...	GET HTTP Req...	341	🟢	8123	377	127	72
10	08:12:14.112	Performance Te...	GET HTTP Req...	277	🟢	8123	377	67	9
11	08:12:14.006	Performance Te...	GET HTTP Req...	323	🟢	8123	377	73	17
12	08:12:14.114	Performance Te...	GET HTTP Req...	315	🟢	8123	377	87	36
13	08:12:14.110	Performance Te...	GET HTTP Req...	248	🟢	8123	377	68	19
14	08:12:14.009	Performance Te...	GET HTTP Req...	243	🟢	8123	377	55	23
15	08:12:14.103	Performance Te...	GET HTTP Req...	253	🟢	8123	377	52	5
16	08:12:14.147	Performance Te...	GET HTTP Req...	292	🟢	8123	377	74	12
17	08:12:14.150	Performance Te...	GET HTTP Req...	279	🟢	8123	377	54	5
18	08:12:14.170	Performance Te...	GET HTTP Req...	373	🟢	8123	377	97	4
19	08:12:14.140	Performance Te...	GET HTTP Req...	279	🟢	8123	377	67	14
20	08:12:14.120	Performance Te...	GET HTTP Req...	340	🟢	8123	377	83	32
21	08:12:14.126	Performance Te...	GET HTTP Req...	334	🟢	8123	377	76	26
22	08:12:14.100	Performance Te...	GET HTTP Req...	304	🟢	8123	377	62	8

Gambar 9. Result in Table Thread Group 3 GET Login (532 Traffic)

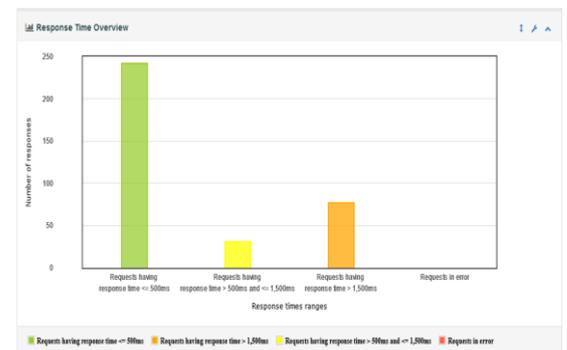
Sample #	Start Time	Thread Name	Label	Sample Timings	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	08:12:38.485	Performance Te...	POST HTTP Req...	221	🟢	7721	404	96	5
2	08:12:38.477	Performance Te...	POST HTTP Req...	281	🟢	7721	404	57	6
3	08:12:38.504	Performance Te...	POST HTTP Req...	348	🟢	7721	404	55	5
4	08:12:38.510	Performance Te...	POST HTTP Req...	230	🟢	7721	404	61	5
5	08:12:38.480	Performance Te...	POST HTTP Req...	273	🟢	7721	404	45	5
6	08:12:38.530	Performance Te...	POST HTTP Req...	246	🟢	7721	404	53	5
7	08:12:38.517	Performance Te...	POST HTTP Req...	280	🟢	7721	404	61	5
8	08:12:38.504	Performance Te...	POST HTTP Req...	223	🟢	7721	404	56	6
9	08:12:38.526	Performance Te...	POST HTTP Req...	288	🟢	7721	404	61	6
10	08:12:38.594	Performance Te...	POST HTTP Req...	232	🟢	7721	404	61	6
11	08:12:38.533	Performance Te...	POST HTTP Req...	274	🟢	7721	404	69	5
12	08:12:38.580	Performance Te...	POST HTTP Req...	274	🟢	7721	404	69	4
13	08:12:38.538	Performance Te...	POST HTTP Req...	311	🟢	7721	404	51	4
14	08:12:38.619	Performance Te...	POST HTTP Req...	245	🟢	7721	404	57	5
15	08:12:38.573	Performance Te...	POST HTTP Req...	283	🟢	7721	404	47	5
16	08:12:38.588	Performance Te...	POST HTTP Req...	301	🟢	7721	404	54	4
17	08:12:38.623	Performance Te...	POST HTTP Req...	301	🟢	7721	404	58	7
18	08:12:38.613	Performance Te...	POST HTTP Req...	359	🟢	7721	404	68	4
19	08:12:38.601	Performance Te...	POST HTTP Req...	362	🟢	7721	404	62	4
20	08:12:38.677	Performance Te...	POST HTTP Req...	362	🟢	7721	404	60	4
21	08:12:38.644	Performance Te...	POST HTTP Req...	349	🟢	7721	404	65	6
22	08:12:38.590	Performance Te...	POST HTTP Req...	406	🟢	7721	404	69	6

Gambar 10. Result in Table Thread Group 3 POST Login (532 Traffic)

Berdasarkan pada Gambar 9 dan Gambar 10 mengenai *Result in Table* menunjukkan bahwa *request* GET dan POST berjalan normal dengan status sukses serta tidak ada *error* yang ditunjukkan pada saat proses, mengenai *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 532 user, dengan 1 *request* baru setiap 1 detik menghasilkan rata-rata 2564 *milisecond* dengan nilai *error* 0%, serta nilai *Throughput* 73.3/sec data ini menunjukkan bahwa *request* GET pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* sangat sibuk, Terkait dengan *Summary Report* dari data yang didapatkan dapat dianalisis dengan menggunakan 532 user, dengan 1 *request* baru setiap 1 detik menghasilkan rata-rata 478 *milisecond* dengan nilai *error* 0%, serta nilai *Throughput* 87.9/sec data ini menunjukkan bahwa *request* POST pada situasi ini dapat bekerja dengan optimal pada kondisi *traffic* sangat sibuk.



Gambar 11. Hasil Response Time Percentiles Thread All Group



Gambar 12. Hasil Response Time Overtime Thread All Group

Berdasarkan pada Gambar 11 *Response Time Percentiles Thread All Group* (1-3) menunjukkan bahwa graph response time mengalami peningkatan berdasarkan jumlah user atau *traffic* yang melakukan request GET atau POST.

Berdasarkan pada Gambar 12 *Response Time Overview Thread All Group* (1-3) menunjukkan bahwa graph response time overview menampilkan hasil yang cukup baik terdapat *trend* cukup tinggi untuk request untuk waktu kurang lebih dari 500ms, terdapat juga request yang lebih dari 1,500ms.

4. DISKUSI

Berdasarkan hasil dari uji testing diatas selanjutnya penulis akan melakukan Tahapan Analisis Data (*Data Analysis*), berdasarkan dengan Tabel 3 mengenai harapan kinerja *website* sesuai dengan *traffic* yang ada berdasarkan pengujian yang dilakukan menggunakan *Apache Jmeter* dengan melihat skenario pengujian pada Tabel 4. maka hasil analisis yang dibandingkan dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan hasil pengujian dengan parameter kinerja yang ditetapkan

Skenario	Total Pengguna	Kondisi Traffic	Kinerja Website
Skenario 1	133	Low	Optimal
Skenario 2	266	Normal	Optimal
Skenario 3	532	High	Cukup Optimal

Dalam proses pengujian peneliti juga melakukan identifikasi masalah kinerja yang terjadi pada *website*, identifikasi ini bertujuan agar melihat kinerja server selama proses pengujian, hasil ini dapat dilihat pada Tabel 6.

Tabel 6. Identifikasi kinerja website

Skenario	Thread Group	Rata-Rata Request user (ms)	Rata-Rata Throughput (/sec)	Kinerja Website
Skenario 1 (133 user)	GET	855 ms	11.5/sec	Optimal
	POST	273 ms	38.6/sec	Optimal
Skenario 2 (266 user)	GET	1256 ms	26.2/sec	Optimal
	POST	315 ms	79.1/sec	Cukup Optimal
Skenario 3 (532 user)	GET	2564 ms	73.3/sec	Cukup Optimal
	POST	478 ms	87.9/sec	Optimal

