

TESIS
ANALISIS PERBANDINGAN APACHE JMETER DAN
LOCUST DALAM PENGUJIAN PERFORMA API SERVICE



disusun oleh

RAHMAT PRASTYO

22.55.1248

Konsentrasi : Digital Transformation Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2025

TESIS
ANALISIS PERBANDINGAN APACHE JMETER DAN
LOCUST DALAM PENGUJIAN PERFORMA API SERVICE

COMPARATIVE ANALYSIS OF APACHE JMETER AND
LOCUST IN API SERVICE PERFORMANCE TESTING

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana
Program Studi *PJJ Informatika*



disusun oleh

RAHMAT PRASTYO

22.55.1248

Konsentrasi : Digital Transformation Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2025

HALAMAN PERSETUJUAN

**ANALISIS PERBANDINGAN APACHE JMETER DAN LOCUST DALAM
PENGUJIAN PERFORMA API SERVICE**

**COMPARATIVE ANALYSIS OF APACHE JMETER AND LOCUST IN
API SERVICE PERFORMANCE TESTING**

yang disusun dan diajukan oleh

Rahmat Prastyo

22.55.1248

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 01 Desember 2025

Dosen Pembimbing,



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PENGESAHAN

**ANALISIS PERBANDINGAN APACHE JMETER DAN LOCUST DALAM
PENGUJIAN PERFORMA API SERVICE**

**COMPARATIVE ANALYSIS OF APACHE JMETER AND LOCUST IN
API SERVICE PERFORMANCE TESTING**

yang disusun dan diajukan oleh

Rahmat Prastyo

22.55.1248

Telah dipertahankan di depan Dewan Penguji
pada tanggal 01 Desember 2025

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Tonny Hidayat, S.Kom., M.Kom., Ph.D.
NIK. 190302182

Dr. Ferry Wahyu Wibowo, S.Si., M.Cs.
NIK. 190302235

Prof. Dr. Kusriani, S.Kom., M.Kom
NIK. 190302106



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 01 Desember 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Rahmat Prastyo
NIM : 22.55.1248

Menyatakan bahwa Tesis dengan judul berikut:

Analisis Perbandingan Apache JMeter Dan Locust Dalam Pengujian Performa API Service

Dosen Pembimbing : Prof. Dr. Kusrini, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 01 Desember 2025

Yang Menyatakan,



Rahmat Prastyo

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT karena berkat Rahmat dan Hidayah-Nya sehingga saya diberi kekuatan dan kemampuan untuk bisa menyelesaikan tesis ini dengan baik. Tesis ini saya persembahkan untuk :

1. Kedua orang tua terkasih Bapak Ibnu Hajar (alm.) dan Ibu Mardiah, S.IP, terima kasih banyak telah memberikan doa, nasehat dan juga dukungan yang tidak terhingg.
2. Kedua mertua Bapak Ujang dan Ibu Imelyati, S.Pd.SD, terima kasih telah memberikan dukungan kepada saya dan mendoakan serta memberikan nasehat kepada saya.
3. Ibu Prof.Dr.Kusrini, M.Kom., dan Bapak Kusnawi, S.Kom., M.Eng., selaku dosen pembimbing saya yang telah memberikan arahan, motivasi, serta bimbingan dalam penyelesaian tesis ini sehingga tesis ini dapat terselesaikan dengan baik.
4. Istri tercinta Gustina Adela Putri, S.E., M.E., terima kasih telah memberikan dukungan yang tiada henti, motivasi serta kesabaran dalam setiap Langkah menyelesaikan Tesis ini.
5. Untuk kedua kakak Fina Nurmita, S.Pd., M.Pd.Mat., dan Muhammad Anggi Pratama, S.Sos., M.A.P., serta kedua adik Andri Atmajaya, S.T., dan Qaisyah Nur Arifah, terima kasih telah mendoakan serta mendukung untuk kelancaran dalam menyelesaikan tesis ini.
6. Serta semua pihak yang telah membantu dan mendukung untuk menyelesaikan tesis ini yang tidak dapat saya sebut satu persatu.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tesis berjudul **“Analisis Perbandingan Apache JMeter dan Locust dalam Pengujian Performa API Service”** sebagai salah satu syarat memperoleh gelar Magister pada Program Studi S2 Informatika, Program Pascasarjana Universitas AMIKOM Yogyakarta.

Penulisan tesis ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. M. Suyanto, MM, selaku Rektor Universitas Amikom Yogyakarta
2. Prof. Dr. Kusriani, M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Amikom Yogyakarta dan juga sebagai dosen pembimbing utama
3. Bapak Kusnawi, S.Kom., M.Eng., selaku Pembimbing Pendamping
4. Tim Dosen Penguji dan semua Dosen Pascasarjana PJJ MTI Universitas Amikom Yogyakarta.
5. Orang tua, Saudara dan Istri Tercinta
6. Pihak lain yang telah memberikan bantuan dan dukungan selama penyusunan tesis ini.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan, sehingga kritik dan saran yang membangun sangat diharapkan demi penyempurnaan karya ini. Semoga tesis ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan.

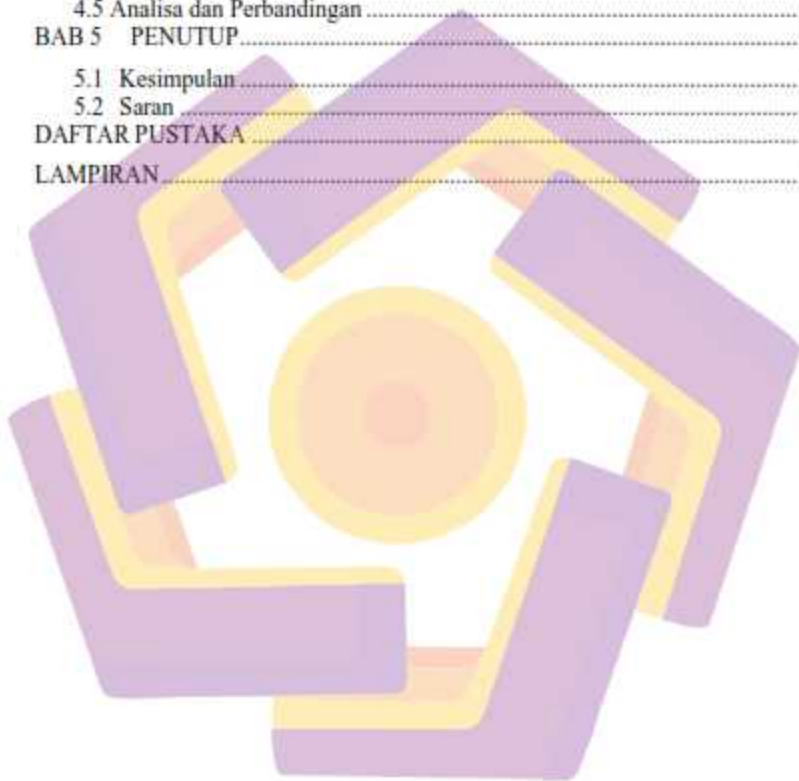
Yogyakarta, 12 Desember 2025

Penulis

DAFTAR ISI

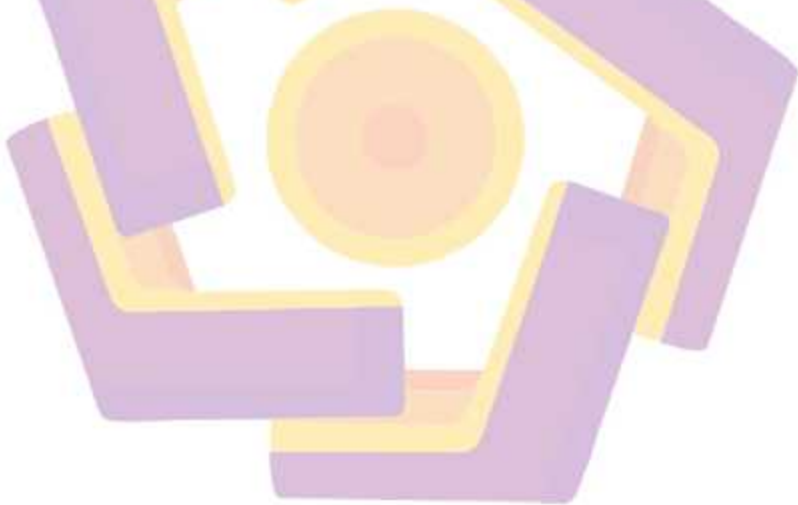
HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN KEASLIAN TESIS.....	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xi
INTISARI.....	xii
ABSTRACT.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian.....	6
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 Tujuan Pustaka.....	7
2.2 Keaslian Penelitian.....	12
2.3 Landasan Teori.....	17
BAB 3 METODE PENELITIAN.....	33
3.1 Jenis, Sifat, dan Pendekatan Penelitian.....	33
3.2 Metode Pengumpulan Data.....	34
3.3 Metode Analisis Data.....	34
3.4 Alur Penelitian.....	36
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN.....	40
4.1 Pengumpulan Data.....	40
4.2 Penyajian Data.....	40
4.3 Apache JMeter.....	42
4.3.1 Skenario Pengujian Apache JMeter.....	43

4.3.2	Eksekusi Apache JMeter.....	47
4.3.3	Analisis Hasil Apache JMeter.....	49
4.4	Locust.....	53
4.4.1	Skenario Pengujian Locust.....	54
4.4.2	Eksekusi Skenario pengujian performance dengan Locust.....	59
4.4.3	Analisis Hasil Locust.....	60
4.5	Analisa dan Perbandingan.....	65
BAB 5	PENUTUP.....	75
5.1	Kesimpulan.....	75
5.2	Saran.....	77
DAFTAR PUSTAKA.....		78
LAMPIRAN.....		82



DAFTAR TABEL

Tabel 2. 1 Matriks literatur review dan posisi penelitian.....	12
Tabel 4. 1 API Service	41
Tabel 4. 2 Mean Hasil Pengujian Apache JMeter	49
Tabel 4. 3 Standard Deviation Apache JMeter	51
Tabel 4. 4 Mean Hasil Pengujian Locust	60
Tabel 4. 5 Standard Deviation Locust.....	63
Tabel 4. 6 Hasil uji Wilcoxon pada API Login.....	69
Tabel 4. 7 Hasil uji Wilcoxon pada API Status Login	70
Tabel 4. 8 Hasil uji Wilcoxon pada API List Jadwal Ujian	70
Tabel 4. 9 Hasil uji Wilcoxon pada API Mulai Ujian	71
Tabel 4. 10 Hasil uji Wilcoxon pada API Daftar Soal Ujian	72
Tabel 4. 11 Hasil uji Wilcoxon pada API Soal Ujian	72
Tabel 4. 12 Hasil uji Wilcoxon pada API Submit Jawaban	73

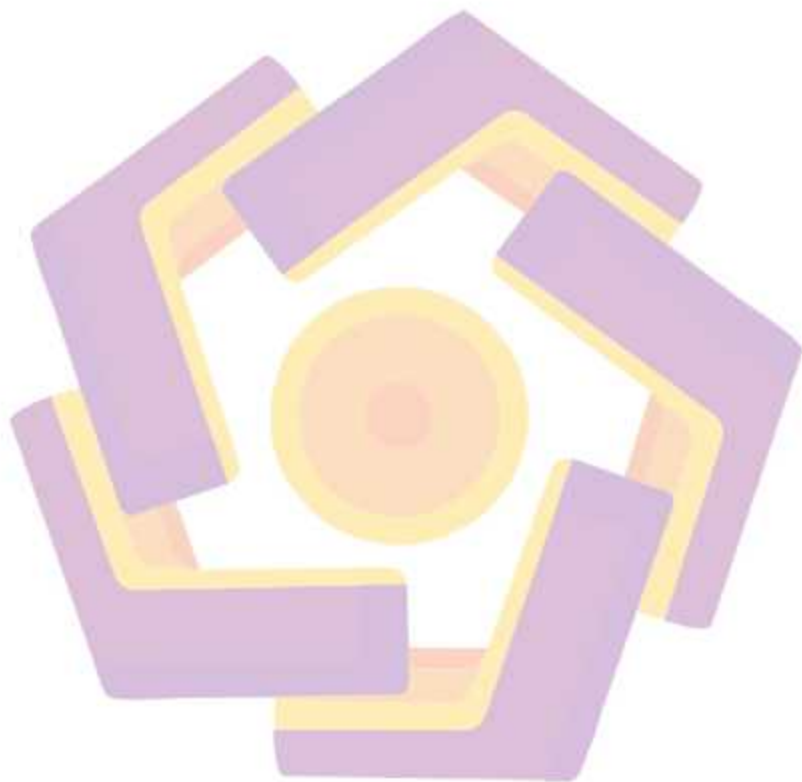


DAFTAR GAMBAR

Gambar 2. 1 Cara Kerja API.....	18
Gambar 2. 2 Alur Pengujian Load Testing	21
Gambar 3. 1 Alur Penelitian.....	36
Gambar 4. 1 Flowchart Pengujian Apache JMeter	42
Gambar 4. 2 Enviroment Service Plan	44
Gambar 4. 3 Skenario Apache JMeter API Login	44
Gambar 4. 4 Skenario Apache JMeter API Status Login	45
Gambar 4. 5 Skenario Apache JMeter API Jadwal Ujian.....	45
Gambar 4. 6 Skenario Apache JMeter API Mulai Ujian	46
Gambar 4. 7 Skenario Apache JMeter API List Soal Ujian.....	46
Gambar 4. 8 Skenario Apache JMeter API Soal Ujian.....	47
Gambar 4. 9 Skenario Apache JMeter API Submit Jawaban.....	47
Gambar 4. 10 Flowchart Pengujian Locust.....	53
Gambar 4. 11 Gambar Grafik Hasil Throughput JMeter dan Locust	74

DAFTAR LAMPIRAN

Lampiran 1 Tabel Hasil Pengujian JMeter	82
Lampiran 2 Tabel Hasil Pengujian Locust.....	92



INTISARI

Arsitektur *microservices* merupakan pendekatan pengembangan perangkat lunak yang memecah sistem menjadi layanan-layanan kecil yang saling terpisah. Meskipun memberikan fleksibilitas dan skalabilitas, arsitektur ini memiliki tantangan dalam memastikan performa setiap layanan tetap stabil ketika diakses secara bersamaan. Oleh karena itu, diperlukan pengujian performa untuk menjamin kualitas layanan, khususnya pada REST API yang menjadi sarana komunikasi antar layanan.

Penelitian ini melakukan pengujian performa terhadap REST API Pijar Sekolah menggunakan dua tools, yaitu Apache JMeter dan Locust. Pengujian difokuskan pada tiga metrik utama, yaitu *response time*, *throughput*, dan *error rate*. Data hasil pengujian kemudian dianalisis dan dibandingkan untuk mengetahui perbedaan performa antara kedua tools. Selain itu, uji statistik *Wilcoxon Signed-Rank Test* digunakan untuk membuktikan apakah perbedaan yang muncul bersifat signifikan.

Hasil penelitian menunjukkan bahwa kedua tools mampu menjalankan pengujian performa dengan baik pada skenario *microservices*. Metrik *average response time* dan *error rate* tidak menunjukkan perbedaan signifikan antara JMeter dan Locust. Namun, pada metrik *maximum response time*, *95th percentile*, *99th percentile*, serta *throughput*, ditemukan perbedaan signifikan yang menunjukkan keunggulan Locust dalam menangani beban tinggi. Dengan demikian, secara keseluruhan, Locust memiliki performa yang lebih optimal dalam skenario yang membutuhkan kemampuan memproses request dalam jumlah besar.

Kata kunci: *Apache JMeter, Locust, REST API, Microservices, Load Testing, Uji Wilcoxon Signed-Rank Test.*

ABSTRACT

The microservices architecture is a software development approach that decomposes a system into small, independent services. Although it offers flexibility and scalability, this architecture presents challenges in ensuring that each service maintains stable performance when accessed concurrently. Therefore, performance testing is required to guarantee service quality, particularly for REST APIs that serve as the communication bridge between services.

This study conducts performance testing on the Pijar Sekolah REST API using two tools, namely Apache JMeter and Locust. The testing focuses on three primary metrics: response time, throughput, and error rate. The resulting data is analyzed and compared to identify performance differences between the two tools. In addition, the Wilcoxon Signed-Rank Test is employed to determine whether the observed differences are statistically significant.

The findings indicate that both tools are capable of executing performance tests effectively in a microservices environment. The metrics of average response time and error rate do not show significant differences between JMeter and Locust. However, for maximum response time, the 95th percentile, the 99th percentile, and throughput, significant differences were identified, demonstrating Locust's superiority in handling high-load scenarios. Overall, Locust delivers more optimal performance in situations requiring the processing of a large number of requests.

Keywords: *Apache JMeter, Locust, REST API, Microservices, Load Testing, Wilcoxon Signed-Rank Test.*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Web services merupakan metodologi utama dalam komunikasi data melalui jaringan. Saat ini, banyak aplikasi web dan aplikasi seluler berkomunikasi melalui API *web services*, di mana REST API menyumbang lebih dari 70% API publik (Dissanayake, 2018). API (*Application Programming Interface*) dikembangkan dengan tujuan agar sebagian atau seluruh fungsi dalam suatu sistem dapat diakses dan digunakan secara terprogram. API pada *Web Services* menyediakan fungsi perangkat lunak sebagai layanan mandiri yang dapat diakses melalui protokol standar (Wang dkk., 2017).

Salah satu arsitektur yang umum digunakan dalam pengembangan Web Service API adalah REST (*Representational State Transfer*). REST merupakan gaya arsitektur di mana klien mengirimkan permintaan dan server memberikan respons terhadap resource yang diminta. API yang dikembangkan dengan REST tidak bergantung pada satu protokol tertentu, meskipun hampir semua layanan REST menggunakan protokol HTTP. Resource tersebut dapat berupa video, halaman web, gambar, maupun format lain yang dikenali oleh sistem komputer (Soni & Ranga, 2019). Web Service API harus memiliki performa yang baik agar mampu melayani permintaan dari berbagai aplikasi dan platform secara optimal.

Dalam beberapa tahun terakhir, *Microservices* menjadi pendekatan arsitektur yang banyak diadopsi dalam pengembangan aplikasi berbasis Web

Service. *Microservices* memecah aplikasi menjadi unit-unit layanan kecil yang bekerja secara independen serta berkomunikasi menggunakan API (Biasin dkk., 2020). Setiap service memiliki fungsi spesifik dan dapat dikembangkan, diuji, serta di *deployment* secara terpisah. Pendekatan ini meningkatkan skalabilitas, fleksibilitas, dan ketangguhan sistem, namun juga meningkatkan kompleksitas terutama dalam hal orkestrasi layanan dan performa komunikasi antar service (Fansha dkk., 2021). Dengan meningkatnya penggunaan *cloud computing* dan *microservices*, serta *distributed architecture systems*, tantangan dalam melakukan pengujian *performance* juga semakin besar (Neelapu, 2024).

Pengujian didefinisikan sebagai proses memastikan bahwa sistem memenuhi persyaratan dan mengidentifikasi perbedaan antara hasil yang diharapkan dan hasil aktual (Umar & Zhanfang, 2019). Salah satu jenis pengujian yang krusial adalah *performance testing*, yaitu pengujian untuk menilai daya tanggap, skalabilitas, dan stabilitas perangkat lunak dalam berbagai kondisi penggunaan (Pargaonkar, 2023).

Beberapa metrik umum dalam pengujian performa antara lain *response time*, *throughput*, *response size*, dan *error rate*. Waktu respons API sendiri dapat dikategorikan menjadi sangat cepat (<100 ms), cepat (100–300 ms), sedang (300–800 ms), lambat (800–1200 ms), dan sangat lambat (>1200 ms) (Odown.com, n.d.). Dalam pengujian performa, diperlukan beberapa tingkat beban agar hasil evaluasi tidak hanya menggambarkan kondisi ideal, tetapi juga menunjukkan kemampuan aplikasi menghadapi berbagai situasi penggunaan (Fansha dkk, 2021). Oleh karena itu, pengujian dibagi menjadi tiga skala, yaitu skala kecil dengan *Concurrent Users*

1 - 100, menengah dengan *Concurrent Users* 100 - 1000, dan besar 1000 – 100.000+. Dalam pengujian *performance* berperan dalam membantu organisasi memenuhi *Service Level Agreements* (SLAs) (Neelapu, 2024). Pembagian ini memungkinkan peneliti untuk mengamati kinerja sistem secara bertahap, mulai dari beban ringan (baseline), beban normal yang mencerminkan kondisi operasional, hingga beban tinggi untuk menguji batas kemampuan aplikasi.

Terdapat beberapa tools open-source yang umum digunakan untuk pengujian performa, di antaranya Apache JMeter, Gatling, K6, dan Locust. Apache JMeter mampu melakukan pengujian performa pada berbagai protokol aplikasi, sedangkan Locust digunakan untuk mengevaluasi perilaku Web Service ketika diakses oleh banyak pengguna secara bersamaan. Tools pengujian ini dapat dibandingkan berdasarkan jenis aplikasi yang didukung, bahasa pemrograman, sistem operasi, format hasil pengujian, dan kemudahan konfigurasi (Dhiman & Sharma, 2016).

Dalam penelitian ini, perbandingan performa dilakukan menggunakan Apache JMeter dan Locust pada layanan REST API berbasis arsitektur *microservices*. Penelitian ini memberikan kontribusi penting bagi literatur terkait metodologi *performance testing*, khususnya dalam konteks sistem terdistribusi yang membutuhkan evaluasi menyeluruh terhadap stabilitas dan efisiensi layanan. Untuk mengetahui apakah terdapat perbedaan performa yang signifikan antara kedua tools tersebut, digunakan metode uji statistik Wilcoxon Signed-Rank Test. Wilcoxon juga memungkinkan analisis yang lebih sensitif terhadap variasi performa pada berbagai metrik, seperti *Error*, *Average Response Time* (ms), *Min*

Response Time (ms), *Max Response Time* (ms), *95th Percentile* (ms), *99th Percentile* (ms), dan *Throughput* (req/s), yang merupakan indikator dalam pengujian pada sistem *microservices*. Dengan demikian, penggunaan *Wilcoxon Signed-Rank Test* dalam penelitian ini tidak hanya meningkatkan validitas hasil komparasi performa antara JMeter dan Locust, tetapi juga memastikan bahwa interpretasi statistik yang diperoleh sesuai dengan karakteristik data pengujian yang bersifat non-linear dan cenderung fluktuatif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas,berikut beberapa rumusan masalah pada penelitian ini yaitu :

1. Bagaimana proses pengujian *performance* Rest API berbasis arsitektur *microservices* dilakukan menggunakan Apache JMeter dan Locust ?
2. Bagaimana hasil pengujian *performance* Rest API *microservice* berdasarkan metrik *response time*, *throughput*, dan Error Rate pada Apache JMeter dan Locust ?
3. Apakah terdapat perbedaan performa yang signifikan antara Apache JMeter dan Locust dalam melakukan pengujian *load testing* terhadap Rest API *microservices* ?
4. Apakah perbedaan performa antara Apache JMeter dan Locust dapat dibuktikan secara statistic menggunakan metode uji *Wilcoxon Signed-Rank Test* ?

1.3 Batasan Masalah

Berikut batasan masalah dalam penelitian ini :

- a) Peneliti akan melakukan pengujian *performance testing* terhadap API Aplikasi Pijar Sekolah yang dibangun menggunakan arsitektur *microservices*.
- b) Peneliti akan mengukur perbedaan dari hasil pengujian dan penggunaan dari Apache JMeter dan Locust. Perbedaan hasil seperti *Error*, *Average Response Time (ms)*, *Min Response Time (ms)*, *Max Response Time (ms)*, *95th Percentile (ms)*, *99th Percentile (ms)*, dan *Throughput (req/s)*.

1.4 Tujuan Penelitian

Tujuan yang akan dicapai oleh peneliti dalam penelitiannya adalah sebagai berikut :

- 1) Untuk menjelaskan proses pengujian performa REST API berbasis *microservices* menggunakan Apache JMeter dan Locust.
- 2) Untuk membandingkan hasil performa Apache JMeter dan Locust dalam melakukan pengujian *load testing* terhadap REST API berbasis *microservices*.
- 3) Untuk menganalisa hasil pengujian *performance* REST API *microservice* berdasarkan parameter *response time*, *throughput*, dan *error rate* menggunakan Apache JMeter dan Locust.

- 4) Untuk mengetahui apakah terdapat perbedaan performa yang signifikan antara Apache JMeter dan Locust dengan menggunakan metode uji statistic Wilcoxon Signed-Rank Test.

1.5 Manfaat Penelitian

Manfaat penelitian dapat dijabarkan menjadi 2 yaitu berdasarkan segi teoritis dan praktis atau manfaat bagi objek penelitian dan bagi peneliti selanjutnya.

- 1) Memberikan referensi ilmiah mengenai perbandingan performa antara Apache JMeter dan Locust open-source.
- 2) Menambah kontribusi pengetahuan dalam bidang pengujian performa system, khusus pada layanan REST API berbasis arsitektur microservice.
- 3) Memberikan panduan bagi developer dan praktisi teknologi informasio dalam melakukan pengujian performa REST API pada arsitektur microservices.

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Beberapa Penelitian yang berkaitan dengan penelitian dalam menguji performa sebuah software adalah sebagai berikut :

1. Penelitian yang dilakukan oleh Alam dan Dew, (2020). Bertujuan untuk menganalisa terhadap *performance* aplikasi Bandungtanginas. Aplikasi mampu melayani sekitar 1.982 posyandu dan sekitar 10.000 mengakses aplikasi bandungtangkis.id. Analisa data dalam menentukan *performance* untuk mendapatkan variable yang akan digunakan seperti Jumlah Pengguna dalam menentukan skenario pengujian. Dalam pengujian tersebut akan mendapatkan hasil *Response Time* dan *Error Rate*. Penerapan *performance testing* dengan menggunakan Apache JMeter. Hasil dari pengujian ketika aplikasi Badung Tangkis di akses dengan jumlah pengguna 50 mendapatkan nilai waktu 25,9 detik, dan pengujian dengan jumlah 100 pengguna mendapatkan nilai waktu 58,7 detik. Hasil *error rate* yang di dapat dari scenario yang di uji adalah 0%. Analisa pengujian dengan hasil respon yang menunjukkan hasil rata-rata dibawah 30 detik sesuai dengan SLA (*Service Level Argeement*). Dalam penelitan tersebut menyimpulkan bahwa aplikasi Bandungtanginas mampu melayani pengguna dengan jumlah 30-50 pengguna.

2. Penelitian yang dilakukan oleh Musthafawi dkk. (2020) bertujuan untuk menguji performa aplikasi website Shopee dengan membandingkan kinerjanya sebelum dan selama masa pandemi. Metode pengujian performa yang digunakan dalam penelitian ini adalah *stress testing*. Analisis dilakukan dengan menerapkan skenario pengujian berdasarkan dua variabel, yaitu jumlah pengguna dan durasi waktu, baik sebelum maupun selama pandemi, untuk mengetahui tingkat kesalahan (*error rate*) yang terjadi seiring meningkatnya jumlah pengguna di masa pandemi. Jumlah pengguna yang digunakan dalam pengujian adalah sebanyak 385 orang dengan durasi waktu pengujian selama 60 detik, baik pada masa sebelum pandemi maupun selama pandemi. Pengujian dilakukan menggunakan alat bantu Apache JMeter. Hasil pengujian menunjukkan bahwa *error rate* sebelum pandemi sebesar 2%, sedangkan selama pandemi meningkat signifikan hingga mencapai 26%.
3. Penelitian yang dilakukan oleh Dhalla (2020) bertujuan untuk menilai dan membandingkan kinerja aplikasi web RESTful yang diimplementasikan menggunakan Spring Boot Java dan MS .NET Core. Untuk menguji kedua kerangka kerja tersebut, dikembangkan empat operasi dasar CRUD (*Create, Read, Update, dan Delete*) menggunakan masing-masing teknologi. Pengujian dilakukan dengan metode *load testing* menggunakan Apache JMeter dalam mode GUI, dengan peningkatan jumlah *virtual users* secara eksponensial dari 1.000 hingga 64.000. Parameter yang diukur dalam pengujian meliputi waktu respons, *error rate*, serta konsumsi sumber daya seperti CPU dan memori dalam implementasi Spring Boot dan MS .NET

Core. Hasil penelitian menunjukkan bahwa MS .NET Core memberikan waktu respons yang lebih cepat dan mengonsumsi lebih sedikit sumber daya dibandingkan dengan Spring Boot Java. Namun, keduanya memiliki kinerja yang sebanding dalam hal tingkat kesalahan pada implementasi RESTful dengan protokol HTTP. Dengan jumlah *virtual users* sebanyak 64.000, tingkat kesalahan pada kedua teknologi tersebut berada pada kisaran 75%–80%.

4. Penelitian yang dilakukan oleh Zahroh dkk. (2022) bertujuan untuk mengembangkan platform bernama Rekaruang dari sisi *back-end*. Bagian *back-end* pada platform aplikasi Rekaruang dikembangkan dan direpresentasikan dalam bentuk *API endpoint* yang kemudian diuji menggunakan metode *unit testing* dan *load testing*. Pengembangan platform ini didasarkan pada pendekatan *Minimum Viable Product* (MVP). Hasil *unit testing* menunjukkan bahwa dari 34 *test case* yang diuji, seluruhnya dinyatakan lolos. Selanjutnya, dilakukan pengujian performa menggunakan perangkat lunak Locust, yang menunjukkan bahwa layanan *microservices* mampu menangani hingga 216 pengguna secara bersamaan. Hasil tersebut dikategorikan baik karena proyeksi jumlah pengguna Rekaruang pada tahun pertama hanya sebanyak 40 pengguna, terdiri dari 20 desainer interior dan 20 klien.
5. Penelitian yang dilakukan oleh Freeman dkk. (2023) membahas mengenai *Tapis Workflows* dari sisi pengembangan *functional API*, model data, dan arsitektur *workflow engine*. Pengembangan dilakukan dengan menerapkan

berbagai teknologi seperti CI/CD, ETL Pipelines, RabbitMQ, MySQL, dan Locust. Analisis performa terhadap alur kerja API diuji menggunakan metode *load testing* dengan bantuan Locust. Pengujian dilakukan dengan maksimum 500 pengguna secara bersamaan, dengan tingkat pengujian sebesar 100 pengguna per detik untuk total 32.520 *requests* selama 5 menit. Hasil dari pengembangan API dengan alur kerja komputasi menunjukkan bahwa sistem tersebut layak digunakan sebagai platform manajemen yang dapat mendukung berbagai kasus penggunaan dalam jalur komputasi penelitian tingkat lanjut. Selain itu, performa sistem menunjukkan tingkat kegagalan *requests* yang sangat rendah, yaitu kurang dari 0,1%.

6. Penelitian yang dilakukan oleh Barczak (2021) bertujuan untuk menganalisis perbandingan kinerja aplikasi yang menggunakan arsitektur *monolith* dan *microservices* melalui metode *stress testing*. Analisis dilakukan dengan membangun aplikasi *WebShopClientService* dalam dua versi, yaitu menggunakan arsitektur *monolith* dan *microservices*. Pengujian performa dilakukan menggunakan perangkat lunak Apache JMeter dan BlazeMeter. Metrik yang dibandingkan dalam pengujian meliputi waktu respons, penggunaan memori (RAM), dan konsumsi CPU. Hasil pengujian pada tes pertama menunjukkan bahwa perbedaan waktu respons antara aplikasi *microservices* dan *monolith* sangat signifikan. Aplikasi klien berbasis *microservices* juga menunjukkan penggunaan memori RAM dan CPU yang lebih tinggi dibandingkan dengan aplikasi *monolith*. Meskipun demikian,

arsitektur *microservices* dinilai lebih populer dan diprediksi akan menjadi standar dalam pengembangan aplikasi di masa mendatang.

7. Penelitian yang dilakukan oleh Abda'u, dkk. (2024) membahas perbandingan kinerja antara Gatling dan Apache JMeter dalam melakukan pengujian beban terhadap sebuah RESTful API. API yang digunakan dalam pengujian adalah API login dari sistem E-Patrol. Analisis perbandingan dilakukan secara kualitatif dan kuantitatif, dengan parameter pengujian yang meliputi waktu respons, *throughput*, dan latensi. Hasil penelitian menunjukkan bahwa Gatling dan Apache JMeter memiliki performa yang sebanding. Dalam aspek waktu respons, Gatling menunjukkan keunggulan, sedangkan Apache JMeter unggul dalam jumlah transaksi per detik (*throughput*). Dengan demikian, jika fokus pengujian adalah kecepatan waktu respons, maka Gatling merupakan pilihan yang lebih baik. Namun, jika yang diutamakan adalah *throughput*, ketersediaan fitur umum, serta dukungan plugin, maka Apache JMeter menjadi pilihan yang lebih sesuai.

2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian

Analisis Perbandingan Apache JMeter dan Locust Dalam Pengujian Performa API Service

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	<i>Comprehensive Review of Load Testing Tools</i>	(Shrivastava & SB, 2020), International Research Journal of Engineering and Technology (IRJET)	Membandingkan tools yang digunakan dalam pengujian beban terhadap service, tools yang dibandingkan antara nya adalah Gatling, Locust, dan Apache JMeter	Setiap tools mempunyai fungsi yang sama dengan pengujian yang dilakukan. Locust mengapatkan keunggulan efisiensi, Gatling mendapatkan keunggulan visualisasi, dan JMeter mendapatkan keunggulan dari	Dapat melakukan penelitian lebih lanjut untuk menganalisa penggunaan resource memori ketika menjalankan software <i>tools performance testing</i> Gatling, Locust, dan Apache JMeter.	Penelitian saat ini juga membandingkan cara kinerja antara Locust dan Apache Jmeter, namun dengan data pada aplikasi yang saat ini berjalan publik yaitu API dari Pijar Sekolah. Penelitian ini menekankan dari hasil <i>response time</i> dan <i>error rete</i> yang di dapat dari Locust dan Apache Jmeter.

				berbagai protokol yang dapat di uji.		
2	<i>Analytical Evaluation of Web Performance Testing Tools: Apache Jmeter and SoapUI</i>	(Tiwari, Upadhyay, Goswami, & Agrawal, 2023), IEEE International Conference on Communication Systems and Network Technologies	Membandingkan 2 tools yang akan digunakan dalam pengujian performa pada aplikasi berbasis Web yaitu Apache JMeter dan SOAP UI.	SoapUI lebih baik dalam pengujian <i>performance</i> untuk project skala kecil. Sedangkan untuk project skala besar JMeter lebih cocok karena memiliki grafik dan laporan yang lebih baik dari SoapUI	Analisa perbandingan yang digunakan dari sisi platform support, bahasa pemrograman support, cloud compability, dan penggunaan pada.	Penelitian tidak menguji perbandingan dengan mengakses halaman website tetapi langsung dengan mengakses API. Sedangkan penelitian yang dirujuk menguji dengan mengakses situs website.
3	<i>Analysis Of Application Performance Testing Using Load Testing and Stress Testing Methods in API Service</i>	(Hendayun, Ginanjar, & Ihsan, 2023), Sisfotek Global	Melakukan pengujian performa dari sebuah API Service dengan menggunakan metode load testing dan stress testing.	Sumber daya server masih ada tersedia ketika ada request dengan 500 CCU, tetapi ada beberapa request yang gagal, yang	Analisa mengenai <i>performance</i> testing dapat mengoptimalkan dari sisi penggunaan CPU dan juga RAM untuk menguji API Service.	Sama- sama melakukan pengujian load testing terhadap API Service menggunakan Apache JMeter. Namun pada penelitian ini juga melakukan pengujian dengan menggunakan Locust.

				berarti gagal dan tidak dibatasi oleh spesifikasi sumber daya tetapi memerlukan optimasi dari sisi service.		
4	<i>A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Application</i>	(Pradcep & Sharma, 2019), International Conference on Artificial Intelligence (AICAI)	Melakukan berbagai pengujian terhadap Aplikasi berbasis Web dengan mengukur performa pada aplikasi yang dibangun. Dengan menggunakan Apache JMeter, Locust, dan HULK Analyzer	Dari hasil dan upaya pengujian yang berbeda, Menggunakan Locust ditemukan relatif cepat dan memiliki waktu yang lebih sedikit waktu eksekusi dibandingkan dengan Apache JMeter berbasis Java.	Analisa perbandingan menggunakan execution time dalam menjalankan Apache JMeter, Locust, dan HULK.	Penelitian saat ini berfokus pada pengujian <i>performance</i> metode load testing dengan menguji data dari API sebuah aplikasi. Penelitian yang di rujuk menggunakan perbandingan data dari mengakses sebuah situs website.

5	<i>Software and Performance Testing Tools</i>	(Srivastava, Kumar, & Singh, 2021), Journal of Informatics Electrical and Electronics Engineering	Melakukan analisa software tools HP LoadRunner dan Apache Jmeter dalam melakukan pengujian performa dengan parameter work time, dependableness, load capability.	LoadRunner memberikan penyelesaian masalah skrip yang lebih cepat dan stabil dalam menjalankan pengujian. Segi biaya penggunaan LoadRunner lebih tinggi dibandingkan dengan menggunakan Apache JMeter.	Skenario yang diuji masih sederhana dengan menggunakan 1 concurrent users dan 15 request.	Pada penelitian ini menganalisa software pengujian <i>performance</i> pada Apache JMeter dan Locust dengan menggunakan metode yang sama yaitu <i>Load Testing</i> . Pada penelitian yang di rujuk menganalisa perbedaan Apache JMeter dan Load Runner.
6	<i>Quality Assurance of Academic Websites using Performance Testing Tools</i>	(Mufarroha, et al., 2023), Technium: Romanian Journal of Applied Sciences and Technology	Melakukan pengujian performa untuk aplikasi website akademik Trunojoyo University dengan menggunakan software Apache	Pengujian dengan simulasi pengguna terbesar, atau 500 pengguna, dihasilkan nilai <i>Throughput</i> tertinggi. Faktor yang paling krusial dalam menentukan seberapa baik server	Perlu dilakukan penelitian lebih lanjut untuk mengidentifikasi error yang di dapat saat melakukan pengujian <i>performance</i> .	Penelitian dilakukan menguji <i>performance</i> dari API sebuah aplikasi pijar sekolah. serta membandingkan alat uji Apache JMeter dan locust. Sedangkan penelitian yang di rujuk menggunakan apache JMeter dalam

			<p>JMeter. Mengidentifikasi kendala performa situs web akademik, menelusuri penyebab ketidakstabilan atau kegagalan. Ketika kondisi beban yang tinggi, dan memastikan situs akademik dapat menangani lalu lintas pengguna yang padat tanpa menurunkan kualitas performa.</p>	<p>dapat mengelola beban yang di terima adalah <i>throughput</i>. Sedangkan kinerja server meningkat seiring dengan meningkatnya nilai <i>Throughput</i>. Dan hasil <i>error rate</i> yang didapat ketika 500 user di uji adalah 44,20%.</p>	<p>pengujian <i>performance</i> dalam mengakses sebuah website academic.</p>
--	--	--	---	--	--

2.3 Landasan Teori

a) Software Quality Assurance

Menurut Goericke (2019), *Software Quality Assurance* (SQA) merupakan upaya yang direncanakan secara sistematis untuk memastikan bahwa suatu produk perangkat lunak memenuhi kriteria kualitas, serta memiliki atribut tambahan yang spesifik sesuai kebutuhan proyek, seperti portabilitas, efisiensi, kemampuan digunakan kembali (*reusability*), dan fleksibilitas.

Hossain (2020) mendefinisikan SQA sebagai sekumpulan aktivitas dan fungsi yang digunakan untuk memantau serta mengendalikan proses pengembangan perangkat lunak agar tetap berada pada jalur pencapaian tujuan proyek, dengan tingkat kepercayaan (*confidence level*) yang diharapkan.

Sementara itu, menurut Pressman (2010), tugas utama SQA meliputi penetapan standar dan prosedur kualitas yang harus diikuti selama proses pengembangan perangkat lunak, melakukan peninjauan (*review*) serta pengujian (*testing*) untuk memastikan bahwa produk yang dikembangkan sesuai dengan standar yang telah ditetapkan, serta mendeteksi dan mencegah *defect* selama proses pengembangan berlangsung.

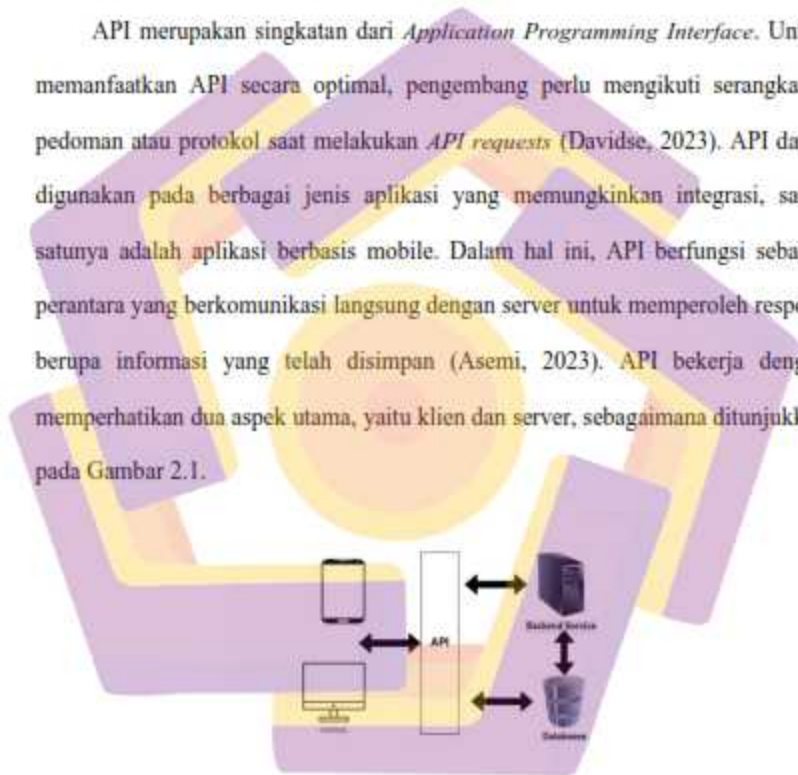
b) Web Service

Web Service merupakan sistem perangkat lunak yang dirancang untuk mendukung interaksi antarmesin melalui jaringan komputer (Susana dkk., 2015). Dengan menggunakan Web Service, berbagai jenis perangkat lunak dapat saling terhubung dan bertukar informasi serta data yang dibutuhkan.

Terdapat dua arsitektur yang paling dikenal dalam pengembangan Web Service, yaitu SOAP (*Simple Object Access Protocol*) dan REST (*Representational State Transfer*) (Ramadhanu & Priandika, 2021).

e) Application Programming Interface

API merupakan singkatan dari *Application Programming Interface*. Untuk memanfaatkan API secara optimal, pengembang perlu mengikuti serangkaian pedoman atau protokol saat melakukan *API requests* (Davidse, 2023). API dapat digunakan pada berbagai jenis aplikasi yang memungkinkan integrasi, salah satunya adalah aplikasi berbasis mobile. Dalam hal ini, API berfungsi sebagai perantara yang berkomunikasi langsung dengan server untuk memperoleh respons berupa informasi yang telah disimpan (Asemi, 2023). API bekerja dengan memperhatikan dua aspek utama, yaitu klien dan server, sebagaimana ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Cara Kerja API

d) REST API

Representational State Transfer (REST) atau *RESTful Web Services* adalah pendekatan untuk menyediakan interoperabilitas antar sistem komputer melalui

jaringan internet. Web service yang sesuai dengan prinsip REST memungkinkan suatu sistem untuk melakukan permintaan (*request*) dalam rangka mengakses dan memanipulasi representasi tekstual dari *web resources* menggunakan seperangkat metode yang telah ditentukan sebelumnya dalam sistem.

Resource (sumber daya) merujuk pada suatu jenis informasi yang dapat diakses, seperti objek aplikasi, rekaman basis data, algoritma, dan sejenisnya. Setiap resource diidentifikasi oleh URI (*Uniform Resource Identifier*). REST mewakili URI dalam bentuk path seperti “/user/name” dan menggunakan metode HTTP seperti GET, PUT, POST, DELETE, HEAD, dan OPTIONS untuk melakukan operasi terhadap resource tersebut (Chen dkk., 2017).

e) Performance Testing

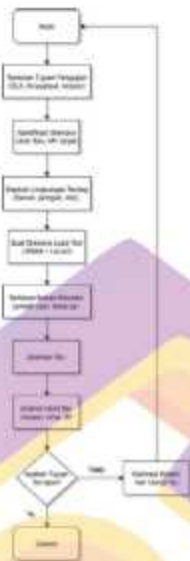
Pengujian performa merupakan jenis pengujian non-fungsional yang digunakan untuk mengukur karakteristik perangkat lunak, seperti *response time*, *page load time*, serta penggunaan sumber daya seperti memori (RAM) dan CPU server. Selain itu, pengujian ini juga bertujuan untuk menentukan ambang batas performa optimal dari perangkat lunak yang dikembangkan.

Hasil dari pengujian performa memberikan gambaran mengenai tingkat kepuasan terhadap pengembangan dan penggunaan suatu aplikasi, yang dianalisis berdasarkan parameter-parameter terkait waktu, seperti *load time*, *access time*, *run time*, *execution time*, *success rate*, *failure frequency*, serta berbagai hambatan teknis yang ditemukan selama pengujian.

Pengujian performa dilakukan untuk mengevaluasi kecepatan, stabilitas, dan skalabilitas suatu aplikasi dalam berbagai skenario penggunaan (Pradeep & Sharma, 2019). Dalam praktiknya, beberapa parameter utama menjadi fokus pengukuran, seperti *response time*, *error rate*, dan *latency* (Barezak, 2021). Pemahaman terhadap parameter-parameter ini sangat penting untuk menilai kemampuan sistem dalam menyediakan layanan yang konsisten saat diakses oleh banyak pengguna secara simultan.

f) Load Testing

Load Testing merupakan salah satu metode dari pengujian performa yang digunakan untuk menguji seberapa baik suatu system mampu menangani beban kerja dalam jumlah tertentu, seperti banyaknya permintaan dari pengguna secara simultan. *Load Testing* adalah jenis pengujian performa yang bertujuan mengevaluasi kemampuan system aplikasi saat menerima sejumlah besar beban pengguna atau permintaan dalam waktu bersamaan. *Load Testing* mengukur kinerja sistem dalam kondisi beban yang diharapkan untuk memastikan sistem merespons dalam jangka waktu yang dapat diterima (Priyanka & Kanna, 2016).



Gambar 2. 2 Alur Pengujian *Load Testing*

Pada Gambar 2.2 menggambarkan prinsip dasar iteratif dalam *load testing*: Uji → Evaluasi → Optimasi → Uji ulang, sampai sistem siap untuk produksi. Proses dimulai dengan menentukan tujuan pengujian, yang mencakup parameter performa seperti *Service Level Agreement (SLA)*, throughput, waktu respon maksimum, dan tingkat kesalahan yang dapat ditoleransi. Setelah tujuan ditetapkan, langkah selanjutnya adalah mengidentifikasi skenario pengujian. Di tahap ini, pengujian menentukan alur pengguna (*user flow*) atau API endpoint yang akan diuji, disesuaikan dengan aktivitas nyata pengguna.

Setelah skenario ditentukan, dilakukan persiapan lingkungan pengujian, yaitu menyiapkan server, database, jaringan, dan infrastruktur lainnya yang diperlukan untuk simulasi. Berikutnya, pengujian akan membuat skenario pengujian

menggunakan tools seperti JMeter atau Locust. Skenario ini mencakup urutan aktivitas pengguna yang disimulasikan selama tes.

Tahap berikutnya adalah menentukan beban simulasi, yaitu berapa banyak pengguna virtual yang akan digunakan, bagaimana pola peningkatan beban (*ramp-up*), serta durasi pengujian. Setelah semua konfigurasi siap, proses pengujian dijalankan untuk melihat bagaimana sistem merespons beban tersebut.

Setelah tes selesai, penguji melakukan analisis terhadap hasil pengujian. Analisis ini mencakup waktu respon rata-rata, tingkat kesalahan, performa server, dan kapasitas sistem. Berdasarkan hasil ini, dilakukan evaluasi: apakah sistem telah memenuhi tujuan pengujian atau belum.

Jika tujuan pengujian belum tercapai, maka dilakukan optimasi sistem, seperti tuning performa aplikasi, perbaikan kode, scaling server, atau penyesuaian konfigurasi. Setelah optimasi, tes diulang dari awal hingga sistem memenuhi standar performa yang ditentukan. Jika semua tujuan sudah tercapai, maka proses pengujian dinyatakan selesai.

Parameter yang akan di uji dalam melakukan pengujian *Load Testing* yaitu :

1. *Response Time* (Waktu Respons)

Waktu respons adalah durasi yang dibutuhkan oleh sistem untuk memberikan respons terhadap suatu permintaan yang dikirim oleh pengguna. Parameter ini merupakan salah satu indikator utama dalam *load testing* karena

mencerminkan seberapa cepat sistem memproses permintaan di bawah kondisi tertentu.

Waktu respons umumnya diukur dalam satuan milidetik (ms) atau detik (s), tergantung pada konteks dan skala aplikasi yang diuji. Dalam praktiknya, waktu respons dapat dibedakan menjadi beberapa metrik penting, yaitu:

- *Minimum Response Time*: waktu respons tercepat yang tercatat selama pengujian berlangsung.
- *Maximum Response Time*: waktu respons terlama yang terjadi dalam periode pengujian.
- *Average Response Time*: nilai rata-rata dari semua waktu respons yang terekam.
- *Percentile Response Time*: nilai waktu respons yang menunjukkan persentil tertentu dalam distribusi data. Sebagai contoh, *95th percentile* menunjukkan bahwa 95% permintaan memiliki waktu respons yang lebih cepat atau sama dengan nilai tersebut, sedangkan 5% sisanya lebih lambat. Ini berguna untuk memahami performa sistem dalam skenario nyata, karena rata-rata saja tidak selalu mencerminkan pengalaman mayoritas pengguna.

2. Throughput

Throughput adalah ukuran yang menunjukkan jumlah permintaan (request) atau transaksi yang berhasil diproses oleh sistem dalam satu periode waktu tertentu. Parameter ini digunakan untuk menggambarkan kapasitas sistem dalam menangani beban kerja, terutama saat dihadapkan pada kondisi lalu lintas yang tinggi. Dalam pengujian performa, *throughput* biasanya

dinyatakan dalam satuan *requests per second* (RPS) atau *transactions per second* (TPS). Nilai *throughput* yang tinggi menandakan bahwa sistem mampu melayani banyak permintaan dalam waktu singkat, sedangkan nilai yang rendah bisa mengindikasikan keterbatasan performa atau potensi *bottleneck*.

3. *Concurrency* (Jumlah Pengguna Bersamaan)

Concurrency mengacu pada jumlah pengguna virtual yang secara bersamaan mengakses sistem atau aplikasi dalam proses pengujian beban. Parameter ini digunakan untuk mensimulasikan kondisi nyata di mana banyak pengguna aktif menggunakan layanan pada waktu yang sama. Dalam konteks *load testing*, *concurrency* merepresentasikan seberapa banyak *thread* atau *user instance* yang dikonfigurasi untuk menjalankan permintaan secara paralel. Pada tool seperti Apache JMeter, parameter ini dikenal sebagai *number of threads*, sementara pada Locust disebut sebagai *number of users*.

4. *Error Rate*

Error rate adalah parameter yang menunjukkan persentase permintaan yang mengalami kegagalan selama pengujian beban berlangsung. Kegagalan ini dapat berupa berbagai jenis kesalahan, seperti respons HTTP dengan status kode 4xx (client error), 5xx (server error), maupun kasus timeout di mana server tidak merespons dalam batas waktu yang ditentukan. Nilai *error rate* biasanya disajikan dalam bentuk persentase (%) dari total permintaan yang dikirim. Misalnya, jika dari 10.000 permintaan terdapat 500 yang gagal, maka *error rate*-nya adalah 5%.

5. CPU & Memory Usage

Penggunaan sumber daya (*resource usage*) adalah parameter yang digunakan untuk memantau kondisi internal sistem saat menghadapi beban tinggi, khususnya dalam hal konsumsi CPU, memori (RAM), disk I/O, dan jaringan (*network usage*). Monitoring ini penting untuk mengetahui apakah performa buruk disebabkan oleh keterbatasan kapasitas perangkat keras atau konfigurasi sistem yang tidak optimal. Selama pengujian beban berlangsung, data *resource usage* dapat dikumpulkan melalui berbagai alat pemantauan seperti Grafana (biasanya dikombinasikan dengan Prometheus atau InfluxDB), serta alat berbasis sistem operasi seperti top, htop, atau *server profiler* lainnya. Visualisasi dari alat-alat ini membantu pengembang dalam mengidentifikasi *bottleneck*, misalnya lonjakan CPU usage atau kebocoran memori saat jumlah permintaan meningkat. Dengan mengamati penggunaan sumber daya selama tes berlangsung, tim pengembang dapat menentukan apakah sistem perlu dioptimalkan secara kode, database, atau perlu dilakukan *scaling* pada infrastruktur.

g) Apache JMeter

Apache JMeter adalah software yang dapat digunakan untuk melakukan pengujian *performance testing*. Apache JMeter dikembangkan oleh Apache Software Foundation (ASF). Software Apache JMeter ini adalah perangkat lunak *open-source* yang dibangun menggunakan Bahasa pemrograman Java dan software tersebut mendukung berbagai jenis *protocol* komunikasi, termasuk HTTP, HTTPS, FTP, dan juga JDBC. Apache JMeter dapat mengukur performance sebuah aplikasi

dengan cara mensimulasikan sejumlah besar pengguna untuk menguji bagaimana aplikasi tersebut menangani beban tersebut (Apache Software Foundation, 2023).

h) Locust

Locust adalah software pengujian beban yang terfokus pada pengujian beban HTTP dan HTTPS. Locust merupakan software open-source yang dibangun menggunakan bahasa pemrograman Python dan di rilis pada tahun 2019. Dengan menggunakan kode Python dapat membuat berbagai skenario pengujian beban yang memiliki fleksibilitas lebih besar. (Locust.io, 2023).

i) Application Performance Monitoring

Application Performance Monitoring (APM) adalah *software tools* yang digunakan untuk proses memantau *performance* sebuah *software application*. Penggunaan APM dapat mengirimkan data *performance* sebuah aplikasi dengan real-time (AWS, 2023). APM tools seperti New Relic, Datadog, AppDynamics dan Elastic focus pada memantau *performance* sebuah aplikasi yang memberikan informasi terkait code-level issues, database queries, and external service dependencies.

j) Statistika

Statistika atau juga bisa disebut dengan metoda statistik, mempunyai peranan yang sangat penting di dalam kehidupan manusia.

Statistika merupakan suatu cabang ilmu dari matematika yang pada dasarnya bukan mengemukakan data atau fakta-fakta, tetapi merupakan ilmu yang hanya mengetahui sebagian dari populasi namun membicarakan keseluruhan populasi

(Danang Sunyoto, 2013). Selanjutnya Statistika adalah cabang ilmu yang mempelajari cara-cara mendeteksi suatu objek, menggambarkan objek, dan menganalisis setiap aspek-aspek yang mempengaruhi objek, untuk disimpulkan secara ilmiah tentang keberadaan objek, sebagai pedoman sains atau pengambilan Keputusan (Siregar ,2013).

Peranan dari statistika, diantaranya:

- a) Memungkinkan pencatatan secara lengkap dari data penyelidikan
- b) Memampukan seorang peneliti untuk bekerja secara berurutan dari awal sampai akhir
- c) Menyediakan cara-cara meringkas data kedalam bentuk yang lebih banyak artinya dan lebih gampang mengerjakannya
- d) Memberikan landasan untuk meramalkan secara ilmiah tentang bagaimana sesuatu gejala akan terjadi dalam kondisi-kondisi yang telah diketahui
- e) Memungkinkan peneliti menganalisa, menguraikan sebab akibat yang kompleks dan rumit yang tanpa statistik akan merupakan persoalan yang membingungkan serta kejadian yang tak teruraikan

Fungsi dari statistika, diantaranya:

- a) Mendeskripsikan data dalam bentuk tak tentu
- b) Menyederhanakan data yang kompleks menjadi data yang mudah dimengerti
- c) Teknik untuk membuat perbandingan

d) Menentukan tingkat hubungan atau peranan antar variabel

e) Mengukur besaran variable

Mean merupakan nilai rata-rata yang diperoleh dengan cara menjumlahkan seluruh data, kemudian membaginya dengan jumlah data yang tersedia.

$$\bar{x} = f_1X_1 + f_2X_2 + f_3X_3 + \dots + f_nX_n$$

$$\bar{x} = \frac{\sum_{i=1}^n f_i X_i}{\sum_{i=1}^n f_i}$$

Keterangan :

$f_i X_i$ = frekuensi untuk nilai x_i yang bersesuaian

x_i = data ke-i.

Median adalah nilai tengah dari sekumpulan data yang telah disusun secara berurutan; apabila jumlah datanya ganjil, median adalah nilai yang tepat berada di tengah, sedangkan jika jumlah datanya genap, median diperoleh dari rata-rata dua nilai yang berada di posisi tengah.

$$\text{Median: } M_e = L_2 + C \cdot \frac{\frac{n}{2} - \sum f_i}{f_{me}}$$

Keterangan:

n = Banyak data

L_2 = Tepi bawah kelas Median

C = Interval kelas

$(\sum f)_2$ = Frekuensi kumulatif sebelum kelas Median

f_{Me} = Frekuensi kelas Median

Modus ialah nilai yang paling sering muncul dalam suatu kumpulan data atau nilai dengan frekuensi kemunculan tertinggi.

$$M_0 = L + C \frac{b_1}{b_1 + b_2}$$

Keterangan:

M_0 = Modus

L = Tepi bawah kelas yang memiliki frekuensi tertinggi (kelas Modus)

C = Interval kelas

b_1 = Frekuensi kelas Modus dikurangi frekuensi kelas sebelumnya

b_2 = Frekuensi kelas Modus dikurangi frekuensi kelas sesudahnya

Simpangan Baku (Standar Deviasi)

Standar deviasi atau simpangan baku merupakan salah satu cara mengukur variasi sekelompok data kuantitatif. Besar harga standar deviasi menunjukkan tingkat varians dari sekumpulan data. Jika harga standar deviasi semakin besar hal ini berarti semakin bervariasi angka-angka pada data kuantitatif tersebut.

Rumusnya:

$$S = \sqrt{\frac{1}{n} \sum f_i (x_i - y)^2}$$

k) Wilcoxon

Uji Wilcoxon biasanya untuk menguji adanya perbedaan kondisi variabel pada sampel yang berpasangan, juga memperhitungkan selisih nilai positif dan negatif, serta lebih unggul dari uji tanda dalam analisis perbedaan rata-rata populasi (Rudianto dkk., 2020).

Uji Wilcoxon adalah pengujian alternatif untuk uji Paired Sample T Test dengan beberapa hal yang mempengaruhi, apabila data yang didapat dari sampel tidak bertipe interval atau rasio atau sebaran data tidak berdistribusi normal. Hanya salah satu syarat saja tidak terpenuhi, maka uji parametrik Paired Sample T Test harus diganti dengan uji non parametrik sebagai alternatifnya yaitu uji Wilcoxon.

Uji Wilcoxon adalah uji dua sampel yang dimana saling berhubungan. Uji Wilcoxon ialah uji non-parametrik yang dimana tidak punya syarat data homogen secara normal untuk membuktikan keefektifan suatu pengujian (Pratiwi et al., 2021).

Salah satu uji statistik yang krusial dalam analisis data adalah uji Wilcoxon, yang digunakan saat asumsi normalitas terganggu atau data yang memiliki sifat ordinal. Uji Wilcoxon menjadi instrumen penting dalam analisis data non-normal atau data yang bersifat ordinal. Uji ini memungkinkan peneliti farmasi untuk membandingkan kondisi atau kelompok tanpa asumsi distribusi normal, mempertahankan keakuratan analisis data dalam eksperimen terkait pengembangan obat (Fatima & Anupama, 2022).

Uji Wilcoxon Signed Rank, yaitu suatu analisis dalam statistik nonparametrik yang digunakan membandingkan dua kondisi yang saling berpasangan, terutama ketika data tidak berdistribusi normal (Afifah, dkk. 2025). Uji ini mengukur apakah median dari perbedaan antar pasangan data signifikan secara statistik (Harris & Hardin, 2013).

Berikut Rumus Wilcoxon signed rank test :

$$Z = \frac{\frac{T - N(N - 1)}{4}}{\sqrt{\frac{N(N + 1)(2N + 1)}{24}}}$$

Keterangan:

- T = nilai statistik Wilcoxon (jumlah ranking dengan tanda tertentu, biasanya yang lebih kecil antara T_+ dan T_-)
- N = jumlah pasangan data (banyaknya sampel setelah mengabaikan nilai nol)
- Z = nilai statistik uji yang akan dibandingkan dengan nilai kritis Z tabel (misalnya 1,96 untuk $\alpha = 0,05$)

Cara pengambilan keputusan:

- Tolak H_0 jika $|Z| > 1,96^{**}$ pada taraf signifikansi 0,05**.
- Artinya, ada perbedaan yang signifikan antara dua kondisi yang dibandingkan.

Uji Wilcoxon merupakan uji non parametrik yang digunakan untuk mengukur perbedaan dua kelompok data berpasangan. Penelitian ini bertujuan untuk mengetahui alasan peneliti menggunakan Uji Wilcoxon (non parametrik).

Secara umum, ada beberapa syarat yang harus dipenuhi dalam melakukan Uji Wilcoxon, yaitu:

1. Data sampel yang tidak berdistribusi normal.
2. Dua Kelompok sampel yang saling berpasangan
3. Sampel berskala data ordinal, atau interval.
4. Jumlah sampel pada kedua kelompok sama.

Pada beberapa kasus, peneliti memilih untuk menggunakan uji non parametrik Wilcoxon, disamping untuk tujuan tertentu, hal tersebut juga dilakukan karena uji non parametrik merupakan alternatif bagi peneliti dalam mengatasi keterbatasan dalam penelitian (Dwivedi et al., 2017; H.-Y. Kim, 2014). Salah satu keterbatasan tersebut adalah jumlah sampel yang diambil hanya sebesar ($N \leq 30$) yang berindikasi data tersebut tidak memenuhi asumsi distribusi normal.

BAB 3 METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Berikut Jenis, Sifat dan Pendekatan Penelitian yang akan dilakukan sebagai berikut :

1) Jenis Penelitian

Penelitian ini merupakan penelitian eksperimen, yaitu penelitian yang dilakukan dengan melakukan pengujian secara langsung terhadap objek penelitian untuk membandingkan hasil yang diperoleh. Dalam penelitian ini, software Apache JMeter dan Locust dibandingkan berdasarkan hasil pengujian performa terhadap REST API service yang digunakan dalam aplikasi Pijar Sekolah.

2) Sifat Penelitian

Sifat penelitian ini adalah deskriptif komparatif, yaitu bertujuan untuk mendeskripsikan serta membandingkan kinerja dua tools pengujian performa, yakni Apache JMeter dan Locust, berdasarkan beberapa parameter performa seperti *response time*, *throughput*, dan *error rate* pada skenario

3) Pendekatan Penelitian

Pendekatan yang digunakan dalam penelitian ini adalah pendekatan kuantitatif.

- Pendekatan kuantitatif digunakan untuk menganalisis data numerik hasil pengujian performa berupa *response time*, *throughput*, dan *error rate* yang disajikan dalam bentuk tabel dan statistik.

3.2 Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dilakukan melalui:

a) Observasi

Observasi dilakukan terhadap API service aplikasi Pijar Sekolah untuk mengidentifikasi fungsionalitas dan endpoint yang digunakan.

Observasi dilakukan dengan memanfaatkan fitur *developer tools* pada browser, dokumentasi API, serta log aktivitas untuk melihat bagaimana API digunakan oleh aplikasi.

b) Wawancara

Wawancara dilakukan kepada pengembang aplikasi Pijar Sekolah untuk memperoleh informasi terkait penggunaan API, frekuensi akses endpoint, skenario workload yang umum terjadi, serta prioritas layanan yang perlu diuji performanya. Informasi ini digunakan dalam perancangan skenario pengujian.

3.3 Metode Analisis Data

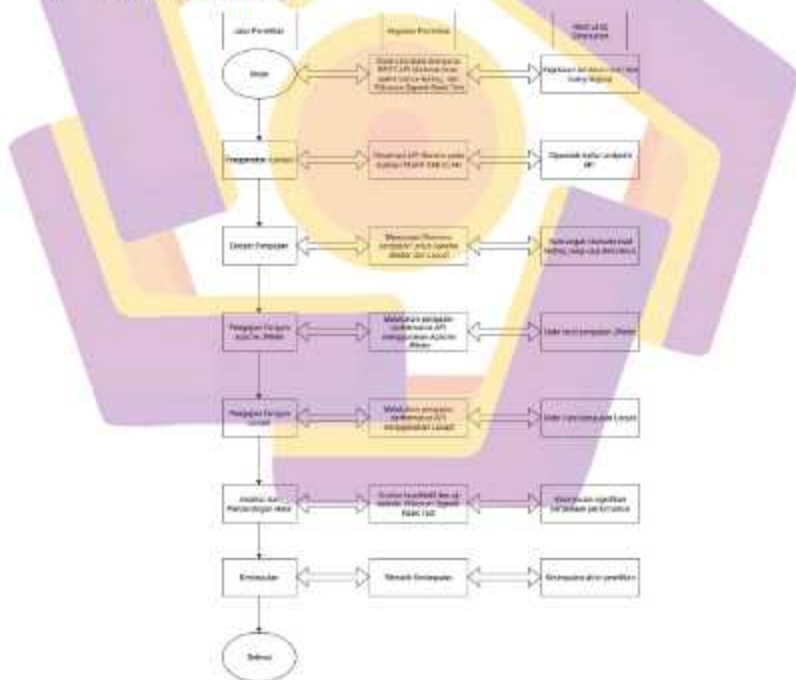
Analisis data dalam penelitian ini dilakukan melalui langkah-langkah sebagai berikut:

- 1) Mengidentifikasi API endpoint dari aplikasi Pijar Sekolah yang memiliki tingkat penggunaan tinggi atau berpotensi menjadi bottleneck.
- 2) Melakukan studi literatur mengenai teori pengujian performa API dan karakteristik tools Apache JMeter dan Locust.
- 3) Menganalisis struktur endpoint, pola request, kapasitas layanan, serta parameter beban untuk menentukan skenario pengujian.
- 4) Menyusun skenario *load testing* terhadap endpoint terpilih, termasuk jumlah virtual users, waktu peningkatan beban (*ramp-up*), dan durasi pengujian.
- 5) Melakukan pengujian performa API menggunakan Apache JMeter dan Locust berdasarkan skenario yang sama.
- 6) Mengumpulkan hasil pengujian dalam bentuk metrik performa: *Average Response Time*, *Minimum Response Time*, *Maximum Response Time*, *Throughput*, dan *Error Rate*.
- 7) Melakukan perbandingan performa antara Apache JMeter dan Locust berdasarkan hasil pengujian yang diperoleh. Perbandingan *Response time* mengungkap apakah salah satu tool cenderung menghasilkan latency tambahan karena cara mereka membuat *load*. *Throughput* menunjukkan tool mana yang lebih efisien sebagai *load generator* dalam kondisi hardware yang sama. *Error rate* membantu memastikan apakah perbedaan hasil disebabkan oleh API atau oleh tool pengujinya. Dengan menganalisis ketiga

aspek ini, penelitian dapat memberikan evaluasi yang objektif, terukur, dan relevan terhadap kinerja API serta kapabilitas kedua *tool load testing* tersebut.

- 8) Melakukan analisis statistik menggunakan metode Wilcoxon Signed-Rank Test untuk menentukan apakah perbedaan performa antara kedua tools bersifat signifikan.
- 9) Menarik kesimpulan berdasarkan hasil analisis data kuantitatif dan evaluasi kualitatif.

3.4 Alur Penelitian



Gambar 3. 1 Alur Penelitian

Alur penelitian ini menjelaskan tahapan-tahapan yang dilakukan dalam proses penelitian untuk membandingkan performa pengujian REST API berbasis microservice menggunakan Apache JMeter dan Locust dengan bantuan analisis statistik Wilcoxon Signed-Rank Test.

Berikut langkah-langkah alur penelitian secara sistematis:

1. Mulai

Tahapan awal penelitian yang menandai dimulainya proses Studi Literatur.

- Melakukan kajian teori mengenai konsep REST API, arsitektur microservices, pengujian performa (*performance testing*), serta alat uji Apache JMeter dan Locust.
- Mempelajari metode analisis statistik Wilcoxon Signed-Rank Test untuk membandingkan dua sampel hasil pengujian.
- Hasil yang diharapkan: Memperoleh dasar teori dan pemahaman yang kuat sebagai acuan penelitian.

2. Pengamatan Aplikasi (Observasi API Service)

- Melakukan analisis terhadap API service aplikasi Pijar Sekolah, untuk mengidentifikasi endpoint yang aktif digunakan.
- Menentukan endpoint dengan tingkat penggunaan tinggi untuk dijadikan objek pengujian.
- Hasil yang diharapkan: Daftar endpoint API dan karakteristik penggunaan setiap endpoint.

3. Desain Pengujian

- Menentukan skenario pengujian performa yang mencakup jumlah *virtual user*, waktu pengujian, dan jenis request.
 - Menyusun rancangan *load testing* yang sama untuk digunakan pada Apache JMeter dan Locust agar hasil bisa dibandingkan secara objektif.
 - Hasil yang diharapkan: Rancangan skenario pengujian yang siap dieksekusi pada kedua tools.
4. Pengujian Menggunakan Apache JMeter
- Menjalankan pengujian performa API menggunakan Apache JMeter berdasarkan skenario yang telah ditetapkan.
 - Merekam hasil pengujian meliputi metrik seperti: *Average Response Time*, *Minimum* dan *Maximum Response Time*, *Throughput* (Requests per Second), *Error Rate*, *Latency*.
 - Hasil yang diharapkan: Data performa API hasil pengujian menggunakan Apache JMeter.
5. Pengujian Menggunakan Locust
- Melakukan pengujian dengan Locust menggunakan skenario yang sama agar hasil dapat dibandingkan dengan JMeter.
 - Mencatat hasil pengujian berdasarkan metrik yang sama seperti pada JMeter.
 - Hasil yang diharapkan: Data performa API hasil pengujian menggunakan Locust.
6. Analisis dan Perbandingan Hasil

- Melakukan analisis kuantitatif terhadap hasil pengujian kedua tools dengan menghitung:
 - i. Nilai rata-rata (mean) setiap parameter performa.
 - ii. Nilai minimum, maksimum, dan deviasi standar untuk tiap metrik.
- Menggunakan uji statistik Wilcoxon Signed-Rank Test untuk menentukan apakah terdapat perbedaan signifikan antara hasil pengujian JMeter dan Locust.
- Hasil yang diharapkan: Kesimpulan sementara mengenai performa relatif kedua tools.

7. Kesimpulan

- Menarik kesimpulan berdasarkan hasil analisis kuantitatif dan statistik.
- Memberikan rekomendasi mengenai tool yang lebih efisien untuk pengujian performa API microservice.
- Hasil yang diharapkan: Kesimpulan akhir penelitian dan saran untuk penelitian lanjutan.

8. Selesai

- Menandai berakhirnya seluruh proses penelitian setelah semua data dianalisis dan kesimpulan disusun.

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Data

Penelitian ini memanfaatkan data dari layanan API (*Application Programming Interface*) yang diperoleh dari Aplikasi Web Pijar Sekolah, sebuah aplikasi yang berperan dalam mendigitalisasi pendidikan dengan menyediakan layanan manajemen sekolah serta mendukung proses belajar dan mengajar. Fokus penelitian ini adalah pada API yang digunakan dalam proses ujian siswa, karena API tersebut merupakan salah satu yang paling sering diakses, terutama saat berlangsungnya ujian serentak oleh banyak siswa, bahkan lintas sekolah.

Pengumpulan data API dilakukan dengan mengakses langsung aplikasi Pijar Sekolah serta melalui konfirmasi kepada tim pengembang aplikasi tersebut. Terdapat tujuh layanan API yang akan digunakan dalam penelitian ini dan ada 3 service. API yang telah dikumpulkan kemudian diproses melalui skenario pengujian yang telah ditetapkan untuk melakukan pengujian performa terhadap layanan API tersebut. Skenario pengujian yang telah ditetapkan akan dibuat dengan menggunakan Apache JMeter dan juga Locust.

4.2 Penyajian Data

Data yang telah dikumpulkan mencakup tujuh layanan API yang akan diuji performanya menggunakan Apache JMeter dan Locust. Ketujuh API tersebut ditampilkan pada Tabel 4.1. Berikut ini adalah penjelasan dari masing-masing layanan API:

1. API Login
Berfungsi agar siswa dapat melakukan proses login ke dalam aplikasi Pijar Sekolah.
2. API Status Login
Digunakan untuk memvalidasi apakah proses login siswa berhasil atau tidak.
3. API List Ujian
Berfungsi untuk menampilkan daftar ujian yang telah diberikan kepada siswa.
4. API Mulai Ujian
Digunakan agar siswa dapat memulai ujian yang tersedia pada aplikasi.
5. API Daftar Soal Ujian
Berfungsi untuk mengakses daftar soal yang terdapat dalam ujian.
6. API Soal Ujian
Digunakan untuk menampilkan detail dari salah satu soal yang ada dalam daftar soal ujian.
7. API Submit Jawaban Ujian
Berfungsi untuk mengirimkan jawaban siswa terhadap soal-soal ujian yang telah dikerjakan.

Tabel 4. 1 API Service

API Service	Akses API Service	Method	Service
API Login	$\${serviceLogin}/student/login$	POST	Auth Service
API Status Login	$\${serviceLogin}/auth/status$	GET	Auth Service
API List Ujian	$\${serviceUjian}/exam/v2/jadwal-ujian-dashboard?page=1&size=4$	GET	Exam Service
API Mulai Ujian	$\${serviceUjian}/exam/v2/test/start-ujian/\${IdUjian}$	GET	Exam Service

API List Soal Ujian	$\${serviceUjian}/exam/v2/daftar-soal?examId=\${IdUjian}&packageId=\${IdPaket}$	GET	Exam Monit Service
API Soal Ujian	$\${serviceUjian}/exam/v2/test/getSoal?idUjian=\${IdUjian}&idPaket=\${IdPaket}&idSoal=\${IdSoal}$	GET	Exam Monit Service
API Submit Jawaban	$\${serviceUjian}/exam/v2/test/store-one-non-akm?idUjian=\${IdUjian}&idPaket=\${IdPaket}&idSoal=\${IdSoal}$	POST	Exam Monit Service

4.3 Apache JMeter

Data API Service yang di dapat akan di proses menggunakan Apache JMeter. Pembuatan scenario pengujian *performance* dari API yang di dapat menggunakan GUI yang di download dari https://jmeter.apache.org/download_jmeter.cgi.

Dalam melakukan pengujian menggunakan Apache JMeter ada beberapa tahap yang perlu dilakukan seperti pada Gambar 4. 1.



Gambar 4. 1 Flowchart Pengujian Apache JMeter

Langkah-langkah pelaksanaan pengujian dengan Apache JMeter pada Gambar 4.1 disajikan sebagai berikut:

1. Dimulai dari perancangan pengujian yang terdiri dari persiapan API yang akan di uji dan persiapan tools Apache JMeter yang dapat di install dari <https://jmeter.apache.org/>
2. Membuat Thread Group dengan menentukan jumlah *Virtual Users / Concurrent users*, *Ramp-up time* dan juga durasi pengujian.
3. Menambahkan HTTP Request dari URL endpoint API dan juga method (GET/POST/ PUT) yang telah ditentukan untuk pengujian serta menambahkan HTTP Header supaya API tersebut bisa di akses seperti token, JSON, dll.
4. Menjalankan pengujian yang telah dibuat dengan cara menjalankan dari GUI atau dapat juga melalui *Command Line Mode (Non GUI)*
5. Mengekspor Hasil Pengujian dengan menyimpan output pengujian ke file CSV/Excel dengan mengambil nilai *avg response time*, *min response time*, *response time 95th Percentile*, *response time 99th Percentile*, *error*, dan *throughput*.

4.3.1 Skenario Pengujian Apache JMeter

Skenario pembuatan pengujian API dimulai dari penentuan environment yang akan digunakan, yaitu environment untuk layanan Login $\{serviceLogin\}$ dan layanan Ujian $\{serviceUjian\}$.

Pembuatan environment ini merupakan langkah awal dalam proses pengujian untuk memastikan bahwa setiap layanan API memiliki konfigurasi yang sesuai sebelum dilakukan pengujian performa. Rincian skenario pembuatan environment tersebut dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Enviroment Service Plan

Setelah pembuatan environment, langkah selanjutnya adalah pembuatan skenario pengujian untuk API Login, seperti yang ditampilkan pada Gambar 4.3.

API Login ini akan dikirimkan request untuk memperoleh token autentikasi siswa yang berhasil login ke dalam aplikasi Pijar Sekolah. Token tersebut akan digunakan untuk mengakses layanan-layanan API lainnya dalam rangkaian pengujian.



Gambar 4. 3 Skenario Apache JMeter API Login

Setelah skenario untuk API Login dibuat, langkah selanjutnya adalah pembuatan skenario pengujian untuk API Status Login, seperti yang ditampilkan pada Gambar 4.4.

API Status Login ini berfungsi untuk memperoleh ID peserta ujian atau ID siswa yang berhasil melakukan login ke dalam aplikasi. Data ID ini akan digunakan pada tahap-tahap pengujian berikutnya.



Gambar 4. 4 Skenario Apache JMeter API Status Login

Selanjutnya dilakukan pembuatan skenario pengujian yang berkaitan dengan layanan ujian, dimulai dari API List Jadwal Ujian. API ini diakses untuk memperoleh ID ujian yang akan digunakan pada pengujian selanjutnya, yaitu API Mulai Ujian, API List Soal, API Soal, dan API Submit Jawaban. Skenario untuk API List Jadwal Ujian ditampilkan pada Gambar 4.5.



Gambar 4. 5 Skenario Apache JMeter API Jadwal Ujian

Setelah skenario untuk API List Jadwal Ujian berhasil dibuat, langkah selanjutnya adalah pembuatan skenario pengujian untuk API Mulai Ujian. API ini digunakan untuk memperoleh ID paket dan ID sertifikasi, yang diperlukan pada pengujian berikutnya, yaitu API List Soal, API Soal, dan API Submit Jawaban. Skenario untuk API Mulai Ujian dapat dilihat pada Gambar 4.6



Gambar 4. 6 Skenario Apache JMeter API Mulai Ujian

Selanjutnya dilakukan pembuatan skenario pengujian untuk API List Soal Ujian. API ini akan dikirimkan request untuk memperoleh ID soal ujian, yang selanjutnya akan digunakan dalam API Soal Ujian dan API Submit Jawaban. Skenario untuk API List Soal Ujian dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Skenario Apache JMeter API List Soal Ujian

Setelah ID soal berhasil diperoleh, langkah selanjutnya adalah pembuatan skenario pengujian untuk API Soal Ujian. API ini akan dikirimkan request untuk mendapatkan data detail soal ujian, yang akan digunakan dalam pembuatan skenario pengujian pada API Submit Jawaban. Skenario untuk API Soal Ujian dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Skenario Apache JMeter API Soal Ujian

Terakhir, dibuat skenario pengujian untuk API Submit Jawaban, yang digunakan untuk menyimpan data jawaban ujian yang telah dikerjakan oleh siswa. Skenario untuk API Submit Jawaban dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Skenario Apache JMeter API Submit Jawaban

4.3.2 Eksekusi Apache JMeter

Setelah proses pembuatan skenario pengujian menggunakan Apache JMeter terhadap layanan-layanan API yang akan diuji, langkah selanjutnya adalah menjalankan skenario tersebut menggunakan perintah baris (*command line*).

Berikut adalah sintaks dasar yang digunakan untuk menjalankan file .jmx pada Apache JMeter:

```
jmeter -n -t [jmx file] -l [results file] -e -o [path to web report folder]
```

Pengujian dilakukan dengan mensimulasikan jumlah siswa yang banyak yang secara bersamaan melakukan ujian melalui aplikasi Pijar Sekolah. Untuk menjalankan pengujian tersebut, digunakan file skenario bernama FileTesisJMeter.jmx dengan perintah sebagai berikut:

```
jmeter -n -t "/TesisJmeter/FileTesisJMeter.jmx" -l  
"/TesisJmeter/hasil/generate.csv" -e -o "/TesisJmeter/hasil/report1"
```

Perintah ini akan:

- Menjalankan JMeter dalam mode non-GUI (-n)
- Menggunakan file skenario FileTesisJMeter.jmx (-t)
- Menyimpan hasil pengujian dalam format CSV (-l)
- Menghasilkan laporan web otomatis (-e -o)

Berdasarkan data pada Lampiran 1 hasil pengujian Apache JMeter sebanyak 30 pengujian (*testing*). JMeter menunjukkan kinerja yang cukup stabil, ditandai dengan jumlah *error* yang relatif kecil (sebagian besar bernilai 0) pada setiap skenario pengujian seperti *Login*, *Status Login*, *List Jadwal Ujian*, *Mulai Ujian*, *Daftar Soal Ujian*, *Soal Ujian*, dan *Submit Jawaban*.

Rata-rata waktu respons (*Average Response Time*) bervariasi antara sekitar 500 ms hingga 6.000 ms, tergantung pada kompleksitas skenario dan beban pengguna yang diuji. Nilai *minimum* dan *maximum response time* menunjukkan adanya fluktuasi, yang wajar terjadi pada pengujian performa berbasis beban tinggi.

Persentil ke-95 dan ke-99 (*95th Percentile* dan *99th Percentile*) menunjukkan konsistensi performa dalam menangani sebagian besar permintaan,

walaupun beberapa pengujian menghasilkan lonjakan waktu respons tinggi pada skenario yang lebih kompleks.

Nilai *Throughput (Request per Second)* umumnya berada di kisaran 2–13, menandakan kemampuan JMeter dalam mempertahankan laju permintaan yang cukup baik di bawah berbagai kondisi beban.

4.3.3 Analisis Hasil Apache JMeter

Berdasarkan hasil 30 kali pengujian menggunakan Apache JMeter, diperoleh rata-rata (*mean*) dan simpangan baku (*standard deviation*) dari setiap metrik utama sebagai berikut:

Tabel 4. 2 Mean Hasil Pengujian Apache JMeter

API Request	Error %	Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	Throughput (req/s)
Login	1,10	2476,38	589,20	10321,40	4690,30	7013,86	5,10
Status Login	1,13	1899,56	107,80	8101,23	4423,69	6218,83	5,22
List Jadwal Ujian	1,43	2391,25	391,10	9553,50	4831,32	6723,64	5,28
Mulai Ujian	1,47	1775,82	79,53	7865,67	4353,41	6420,31	5,48
Daftar Soal Ujian	1,50	1793,26	122,33	8651,10	4208,14	6307,81	5,56
Soal Ujian	1,50	1843,85	72,13	8896,93	4208,38	6602,34	5,67
Submit Jawaban	1,77	1810,19	49,70	7900,10	4219,88	6308,71	5,79

Hasil *mean* pada pengujian menggunakan Apache JMeter yang ditunjukkan pada Tabel 4.2. Berdasarkan hasil pengujian performa menggunakan JMeter, nilai *mean Error Rate* pada seluruh API berada pada kisaran 1,10% hingga 1,77%. Nilai ini menunjukkan bahwa tingkat kesalahan yang terjadi selama proses permintaan relatif rendah dan masih dalam batas wajar. API *Login* memiliki rata-rata error terendah sebesar 1,10%, sedangkan nilai tertinggi terdapat pada API *Submit Jawaban* sebesar 1,77%.

Pada metrik *Average Response Time*, seluruh API menunjukkan nilai rata-rata antara 1.775,82 ms hingga 2.476,38 ms. API *Login* memiliki waktu tanggap rata-rata paling tinggi yaitu 2.476,38 ms, sedangkan waktu tanggap rata-rata tercepat terjadi pada API *Mulai Ujian* dengan nilai 1.775,82 ms. Perbedaan ini mengindikasikan bahwa proses otentikasi pada API *Login* membutuhkan waktu lebih panjang dibandingkan API lainnya.

Untuk *Minimum Response Time*, nilai *mean* yang diperoleh berkisar antara 49,70 ms sampai 589,20 ms. API *Submit Jawaban* menunjukkan waktu respon minimum paling cepat (49,70 ms), sedangkan API *Login* kembali menjadi yang paling lambat dengan nilai minimum 589,20 ms. Hal ini menunjukkan bahwa API *Login* memiliki proses awal yang lebih kompleks.

Pada metrik *Maximum Response Time*, nilai *mean* berada pada rentang 7.865,67 ms hingga 10.321,40 ms. API *Login* menunjukkan nilai maksimum tertinggi, sementara API *Mulai Ujian* memiliki nilai maksimum terendah. Ini menandakan bahwa beberapa request pada API *Login* dapat mengalami lonjakan waktu proses yang jauh lebih tinggi dibandingkan API lainnya.

Nilai *mean 95th Percentile* berada pada kisaran 4.208,14 ms hingga 4.831,32 ms. *API List Jadwal Ujian* mencatat nilai *95th percentile* tertinggi, sedangkan *API Daftar Soal Ujian* memiliki nilai terendah. Secara umum, sebagian besar permintaan pada seluruh API masih berada di bawah kisaran 4 hingga 4,8 detik pada persentil tinggi.

Untuk *99th Percentile*, nilai mean berada pada kisaran 6.218,83 ms hingga 7.013,86 ms. *API Login kembali* menunjukkan nilai *99th percentile* tertinggi dengan 7.013,86 ms, mengindikasikan adanya kemungkinan *bottleneck* pada kondisi beban ekstrem.

Terakhir, pada metrik *Throughput*, seluruh API memiliki kemampuan memproses request antara 5,10 hingga 5,79 request per detik. *API Submit Jawaban* memiliki *throughput* tertinggi sebesar 5,79 req/s, sedangkan *API Login* berada pada posisi terendah dengan nilai 5,10 req/s. Temuan ini konsisten dengan metrik waktu respon, di mana *API Login* cenderung memiliki performa paling berat.

Tabel 4. 3 Standard Deviation Apache JMeter

API Request	Error	Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	Throughput (req/s)	
Login	Mean	0,3	2476,38	589,2	10321,4	4690,3	7013,86	5,1
	StdDev	1,02	1163,33	412,92	6944,1	2918,94	4153,95	1,86
	K-	0,72	1313,05	176,28	3377,3	1771,36	2859,91	3,23
	K+	1,32	3639,71	1002,12	17265,5	7609,24	11167,8	6,96
Status Login	Mean	0,33	1899,56	107,8	8101,23	4423,69	6218,83	5,22
	StdDev	1,03	1294,69	77,69	6699,06	3086,35	4855,67	1,99
	K-	-0,7	604,87	30,11	1402,17	1337,34	1363,16	3,23
	K+	1,36	3194,24	185,49	14800,29	7510,05	11074,5	7,22

List Jadwal Ujian	Mean	1,13	2391,25	391,1	9553,5	4831,32	6723,64	5,28
	StdDev	5,13	957,11	382,86	6571,25	2751,78	4933	2,1
	K-	-4	1434,14	8,24	2982,25	2079,54	1790,64	3,18
	K+	6,26	3348,36	773,96	16124,75	7583,09	11656,64	7,37
Mulai Ujian	Mean	1,47	1775,82	79,53	7865,67	4353,41	6420,31	5,48
	StdDev	5,37	1262,5	93,19	6295,1	2953,63	5585,35	2,25
	K-	3,91	513,32	-13,66	1570,57	1399,78	834,97	3,23
	K+	6,84	3038,32	172,73	14160,77	7307,05	12005,66	7,74
Daftar Soal Ujian	Mean	1,5	1793,26	122,33	8651,1	4208,14	6307,81	5,56
	StdDev	5,37	1340,95	397,64	6690,17	3191,01	6159,18	2,29
	K-	3,87	452,31	-275,31	1960,93	1017,14	148,63	3,27
	K+	6,87	3134,21	519,97	15341,27	7399,15	12467	7,86
Soal Ujian	Mean	1,5	1843,85	72,13	8896,93	4208,38	6602,34	5,67
	StdDev	5,37	1316,3	144,97	7081,43	2907,95	5613,95	2,37
	K-	3,87	527,56	-72,84	1815,51	1300,43	988,39	3,3
	K+	6,87	3160,15	217,1	15978,36	7116,33	12216,29	8,05
Submit Jawaban	Mean	1,77	1810,19	49,7	7900,1	4219,88	6308,71	5,79
	StdDev	5,39	1208,8	15,01	6495,31	2948,5	5005,29	2,46
	K-	3,62	601,39	34,69	1404,79	1271,38	1303,42	3,32
	K+	7,15	3018,99	64,71	14395,41	7168,38	11314,01	8,25

Dalam konteks *performance testing*, standar deviasi pada Tabel 4.3 menunjukkan konsistensi waktu respons — seberapa stabil API dalam merespons permintaan pengguna.

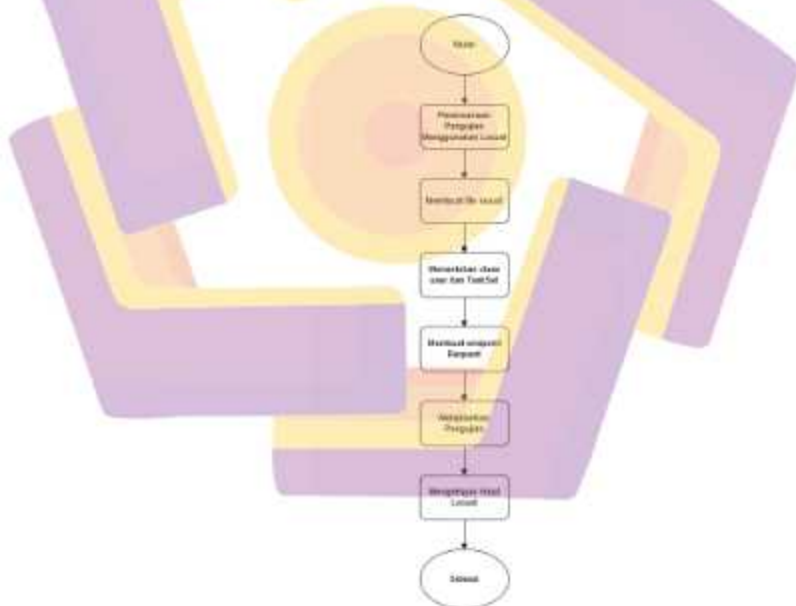
- Nilai standar deviasi kecil → waktu respons stabil dan sistem konsisten.

- Nilai standar deviasi besar → fluktuasi tinggi, artinya ada permintaan yang sangat cepat dan ada juga yang sangat lambat (tidak stabil).

4.4 Locust

Data layanan API Service yang telah diperoleh selanjutnya akan diproses menggunakan Locust. Pembuatan skenario pengujian performa terhadap layanan-layanan API tersebut dilakukan dengan menggunakan script yang ditulis dalam bahasa pemrograman Python.

Pada pelaksanaan pengujian menggunakan locust ada beberapa tahap yang perlu dilakukan seperti pada Gambar 4.10



Gambar 4. 10 Flowchart Pengujian Locust

Langkah-langkah pelaksanaan pengujian dengan Locust pada Gambar 4.10 disajikan sebagai berikut:

1. Dimulai dari perancangan pengujian yang terdiri dari persiapan API yang akan di uji dan persiapan tools tools Locust dengan menginstall Python 3.x kemudian install locust dengan menggunakan Command `pip install locust`.
2. Membuat locust file dengan extension .py seperti `locustfile.py` dan install library yang di butuhkan seperti : `Json, CSV, pandas, dll`.
3. File `locustfile.py` mendeskripsikan API yang diuji dan mengatur header serta payload yang dibutuhkan di masukan kedalam class karena locust menggunakan *class-based approach* untuk mendeskripsikan perilaku user. Membuat *task-set* yang dianggap sebagai *task* yang akan dijalankan berulang oleh *virtual user*.
4. Didalam *Task-Set* menentukan endpoint API request dengan method (`GET/POST/PUT`) yang akan di uji.
5. Ketika endpoint API Request sudah di buat kedalam satu file locust, Skenario tersebut dapat dijalankan dengan perintah `locust -f locustfile.py`.
6. Mengekspor Hasil Pengujian Locust dengan menyimpan output pengujian ke file CSV/Excel dengan mengambil nilai *avg response time, min response time, response time 95th Percentile, response time 99th Percentile, error, dan throughput*.

4.4.1 Skenario Pengujian Locust

Berikut ini adalah kode pengujian yang telah dibuat menggunakan Locust, dengan alur skenario yang sama seperti yang telah diterapkan pada pengujian menggunakan Apache JMeter.

Skenario pengujian dimulai dari:

- API Login
- API Status Login
- API List Jadwal Ujian

- API Mulai Ujian
- API List Soal Ujian
- API Soal Ujian
- API Submit Jawaban

Semua API tersebut diuji secara berurutan untuk mensimulasikan proses ujian oleh siswa melalui aplikasi Pijar Sekolah, dengan menggunakan pendekatan berbasis kode Python pada Locust.

Script Python untuk menguji performance API menggunakan Locust

```
import json
import csv
from locust import HttpUser, TaskSet, task, between

# ----- 1 LOAD DATA USER DARI FILE CSV -----
# Format CSV harus berisi kolom: username,password
with open(r"C:\Users\andri\Music\IS_Locust\Ujian_Siswa>Login\user_data_siswa.csv",
newline="", encoding="utf-8") as csvfile:
    reader = csv.DictReader(csvfile)
    user_data_list = list(reader)

# Global counter agar tiap user ambil baris berbeda dari CSV
user_counter = 0

class MyTaskSet(TaskSet):
    def on_start(self):
        """Dipanggil ketika virtual user dimulai"""
        global user_counter

        # Ambil data user berdasarkan urutan (berulang jika VU > jumlah data)
        self.user_index = user_counter % len(user_data_list)
        user_counter += 1
        self.user_data = user_data_list[self.user_index]

        # Host API
        self.host1 = "https://auth-api.pijarsekolah.id"
        self.host2 = "https://api.pijarsekolah.id"

        # Jalankan login di awal
        self.token = self.login_siswa()
        if self.token:
            self.setup_ids()
        else:
            print(f"🚫 Login gagal untuk user: {self.user_data['username']}")

        # Simpan urutan task
        self.current_task = 0
        self.tasks_list = [
            self.status_siswa,
```

```

        self.list_ujian,
        self.mulai_ujian,
        self.daftar_soal,
        self.get_soal,
        self.submit_jawaban
    ]

# ----- LOGIN SISWA -----
def login_siswa(self):
    payload = {
        "username": self.user_data["username"],
        "password": self.user_data["password"],
        "npsn": "66116611"
    }
    headers = {
        "Content-Type": "application/json",
        "Authorization": "Basic *****"
    }

    with self.client.post(f'{self.host1}/student/login', headers=headers, json=payload,
catch_response=True) as response:
        if response.status_code == 200:
            try:
                data = response.json()
                token = data.get("data", {}).get("token")
                if token:
                    print(f"✅ Login sukses: {self.user_data['username']}")
                    response.success()
                    return token
                else:
                    response.failure("Token tidak ditemukan")
            except json.JSONDecodeError:
                response.failure("Invalid JSON response saat login")
            else:
                response.failure(f"Login gagal ({response.status_code}) - {self.user_data['username']}")
                return None

# ----- SETUP ID UJIAN -----
def setup_ids(self):
    self.idUjian = self.list_ujian()
    self.idSertifikasi = self.idUjian
    self.idPeserta = self.idUjian
    self.idPaket = self.mulai_ujian()
    self.idSoal = self.daftar_soal()

# ----- TASK 1: STATUS SISWA -----
@task
def run_tasks_in_order(self):
    """Menjalankan task berurutan"""
    if self.current_task < len(self.tasks_list):
        self.tasks_list[self.current_task]()
        self.current_task += 1
    else:
        self.current_task = 0

def status_siswa(self):
    if not self.token:

```

```

        return
        headers = {"Authorization": f"Bearer {self.token}"}
        with self.client.get(f"{self.host1}/student/status", headers=headers, catch_response=True)
as response:
    if response.status_code == 200:
        response.success()
    else:
        response.failure(f"Status siswa gagal: {response.status_code}")

# ----- TASK 2: LIST UJIAN -----
def list_ujian(self):
    if not self.token:
        return None
    headers = {"Authorization": f"Bearer {self.token}"}
    with self.client.get(f"{self.host2}/exam/v2/jadwal-ujian-dashboard?page=1&size=4",
        headers=headers, catch_response=True) as response:
        if response.status_code == 200:
            try:
                dataUjian = response.json().get("data", [])
                if not dataUjian:
                    response.failure("Data ujian kosong")
                    return None
                idUjian = dataUjian[0].get("id_ujian")
                print(f"👤 {self.user_data['username']} - idUjian: {idUjian}")
                response.success()
                return idUjian
            except Exception:
                response.failure("Error parsing list ujian")
        else:
            response.failure(f"Gagal ambil jadwal ujian ( {response.status_code} )")
    return None

# ----- TASK 3: MULAI UJIAN -----
def mulai_ujian(self):
    if not self.token or not self.idUjian:
        return None
    headers = {"Authorization": f"Bearer {self.token}"}
    with self.client.get(f"{self.host2}/exam/v2/test/start-ujian/{self.idUjian}",
        headers=headers, catch_response=True) as response:
        if response.status_code == 200:
            try:
                meta = response.json().get("meta", {})
                idPaket = meta.get("idPaket")
                if idPaket:
                    print(f"👉 idPaket: {idPaket}")
                    response.success()
                    return idPaket
            except:
                response.failure("idPaket tidak ditemukan")
        except json.JSONDecodeError:
            response.failure("Invalid JSON response")
        else:
            response.failure(f"Gagal start ujian: {response.status_code}")
    return None

# ----- TASK 4: DAFTAR SOAL -----
def daftar_soal(self):
    if not self.token or not self.idUjian or not self.idPaket:

```

```

return None
headers = {"Authorization": f"Bearer {self.token}"}
params = {"examId": self.idUjian, "packageId": self.idPaket}
with self.client.get(f"{self.host2}/exam/v2/daftar-soal", params=params,
                    headers=headers, catch_response=True) as response:
    if response.status_code == 200:
        try:
            soal = response.json().get("data", [])
            if not soal:
                response.failure("Daftar soal kosong")
                return None
            idSoal = soal[0].get("_id")
            print(f"👉 idSoal: {idSoal}")
            response.success()
            return idSoal
        except json.JSONDecodeError:
            response.failure("Invalid JSON response daftar soal")
    else:
        response.failure(f"Gagal ambil daftar soal ({response.status_code}")
    return None

# ----- TASK 5: GET SOAL -----
def get_soal(self):
    if not self.token:
        return
    headers = {"Authorization": f"Bearer {self.token}"}
    params = {"idUjian": self.idUjian, "idPaket": self.idPaket, "idSoal": self.idSoal}
    with self.client.get(f"{self.host2}/exam/v2/test/getSoal",
                        params=params, headers=headers, catch_response=True) as response:
        if response.status_code == 200:
            response.success()
        else:
            response.failure(f"Gagal ambil soal: {response.status_code}")

# ----- TASK 6: SUBMIT JAWABAN -----
def submit_jawaban(self):
    payload = {
        "idPeserta": self.idPeserta,
        "idUjian": self.idUjian,
        "idSertifikasi": self.idSertifikasi,
        "idPaket": self.idPaket,
        "questionId": self.idSoal,
        "type_soal": "pil_ganda",
        "answer": {"submission": "<p>Matahari</p>", "choice": 1},
        "ragu": 0,
        "no_soal": 1,
        "lastAnswer": False
    }
    headers = {
        "Content-Type": "application/json",
        "Authorization": f"Bearer {self.token}"
    }

    with self.client.post(f"{self.host2}/exam/v2/test/store-one-non-akm", headers=headers,
                          json=payload, catch_response=True) as response:
        if response.status_code == 200:
            print("👉 Jawaban tersubmit")
            response.success()

```

```
else:
    response.failure("Gagal submit jawaban")

class MyUser(HttpUser):
    tasks = [MyTaskSet]
    wait_time = between(1, 3)
```

Script ini digunakan untuk melakukan pengujian performa (*load testing*) pada REST API ujian siswa (Pijar Sekolah) menggunakan Locust. Setiap virtual user (VU) akan:

1. Login sebagai siswa (data login diambil dari CSV),
2. Mengakses serangkaian endpoint API ujian secara berurutan,
3. Mengirimkan jawaban ke server.

4.4.2 Eksekusi Skenario pengujian performance dengan Locust

Setelah pembuatan scenario pengujian menggunakan Locust terhadap API yang akan di uji berikut script untuk run file .py yang sudah di buat.

Pengujian dilakukan dengan mensimulasikan jumlah user siswa yang banyak saat melakukan ujian menggunakan pijar sekolah dengan menjalankan script FileLocust.py.

Berdasarkan hasil 30 kali pengujian pada Lampiran 2 menggunakan Locust, dapat disimpulkan bahwa performa sistem menunjukkan kestabilan yang baik dengan tingkat error yang sangat rendah hingga hampir tidak ada. Hal ini mengindikasikan bahwa sistem mampu menanggapi beban uji dengan efektif tanpa banyak kegagalan permintaan.

Nilai rata-rata waktu respons (*Average Response Time*) menunjukkan variasi tergantung pada tingkat beban. Pada saat jumlah pengguna sedikit, waktu respons relatif cepat, namun mengalami peningkatan seiring dengan bertambahnya jumlah pengguna. Meskipun demikian, peningkatan tersebut masih berada dalam batas yang wajar dan tidak menimbulkan gangguan signifikan terhadap kestabilan sistem.

Nilai persentil ke-95 dan ke-99 juga memperlihatkan bahwa sebagian besar permintaan diproses dengan waktu respons yang konsisten, meskipun terdapat beberapa *outlier* pada saat beban tinggi.

Selain itu, nilai *Throughput* (*Request per Second*) cenderung stabil di setiap pengujian, menunjukkan bahwa sistem mampu mempertahankan *throughput* secara konsisten.

Secara keseluruhan, hasil pengujian menggunakan *Locust* menunjukkan bahwa sistem memiliki performa yang stabil, waktu respons yang cukup baik, serta tingkat keandalan tinggi dalam menangani beban uji yang bervariasi.

4.4.3 Analisis Hasil *Locust*

Hasil pengujian dengan *Locust* menunjukkan karakteristik yang sedikit berbeda karena arsitektur asynchronous berbasis event loop. Ringkasan hasil statistik dari 30 kali pengujian disajikan berikut ini:

Tabel 4. 4 Mean Hasil Pengujian *Locust*

API Request	Error %	Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	Throughput (req/s)
Login	1,37	8630,38	203,63	44224,20	30028,33	39360,00	0,99

Status Login	1,33	2150,04	40,50	27825,55	7595,70	12570,33	8,80
List Jadwal Ujian	2,03	2623,48	38,41	40566,80	9947,67	22904,00	9,59
Mulai Ujian	1,50	2646,81	36,12	22154,27	6770,33	6945,29	9,43
Daftar Soal Ujian	1,27	2622,34	35,33	27378,90	6664,31	7178,71	9,33
Soal Ujian	1,53	2849,42	37,73	25729,23	7119,16	7275,95	8,23
Submit Jawaban	1,60	2090,87	43,13	20112,28	7326,13	11517,67	8,09

Pengujian performa dilakukan menggunakan tool Locust dengan beban tinggi yang melibatkan ratusan *Concurrent Users* (CCU). Data pada Tabel 4.6 yang diperoleh merupakan hasil perhitungan rata-rata (*mean*) dari 30 kali pengujian untuk setiap endpoint utama pada layanan API sistem.

Hasil pengujian performa dengan menggunakan locust menunjukkan bahwa setiap endpoint API memiliki karakteristik waktu respon dan tingkat error yang berbeda. Endpoint Login memiliki performa paling berat dengan rata-rata waktu respon sebesar 8630,38 ms serta nilai *percentile* ke-95 mencapai 30.028,33 ms, yang menunjukkan adanya latensi tinggi pada sebagian besar permintaan. Selain itu, nilai maksimum mencapai 44.224,20 ms, sehingga endpoint ini dapat dikategorikan sebagai bottleneck utama. Meskipun persentase error relatif kecil (1,37%), throughput-nya hanya 0,99 request per detik, terendah dibanding endpoint lain.

Endpoint Status Login memiliki rata-rata waktu respon sebesar 2150,04 ms dengan tingkat *error* 1,33% dan *throughput* 8,80 request per detik. Meskipun lebih stabil dibanding endpoint Login, masih terdapat lonjakan waktu respon hingga 27.825,55 ms, terlihat dari nilai maksimum dan nilai *percentile* ke-95 sebesar 7595,70 ms.

Endpoint List Jadwal Ujian menunjukkan rata-rata waktu respon sebesar 2623,48 ms dengan tingkat *error* 2,03%, tertinggi di antara semua endpoint. Nilai maksimum mencapai 40.566,80 ms, dan nilai *percentile* ke-95 sebesar 9947,67 ms, menandakan adanya variasi waktu respon yang cukup besar, meskipun *throughput* termasuk tinggi yaitu 9,59 request per detik.

Selanjutnya, endpoint Mulai Ujian memiliki rata-rata waktu respon 2646,81 ms dengan tingkat *error* 1,50%. Nilai maksimum mencapai 22.154,27 ms, sementara nilai *percentile* ke-95 berada pada 6770,33 ms, dan *throughput* sebesar 9,43 request per detik, menunjukkan bahwa endpoint ini cukup stabil namun masih memiliki latensi menengah.

Endpoint Daftar Soal Ujian menghasilkan rata-rata waktu respon 2622,34 ms, dengan tingkat *error* paling rendah yaitu 1,27%. Nilai maksimum berada pada 27.378,90 ms, sedangkan nilai *percentile* ke-95 adalah 6664,31 ms. Dengan *throughput* 9,33 request per detik, endpoint ini tergolong stabil dan efisien.

Untuk endpoint Soal Ujian, rata-rata waktu respon tercatat 2849,42 ms dengan tingkat *error* 1,53%, serta *throughput* 8,23 request per detik. Latensi masih terlihat dari nilai maksimum 25.729,23 ms dan nilai *percentile* ke-95 sebesar 7119,16 ms, menandakan adanya beban proses yang lebih besar saat memuat soal.

Endpoint Submit Jawaban merupakan salah satu yang paling cepat dengan rata-rata waktu respon 2090,87 ms dan *throughput* 8,09 request per detik. Nilai error berada pada 1,60%, sedangkan nilai maksimum mencapai 20.112,28 ms. Dengan nilai *percentile* ke-95 sebesar 7326,13 ms, endpoint ini masih memiliki beberapa permintaan lambat, namun performanya lebih baik dibanding beberapa endpoint lainnya.

Tabel 4. 5 Standard Deviation Locust

API Request	Error	Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	Throughput (req/s)	
Login	Mean	1	8630,38	203,63	44224,2	30028,33	39360	0,99
	StdDev	2	4829,31	152,35	18098,41	13728,31	16370,72	0,22
	K-	-1	3801,07	51,28	26125,79	16300,02	22989,28	0,77
	K+	4	13459,7	355,98	62322,62	43756,64	55730,72	1,21
Status Login	Mean	1,33	2150,04	40,5	27825,55	7595,7	12570,33	8,8
	StdDev	3,51	1225,56	10,64	14038,97	3119,38	5233,03	3,38
	K-	-	924,48	29,86	13786,58	4476,32	7337,3	5,42
	K+	2,17	4,84	3375,6	51,14	41864,51	10715,08	17803,36
List Jadwal Ujian	Mean	2,03	2623,48	38,41	40566,8	9947,67	22904	9,59
	StdDev	4,73	1599,39	13,81	16154,46	5439,46	13606,55	3,39
	K-	-2,7	1024,1	24,6	24412,34	4508,2	9297,45	6,19
	K+	6,76	4222,87	52,23	56721,26	15387,13	36510,55	12,98
Mulai Ujian	Mean	1,5	2646,81	36,12	22154,27	6770,33	6945,29	9,43
	StdDev	4,2	1536,79	11,72	15845,85	2946,65	2958,55	3,4
	K-	-2,7	1110,02	24,4	6308,42	3823,68	3986,74	6,03
	K+	5,7	4183,6	47,84	38000,13	9716,97	9903,84	12,83
Daftar Soal Ujian	Mean	1,27	2622,34	35,33	27378,9	6664,31	7178,71	9,33
	StdDev	3,48	1584,96	8,97	16416,12	3192,88	3487,95	3,49
	K-	-	1037,38	26,36	10962,79	3471,43	3690,75	5,83
	K+	2,22	4,75	4207,3	44,31	43795,02	9857,18	10666,66

Soal Ujian	Mean	1,53	2849,42	37,73	25729,23	7119,16	7275,95	8,23
	StdDev	4,17	1660,78	15,16	12587,51	3167,59	3197,2	3,37
	K-	2,63	1188,64	22,57	13141,72	3951,57	4078,75	4,86
	K+	5,7	4510,2	52,9	38316,74	10286,76	10473,15	11,59
Submit Jawaban	Mean	1,6	2090,87	43,13	20112,28	7326,13	11517,67	8,09
	StdDev	4,26	1191,19	12,88	11865,91	3024,44	5106,92	3,37
	K-	2,66	899,68	30,25	8246,37	4301,69	6410,75	4,72
	K+	5,86	3282,06	56,01	31978,18	10350,57	16624,58	11,46

Standar deviasi untuk pengujian *performance* API menggunakan Locust pada Tabel 4.4 Standar deviasi pada metrik error menunjukkan seberapa besar fluktuasi jumlah permintaan yang gagal selama pengujian. Nilai yang tinggi mengindikasikan ketidakstabilan layanan, karena tingkat kegagalan tidak konsisten pada tiap siklus pengujian. Semakin rendah standar deviasi error, semakin stabil API dalam menangani beban. Standar deviasi pada waktu respons rata-rata menggambarkan variasi waktu respons antar permintaan. Nilai StdDev yang besar (lebih dari 1 detik) menunjukkan bahwa performa API tidak konsisten—kadang merespons cepat, namun pada kondisi tertentu terjadi lonjakan waktu respons yang signifikan. Semakin besar StdDev, semakin tidak stabil performa API.

Percentile 95 menunjukkan waktu respons terburuk yang masih dialami oleh 95% request. Jika StdDev P95 tinggi, berarti nilai P95 berubah-ubah secara signifikan antara satu interval pengujian dan interval lainnya, menandakan performa API sering mengalami lonjakan beban.

Percentile 99 mewakili waktu respons pada kondisi yang paling berat. StdDev yang besar pada P99 mengindikasikan bahwa kondisi terburuk API

berubah-ubah dan tidak konsisten. Ini menunjukkan gangguan signifikan pada sebagian kecil request, yang dapat menyebabkan slowdown sistem jika terjadi secara terus-menerus.

Standar deviasi *throughput* menunjukkan variasi jumlah request yang dapat diproses API per detik. StdDev yang tinggi menandakan kapasitas pemrosesan tidak stabil, kadang API mampu menangani banyak request, namun pada kondisi lain terjadi penurunan tajam.

4.5 Analisa dan Perbandingan

Untuk mengetahui apakah terdapat perbedaan yang signifikan antara hasil pengujian performa API menggunakan Apache JMeter dan Locust, dilakukan analisis statistik non-parametrik menggunakan Uji Wilcoxon Signed-Rank Test. Metode ini digunakan karena data hasil pengujian tidak berdistribusi normal serta berasal dari dua sampel yang saling berpasangan, yaitu hasil pengujian pada endpoint yang sama menggunakan dua alat yang berbeda.

Uji Wilcoxon dilakukan terhadap variabel utama yang menjadi indikator performa, yaitu *Average Response Time*, *Error Rate*, dan *Throughput*. Kriteria pengambilan keputusan adalah sebagai berikut:

- Jika nilai $p\text{-value} < 0,05$, maka terdapat perbedaan signifikan antara hasil pengujian menggunakan JMeter dan Locust.
- Jika $p\text{-value} \geq 0,05$, maka tidak terdapat perbedaan signifikan di antara keduanya.

Analisis dilakukan menggunakan bahasa pemrograman Python, memanfaatkan pustaka statistik `scipy.stats`.

Script Python untuk mengeksekusi hasil uji dengan menggunakan Wilcoxon

```
from scipy.stats import wilcoxon
import pandas as pd
import numpy as np

# -----
# Hitung Z Wilcoxon (N > 25)
# -----

def compute_wilcoxon_z(x, y):
    diffs = x - y
    diffs_nonzero = diffs[diffs != 0]
    n = len(diffs_nonzero)

    # ranking absolute differences
    ranks = pd.Series(np.abs(diffs_nonzero)).rank()

    # jumlahkan rank bertanda positif & negatif
    W_plus = ranks[diffs_nonzero > 0].sum()
    W_minus = ranks[diffs_nonzero < 0].sum()

    # Statistik W = nilai terkecil
    W = min(W_plus, W_minus)

    # mean dan standar deviasi distribusi normal aproksimasi
    mu_W = n * (n + 1) / 4
    sigma_W = np.sqrt(n * (n + 1) * (2 * n + 1) / 24)

    # Z statistic
    Z = (W - mu_W) / sigma_W

    return W, Z

# -----
# Interpretasi dan Output
# -----

def interpret_wilcoxon_result(result, x, y, alpha=0.05, label_x="Sampel 1", label_y="Sampel 2",
metric_name="Metrik"):
    stat = result.statistic
    pval = result.pvalue
```

```

# Format P-value tanpa notasi ekponensial
pval_str = format(pval, "%.6f")

# Hitung Z statistic jika N > 25
n = len(x)
if n > 25:
    W_manual, Z = compute_wilcoxon_z(x, y)
else:
    W_manual, Z = stat, None

# Kesimpulan
if pval <= alpha:
    conclusion = "Tolak H0 (ada perbedaan signifikan)"
    meaning = f"Hasil menunjukkan bahwa {label_x} dan {label_y} memiliki perbedaan signifikan pada metrik '{metric_name}'."
else:
    conclusion = "Gagal tolak H0 (tidak ada perbedaan signifikan)"
    meaning = f"Hasil menunjukkan bahwa {label_x} dan {label_y} tidak memiliki perbedaan signifikan pada metrik '{metric_name}'."

# ----- Output -----
print(f"📊 Hasil Uji Wilcoxon")
print("-----")
print(f"Metrik      : {metric_name}")
print(f"Kelompok    : {label_x} vs {label_y}")
if Z is not None:
    print(f"Z-statistic   : {Z:.4f}")
print(f"P-value      : {pval_str}")
print(f"Alpha (α)    : {alpha}")
print(f"Kesimpulan   : {conclusion}")
print(f"Interpretasi : {meaning}")
print("-----n")

return {
    "Metrik": metric_name,
    "Z": Z,
    "p-value": float(pval_str),
    "Signifikan": pval <= alpha,
    "Kesimpulan": conclusion
}

```

```

# -----
# Load Data Excel
# -----
file_path = "path/to/your/data.xlsx"
df1 = pd.read_excel(file_path, sheet_name="Sheet1") # JMeter
df2 = pd.read_excel(file_path, sheet_name="Sheet2") # Locust

# Daftar metrik
metrics = ["Error", "Avg Response Time", "Min", "Max", "95th pct", "99th pct", "Throughput"]

results = []

# -----
# Jalankan Uji Wilcoxon
# -----
for metric in metrics:
    if metric in df1.columns and metric in df2.columns:
        x = df1[metric]
        y = df2[metric]

        result = wilcoxon(x, y)
        res = interpret_wilcoxon_result(result, x, y, label_x="JMeter", label_y="Locust",
metric_name=metric)
        results.append(res)
    else:
        print(f"⚠ Kolom '{metric}' tidak ditemukan di salah satu sheet.\n")

# -----
# Simpan Ringkasan ke Excel
# -----
summary_df = pd.DataFrame(results)

print("📄 Ringkasan Hasil Uji Wilcoxon:")
print(summary_df)

output_path = "path/to/save/wilcoxon_summary.xlsx"
summary_df.to_excel(output_path, index=False)

print(f"✅ Hasil uji Wilcoxon telah disimpan ke file Excel:\n{output_path}")

```

Kode di atas berfungsi untuk membandingkan hasil pengujian performa antara Apache JMeter dan Locust menggunakan uji statistik Wilcoxon Signed-Rank Test pada beberapa metrik seperti *Error*, *Average Response Time*, *Min*, *Max*, *95th Percentile*, *99th Percentile*, dan *Throughput*.

Pertama, program membaca data dari dua *sheet* dalam file Excel, yaitu Sheet1 untuk data JMeter dan Sheet2 untuk data Locust. Kemudian, untuk setiap metrik yang terdaftar, kode menjalankan uji Wilcoxon untuk melihat apakah terdapat perbedaan yang signifikan antara hasil pengujian kedua alat tersebut.

Fungsi `interpret_wilcoxon_result()` digunakan untuk menampilkan hasil uji secara terperinci, termasuk nilai statistik, *p-value*, kesimpulan, dan interpretasi apakah perbedaan tersebut signifikan atau tidak. Jika sampel berpasangan lebih besar dari 25, maka distribusinya dianggap akan mendekati distribusi normal. Untuk itu digunakan Z sebagai Uji Statistiknya.

Hasil analisis dengan menggunakan uji Wilcoxon, uji tersebut digunakan untuk menguji apakah terdapat perbedaan dalam pengujian JMeter dan Locust dalam mengukur *Error*, *Average Response Time*, *Min*, *Max*, *95th Percentile*, *99th Percentile*, dan *Throughput*. Pada setiap API yang di uji.

Pada API Login untuk error mendapatkan hasil bahwa Gagal tolak H_0 yang artinya tidak ada perbedaan signifikan dari hasil test menggunakan JMeter dan Locust yang di tampilkan pada Tabel 4.6.

Tabel 4. 6 Hasil uji Wilcoxon pada API Login

Metrik	Z	p-value	Signifikan	Kesimpulan
Error	-1,76068	0,07697	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	-4,57646	0	TRUE	Tolak H_0 (ada perbedaan signifikan)

Min Response Time	-3,79486	0,000044	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	-4,741	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-4,72043	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	-4,741	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	-4,78214	0	TRUE	Tolak H_0 (ada perbedaan signifikan)

Selanjutnya untuk hasil perbedaan uji pada API Status Login mendapatkan hasil seperti pada Tabel 4.7.

Tabel 4. 7 Hasil uji Wilcoxon pada API Status Login

Metrik	Z	p-value	Signifikan	Kesimpulan
Error	0,81544	0,413302	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	0,60677	0,556113	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Min Response Time	-4,2885	0,000001	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	4,45305	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	3,75372	0,000056	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	3,85656	0,00003	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	4,16509	0,000004	TRUE	Tolak H_0 (ada perbedaan signifikan)

Untuk hasil pengujian pada API Status Login ada 2 metrik yang mendapatkan hasil Gagal Tolak H_0 yang berarti tidak ada perbedaan antara hasil dari JMeter dan Locust yaitu metrik Error dan Avg Response Time.

Tabel 4. 8 Hasil uji Wilcoxon pada API List Jadwal Ujian

Metrik	Z	p-value	Signifikan	Kesimpulan
Error	0,82369	0,409215	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	-0,1954	0,855272	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)

Min Response Time	-	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	-	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-	0,000003	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	-	0	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	-	0	TRUE	Tolak H_0 (ada perbedaan signifikan)

Pada hasil pengujian Wilcoxon untuk API List Jadwal Ujian pada Tabel 4.8 mendapatkan hasil yang sama dengan API Status Login yang dimana untuk metrik Error dan *Avg Response Time* mendapatkan Gagal Tolak H_0 .

Untuk hasil perbandingan pada API Mulai Ujian mendapatkan 3 metrik dengan Gagal Tolak H_0 yaitu Error, *95th Percentile* dan *99th Percentile* seperti yang terlihat pada tabel 4.9.

Tabel 4.9 Hasil uji Wilcoxon pada API Mulai Ujian

Metrik	Z	p-value	Signifikan	Kesimpulan
Error	-0,4001	0,688646	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	-2,70474	0,005776	TRUE	Tolak H_0 (ada perbedaan signifikan)
Min Response Time	-3,5069	0,000209	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	-4,14452	0,000004	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-1,05927	0,298844	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
99th Percentile	-1,69689	0,09195	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Throughput	-4,43248	0	TRUE	Tolak H_0 (ada perbedaan signifikan)

Pada pengujian perbandingan pada API Daftar Soal Ujian yang di jalankan menggunakan kode python mendapatkan hasil bahwa terdapat 1 metrik dengan menghasilkan p-value 0,89 yang berarti hasil dari perbandingan adalah Gagal Tolak

H_0 yang artinya tidak ada perbedaan signifikan antara JMeter dan Locust pada Tabel

4.10.

Tabel 4. 10 Hasil uji Wilcoxon pada API Daftar Soal Ujian

Metrik	Z	p-value	Signifikan	Kesimpulan	
Error	-	0,13337	0,893642	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	-	2,43735	0,013663	TRUE	Tolak H_0 (ada perbedaan signifikan)
Min Response Time	-	3,26008	0,000667	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	-	4,14452	0,000004	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-	3,30122	0,000555	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	-	2,39621	0,01546	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	-	4,20623	0,000003	TRUE	Tolak H_0 (ada perbedaan signifikan)

Hasil pengujian perbandingan pada API Soal Ujian yang terlihat pada Tabel 4.11. Pada metrik Error mendapatkan p-value 0,72 yang berarti tidak ada perbedaan antara hasil pengujian JMeter dan Locust untuk mendapatkan response Error pada API tersebut.

Tabel 4. 11 Hasil uji Wilcoxon pada API Soal Ujian

Metrik	Z	p-value	Signifikan	Kesimpulan	
Error	-	0,35301	0,722955	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	-	2,86928	0,003223	TRUE	Tolak H_0 (ada perbedaan signifikan)
Min Response Time	-	-2,6636	0,00664	TRUE	Tolak H_0 (ada perbedaan signifikan)
Max Response Time	-	-4,2885	0,000001	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-	3,60974	0,000123	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	-	2,06712	0,038418	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	-	3,40406	0,000345	TRUE	Tolak H_0 (ada perbedaan signifikan)

API terakhir Adalah API Submit Jawaban perbandingan pengujian API Submit Jawaban dilihat pada tabel 4.12 dengan mendapatkan hasil ada 3 metrik yang mendapatkan p-value < 0,05 yaitu metrik *Error*, *Avg Response Time*, dan juga *Min Response Time* yang berarti ketiga metrik ini mendapatkan hasil Gagal Tolak H_0 .

Tabel 4. 12 Hasil uji Wilcoxon pada API Submit Jawaban

Metrik	Z	p-value	Signifikan	Kesimpulan
Error	0,26036	0,792153	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Avg Response Time	0,89472	0,381798	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Min Response Time	1,45007	0,151887	FALSE	Gagal tolak H_0 (tidak ada perbedaan signifikan)
Max Response Time	4,06225	0,000008	TRUE	Tolak H_0 (ada perbedaan signifikan)
95th Percentile	-3,8977	0,000024	TRUE	Tolak H_0 (ada perbedaan signifikan)
99th Percentile	3,60974	0,000123	TRUE	Tolak H_0 (ada perbedaan signifikan)
Throughput	3,19838	0,000872	TRUE	Tolak H_0 (ada perbedaan signifikan)

Dari hasil pengujian perbandingan antara hasil performa dari Apache JMeter dan Locust semua API pada metrik error tidak ada perbedaan dalam mengukur performa error yang di dapat. Namu terdapat 1 metrik yang hasil selalu berbeda pada setiap pengujian yaitu *Throughput* (req/s). *Throughput* adalah metrik yang menunjukkan jumlah permintaan (request) yang dapat diproses oleh sistem setiap detik.



Gambar 4. 11 Gambar Grafik Hasil Throughput JMeter dan Locust
 Pada Gambar 4.11 hasil yang didapat pada untuk mengukur pada metrik

Throughput Locust lebih tinggi di bandingkan dengan JMeter. Dalam konteks penelitian performa, *throughput* dipengaruhi secara langsung oleh waktu respons. Ketika waktu respons meningkat, sistem membutuhkan waktu lebih lama untuk menyelesaikan setiap permintaan, sehingga jumlah permintaan yang dapat diproses per detik menurun. Oleh karena itu, semakin tinggi waktu respons, *throughput* cenderung semakin rendah, dan sebaliknya, jika waktu respons rendah, *throughput* akan meningkat.

Rendahnya *throughput* biasanya menunjukkan bahwa sistem mengalami hambatan pemrosesan, seperti beban server yang tinggi, antrian permintaan, atau proses backend yang kompleks. Sebaliknya, *throughput* yang tinggi mencerminkan kondisi sistem yang stabil, efisien, dan mampu menangani permintaan dengan cepat.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai *Analisis Perbandingan Apache JMeter dan Locust dalam Pengujian Performa API Service*, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Proses pengujian performa REST API berbasis arsitektur *microservice* dilakukan dengan tahapan yang sama pada kedua tools, meliputi: perancangan skenario uji (*test plan*), penentuan jumlah *virtual users*, durasi pengujian, serta pemilihan *endpoint API* yang diuji. Apache JMeter menggunakan pendekatan *thread-based* untuk simulasi pengguna, sedangkan Locust menggunakan pendekatan *event-driven asynchronous*, sehingga dapat menangani lebih banyak pengguna dengan konsumsi sumber daya lebih efisien. Seluruh skenario pengujian dilaksanakan pada kondisi lingkungan dan parameter yang seragam untuk menjaga validitas hasil.
2. Hasil Pengujian Berdasarkan *Response Time*, *Throughput*, dan *Error Rate* Hasil pengujian menunjukkan bahwa kedua tools menghasilkan nilai *response time* rata-rata dan *error rate* yang relatif serupa pada sebagian besar API. Namun demikian, terdapat perbedaan yang lebih jelas pada metrik ekstrem seperti *maksimum response time*, *95th percentile*, dan *99th percentile*. Selain itu, Locust secara konsisten menghasilkan nilai *throughput* yang lebih tinggi dibandingkan JMeter.

yang menunjukkan kemampuan Locust dalam menangani request per detik secara lebih efisien.

3. Perbedaan Performa antara Apache JMeter dan Locust dalam *Load Testing* REST API Berdasarkan hasil analisis komparatif, perbedaan performa antara JMeter dan Locust tidak terjadi pada seluruh metrik. Performa keduanya cenderung setara untuk metrik *error rate* dan rata-rata *response time*. Namun, pada metrik *throughput* dan beberapa metrik respons tinggi (*peak*), Locust menunjukkan performa yang lebih unggul. Hal ini mengindikasikan bahwa masing-masing tools memiliki karakteristik performa yang berbeda dalam menangani beban tertentu. Pengujian statistik dilakukan menggunakan metode Wilcoxon Signed-Rank Test karena data bersifat non-parametrik dan berpasangan.
4. Pembuktian Perbedaan Performa dengan Uji Wilcoxon Signed-Rank Test Hasil uji statistik Wilcoxon Signed-Rank Test menunjukkan bahwa tidak terdapat perbedaan signifikan pada metrik *error rate* dan *average response time* antara JMeter dan Locust. Sebaliknya, pada metrik *maximum response time*, *95th percentile*, *99th percentile*, dan *throughput*, terdapat perbedaan signifikan antara kedua tools. Dengan demikian, secara statistik dapat disimpulkan bahwa sebagian metrik performa menunjukkan perbedaan nyata, sehingga keduanya tidak sepenuhnya memiliki performa yang sama.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penulis memberikan beberapa saran sebagai berikut:

1. Saran bagi Pengembang dan Praktisi

Pengujian performa REST API berbasis *microservice* sebaiknya disesuaikan dengan kebutuhan skala sistem:

- Locust direkomendasikan untuk pengujian skala besar yang membutuhkan efisiensi sumber daya dan fleksibilitas *scripting*.
- JMeter lebih cocok digunakan pada pengujian skala kecil hingga menengah atau ketika dibutuhkan visualisasi grafik hasil pengujian secara cepat melalui antarmuka GUI.

2. Saran untuk Penelitian Selanjutnya

Penelitian mendatang dapat dikembangkan dengan:

- Menambahkan parameter lain seperti *CPU usage*, *memory usage*, dan *network latency* untuk memperluas analisis performa.
- Melakukan pengujian pada arsitektur *cloud-native* untuk menganalisis skalabilitas dan keandalan pada lingkungan terdistribusi.
- Menggunakan metode statistik lain (misalnya Mann-Whitney U atau Kruskal-Wallis) untuk memperkuat analisis hasil uji non-parametrik.
- Sebaiknya melakukan pengujian performa pada beberapa platform API, untuk mendapatkan performa dari instrumen ukur.
- Melakukan penelitian terhadap faktor yang dapat mempengaruhi perbedaan antara JMeter dan Locust

DAFTAR PUSTAKA

- [1] P. D. Abda'u, A. Susanto, A. R. Supriyono, and D. N. Prasetyani, "Perbandingan kinerja antara Gatling dan Apache JMeter pada uji beban RESTful API," *Infotekmesin*, pp. 211–215, 2024.
- [2] E. N. Alam and F. Dewi, "Performance testing analysis of BandunGTanginas application with JMeter," *Int. J. Innov. Enterprise Syst.*, pp. 146–155, 2022.
- [3] H. Asemi, "A study on API security pentesting," California Polytechnic State University, 2023.
- [4] A. Barezak, "Performance comparison of monolith and microservices based applications," in *Proc. 25th World Multi-Conf. Systemics, Cybernetics and Informatics*, 2021, pp. 120–125.
- [5] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "RESTful API architecture based on Laravel framework," *Purpose-Led Publishing*, 2017.
- [6] H. K. Dhalla, "A performance comparison of RESTful applications implemented in Spring Boot Java and MS.NET Core," in *Virtual Conf. Eng., Sci., Technol. (ViCEST)*, 2020.
- [7] S. Dhiman and P. Sharma, "Performance testing: A comparative study and analysis of web service testing tools," *Int. J. Comput. Sci. Mobile Comput.*, pp. 507–512, 2016.
- [8] D. P. Dissanayake, "REST API service middleware," Univ. of Colombo School of Computing, 2018.
- [9] N. Freeman, R. Cardone, J. Stubbs, and C. Garcia, "Detailed functional overview of an API and workflow engine for scientific research computing," *Pract. Exp. Adv. Res. Comput.*, pp. 11–17, 2023.
- [10] M. Hendayun, A. Ginanjar, and Y. Ihsan, "Analysis of application performance testing using load testing and stress testing methods in API service," *SISFOTEK Global*, pp. 28–34, 2023.
- [11] Z. M. Jiang, "Automated analysis of load testing results," in *Proc. Int. Symp. Softw. Test. Anal.*, 2010, pp. 143–146.
- [12] D. M. Kaushik and P. Fageria, "Performance testing tools: A comparative study," *Int. J. Innov. Sci., Eng. Technol.*, 2014.
- [13] F. A. Mufarroha *et al.*, "Quality assurance of academic websites using performance testing tools," *Romanian J. Appl. Sci. Technol.*, pp. 226–233, 2023.
- [14] A. Z. Musthafawi, A. Mas'adah, Sukmadiningtyas, and F. Ramdani, "Performance testing on the Shopee website in the pandemic period of

- COVID-19," in *Sustainable Information Eng. Technol. (SIET)*, pp. 195–199, 2020.
- [15] S. Pargaonkar, "A comparative review of performance testing methodologies and best practices: Software quality engineering," *Int. J. Sci. Res.*, pp. 2010–2014, 2023.
- [16] D. I. Permatasari *et al.*, "Pengujian aplikasi menggunakan metode load testing dengan Apache JMeter pada sistem informasi pertanian," *J. Sist. Teknol. Informasi*, pp. 135–139, 2020.
- [17] S. Pradeep and Y. K. Sharma, "A pragmatic evaluation of stress and performance testing technologies for web-based application," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, pp. 399–403, 2019.
- [18] R. Priyanka and M. Kanna, "A study on performance testing based on web application," *Int. J. Adv. Eng. Res. Sci.*, pp. 161–165, 2016.
- [19] P. B. Ramadhanu and A. T. Priandika, "Rancang bangun web service API aplikasi sentralisasi produk UMKM pada UPTD PLUT KUMKM Provinsi Lampung," *J. Teknologi dan Sistem Informasi*, pp. 59–64, 2021.
- [20] S. Shrivastava and P. P. SB, "Comprehensive review of load testing tools," *Int. Res. J. Eng. Technol. (IRJET)*, pp. 3392–3395, 2020.
- [21] A. Soni and V. Ranga, "API features individualizing of web services: REST and SOAP," *Int. J. Innov. Technol. Explor. Eng.*, pp. 664–671, 2019.
- [22] N. Srivastava, U. Kumar, and P. Singh, "Software and performance testing tools," *J. Informatics Electr. Electron. Eng.*, pp. 1–12, 2021.
- [23] R. Susana, A. Nugraha, and D. Nataliana, "Perancangan dan realisasi web-based data logging system menggunakan ATmega16 melalui Hypertext Transfer Protocol (HTTP)," *J. ELKOMIKA Itenas*, 2015.
- [24] V. Tiwari, S. Upadhyay, J. K. Goswami, and S. Agrawal, "Analytical evaluation of web performance testing tools: Apache JMeter and SoapUI," in *Int. Conf. