

Pengujian Load Testing Website Jurusan Teknik Informatika Politeknik Negeri Bengkalis

Harry Setiawan¹, Depandi Enda²

^{1,2}Rekayasa Perangkat Lunak/Politeknik Negeri Bengkalis

e-mail: 1sstwnn@gmail.com, 2depandienda@polbeng.ac.id

Abstract – The Bengkalis State Polytechnic Informatics Engineering department website is an academic site regarding information on study programs and Informatics Engineering majors at the Bengkalis State Polytechnic. However, since it was built in 2019, the website has never been tested. The aim of this research is to test the Bengkalis State Polytechnic Information Engineering department website using the load testing method. Load testing aims to identify the system's maximum capacity for accommodating the number of users. The stages begin with scenario modeling, Apache JMeter testing, and web optimization. Testing looks at aspects of load time, latency, throughput, error rate and response time, by comparing server results. Before optimization of the 10-50users test results, the localhost values were 42.6ms, 150.18ms, 15.16s, 0%, and 150ms. Shared hosting 202ms, 472ms, 45ms, 7.5s, 0%, and 1091ms. VPS hosting 202ms, 472.45ms, 7.5s, 0%, 1091ms. The comparison is stopped the load on it is having problems. After optimization, 10-50users test results, localhost values are 42ms, 54.8ms, 15s, 0%, and 54.6ms. Shared hosting 362ms, 126.86ms, 12.7s, 0%, and 441ms. VPS hosting 531ms, 220ms, 42s, 0%, 608ms. Test results of 100-300users average localhost values of 89ms, 121ms, 69s, 0%, and 121ms. Shared hosting 623ms, 2100ms, 33s, 0%, 4615ms. VPS hosting 412ms, 126ms, 76s, 0%, and 454ms. Comparison stopped load above encountered an error. It can be concluded that the maximum limit after optimizing shared hosting is 300 users, the value obtained increases significantly, the optimization results also have an impact on localhost and VPS hosting.

Keywords – Apache JMeter, Load Testing, Website.

Abstrak – Website jurusan Teknik Informatika Politeknik Negeri Bengkalis adalah situs akademik seputar informasi program studi dan jurusan Teknik Informatika Politeknik Negeri Bengkalis. Namun, sejak dibangun pada tahun 2019, website tersebut belum pernah diuji. Tujuan dari penelitian ini untuk menguji website jurusan Teknik Informatika Politeknik Negeri Bengkalis dengan metode pengujian beban. Pengujian beban bertujuan mengidentifikasi maksimum pada sistem dalam menampung jumlah pengguna. Tahapan dimulai pemodelan skenario, pengujian Apache JMeter, dan optimasi web. Pengujian mengamati aspek *load time*, *latency*, *throughput*, *error rate*, dan *response time*, dengan membandingkan hasil perolehan server. Sebelum optimasi hasil uji 10-50users, nilai localhost adalah 42,6ms, 150,18ms, 15,16s, 0%, dan 150ms. Shared hosting 202ms, 472ms, 45ms, 7,5s, 0%, dan 1091ms. VPS hosting 202ms, 472,45ms, 7,5s, 0%, 1091ms. Perbandingan dihentikan beban di atasnya mengalami masalah. Setelah optimasi, hasil uji 10-50users, nilai localhost adalah 42ms, 54,8ms, 15s, 0%, dan 54,6ms. Shared hosting 362ms, 126,86ms, 12,7s, 0%, dan 441ms. VPS hosting 531ms, 220ms, 42s, 0%, 608ms. Hasil uji 100-300users rata-rata nilai localhost 89ms, 121ms, 69s, 0%, dan 121ms. Shared hosting 623ms, 2100ms, 33s, 0%, 4615ms. VPS hosting 412ms, 126ms, 76s, 0%, dan 454ms. Perbandingan dihentikan beban di atas mengalami kesalahan. Dapat disimpulkan batas maksimum setelah optimasi shared hosting 300users, nilai yang diperoleh meningkat signifikan, hasil optimasi juga berdampak pada localhost dan VPS hosting.

Kata Kunci – Apache JMeter, Pengujian Beban, Situs Web.

I. PENDAHULUAN

Website jurusan Teknik Informatika Politeknik Negeri Bengkalis adalah situs akademik yang menampilkan berbagai informasi mengenai program studi dan jurusan Teknik Informatika Politeknik Negeri Bengkalis. Website merupakan kumpulan hyperlink yang menghubungkan satu ke alamat yang lainnya menggunakan bahasa pemrograman HTML (*Hypertext Markup Language*)[1]. Situs web ini dibangun pada tahun 2019, sejak dibangun website ini selalu mengalami masalah pada akses jumlah pengguna yang banyak, situs selalu mengalami *down* dan berujung tidak dapat diakses. Hal ini dapat disebabkan karena penggunaan server ataupun penggunaan sumber daya yang begitu besar. Dengan begitu perlu dilakukan pengujian lebih lanjut terhadap sistem untuk mengetahui batas maksimum pengguna yang dapat mengakses secara bersamaan dan dibuatkan rekomendasi penggunaan server yang lebih baik.

Performa pada sebuah situs web dapat dipengaruhi oleh jumlah pengguna. Jumlah pengguna yang mengakses situs web dapat berdampak pada kinerja. Agar pengguna dapat mengunjungi suatu situs web, diperlukan server web

yang dapat bekerja pada beban tinggi. Hal ini situs web dapat menangani pengguna dengan baik[2]. *Apache JMeter* adalah salah satu teknologi yang dapat digunakan untuk pengujian kinerja[3].

Pengujian kinerja dipisahkan menjadi tiga kategori: *load test*, *strength test* dan *stress test*[4]. Tujuan dasar dari pengujian adalah untuk membangun pengujian yang secara teratur mengungkapkan jenis masalah dengan sedikit usaha dan waktu[5].

Pendekatan pengujian penelitian ini adalah uji beban, pengujian beban adalah jenis pengujian kinerja yang mengukur respon sistem terhadap kondisi beban yang berbeda. Temuan pengujian beban dapat digunakan untuk menilai bagaimana sistem beroperasi ketika beberapa pengguna mengaksesnya secara bersamaan[6].

Permatasari dkk. melakukan pengujian beban pada aplikasi MeTANI, dan hasilnya menunjukkan bahwa target pengujian waktu pemuatan dan memori proses tercapai[6]. Tejaya dkk, melakukan *load testing* hasil dari *load time* pada skenario pertama dan kedua sudah memenuhi target pengujian dibawah 3 detik[5]. Agustika dkk, juga melakukan *load testing* dengan hasil *response time* kurang dari 1.0 detik pada aplikasi Greenwallet[2]. Setiawan dkk, melakukan pengujian beban pada API, hasil waktu respon dan *throughput* mengalami penurunan seiring bertambahnya *user*, namun dapat menangani *request* ditunjukkan dengan *error* 0.0%[7]. Fandy, dkk melakukan uji beban terhadap penerapan layanan *Virtual Private Server (VPS)*, temuan yang didapatkan dari pengujian *web server* pada layanan *Elastic Cloud Computel* mempunyai kinerja yang lebih unggul dibandingkan layanan server lainnya[8].

Berdasarkan permasalahan yang telah dijelaskan sebelumnya, pada penelitian ini bertujuan untuk melakukan pengujian *website* jurusan Teknik Informatika Politeknik Negeri Bengkalis, dengan menggunakan metode pengujian beban. Pengujian beban ini sangat cocok untuk menyelesaikan permasalahan yang telah dijelaskan sebelumnya, dengan memvisualisasi jumlah pengguna yang mengakses secara bersamaan, pengguna yang divisualisasikan menggunakan alat pengujian beban yakni *Apache JMeter*. *Apache JMeter* merupakan sebuah aplikasi *open source* dengan tampilan sederhana yang mudah dipahami dan dioperasikan, aplikasi ini dapat digunakan sebagai alat pengujian beban dan untuk mengetahui nilai dari aspek yang diuji pada situs web. Pada penelitian ini pengujian beban berfokus pada aspek nilai dari *load time*, *latency*, *throughput*, *error rate* dan *response time*, pengujian beban dimulai dengan dua sesi pengujian sebelum optimasi *website* dan pengujian setelah optimasi *website*. Hasil yang diperoleh akan dibandingkan dengan tiga server yang digunakan yakni *localhost*, *shared hosting* dan *vps hosting*. Hasil dari penelitian dapat dijadikan sebagai arahan untuk mengevaluasi dari sistem ataupun penggunaan layanan server pada *website* jurusan Teknik Informatika Politeknik Negeri Bengkalis.

II. PENELITIAN YANG TERKAIT

Penelitian sebelumnya meneliti waktu muat, khususnya kecepatan muat ideal situs web *Invitees*. Rata-rata *load time website* *Invitees* pada skenario pertama adalah 2,7 detik, sedangkan pada skenario kedua adalah 1,7 detik. Hasil yang diperoleh mencapai target pengujian dalam waktu kurang dari 3 detik[5].

Studi lain berfokus pada tiga indikator: tingkat kesalahan, waktu reaksi, dan *throughput*. Pengujian dilakukan pada lima titik akhir API. Waktu respons dan kinerja *throughput* menurun seiring meningkatnya beban permintaan pengguna, namun setiap titik akhir menangani permintaan secara memadai dengan tingkat kesalahan 0,0% [7].

Penelitian ini melakukan pengujian pada empat parameter yakni *throughput*, *response time*, *latency* dan *resource utilization* untuk mengetahui performa server AWS. Hasil yang diperoleh pada AWS, server web berdasarkan layanan *Elastic Cloud Compute* mengungguli penyediaan layanan lainnya [9]

Pengujian sebelumnya di *website* IAIN Salatiga tidak ditemukan kendala, penggunaan *CPU* tertinggi sekitar 6-11% dan penggunaan *RAM* sekitar 2-4%. Meskipun terdapat hambatan pada saat pengujian dengan 50 pengguna virtual dan jangka waktu pengujian 5 menit, namun *website* IAIN Salatiga tidak terdapat permasalahan yang buruk[10].

Kajian lain pengujian pada performa API aplikasi DiTenun, Pengukuran baseline web service, peneliti memperoleh statistik *response time* yang digunakan sebagai acuan API yang dijalankan pada sistem di server peneliti[11].

Penelitian terdahulu yang dilakukan pengujian untuk menganalisis *processor* yang dibutuhkan pada komputer laboratorium. Temuan yang dikumpulkan mengusulkan *processor* yang memenuhi standar komputer laboratorium[12]

Penelitian selanjutnya melakukan pengujian situs web Greenwallet PT. Terminal Teluk Lamong menguji kinerja waktu respons server *localhost* dan *hosting*. Hasil pengujian mencapai target waktu respon server hosting kurang dari 1.0. Salah satu fitur, khususnya tindakan registrasi, tidak pernah melebihi sasaran pengujian dalam skenario ini, sehingga menghasilkan waktu respons lebih dari satu detik[2]

Penelitian lain pengujian pada *website* Universitas Pendidikan Ganesha, untuk menguji kecepatan server *cloud*. Pengujian pertama dengan program *GTmetrix* menghasilkan peringkat kinerja F (47%) untuk *PageSpeed* dan E (53%) untuk *YSlow*, serta waktu akses 2,5 detik dan ukuran total 1,22 MB untuk halaman yang ditargetkan. Untuk temuan pengujian beban, kami mencapai hasil simulasi sebanyak 55.138 permintaan dengan tidak ada kegagalan,

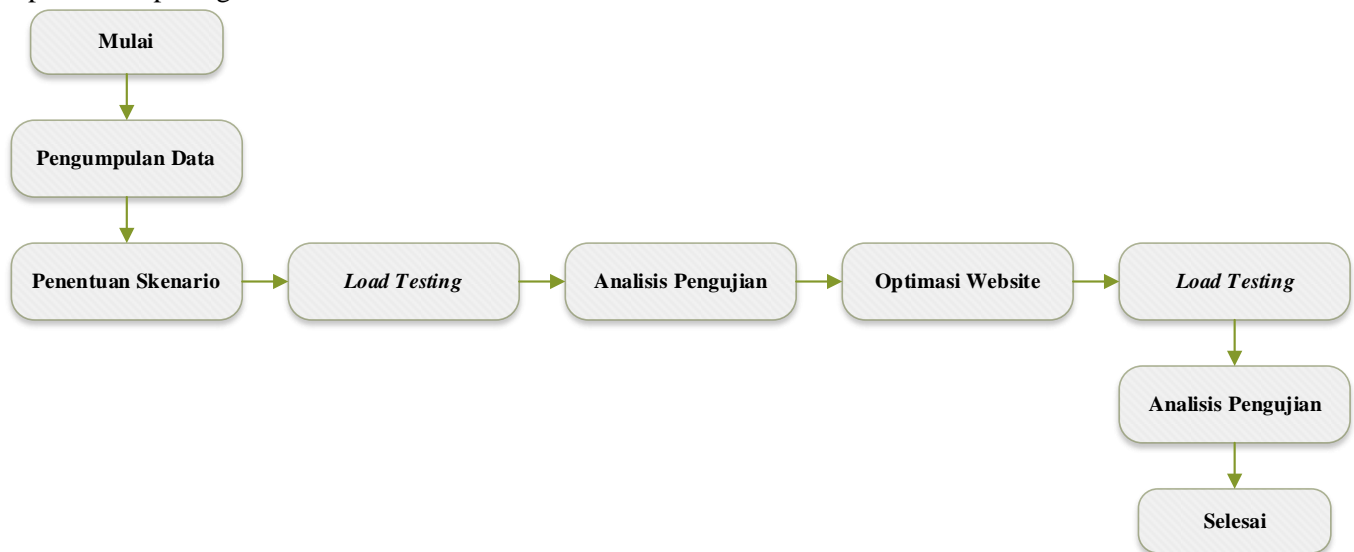
jumlah maksimum permintaan dapat dilakukan selama per/detik ialah 87 pengguna, dan waktu respon mencapai 87,84ms[13].

Penelitian ini melakukan uji aplikasi MeTANI berfokus pada *loading time* dan *Hasil* memori proses dan waktu proses dapat dipelajari; memori proses tidak lebih dari 400MB, dan waktu pemuatan tidak lebih dari 3 detik. Kedua target pengujian telah terpenuhi sesuai rencana pengujian[6].

III. METODE PENELITIAN

A. Metode Penelitian

Penelitian ini meliputi delapan tahapan, yang diawali dengan pengumpulan data, penentuan scenario, melakukan pengujian *load testing*, menganalisis hasil pengujian sebelum optimasi sistem web, mengoptimasi sistem web, melakukan pengujian *load testing* ulang dan analisis hasil pengujian setelah optimasi sistem web. Tahapan penelitian dapat dilihat pada gambar 1.



Gambar 1. Tahapan Penelitian *Load Testing*

B. Pengumpulan Data

Penelitian ini diawali mengumpulkan data dari statistik pemakaian layanan server situs web jurusan Teknik Informatika Politeknik Negeri Bengkalis. Pada gambar 2.

Statistics
Views Domains
0 / 0
Domains Add-On
0 / 0
Penggunaan Disk
82,32 MB / 1,000 MB (8,23%)
Kecala
MySQL Disk Usage
4,25 MB / 10,2 MB (4,17%)
Bandwidth
1,16 GB / 97,66 GB (1,19%)
Penggunaan File
20,071 / -
Subdomain
0 / -
Alur Email
3 / -
Mailing List
0 / -
Keanggotaan Cpanel
0 / -
Pelanggan
0 / -
Alur email
0 / -
Alur FTP
2 / -

Gambar 2. Statistik *Cpanel*

C. Penentuan Skenario

Setelah *tahap* pengumpulan data, dilanjutkan dengan tahap penentuan skenario. Menentukan skenario pengujian berdasarkan pada statistik penggunaan layanan server. skenario tersebut menjadi tiga keadaan, yaitu kondisi *baseline*, situs web diakses pengguna pada saat layanan dimanfaatkan secara luas. Kondisi *peak*, situs web diakses pada saat layanan digunakan dan kondisi *stress*, situs web diakses pengguna ketika dua kali lebih banyak dari sebelumnya[14]. Alur skenario pada tabel 1.

Tabel 1. *Test Case* Pengujian

Skenario	Setting Thread Group	Server	Waktu Pelaksanaan
----------	----------------------	--------	-------------------

1	10, 30, 50 threads, 2 ramp up, 1 loop	Localhost	-
2		Shared	09.00-11.00
3		VPS	13.00-15.00
4	100, 300, 500 threads, 5 ramp up, 1 loop	Localhost	-
5		Shared	09.00-11.00
6		VPS	13.00-15.00
7	1000, 3000, 5000 threads, 15 ramp up, 1 loop	Localhost	-
8		Shared	09.00-11.00
9		VPS	13.00-15.00

D. Load Testing

Pengujian beban adalah pengujian kinerja yang mengukur reaksi sistem dalam berbagai situasi dan beban. Pengujian beban dilakukan untuk menguji bagaimana sistem dapat mengelola masalah atau beban yang diuji, tergantung pada kondisinya[15]. Pengujian dilakukan sesuai dengan skenario yang telah ditetapkan, dengan menggunakan tools Apache JMeter. Pengujian berfokus pada lima parameter yakni *load time*, *latency*, *throughput*, *error rate* dan *response time* pada pengujian *website* jurusan Teknik Informatika Politeknik Negeri Bengkalis. Setelah dilakukan pengujian dilakukan perhitungan dengan rumus sebagai berikut:

1. Load time

WebsiteBuilderExpert menemukan bahwa *load time* terbaik untuk situs web non-komersial adalah 3 detik. Lebih lanjut menurut penelitian Cami Bird, kecepatan pemuatan halaman yang optimal untuk mesin pencari adalah kurang dari 3 detik.

2. Latency

Waktu *latency* (*lag time*) yang ideal pada sebuah *website* adalah kurang dari 1 detik. Dengan perhitungan nilai *latency* rumusan sebagai berikut:

$$Latency = \frac{\text{jumlah seluruh nilai latency}}{\text{jumlah sample}}$$

3. Throughput

Throughput adalah jumlah permintaan pelanggan yang diproses oleh sistem per satuan waktu. Dengan perhitungan nilai *throughput* rumusan sebagai berikut:

$$Throughput = \frac{\text{jumlah request}}{\text{waktu rata – rata request}}$$

4. Error rate

Error rate pada sebuah situs web dapat diartikan dengan perbandingan antara jumlah prediksi yang salah dengan total prediksi yang dilakukan oleh sistem. Dengan perhitungan nilai *error rate* rumusan sebagai berikut:

$$Error Rate = \left(\frac{\text{jumlah sample yang gagal}}{\text{total sample}} \right) \times 100\%$$

5. Response time

Waktu respons 0,1 detik membuat pengguna merasa bahwa sistem merespons secara instan, tanpa memerlukan umpan balik khusus kecuali untuk menampilkan hasilnya. Dengan perhitungan nilai *response time* rumusan sebagai berikut:

$$Response Time = \frac{\text{jumlah seluruh nilai sample time}}{\text{jumlah sample}}$$

E. Analisis Pengujian

Setelah pengujian *load testing* berlangsung maka hasil pengujian akan dilakukan analisis pengujian dan rekomendasi perbaikan yang didapat dari analisa pengujian[16]. Dengan dilakukan analisis lebih lanjut dapat dilakukan perbaikan menyesuaikan standar waktu yang ideal bagi sebuah sistem *website*[17]. Gambaran tabel hasil pengujian beban terlihat pada tabel 2.

Tabel 2. Hasil Pengujian

Skenario	Thread Group	Server	Load Time	Latency	Throughput	Error Rate (%)	Response Time (ms)
Skenario 1							

Skenario 2							
Skenario 3							
Skenario ..							
Skenario 8							
Skenario 9							

F. Optimasi Website

Setelah tahapan analisis pengujian, pengujian dilakukan untuk mengoptimasi *website* menemukan kesalahan atau kekurangan dari sistem untuk kemudian diperbaiki[18]. Kecepatan *website* mempengaruhi pengalaman pengguna, karena itu penting melakukan pengujian performa dan melakukan optimasi performa *website* agar optimal[19]. Untuk melakukan optimasi pada *website* dapat dilakukan dengan penggunaan aplikasi meliputi *image compression*, *minify css*, *html* dan *javascript* sehingga dapat meningkatkan kecepatan dan kestabilan *website*[20][21]. Optimasi dilakukan pada sisi *front website*, dengan metode meminimalkan ukuran file[22]. Proses mengoptimasi *website* dilakukan pada kode *html*, *css* dan *javascript*. Selanjutnya dilakukan pengujian beban kembali dan perbandingan hasil sebelum optimasi dan sesudah optimasi.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Load Testing

Pengujian beban pada *website* jurusan Teknik Informatika Politeknik Negeri Bengkalis, dilakukan pengujian pada alamat url untuk menghasilkan *report* dari parameter *load time*, *latency*, *throughput*, *error rate* dan *response time*.

B. Test Plan Load Testing

Penelitian ini berfokus pada lima parameter yang akan diuji, yakni *load time*, *latency*, *throughput*, *error rate* dan *response time* dapat dilihat pada tabel 3.

Tabel 3. Konfigurasi *Thread Group*

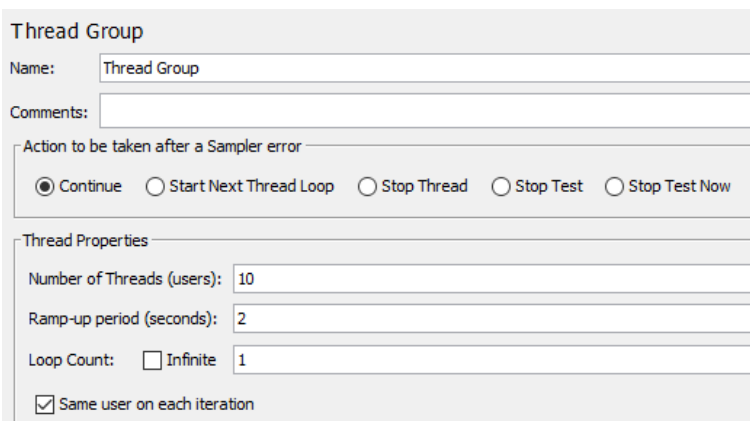
<i>Number of threads (users)</i>	<i>Ramp-up period (seconds)</i>	<i>Loop count</i>
10 users	2 seconds	1
30 users		
50 users		
100 users	5 seconds	
300 users		
500 users		
1000 users	15 seconds	
3000 users		
5000 users		

C. Proses Pengujian

Bagian ini dilakukan tahapan konfigurasi sesuai *test plan*, proses pengujian diawali dengan konfigurasi *thread group*, *http request* serta menampilkan *summary report* dan *view results in table* pada tools *Apache JMeter*.

1. Konfigurasi *thread group*

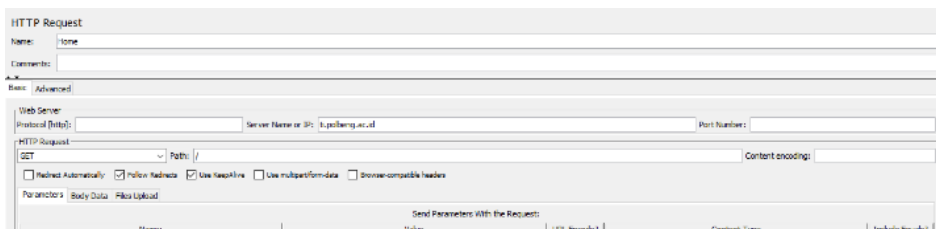
Thread group merupakan tabel untuk mensimulasikan proses pengujian dengan beberapa kolom yang dapat dikonfigurasi, konfigurasi disesuaikan pada *test plan* gambar 3.



Gambar 3. Konfigurasi *Thread Group*

2. Konfigurasi *http request*

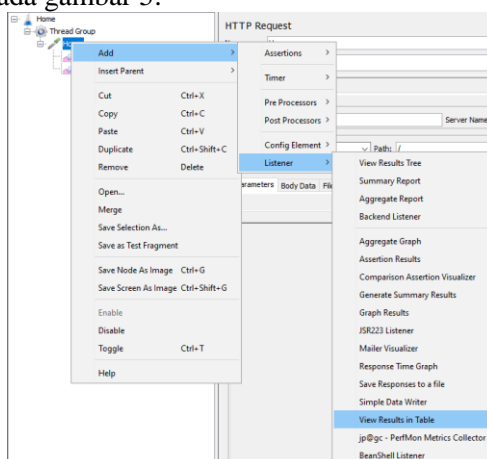
Setelah konfigurasi pada *thread group*, selanjutnya melakukan konfigurasi alamat *url* dan *path website* pada tabel konfigurasi *http request* dapat dilihat pada gambar 4.



Gambar 4. Konfigurasi *HTTP Request*

3. Menampilkan *view listener*

Sebelum memulai pengujian tambahkan halaman hasil pengujian pada menu *listener*, pilih *summary report* dan *view results in table*. Dapat dilihat pada gambar 5.



Gambar 5. Konfigurasi *Listener*

D. Hasil Pengujian

Hasil pengujian sebelum mengoptimasi *website* selanjutnya akan dilakukan analisa pada tabel 4.

Tabel 4. Hasil Pengujian Sebelum Optimasi

Skenario	Thread Group	Server	Load Time	Latency	Throughput (rps)	Error Rate (%)	Response Time (ms)
1	10 user	localhost	44	340,5	5.5	0.00	340
		Shared Host	205	824,8	3.4	0.00	1750

2	30 user	VPS Host	380	135,5	4.5	0.00	434
		Localhost	42	57,66	15.0	0.00	58
		Shared Host	197	299,1	10.0	0.00	872
3	50 user	VPS Host	371	122,03	12.8	0.00	421
		Localhost	42	52,38	25.0	0.00	52
		Shared Host	204	293,46	9.0	0.00	652
4	100 user	VPS Host	392	135,16	21.0	0.00	481
		Localhost	42	51,46	3.3	0.00	324
		Shared Host	52	751,14	11.5	64.0	2693
5	300 user	VPS Host	401	149,65	16.6	0.00	568
		Localhost	45	53,62	58.5	0.00	54
		Shared Host	51	4,51	58.4	93.33	196
6	500 user	VPS Host	450	413,96	42.7	0.00	1735
		Localhost	78	651,63	91.3	0.00	656
		Shared Host	51	75,44	29.6	92.80	3451
7	1000 user	VPS Host	592	785,19	33.5	0.00	5157
		Localhost	47	58,04	66.5	0.00	56
		Shared Host	51	2,80	64.4	96.90	99
8	3000 user	VPS Host	1062	2195,96	13.2	21.90	20673
		Localhost	0	61,11	112.1	27.90	6123
		Shared Host	50	0,9453	83.3	99.13	12915
9	5000 user	VPS Host	1922	4236,84	19.5	60.13	56791
		Localhost	0	4582,2	158.7	45.68	4584
		Shared Host	51	10733,7	151.1	97.88	11749
		VPS Host	692	4719,37	21.3	65.28	71209

Hasil pengujian layanan *shared hosting*, terlihat jelas bahwa ketika jumlah pengguna berjalan antara 10 hingga 50, dengan satu *request* setiap 2 *second*, kinerja layanan cukup baik. *Load time* yang dilaporkan sebesar 204ms dianggap baik, dan *latency* 293,46 menunjukkan bahwa server dapat menangani jaringan secara efektif. Meskipun *throughput* hanya mencapai 9.0, namun tidak ditemukan *error*, dan *response time* yang kurang dari 3 detik, bahwa layanan ini mampu mengelola beban saat ini.

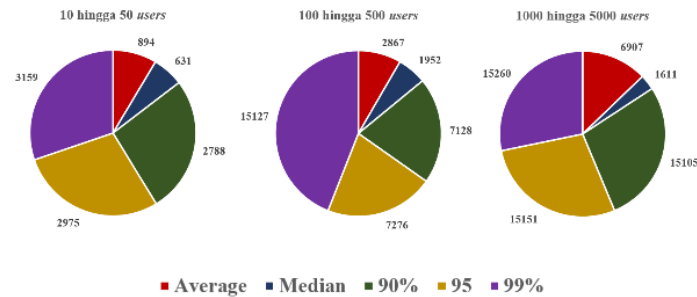
Namun, ketika jumlah pengguna meningkat dari 100 menjadi 500 dengan satu *request* setiap 5 *second*, kualitas layanan menurun secara signifikan. *Load time* 51ms hasil yang dinyatakan tidak baik, dikarenakan hasil *error* yang tinggi sebesar 92,80% dan *response time* 3451ms menunjukkan bahwa *shared hosting* tidak dapat menangani beban kerja melebihi 50 pengguna.

Hal serupa terjadi ketika jumlah pengguna mencapai 1000 hingga 5000, dengan satu *request* setiap 15 *second*. Meskipun *load time* dan *throughput* tampak lebih rendah, namun, *latency* 10733,7 dan *error rate* 97,88% menunjukkan bahwa *shared hosting* tidak mampu mengelola beban saat ini, sehingga mengakibatkan waktu reaksi yang sangat tinggi.

Analisis terhadap *localhost* dan *VPS hosting* menunjukkan kinerja yang unggul, terutama pada beban pengujian yang berat, di mana keduanya dapat mempertahankan kualitas layanan yang layak tanpa mendeteksi kesalahan besar.

Grafik pengujian kinerja web menunjukkan bahwa seiring bertambahnya jumlah pengguna, waktu *respons* rata-rata juga meningkat. Pada rentang 10 hingga 50 pengguna, waktu *respons* rata-rata adalah sekitar 894 milidetik, dengan sebagian besar waktu *respons* kurang dari 3000 milidetik, namun 1% kasus melampaui 3159 milidetik.

Sementara itu, jika jumlah pengguna bertambah dari 100 menjadi 500, waktu *respons* rata-rata mencapai 2867 milidetik, dengan sebagian besar waktu *respons* berada di bawah 8000 milidetik, sementara ada kalanya waktu *respons* lebih lama, terutama di sekitar persentil 99%. Dalam skenario pengujian dengan 1000 hingga 5000 pengguna, waktu *respons* rata-rata mencapai 6907 milidetik, dengan sebagian besar waktu *respons* tetap di bawah 15260 milidetik, namun seperti yang disebutkan sebelumnya, masih ada sejumlah kecil kasus yang waktu *respons* cukup tinggi, khususnya pada persentil 99%. Investigasi ini mengungkapkan bahwa ketika jumlah pengguna meningkat, kinerja web menurun, yang dapat berdampak pada pengalaman pengguna, terutama dalam kondisi beban tinggi. Berikut grafik perbandingan dilihat pada gambar 6.



Gambar 6. Grafik Pengujian

E. Optimasi Website

Mengoptimasi *website* dilakukan dengan cara optimasi pada bagian *script css, html, javascript* dan *file gambar*. Bagian tersebut akan berpengaruh pada saat pertama kali proses *load* pada *website*.

Berbagai alat internet digunakan untuk membantu proses optimasi *website*. Pertama, codebeautify.org menyediakan alat untuk mengecilkan *HTML* dengan menghilangkan baris dan menggabungkannya menjadi satu baris kode. Kedua, minifier.org digunakan untuk memperkecil ukuran file seperti *JavaScript* dan *CSS* dengan menghapus karakter yang tidak diperlukan seperti spasi dan baris tambahan. Ketiga, loveimg.com adalah situs web gratis yang memungkinkan untuk mengedit, mengonversi, dan mengoptimalkan foto secara *online*.

Front-end sering kali menggunakan kode *HTML* yang tidak dioptimalkan, seperti contoh dari *file index.html* ini. Kode ini memiliki pengaruh besar pada cara memuat *website*. Untuk meningkatkan performa *website*, seluruh kode *HTML* yang berdampak pada *front-end* akan dioptimalkan. Kode *HTML* dapat dilihat pada gambar 7.

```

application > <> index.html > <html>
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>403 Forbidden</title>
5     <meta http-equiv="Refresh" content="5; url=http://ti.polbeng.ac.id/" />
6   </head>
7   <body>
8     <p>Directory access is forbidden.</p>
9   </body>
10 </html>

```

Gambar 7. Sebelum Optimasi *Script HTML*

Hasil yang dilakukan dari memodifikasi kode pada situs codebeautify.org. Sebelumnya jumlah baris kode cukup panjang, namun setelah optimasi, baris-baris ini dihilangkan, dan kode dikurangi menjadi satu baris. Optimasi dilakukan pada setiap kode *HTML* yang memengaruhi *front-end* untuk meningkatkan muat halaman. Hasil optimasi kode *HTML* dapat dilihat pada gambar 8.

```

application > <> index.html > <html>
1 <!DOCTYPE html><html><head><title>403 Forbidden</title><meta http-equiv="Refresh" content="5; url=http://ti.polbeng.ac.id/" /></head><body><p>Directory access is forbidden.</p></body></html>

```

Gambar 8. Setelah Optimasi *Script HTML*

Kode *javascript* juga dapat berdampak pada waktu muat buka halaman. Untuk meningkatkan kinerja situs web, semua kode *javascript* yang memengaruhi *front-end* akan dioptimalkan. Kode dapat dilihat pada gambar 9.


```

assets > web > js > # app-1.js > ...
1 function socialShare() {
2   $(window).scroll(function () {
3     var e = $("#content").position().top,
4     s = $(".single-post").height() / 1.8;
5     $(".share-box-wrapper").height(s);
6     $(this).scrollTop() > e ? $(".share-box-wrapper .share-box").addClass("sticky") : $(".share-box-wrapper .share-box").
7     removeClass("sticky");
8     $(this).scrollTop() > s ? $(".share-box-wrapper .share-box").addClass("holdy") : $(".share-box-wrapper .share-box").
9     removeClass("holdy");
10  });
11  $(document).ready(function () {
12    function e() {
13      $(window).width() < 1199 &&
14      $(".navbar ul li.dropdown>.caret").on("click", function () {
15        $(".navbar ul li.dropdown").removeClass("active"), $(this).parent().parent().addClass("active"), e.preventDefault
16        ();
17      });
18    }
19    $(window).scroll(function () {
20      var e = $(window).scrollTop(),
21      s = $(".header");
22      e > 15 ? s.addClass("sticky-header") : s.removeClass("sticky-header");
23    });
24    $(".navbar .dropdown").on("click", function () {
25      $(".navbar .dropdown").removeClass("active"), $(this).addClass("active");
26    });
27    $(".btn-tabmenu").on("click", function () {
28      $(this).toggleClass("show-menu"), $(this).hasClass("show-menu") ? $(".widget_menu").slideUp() : $(".widget_menu").
29      slideDown();
30    });
31  });
32  });

```

Gambar 9. Sebelum Optimasi Script Javascript

Setelah optimasi menggunakan situs minifier.org. Sebelumnya kode, memiliki beberapa baris, dikurangi menjadi satu baris kode. Hal ini dapat meningkatkan performa web, proses ini dilakukan pada setiap kode *javascript* yang memengaruhi *front-end* web. Dapat dilihat pada gambar 10.

```

assets > web > js > # app-1.js > ...
1 function socialShare(){$(window).scroll((function(){var e=$("#content").position().top,s=($(".single-post").height())/1.8;$(".
share-box-wrapper").height(s);$(this).scrollTop()>e?($(".share-box-wrapper .share-box").addClass("sticky");$(".share-box-wrapper .
share-box").removeClass("sticky");$(".share-box-wrapper .share-box").addClass("holdy");$(".share-box-wrapper .share-box").re
moveClass("holdy");$(document).ready((function(){function e(){$(window).width()<1199&&$(".navbar ul li.dropdown>.caret").
on("click",function(){$(".navbar ul li.dropdown").removeClass("active");$(this).parent().parent().addClass("active");e.preventDefault
().addClass("active");e.preventDefault();})$(window).scroll((function(){var e=$(window).scrollTop(),s=($(".header");e>15?s.add
Class("sticky-header");s.removeClass("sticky-header");$(".navbar .dropdown").on("click",function){$(".navbar .dropdown").
removeClass("active");$(this).addClass("active");$(".btn-tabmenu").on("click",function){$(this).toggleClass("show-menu");$
(this).hasClass("show-menu")?$(".widget_menu").slideUp():$(".widget_menu").slideDown();})})})})});

```

Gambar 10. Sesudah Optimasi Script Javascript

Optimasi kode *css* juga menjadi aspek penting dalam proses muat *website*. Semua kode *css* yang mempengaruhi *front-end* web akan dioptimalkan untuk meningkatkan kecepatan halaman. Kode dapat dilihat pada gambar 11.

```

assets > web > css > # menu.css
1 .dropdown-submenu {
2   position: relative;
3 }
4 .dropdown-submenu > .dropdown-menu {
5   top: -1px;
6   left: 100%;
7 }
8 .dropdown-submenu: hover > .dropdown-menu {
9   display: block;
10 }
11 .dropdown-submenu > a: after {
12   display: block;
13   content: " ";
14   float: right;
15   width: 0;
16   height: 0;
17   border-color: transparent;
18   border-style: solid;
19   border-width: 5px 0 5px 5px;
20   border-left-color: #ccc;
21   margin-top: 5px;
22   margin-right: -10px;
23 }

```

Gambar 11. Sebelum Optimasi Script CSS

Setelah optimasi menggunakan situs web minifier.org. Dengan menghilangkan jumlah baris-baris kode, dan menjadikan satu baris kode. Proses optimasi dilakukan keseluruhan kode *css* yang memengaruhi *front-end* situs web untuk mengoptimalkan kecepatan halaman. Kode dapat dilihat pada gambar 12.

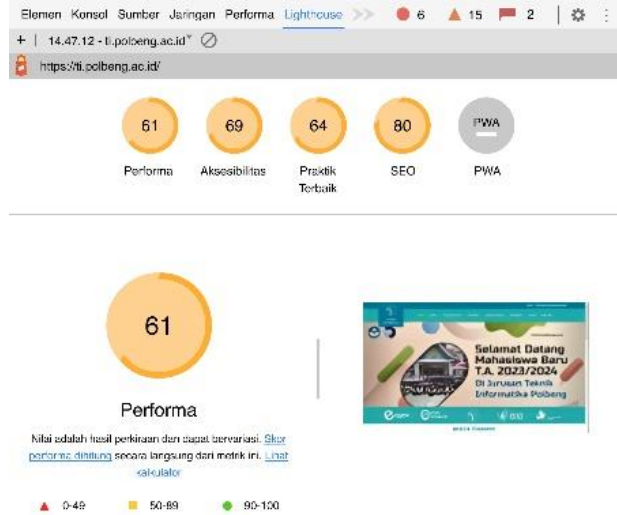
```

assets > web > css > # menu.css
1 .dropdown-submenu(position:relative),.dropdown-submenu>.dropdown-menu(top:-1px;left:100%;).dropdown-submenu: hover>.dropdown-menu
(display:block),.dropdown-submenu>a: after(display:block;content:" ";float:right;width:0;height:0;border-color:transparent;
border-style:solid;border-width:5px 0 5px 5px;border-left-color:#ccc;margin-top:5px;margin-right:-10px;).
dropdown-submenu: hover>a: after(border-left-color:#fff);.dropdown-submenu.pull-left(float:none);.dropdown-submenu.pull-left>.
dropdown-menu(left:-100%;margin-left:10px;-webkit-border-radius:6px 0 6px 6px;-moz-border-radius:6px 0 6px 6px;border-radius:6px 0
6px 6px);

```

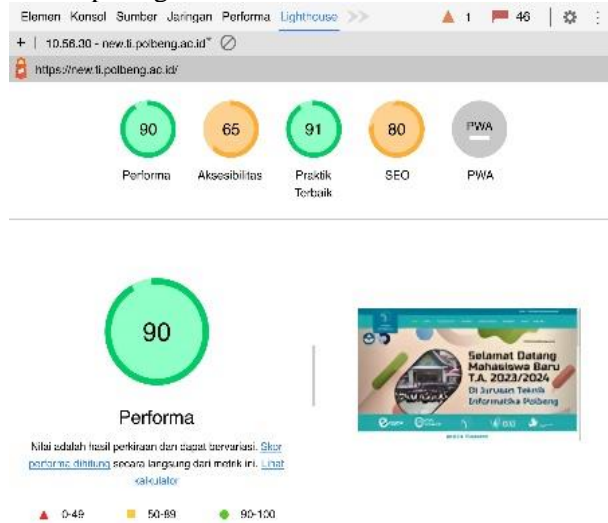
Gambar 12. Sebelum Optimasi Script CSS

Sebelum dilakukan optimasi kode *html*, *javascript*, dan *css* secara keseluruhan, kinerja web Jurusan Teknik Informatika Politeknik Negeri Bengkalis mendapat skor 61 berdasarkan hasil pengujian kinerja dari *tool lighthouse* yang digunakan dari browser *google chrome*. Hasil performa dapat dilihat pada gambar 13.



Gambar 13. Performa Web Sebelum Optimasi

Setelah dilakukan optimasi, hasil uji ulang *lighthouse* menunjukkan peningkatan kinerja yang cukup besar, mencapai angka 90. Optimasi pada kode *html*, *javascript*, dan *css* berdampak signifikan terhadap kinerja web. Hasil performa setelah optimasi dapat dilihat pada gambar 14.



Gambar 14. Performa Web Setelah Optimasi

F. Hasil Pengujian Setelah Optimasi

Hasil uji setelah mengoptimasi *website* selanjutnya akan dilakukan analisa pada tabel 5.

Tabel 5. Hasil Pengujian Setelah Optimasi

Skenario	Thread Group	Server	Hasil Setelah Optimasi				
			Load Time	Latency	Throughput (rps)	Error Rate (%)	Response Time (ms)
1	10 user	Localhost	43	57,9	5.4	0.00	58
		Shared Host	371	124,5	4.4	0.00	446
		VPS Host	179	67,9	5.0	0.00	195
2	30 user	Localhost	41	54,46	15.0	0.00	54
		Shared Host	336	122,6	12.6	0.00	425
		VPS Host	176	75,5	14.1	0.00	204
3	50 user	Localhost	43	52,14	24.8	0.00	52

Skenario	Thread Group	Server	Hasil Setelah Optimasi				
			Load Time	Latency	Throughput (rps)	Error Rate (%)	Response Time (ms)
		Shared Host	379	133,48	21.3	0.00	452
		VPS Host	176	77,24	23.4	0.00	209
4	100 user	Localhost	42	61,4	20.2	0.00	61
		Shared Host	307	107,68	18.7	0.00	373
		VPS Host	217	65,47	19.3	0.00	240
5	300 user	Localhost	47	60,17	49.2	0.00	60
		Shared Host	316	1992,88	14.7	0.00	4242
		VPS Host	195	61,39	56.8	0.00	214
6	500 user	Localhost	87	486,15	93.7	0.00	487
		Shared Host	313	4877,28	19.3	4.20	8686
		VPS Host	196	57,38	95.2	0.00	211
7	1000 user	Localhost	50	68,41	66.5	0.00	68
		Shared Host	327	1812,34	14.1	68.20	21837
		VPS Host	195	58,47	65.8	0.00	224
8	3000 user	Localhost	0	6165,42	101.2	26.90	6170
		Shared Host	322	1599,73	53.2	90.87	22966
		VPS Host	200	89,37	179.1	0.00	361
9	5000 user	Localhost	0	4348,79	180.8	60.08	4357
		Shared Host	316	1226,89	81.1	88.32	23302
		VPS Host	424	443,02	113.1	0.14	2953

Analisis pengujian setelah optimasi pada *thread* 10 hingga 50 untuk layanan *shared hosting* menunjukkan peningkatan kinerja yang cukup besar. Parameter seperti *load time*, *latency*, *throughput*, *error rate* dan *response time* telah mengalami peningkatan yang signifikan. *Load time* dan *response time* yang diperoleh dibawah 3 detik, *latency* dan *throughput* diperoleh juga sudah baik. Selama pengujian berlangsung tidak terjadi *error*.

Namun, ketika beban ditingkatkan dari 100 menjadi 500 *thread*, hasil yang diperoleh berbeda. Meskipun *load time* tetap sangat baik, *latency* dan *throughput* mengalami penurunan kinerja yang signifikan, terutama saat memuat lebih dari 300 *thread*. *Response time* juga melambat, melampaui waktu diatas 3 detik. Dengan demikian, pengujian diatas 300 *thread* akan selalu mengalami *error* pada layanan *shared hosting*.

Ketika beban ditingkatkan menjadi 1000 hingga 5000 *thread*, kinerja layanan *shared hosting* menurun secara signifikan. *Latency*, *throughput*, *error rate* dan *response time* semuanya menunjukkan bahwa server tidak mampu menampung beban. Sejumlah besar mengalami *error*, bahwa layanan tidak mampu menangani permintaan yang dibuat. Artinya batas maksimal beban pada layanan *shared hosting* 300 pengguna.

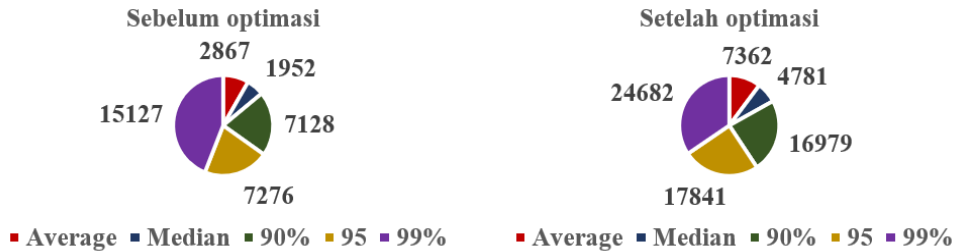
Secara umum, meskipun peningkatan kinerja pada *thread* 10 hingga 300, peningkatan beban dapat mengakibatkan penurunan kinerja layanan *shared hosting*. Untuk mengatasi hal ini, perlu dilakukan penyesuaian lebih lanjut atau mempertimbangkan penggunaan solusi hosting yang lebih kuat untuk menangani beban yang lebih tinggi.

Hasil pengujian sebelum optimasi memperoleh angka yang tinggi. Namun, setelah optimasi, kinerja meningkat secara signifikan, serta penurunan nilai parameter tertentu. Grafik dapat pada gambar 15.



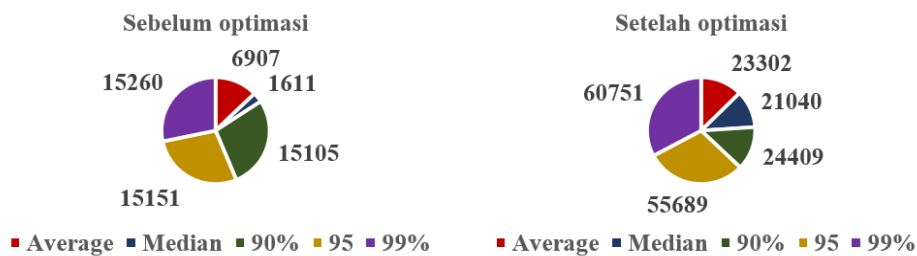
Gambar 15. Grafik 10 Sampai 50 Users

Sebelum dilakukan optimasi, pengujian menunjukkan nilai yang tinggi dengan *average* 2867 dan *median* 1952. Namun masih banyak error yang tidak dicatat sehingga penghitungan hanya berdasarkan nilai keberhasilan. Persentase tertinggi yang mencapai 90%, 95%, dan 99% berturut-turut adalah 7128, 7276, dan 15127. Setelah dilakukan optimasi, terjadi peningkatan performa yang signifikan dengan rata-rata meningkat menjadi 7362 dan *median* menjadi 4781. Persentase tertinggi juga meningkat masing-masing menjadi 16979, 17841, dan 24682. Meskipun data setelah optimasi menunjukkan perbaikan, namun perlu diingat bahwa data sebelumnya banyak dipengaruhi oleh kesalahan yang tidak tercatat, sehingga tidak dapat dikatakan dengan pasti bahwa data sebelum optimasi lebih bagus. Perbandingan grafik pada gambar 16.



Gambar 16 Grafik 100 Sampai 500 Users

Sebelum optimasi, pengujian menghasilkan hasil yang tinggi, dengan rata-rata 6907 dan *median* 1611. Namun, beberapa kegagalan terjadi selama pengujian, oleh karena itu penghitungan hanya didasarkan pada data yang berhasil, dengan kesalahan yang dilaporkan sebesar 0. Setelah pengoptimalan, kinerja meningkat secara signifikan, dengan nilai rata-rata 23302 dan *median* 21040. Meskipun data setelah optimasi menunjukkan peningkatan nilai secara keseluruhan, data sebelumnya tidak dapat dianggap unggul karena banyaknya kesalahan tidak tercatat. Perbandingan grafik pada gambar 17.



Gambar 17. Grafik 1000 Sampai 5000 Users

G. Hasil Penelitian

Penelitian ini melakukan pengujian beban untuk mengetahui batas maksimum akses pengguna secara bersamaan dari *website* jurusan Teknik Informatika Politeknik Negeri Bengkalis. Tiga jenis layanan *server* diuji sebelum dan sesudah optimasi: *localhost*, *shared hosting* dan *vps hosting*. Sebelum dilakukan optimasi, Pengujian awal sebelum perbaikan sistem adalah pengujian beban dengan 10-50 pengguna. Nilai rata-rata untuk setiap *server localhost* adalah 42.6ms, 150.18ms, 15.16s, 0%, dan 150ms. *Shared hosting*: 202 ms, 472 ms, 45 ms, 7,5 s, 0%, dan 1091 ms. *VPS hosting*: 202 ms, 472,45 ms, 7,5 s, 0%, dan 1091 ms. Ketika terdapat lebih dari 50 pengguna, terjadi kesalahan yang menyebabkan perbandingan nilai untuk setiap *server* terhenti. Pengujian berikut dilakukan setelah melakukan optimasi *website* dengan pendekatan merangkum baris kode dan memperkecil ukuran gambar. Hasil pengujian untuk beban 10-50 pengguna menunjukkan bahwa nilainya meningkat drastis, dengan nilai rata-rata *localhost* sebesar 42 ms, 54,8 ms, 15 s, 0%, dan 54,6 mstk. *Shared hosting*: 362 ms, 126,86 ms, 12,7 s, 0%, dan 441 ms. *VPS hosting*: 531 ms, 220 ms, 42 s, 0%, 608 ms. Pengujian dilanjutkan dengan beban hingga 500 pengguna, namun perbandingan nilai yang diperoleh pada setiap *server* dibatasi hingga 300 pengguna karena kegagalan pada beban yang lebih tinggi. Hasil pengujian untuk 100-300 pengguna menunjukkan nilai rata-rata *localhost* sebesar 89ms, 121ms, 69s, 0%, dan 121ms. *Shared hosting*: 623 ms, 2100 ms, 33 s, 0%, 4615 ms. *VPS hosting*: 412 ms, 126 ms, 76 s, 0%. dan 454ms. Dapat disimpulkan bahwa mengoptimasi *website* dapat berpengaruh baik terhadap kinerja *website* dan *server*. Setelah optimasi, hasil dari *load time*, *latency*, *throughput* dan *response time* meningkat secara signifikan

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengujian performa dengan menggunakan metode pengujian beban, dinyatakan dapat mengetahui batas maksimum sistem dalam menampung jumlah pengguna yang akses secara bersamaan. Hal ini dapat dipastikan dengan hasil aspek *load time*, *latency*, *throughput*, *error rate* dan *response time*. Pada pengujian pertama sebelum dilakukan optimasi pada sistem, hasil pengujian beban dengan *users* 10-50. Nilai rata-rata yang diperoleh setiap servernya *localhost* adalah 42,6ms,

150,18ms, 15,16s, 0%, dan 150ms. *Shared hosting* 202ms, 472ms, 45ms, 7,5s, 0%, dan 1091ms. *VPS hosting* 202ms, 472,45ms, 7,5s, 0%, 1091ms. Hasil ini merupakan batas maksimum sistem dalam menampung jumlah pengguna yang akses secara bersamaan, yakni sebanyak 50 users.

Pengujian selanjutnya dilakukan setelah optimasi pada *website* dengan teknik meringkas baris code dan meringankan ukuran gambar. Hasil pengujian pada beban 10-50 users diperoleh mendapatkan nilai meningkat signifikan dengan perolehan nilai rata-rata *localhost* adalah 42ms, 54.8ms, 15s, 0%, dan 54.6ms. *Shared hosting* 362ms, 126,86ms, 12,7s, 0%, dan 441ms. *VPS hosting* 531ms, 220ms, 42s, 0%, 608ms. Pengujian dilanjutkan dengan jumlah beban hingga 500 users, namun perbandingan perolehan nilai pada tiap server hanya pada beban 300 users dikarenakan beban diatas 300 users mengalami kesalahan. Hasil uji 100-300users rata-rata nilai *localhost* 89ms, 121ms, 69s, 0%, dan 121ms. *Shared hosting* 623ms, 2100ms, 33s, 0%, 4615ms. *VPS hosting* 412ms, 126ms, 76s, 0%, dan 454ms. Dapat disimpulkan batas maksimum setelah optimasi *shared hosting* 300users, nilai yang diperoleh meningkat signifikan, hasil optimasi juga berdampak pada perolehan nilai *localhost* dan *VPS hosting*.

Untuk penelitian selanjutnya disarankan dapat dilakukan pengujian dan analisis keamanan basis data dengan alat seperti Acunetix, SQLMap, dan Burp Suite. Dapat digunakan untuk menjalankan pemindaian kerentanan injeksi SQL, mengungkap kerentanan yang ada, dan melakukan pengujian penetrasi aplikasi online. Tujuan ini dapat mengetahui apakah batas maksimum pengujian beban yang dilakukan sebelumnya dikarenakan sistem mengidentifikasi sebagai serangan *bot* atau *ddos*.

DAFTAR PUSTAKA

- [1] I. G. N. Suryantara, *Merancang Aplikasi dengan Metodologi Extreme Programmings*. 2017. [Online]. Available: <https://www.researchgate.net/publication/323906989>
- [2] C. P. Agustika, W. S. Saputra, and M. Idhom, "Pengujian Aplikasi Greenwallet Dengan Metode Load Testing dan Apache Jmeter," *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 2, no. 2, pp. 190–195, 2021.
- [3] R. Khan and M. Amjad, "Web application's performance testing using HP LoadRunner and CA Wily introscope tools," *In 2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 802–806, Jan. 2016, doi: 10.1109/CCAA.2016.7813849.
- [4] K. Zhu, J. Fu, and Y. Li, "Research the Performance Testing and Performance Improvement Strategy in Web Application," *20i 0 2nd international Conference on Education Technology and Computer (ICETC)*, pp. 328–332, 2010.
- [5] W. Tejaya, S. Rahman, and A. Munir, "Pengujian Website Invitees Menggunakan Metode Load Testing Dengan Apache Jmeter," *Kharisma Tech*, vol. 18, no. 1, pp. 99–112, 2023, [Online]. Available: <https://jurnal.kharisma.ac.id/kharismatech/>
- [6] D. I. Permatasari et al., "Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian," *Jurnal Sistem Informasi dan Teknologi*, vol. 8, no. 1, pp. 135–139, 2020.
- [7] G. H. Setiawan, I. M. B. Adnyana, and K. Budiarta, "Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing," *Seminar Nasional CORIS 2022*, pp. 539–542, 2022.
- [8] Fandy, Rosmasari, and G. M. Putra, "Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS)," *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 1, no. 1, pp. 21–35, Jun. 2022, doi: 10.30872/atasi.v1i1.45.
- [9] Fandy, Rosmasari, and G. M. Putra, "Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS)," *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 1, no. 1, pp. 21–35, Jun. 2022, doi: 10.30872/atasi.v1i1.45.
- [10] A. Suprpto and D. Sasongko, "Evaluasi Performa Website Berdasarkan Pengujian Beban Dan Stress Menggunakan Loadimpact (Studi Kasus Website Iain Salatiga)," *Network Engineering Research Operation*, vol. 6, no. 1, pp. 31–37, 2021, [Online]. Available: <https://iainsalatiga.ac.id>
- [11] A. C. Barus, J. Harungguan, and E. Manulu, "Pengujian API Website Untuk Perbaikan Performansi Aplikasi Ditenun," *Journal of Applied Technology and Informatics*, vol. 1, no. 3, pp. 14–21, 2021.
- [12] D. Ferdiansyah, A. Riaunanda Kamal, S. Alas, and F. Mulyanto, "Pengujian Multicore Pada Processor Terhadap Performansi Server Virtualisasi Menggunakan Metode Load Testing," *Journal of Information Systems and Informatics*, vol. 3, no. 4, 2021, [Online]. Available: <http://journal-isi.org/index.php/isi>
- [13] I. M. E. Listartha, "Pengujian Performa dan Tingkat Stress Pada Website Legalisir Ijasah Online Universitas Pendidikan Ganesha," *Jurnal Elektro Luceat*, vol. 6, no. 1, 2020.
- [14] "5 Load Testing Scenarios for Your API," <https://www.soapui.org/learn/load-testing/5-load-testing-scenarios-for-your-api/>.
- [15] D. I. Permatasari, B. Santoso, and N. Ningtias, "Pengukuran Throughput Load Testing Menggunakan Test Case Sampling Gorilla Testing," *Seminar Nasional Sistem Informasi (SENASIF)*, vol. 3, pp. 2008–2014, 2019.
- [16] N. P. K. Widyani, A. A. K. O. Sudana, and I. N. Piarsa, "Pengujian Performa Sistem Informasi Perpustakaan Online pada Universitas Hindu Indonesia (Astakali UNHI) Menggunakan Tools GTmetrix," *Jurnal Ilmiah Teknologi dan Komputer*, vol. 2, no. 3, 2021, [Online]. Available: <https://GTmetrix.com>.
- [17] C. Bird, "What Is a Good Page Load Time for SEO — How Fast Is Fast Enough?," https://www.semrush.com/blog/page-speed/?kw=&cmp=AA_SRCH_DSA_Blog_Core_BU_EN&label=dsa_pagefeed&Network.
- [18] D. Andriansyah, "Performance dan Stress Testing Dalam Mengoptimasi Website," *Computer Based Information System Journal*, vol. 7, no. 1, pp. 23–28, 2019, [Online]. Available: <http://ejournal.upbatam.ac.id/index.php/cbishttp://ejournal.upbatam.ac.id/index.php/cbis>
- [19] F. J. Wangsa, Marlina, and Renny, "Optimasi Website Toko Kerja Menggunakan Uji Performa Google Pagespeed Insights," *KHARISMA Tech*, vol. 18, no. 2, pp. 41–54, 2023, [Online]. Available: <https://tech.kharisma.ac.id>
- [20] A. P. D. G. Andini, D. Wahyuningsih, and M. Yunus, "Analisis Dan Peningkatan Performa Aplikasi Berbasis Website Menggunakan Stress Tools Gtmetrix," *TEMATIK*, vol. 9, no. 2, pp. 191–201, Dec. 2022, doi: 10.38204/tematik.v9i2.1071.
- [21] A. I A A A, "Review Teknik Optimasi Performa pada Front End Website," *Universitas Muhammadiyah Malang*, 2021.
- [22] Y. T. Arumoad, W. L. Y.S, and T. Susyanto, "Optimasi Kinerja Mobile Website Dengan Teknik Front-End Optimization Pada Toko Online Imperial Parfum," *Jurnal TIKomSiN*, vol. 2, no. 2, pp. 53–59, 2014.