

# PERFORMANCE TESTING SISTEM UJIAN ONLINE MENGUNAKAN JMETER PADA LINGKUNGAN VIRTUAL

Ade Ismail<sup>1</sup>, Ahmadi Yuli Ananta<sup>2</sup>, Sofyan Noor Arief<sup>3</sup>, Elok Nur Hamdana<sup>4</sup>

<sup>1,2,3</sup> Jurusan Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>aismail@polinema.ac.id, <sup>2</sup>ahmadi@polinema.ac.id, <sup>3</sup>[sofyan@polinema.ac.id](mailto:sofyan@polinema.ac.id), <sup>4</sup>elokhamdana@polinema.ac.id

## Abstrak

*Performance testing* adalah proses pengujian perangkat lunak yang digunakan untuk menguji kecepatan, waktu *respons*, *stabilitas*, *keandalan*, skalabilitas, dan penggunaan *resource* perangkat lunak dibawah beban kerja tertentu. Semakin tinggi nilai yang didapat dari hasil pengujian *performance testing* akan semakin baik bagi *server* dalam menangani *request*. *Tools* yang digunakan dalam pengujian adalah Apache Jmeter dengan *test plan* yang direncanakan seperti aktivitas pengguna dalam keadaan sebenarnya seperti *login*, membuka, mengerjakan dan menjawab soal *test*, kemudian Kembali ke *dashboard* dan *logout* dari sistem. Penelitian ini membantu menentukan bagaimana *respons* perangkat lunak ketika beberapa pengguna mengakses sistem secara bersamaan. Pengujian menggunakan *load testing* yang mensimulasikan beban pengguna yang sebenarnya pada perangkat lunak. Pengujian bertujuan mencari tau berapa banyak *pengguna* yang dapat ditangani oleh sistem dalam konfigurasi tertentu. Perangkat lunak ujian online yang digunakan adalah TCEXam yang merupakan sistem ujian online berbasis komputer yang banyak digunakan oleh institusi pendidikan. Dalam pengujian ini target pengguna yang harus bisa ditangani oleh perangkat lunak sebanyak 1000 *pengguna* dan dilakukan pembatasan spesifikasi *server*. Penyesuaian *resource* dibutuhkan untuk mengetahui *resource* ideal yang dibutuhkan bagi 1000 *pengguna*. Konfigurasi dilakukan dengan model virtualisasi sehingga memudahkan dalam melakukan pengaturan spesifikasi server berdasarkan kebutuhan pengujian. Hasil pengujian *performance testing* menggunakan Apache Jmeter menunjukkan bahwa dengan spesifikasi paling tinggi saja masih terdapat error dalam load testing sehingga target 1000 pengguna belum dapat terpenuhi.

**Kata kunci** : performance testing, load testing, virtualisasi, Jmeter.

## 1. Pendahuluan

Ujian merupakan cara terbatas untuk mengukur kemampuan seseorang. Pelaksanaan ujian dimaksudkan untuk mengukur pengetahuan seseorang atau peserta didik dan digunakan sebagai alat evaluasi untuk menilai berapa jauh pengetahuan sudah dikuasai dan keterampilan yang sudah diperoleh (Bernadette, 2005).

Sistem ujian online merupakan sistem ujian berbasis web yang dilaksanakan secara online atau menggunakan komputer dengan akses internet (Andika et al., 2017). Dalam pelaksanaannya, ujian online berbasis web ini ternyata tidak semudah kelihatannya, butuh banyak persiapan baik dari infrastruktur, software dan aturan-aturan mengenai penggunaan sistem ujian ini. Untuk itu rencana penambahan *resource server* menjadi salah satu langkah tepat yang dapat diambil jika ingin tetap menggunakan sistem ujian berbasis online ini.

Namun pada kenyataannya penambahan *resource server* belum tentu langsung dapat meningkatkan kinerja dari sistem ujian online. Untuk itu perlu dilakukan analisa mengenai kinerja sistem ujian online baik sebelum ditambahkan *resource server* maupun setelah ditambahkan agar penggunaan *server* menjadi lebih optimal dan sesuai dengan yang

diharapkan. Untuk itu diperlukan perencanaan yang baik dalam mengimplementasikan teknologi ini. Jika teknologi yang digunakan sesuai maka penggunaan sistem ujian online ini akan memberikan dampak *behavioral intention*, *user satisfaction* dan *net benefit* (Andhica & Irwan, 2018).

Menguji program berarti menjalankannya dengan serangkaian input yang dipilih dan memeriksa apakah programnya berperilaku dengan cara yang diharapkan; jika tidak, maka bug telah terdeteksi. Salah satu penyebabnya munculnya bug adalah karena program dibuat oleh manusia yang dimana mungkin terdapat *missunderstood* terhadap kebutuhan perangkat lunak (Fraser & Rojas, 2019). Tujuan dari pengujian adalah untuk menemukan bug sebanyak mungkin, tetapi ini adalah tugas yang sulit karena tidak mungkin untuk menjalankan semua kemungkinan tes pada suatu program (Alankar et al., 2020).

Pengujian berbeda dengan *debugging*. Pengujian lebih dari sekadar *debugging*. Pengujian tidak hanya digunakan untuk menemukan cacat dan memperbaikinya, ini juga digunakan di validasi, proses verifikasi dan pengukuran (Permatasari et al., 2019). Pengujian juga dilakukan sebagai bentuk menentukan kualitas dan keandalan produk (Barraod et al., 2021)

Teknik Pengujian Perangkat Lunak dapat dibagi menjadi dua jenis(Lewis, 2016): yaitu pengujian secara manual dan pengujian secara otomatis. Pengujian secara otomatis juga disebut sebagai pengujian dinamis yang bertujuan salah satunya menguji performa sistem.

TCEXAM adalah sistem Open Source untuk ujian elektronik (juga dikenal sebagai CBA - *Computer-Based Assessment*, CBT - *Computer-Based Testing* atau e-exam) yang memungkinkan pendidik dan pelatih untuk menulis, menjadwalkan, menyampaikan, dan melaporkan kuis, tes dan ujian(TCEXAM, n.d.).Penggunaan TCEXAM sebagai aplikasi ujian online berbasis web juga dapat meningkatkan efektivitas bagi siswa dalam melakukan evaluasi pembelajarannya jika dibandingkan dengan sistem tradisional seperti ujian berbasis kertas(Ambiyar et al., 2019).

Dalam penelitian ini akan dilakukan pengujian *performance testing* terhadap kinerja server. model pengujian menggunakan *load testing* yang mensimulasikan beban pengguna yang sebenarnya pada perangkat lunak. Pengujian bertujuan mencari batas kemampuan *server* dalam menganani banyak pengguna dalam konfigurasi tertentu. Apache Jmeter dengan *test plan* yang direncanakan seperti aktivitas pengguna dalam keadaan sebenarnya seperti *login*, membuka, mengerjakan dan menjawab soal *test*, kemudian Kembali ke *dashboard* dan *logout* dari sistem. Hasil dari *performance testing* akan menjadi acuan untuk konfigurasi pengetesan berikutnya. Target pengguna yang harus bisa ditangani ditargetkan sebanyak 1000 pengguna. Konfigurasi dilakukan dengan model virtualisasi sehingga memudahkan dalam melakukan pengaturan spesifikasi *server* berdasarkan kebutuhan pengujian.

## 2. Tinjauan Literatur

Pada penelitian sebelumnya yang berkaitan dengan performa dari sebuah aplikasi berbasis web, dikatakan bahwa performa dari aplikasi berbasis web adalah waktu nilai *response time* dan nilai *throughput* dari sebuah *request*, semakin kecil nilai *response time* dan semakin besarnya nilai *throughput* maka kinerja dari aplikasi web akan semakin baik(Andhica & Irwan, 2018). Performa yang baik ini tentu akan mengoptimalkan penggunaan aplikasi dari sisi kecepatan akses. Kecepatan akses yang baik dapat membuat pengguna menjadi lebih nyaman saat menggunakan aplikasi(Mardiana, 2020).

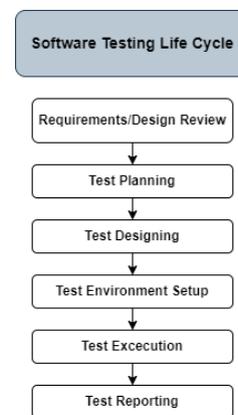
Keterbatasan *resource* dalam pengujian performa aplikasi seringkali menjadi kendala sehingga hasil pengujian biasanya terbatas, dalam penelitian ini kami menggunakan lingkungan virtual yang *resource* bisa disesuaikan sehingga bisa menerapkan beberapa model dari pengujian, salah satunya adalah penggunaan *load balance* sebagai pengatur distribusi *traffic* dair beberapa server. Penggunaan *load balance* ini dapat meningkatkan

*performance* aplikasi web dengan memanfaatkan *resource*(Mbarek & Mosorov, 2018).

Aplikasi Jmeter merupakan aplikasi pengujian kinerja yang fleksibel yang dapat mensimulasikan banyak pengguna aplikasi secara bersamaan sehingga dapat dilakukan evaluasi terhadap kinerja aplikasi(Apache, 2014). Pengujian dengan jumlah pengguna yang dapat disesuaikan ini akan sangat berguna bagi organisasi karena dapat menggambarkan kondisi yang sebenarnya(Wang & Wu, 2019).Pada pengujiannya, aplikasi seperti Jmeter dirancang untuk menangkap kinerja pada tingkat antar muka pengguna dan sistem itu sendiri. Menggunakan aplikasi seperti ini dapat membantu menghemat waktu dan biaya organisasi(Agnihotri & Phalnikar, 2018).

## 3. Metode Penelitian

Penelitian ini dilakukan dengan tahapan pengujian seperti ditunjukkan pada gambar 1. Dimana sebelum pengujian dilakukan, diperlukan beberapa persiapan pengujian seperti menentukan tujuan dan rencana pengujian, menyiapkan lingkungan pengujian dimana lingkungan berupa virtual server serta menyiapkan skenario pengujian dari beban dan pengguna. Selanjutnya dilakukan konfigurasi dari aplikasi Jmeter dan konfigurasi sampler. Hasil analisis didapatkan dari hasil pengujian. Langkah terakhir adalah dokumntasi hasil pengujian.



Gambar 1. Tahapan Pengujian

Dalam pengujian ini disusun beberapa skenario rancangan model *server* yang akan digunakan dalam pengujian. Pembuatan model ini bertujuan untuk membatasi spesifikasi mesin virtual sehingga pengujian ini lebih bisa fokus untuk dilakukan. Rancangan model *server* dapat dilihat pada tabel 1.

Tabel 1. Rancangan model server

Model Server	Keterangan	Spesifikasi Server
Model A	Server Existing sebelum dilakukan penambahan resource	2 Core CPU dan 2GB Memory

<b>Model B</b>	Server Existing setelah dilakukan penambahan resource	8 Core CPU dan 8GB Memory
<b>Model C</b>	Server lain yang dibuat dengan melakukan tuning pada DBMS	8 Core CPU dan 8GB Memory

Pengujian ini juga membatasi aktivitas pengguna dalam menggunakan sistem ujian online. Aktivitas yang dilakukan pengguna disimulasikan seperti peserta ujian yang akan melaksanakan ujian seperti : login, membuka halaman *dashboard*, melihat list ujian yang tersedia, melaksanakan ujian, menghentikan ujian, kembali ke *dashboard*, dan *logout* dari sistem. Total ada 9 aktivitas *pengguna* dalam pengujian ini.

Selanjutnya disusun skenario *pengguna* dan waktu, dalam pengujian ini menggunakan skenario 1000 pengguna dengan waktu pengujian sebanyak 100 detik. Artinya dalam 100 detik akan ada 100 pengguna yang coba melakukan *request*. Ini berarti setiap *pengguna* memiliki waktu 0,1 detik untuk melakukan *request* ke sistem ujian online. Waktu 100 detik adalah waktu maksimal dimana dalam ujian online keterlambatan waktu diatas 2 menit menjadi kerugian bagi peserta ujian.