Berdasarkan hasil pengujian dan analisis sebelumnya, untuk mendapatkan hasil yang lebih komprehensif, peneliti juga melakukan perbandingan dengan pengujian serupa yang menggunakan metode yang sama. Peneliti menemukan sebuah jurnal dengan judul "Pengujian Aplikasi *Greenwallet* Dengan Metode *Load Testing* Dan *Apache Jmeter*" oleh Cahaya Putri Agustika, Wahyu SJ, dan Mohammad Idhom [24], dimana peneliti menemukan beberapa hal setelah membaca jurnal tersebut dan membandingkannya dengan pengujian yang

dilakukan penulis, yaitu: 1) Pada penelitian ini, metodologi penelitian yang digunakan tergolong cukup lengkap dan jelas, mengadopsi metode kuantitatif dan kualitatif serta mengikuti tahapan metode penelitian yang sistematis. Hal ini mempermudah para pembaca untuk memahami alur dari proses pengujian pada penelitian ini, 2) Tidak ditemukan data jumlah *traffic* perhari yang menjadi dasar sebagai pembuatan skenario pengujian pada jurnal "Pengujian Aplikasi *Greenwallet* Dengan Metode *Load Testing* Dan *Apache Jmeter*", jurnal penelitian ini memuat hal tersebut sehingga menjadi dasar yang kuat dalam perhitungan *traffic* dalam membuat skenario pengujian, 3) Penelitian ini menggunakan 2 metode yaitu POST dan GET dengan menghitung seluruh rata-rata request dan rata-rata *Throughput* yang terjadi dengan cara yang lebih simpel, 4) Pengujian hanya dilakukan pada 10 user dalam 2 skenario pengujian pada jurnal Pengujian Aplikasi *Greenwallet* Dengan Metode *Load Testing* Dan *Apache Jmeter*", pada jurnal ini menggunakan 3 skenario dengan perbedaan jumlah user yang digunakan, yaitu Skenario 1 dengan 133 user, Skenario 2 dengan 266 user dan skenario 3 dengan 532 user, berdasarkan hal ini menurut peneliti pengujian dapat dilakukan dengan akurat, 5) Hasil yang diberikan menurut peneliti dalam penelitian "Pengujian Aplikasi *Greenwallet* Dengan Metode *Load Testing* Dan *Apache Jmeter*" hanya menampilkan hasil goals yang ingin dicapai tanpa memberikan kejelasan apa saja yang terjadi termasuk kinerja dari *website* ketika proses pengujian, pada penelitian ini penulis memberikan secara jelas bagaimana kinerja *website* berjalan apakah optimal, cukup optimal atau tidak optimal.

Secara tertulis penulis mempunyai pendapat terkait penelitian ini sangat jelas dan tepat, dengan tahapan metode penelitian yang dibuat dengan menggunakan metode kualitatif dan kuantitatif yang didukung oleh metode *Load Testing* dan *Apache Jmeter*, hasil dari penelitian ini mencerminkan dengan judul dan latar belakang masalah yang sedang dihadapi oleh Universitas XYZ.

5. KESIMPULAN

Hasil dari penelitian *load testing* pada *website* mahasiswa universitas XYZ, khususnya pada fungsionalitas inti, menunjukkan bahwa *website* beroperasi secara optimal dalam berbagai skenario pengujian Pada skenario pertama, yang mewakili situasi normal atau *baseline testing*, *website* berhasil memenuhi target pengujian. Waktu rata-rata untuk permintaan GET adalah sekitar 0.855 detik, yang jauh di bawah batas waktu maksimal yang ditetapkan sebesar 3 detik. Demikian juga, permintaan POST memerlukan waktu rata-rata sekitar 0.273 detik dan tidak ada respon yang mengalami kegagalan. Pada skenario kedua, yang mencerminkan kondisi lalu lintas tinggi atau *peak testing*, *website* juga berkinerja optimal. Waktu rata-rata untuk permintaan GET

adalah sekitar 1.256 detik, dan waktu rata-rata untuk permintaan POST adalah sekitar 0.315 detik, semuanya tetap di bawah batas waktu 3 detik yang ditetapkan, dan tidak ada kegagalan respon. Skenario ketiga, yang mewakili situasi stres atau *stress testing*, menunjukkan bahwa *website* masih beroperasi dengan optimal. Waktu rata-rata untuk permintaan GET adalah sekitar 2.564 detik, yang meskipun lebih lama dari skenario sebelumnya, masih berada di bawah batas waktu 3 detik. Tidak ada kesalahan atau kegagalan respon yang terdeteksi. Kesimpulannya, hasil pengujian menunjukkan bahwa *website* berfungsi dengan baik dalam berbagai kondisi penggunaan layanan, termasuk kondisi berat. Rata-rata total waktu respons untuk permintaan GET dari ketiga skenario adalah sekitar 1.558 detik, sementara rata-rata total waktu respons untuk permintaan POST adalah sekitar 0.355 detik. Perlu diperhatikan bahwa jumlah *thread* atau pengguna dapat mempengaruhi waktu respons permintaan, dan hal ini juga dipengaruhi oleh jumlah mahasiswa yang menggunakan *website* universitas. Faktor-faktor ini memiliki dampak signifikan pada kinerja *website*.

UCAPAN TERIMA KASIH

Terima kasih kepada seluruh pihak yang telah membantu dalam penulisan ini serta kepada Tuhan Yesus yang selalu memberi kekuatan dan kemurahan.

DAFTAR PUSTAKA

- [1] R. S. Samosir and E. Shahnaz Aprilia, "Teknologi Web sebagai Media Pelestarian Budaya Batak Toba Secara Khusus Marga Samosir Se-JABODETABEK," *KALBISCIENTIA J. Sains dan Teknol.*, vol. 10, no. 1, pp. 90–97, 2023, doi: 10.53008/kalbiscientia.v10i1.2103.
- [2] P. Website *et al.*, "Jurnal Pengabdian Masyarakat Bidang Sains dan Teknologi Framework Bootstrap," *Media Cetak*, vol. 2, no. 1, pp. 136–144, 2023, doi: 10.55123/abdikan.v2i1.1721.
- [3] R. Rizal and A. Rahmatulloh, "MIND (Multimedia Artificial Intelligent Networking Database Pengukuran Kinerja Sistem Informasi Penerimaan Mahasiswa Baru Menggunakan GTMetric, WebAIM dan LoadView," *J. MIND J. / ISSN*, vol. 8, no. 1, pp. 107–118, 2023, [Online]. Available: <https://doi.org/10.26760/mindjournal.v8i1.107-118>
- [4] S. H. H. Ilyas, St. Hajrah Mansyur, and Harlinda, "Analysis of Learning Web Application Testing With Weighting Product and Ordered Weighting Averaging on Group Decision Maker," *J. Tek. Inform.*, vol. 3, no. 6, pp. 1513–1522, 2022, doi: 10.20884/1.jutif.2022.3.6.333.
- [5] W. Hadinata, L. Stianingsih, and K. Tangerang, "ANALISIS PERBANDINGAN PERFORMA RESTFUL API ANTARA EXPRESS . JS DENGAN LARAVEL FRAMEWORK," vol. 12, no. 1, pp. 531–540, 2024.
- [6] D. I. Permatasari, "Penguujian 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.
- [7] P. Diantono, A. Susanto, A. R. Supriyono, and D. N. Prasetyanti, "Perbandingan Kinerja Antara Gatling dan Apache JMeter pada Uji Beban RESTful API," vol. 15, no. 01, pp. 211–215, 2024, doi: 10.35970/infotekmesin.v15i1.2176.
- [8] M. K. Abdillah and A. R. Perdanakusuma, "Analisis Kualitas Website XYZ . com menggunakan Model ISO / IEC 25010 Product Quality," vol. 8, no. 1, pp. 41–50, 2024.
- [9] I. Yatini, F. W. Nurwiyati, and K. Anam, "Performa Microframework Php Pada Rest Api Menggunakan Metode Load Testing," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 19, no. 2, pp. 12–20, 2023, doi: 10.61805/fahma.v19i2.55.
- [10] I. Indrianto, "Performance Testing on Web Information System Using Apache Jmeter and Blazemeter," *J. Ilm. Ilmu Terap. Univ. Jambi*, vol. 7, no. 2, pp. 138–149, 2023, doi: 10.22437/jiutuj.v7i2.28440.
- [11] A. G. A. Muammarulloh and N. A. Wiyani, "Analisis SWOT Implementasi Website Rapor Digital Madrasah Dalam Meningkatkan Kualitas Lembaga di MA Minat Kesugihan," *J. Ilmu Sos. dan Pendidik.*, vol. 7, no. 3, pp. 2451–2461, 2023, doi: 10.58258/jisip.v7i1.5458/http.
- [12] T. Irmina Nangameka and E. Erfiana Amir, "Uji Testing Kualitas Website Pemerintah Daerah Dengan GTMETRIX Penulis: conditions of the Creative Commons Attribution-ShareAlike 4.0 International License-(CC-BY-SA)," *J. Terap. Pemerintah. Minangkabau*, vol. 3, no. 2, pp. 118–136, 2023, [Online]. Available: <https://creativecommons.org/licenses/by-sa/4.0/>
- [13] A. Purnomo, "Software Testing Aplikasi Website PT Gramedia Menggunakan Metode Blackbox pada PT WGS Bandung," *E-Journal Univ. Dianapura*, vol. 91, no. 3, pp. 399–404, 2023.
- [14] W. Tejaya, S. Rahman, A. Munir, S. Web, A. Jmeter, and W. Muat, "Penguujian Website

- Invitees Menggunakan Metode Load,” *J. Kharisma tech*, no. 01, pp. 99–112, 2023.
- [15] A. Ismail, Ahmadi Yuli Ananta, Sofyan Noor Arief, and Elok Nur Hamdana, “Performance Testing Sistem Ujian Online Menggunakan Jmeter Pada Lingkungan Virtual,” *J. Inform. Polinema*, vol. 9, no. 2, pp. 159–164, 2023, doi: 10.33795/jip.v9i2.1190.
- [16] Ardiansyah, Risnita, and M. S. Jailani, “Teknik Pengumpulan Data Dan Instrumen Penelitian Ilmiah Pendidikan Pada Pendekatan Kualitatif dan Kuantitatif,” *J. IHSAN J. Pendidik. Islam*, vol. 1, no. 2, pp. 1–9, 2023, doi: 10.61104/ihsan.v1i2.57.
- [17] N. Hikmah, A. Suradika, and R. A. Ahmad Gunadi, “Metode Agile Untuk Meningkatkan Kreativitas Guru Melalui Berbagi Pengetahuan (Knowledge Sharing) (Studi Kasus: Sdn Cipulir 03 Kebayoran Lama, Jakarta,” *Instruksional*, vol. 3, no. 1, p. 30, 2021, doi: 10.24853/instruksional.3.1.30-39.
- [18] A. P. N. Andre and S. Budi, “ANALISIS PENGARUH PRODUK, HARGA, PROMOSI, LOKASI TERHADAP MINAT BELI PAKAIAN DI MARKETPLACE SHOPEE.” pp. 1595–1600, 2022. [Online]. Available: <https://proceeding.unikal.ac.id/index.php/un/article/view/1344>
- [19] A. COCIORVA, “Website Performance Measurements and Related Analysis,” *Inform. Econ.*, vol. 27, no. 4/2023, pp. 16–31, 2023, doi: 10.24818/issn14531305/27.4.2023.02.
- [20] A. Akbarrizky, A. R. Daman Huri, B. Wisnuadhi, and L. H. Firdaus, “Optimasi micro frontend website untuk meningkatkan load times: teknik, tantangan, dan best practice,” *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 4, no. 2, pp. 366–375, 2023, doi: 10.37859/coscitech.v4i2.5205.
- [21] S. Gobel, T. Abdillah, and I. R. Padiku, “Sistem Informasi Geografis Sebaran Penderita Stunting Berbasis Web di Kabupaten Pohuwato,” *J. Syst. Inf. Technol.*, vol. 3, no. 1, pp. 11–21, 2023.
- [22] S. Pragestu, H. Sujaini, and E. Faja, “Analisis Skalabilitas Web Server Apache Tomcat , Node . Js Dan Go Pada Protokol Hypertext Transfer Protocol (HTTP) Dan Message Queue Telemetry Transport (MQTT) Scalability Analysis of Apache Tomcat , Node . Js and Go Web Servers on Hypertext Transfer,” *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 11, no. 4, pp. 605–611, 2023, doi: 10.26418/justin.v11i4.71607.
- [23] N. Qomariyah, E. Subyantoro, and I. Aswardi, “Penelitian Pendahuluan tentang Pengukuran Performance dan Load Testing pada Learning Management System (LMS) A Preliminary Investigation of Performance and Load Testing for Learning Management System (LMS),” vol. 1, no. 2, pp. 122–126, 2023, doi: 10.25181/rt.v1i2.3134.
- [24] C. P. Agustika, W. S. J. Saputra, and M. Idhom, “Pengujian Aplikasi Greenwallet Dengan,” vol. 2, no. 2, pp. 190–195, 2021.