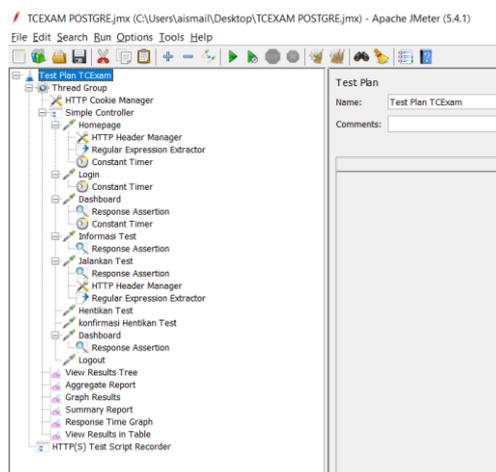
Target pengujian sistem ujian online ini adalah mencari *throughput*, *response time*, dan *resource utilization* seperti *cpu* dan *memory* yang terpakai saat pengujian dilakukan. *Throughput* mewakili kemampuan server untuk menangani beban berat. Semakin tinggi *Throughput*, semakin baik kinerja server (Permatasari et al., 2019). Pengujian juga membandingkan nilai *error* pada setiap model *server* yang diujikan.

#### 4. Hasil dan Pembahasan

##### 4.1 Membuat Test Plan

Test plan yang dibuat untuk pengujian ini terdiri dari 9 sampler *HTTP Request* yang mencerminkan aktivitas yang dilakukan oleh pengguna dalam menggunakan sistem ujian online. Sampler *HTTP Request* ini diantaranya: *Homepage*, *Login*, *Dashboard*, *Informasi Test*, *Jalankan Test*, *Hentikan Test*, *Konfirmasi Hentikan Test*, *Dashboard*, dan *Logout*. Sampler-sampler ini seperti yang ditunjukkan gambar 2.

Terdapat beberapa pengaturan kunci yang menjadikan *Jmeter* ini dapat bertindak layaknya pengguna sungguhan. Pengaturan ini diantaranya adalah pengaturan alamat *host* tujuan, metode *request*, *response assertion* dan *regular expression extractor*.



Gambar 2. Test Plan *Jmeter* untuk sistem ujian online.

##### 4.2 Menjalankan Pengujian

Pengujian dilakukan mulai dari model A,B, dan C. Setelah dilakukan penyesuaian spesifikasi *server* berdasarkan model masing-masing. Hasil dari pengujian untuk model A dapat dilihat pada tabel 2, table 3 dan table 4.

Tabel 2. Hasil Pengujian Model A

Request	Execution	Response Times (ms)		Throughput
Label	#Samples	Error %	Average	Transaction/s
<b>Total</b>	10022	38.45 %	11907.43	41.97
<b>Dashboard 1</b>	1000	75.70 %	18540.64	5.11
<b>Dashboard 2</b>	1000	51.70 %	8884.33	5.09
<b>Hentikan Test</b>	1000	9.00%	6258.08	5.09
<b>Hentikan Test-0</b>	1	0.00%	116.00	8.62
<b>Hentikan Test-1</b>	1	0.00%	35.00	28.57
<b>Homepage</b>	1000	34.40 %	13915.78	7.66
<b>Informasi Test</b>	1000	65.70 %	15381.69	5.00
<b>Jalankan Test</b>	1000	58.00 %	9484.81	5.05
<b>konfirmasi Hentikan Test</b>	1000	3.70%	3106.81	5.09
<b>Login</b>	1000	66.90 %	26869.63	5.78
<b>Login-0</b>	510	0.00%	14189.29	3.35
<b>Login-1</b>	510	35.10 %	17454.00	2.95
<b>Logout</b>	1000	2.30%	756.26	5.41

Tabel 3. Hasil Pengujian Model B

Request	Execution	Response Times (ms)		Throughput
Label	#Samples	Error %	Average	Transaction/s
<b>Total</b>	10390	21.94 %	6371.71	58.26

Dashboard 1	1000	48.10 %	12188 .14	6.78
Dashboard 2	1000	31.30 %	3606. 74	7.08
Hentikan Test	1000	0.30%	228.0 4	7.07
Homepage	1000	21.00 %	9566. 26	7.87
Informasi Test	1000	35.60 %	6318. 03	6.85
Jalankan Test	1000	31.10 %	1031. 95	7.05
konfirmasi Hentikan Test	1000	0.10%	104.8 8	7.07
Login	1000	45.40 %	19897 .62	6.76
Login-0	695	0.00%	7957. 04	5.04
Login-1	695	21.44 %	10957 .50	4.71
Logout	1000	0.20%	114.7 9	7.36

Tabel 4. Hasil Pengujian Model C

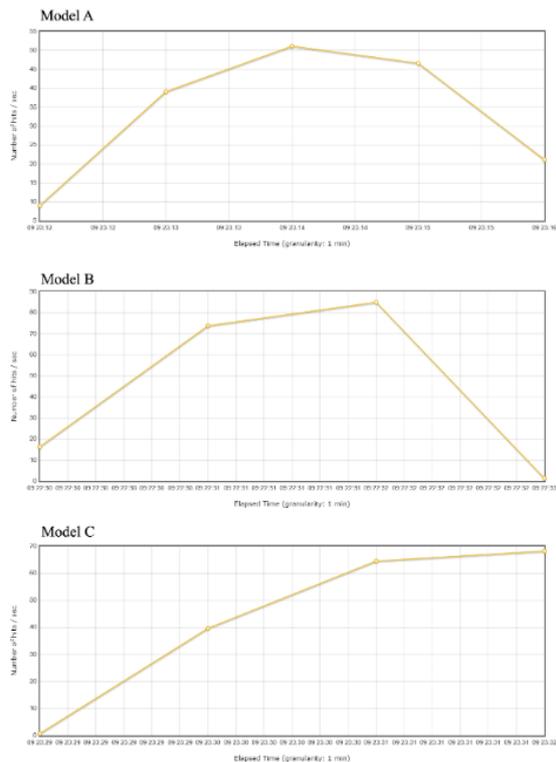
Request Label	Execution #Samples	Response Times (ms)		Throughput Transaction/s
		Error %	Average	
Total	10594	10.88 %	3195. 75	67.81
Dashboard 1	1000	20.60 %	4979. 75	7.98
Dashboard 2	1000	20.50 %	3114. 41	8.03
Hentikan Test	1000	0.00%	88.52	8.04
Homepage	1000	13.10 %	7215. 19	8.33
Informasi Test	1000	20.50 %	3797. 24	7.97
Jalankan Test	1000	20.30 %	166.3 7	8.03
konfirmasi Hentikan Test	1000	0.00%	30.38	8.04
Login	1000	20.30 %	9359. 49	7.98
Login-0	797	0.00%	6295. 96	6.36
Login-1	797	0.00%	15.43	6.37
Logout	1000	0.00%	74.28	8.31

4.3 Analisa Hasil Pengujian

Selanjutnya dari hasil pengujian ini dilakukan analisa hasil pengujian berupa troughput, response time, dan resource utilization

a. Analisa Troughput

Troughput adalah jumlah data yang benar-benar terkirim pada waktu tertentu. Dari hasil pengujian yang sudah dilakukan. didapatkan troughput pada setiap model yang dapat dilihat pada gambar 3.



Gambar 3. Hasil Pengujian Troughput

Dari ketiga hasil tersebut di ketahui bahwa troughput untuk model C memiliki nilai *troughput* yang paling tinggi diangka 85,03 hit/sec dimana ini berarti *server* dapat menangani *request* dari pengguna sebanyak 85,03 per detik. Dan untuk model B nilai *troughput*nya didapat pada angka 68,23 hit/sec yang berarti *server* dapat menangani *request* dari pengguna sebanyak 68,23 per detik. Sedangkan untuk model A memiliki nilai *troughput* terkecil dengan angka angka 51,2 hit/sec yang berarti *server* dapat menangani *request* dari pengguna sebanyak 51,2 per detik.

Dengan spesifikasi *server* yang ditingkatkan sebanyak 4 kali dari *resource* awal, peningkatan performance yang terjadi pada model yang sama jika dilihat dari *troughput* yang dicapai adalah meningkat sekitar 33,26 %, kenaikan ini tidak sebanding dengan *resource* yang ditingkatkan. Setelah dilakukan konfigurasi tambahan berupa mengganti DBMS yang sebelumnya menggunakan mysql menjadi postgresql, terjadi peningkatan sebesar 66.07% dari kondisi awal sebelum penambahan. Hasil ini lebih baik jika dibandingkan dengan kondisi tanpa dilakukan konfigurasi.

b. Analisa Response Time

Response time adalah waktu tanggap yg diberikan oleh antar aplikasi ketika pengguna merequest/ mengirim permintaan ke server. Dari hasil pengujian yang sudah dilakukan, didapatkan *response time* pada setiap model yang dapat dilihat pada gambar 4.



Gambar 4. Hasil Pengujian Response Time

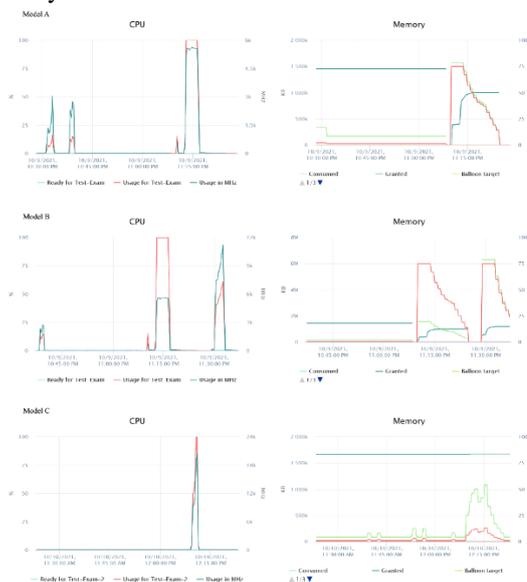
Dari ketiga hasil tersebut di ketahui bahwa *response time* untuk model A memiliki nilai *response time*  $\leq 500ms$  sebanyak 1533 *responses*,  $> 500ms \leq 1,500 ms$  sebanyak 191 *responses*,  $> 1,500ms$  sebanyak 4445 *responses*, dan 3853 *responses error*. Model B memiliki nilai *response time*  $\leq 500ms$  sebanyak 3893 *responses*,  $> 500ms \leq 1,500 ms$  sebanyak 121 *responses*,  $> 1,500ms$  sebanyak 4096 *responses*, dan 2280 *responses error*. Kemudian model C memiliki nilai *response time*  $\leq 500ms$  sebanyak 4989 *responses*,  $> 500ms \leq 1,500 ms$  sebanyak 170 *responses*,  $> 1,500ms$  sebanyak 4282 *responses*, dan 1153 *responses error*. Nilai Model C ini menjadi nilai yang paling besar jika dibandingkan dengan model A dan B, meskipun masih terdapat *responses error* dalam model ini.

**c. Analisa Resource Utilization**

*Resource utilization* adalah jumlah *resource* yang terpakai pada saat pengujian. *Statistic* ini menentukan seberapa tinggi *hardware* bekerja dalam setiap model. Dari pengujian ini didapatkan *resource utilization* pada setiap model yang ditunjukkan pada gambar 5.

Dari ketiga hasil tersebut di ketahui bahwa *resource utilization* pada model A adalah yang tertinggi dengan penggunaan CPU mencapai 100% dan memory mencapai 75% selama pengujian berlangsung. Pada model B penggunaan CPU mencapai 60% dan memory mencapai 75% selama pengujian berlangsung. Sedangkan pada model C,

penggunaan CPU mencapai 100% dan memory sebanyak 50%.



Gambar 5. Hasil Pengujian Resource Utilization

Dari hasil ini bisa dilihat perbandingan antara model A dan model B dimana aplikasi masih menggunakan teknologi server yang sama, namun *resource* yang berbeda. *Resource* CPU yang digunakan pada model B masih memenuhi kebutuhan pengujian untuk 1000 pengguna, walaupun terdapat nilai error yang cukup besar. Sedangkan pada perbandingan antara model B dan C adalah dengan menggunakan teknologi yang berbeda pada setiap server, didapat hasil berupa berkurangnya *error*, namun terjadi peningkatan penggunaan *resource* CPU yang mencapai 100%.

**5. Kesimpulan**

Berdasarkan hasil analisa yang sudah dilakukan penambahan *resource server* yang dilakukan ternyata tidak berdampak signifikan terhadap peningkatan kinerja sistem ujian online jika tidak dilakukan pengaturan tambahan pada teknologi aplikasi. Adanya batasan pada *hardware* menjadikannya perlu untuk mempertimbangkan penggunaan pembagi beban komputasi (*load balancer*) dan memisahkan server database dengan server aplikasi. Dengan menerapkan arsitektur yang modular (pemisahan antara server aplikasi dengan *server database*) serta menambahkan pembagi beban komputasi. Dengan mengganti aplikasi *database* saja, jumlah error request yang muncul menjadi semakin kecil (mencapai 10,88% untuk 1000 request). Peningkatan *throughput server* dapat dicapai dengan menurunkan jumlah *error request* yang terjadi. Target pengujian untuk 1000 pengguna belum tercapai.

**Daftar Pustaka**

Agnihotri, J., & Phalnikar, R. (2018). Development of performance testing suite using apache

- JMeter. *Advances in Intelligent Systems and Computing*. [https://doi.org/10.1007/978-981-10-7245-1\\_32](https://doi.org/10.1007/978-981-10-7245-1_32)
- Alankar, B., Sharma, G., Kaur, H., Valverde, R., & Chang, V. (2020). Experimental setup for investigating the efficient load balancing algorithms on virtual cloud. *Sensors (Switzerland)*, 20(24). <https://doi.org/10.3390/s20247342>
- Ambiyar, Hamzah, M. L., Purwati, A. A., & Saputra, E. (2019). Computer based test using tcexam as an instrument learning evaluation. *International Journal of Scientific and Technology Research*.
- Andhica, I. Y., & Irwan, D. (2018). Performa Kinerja Web Server Berbasis Ubuntu Linux Dan Turnkey Linux. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 5(2). <https://doi.org/10.33558/piksel.v5i2.269>
- Andika, E., Djajasukma, D., & Heryanto, H. (2017). Analisis Manfaat Penerapan Sistem Informasi Ujian Online: Studi Kasus SMK Pasim Plus. *Jurnal Teknologi Rekayasa*, 2(1), 47. <https://doi.org/10.31544/jtera.v2.i1.2017.47-54>
- Apache. (2014). Apache JMeter - Apache JMeter™. In *Apache JMeter*.
- Barraood, S. O., Mohd, H., & Baharom, F. (2021). Test Case Quality Factors: Content Analysis of Software Testing Websites. *Webology*, 18(SpecialIssue). <https://doi.org/10.14704/WEB/V18SI01/WEB18007>
- Bernadette, T. (2005). *Melatih anak berpikir seperti jenius: menemukan dan mengembangkan bakat yang ada pada setiap anak*. Gramedia Pustaka Utama.
- Fraser, G., & Rojas, J. M. (2019). Software testing. In *Handbook of Software Engineering*. [https://doi.org/10.1007/978-3-030-00262-6\\_4](https://doi.org/10.1007/978-3-030-00262-6_4)
- Lewis, W. E. (2016). Software testing and continuous quality improvement: Third edition. In *Software Testing and Continuous Quality Improvement: Third Edition*.
- Mardiana, M. (2020). Implementasi User Satisfaction Model Dalam Mengukur Kualitas Website. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 19(2). <https://doi.org/10.30812/matrik.v19i2.711>
- Mbarek, F., & Mosorov, V. (2018). Load balancing algorithms in heterogeneous web cluster. *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*. <https://doi.org/10.1109/IIPHDW.2018.8388358>
- Permatasari, D., Santoso, B., ... N. N.-S. N., & 2019, undefined. (2019). Pengukuran throughput load testing menggunakan test case sampling gorilla testing. *Jurnalfti.Unmer.Ac.Id*, 4. <https://www.jurnalfti.unmer.ac.id/index.php/nasif/article/view/268>
- TCEXAM. (n.d.). Retrieved January 23, 2021, from [tcexam.org](http://tcexam.org)
- Wang, J., & Wu, J. (2019). Research on performance automation testing technology based on JMeter. *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*. <https://doi.org/10.1109/ICRIS.2019.00023>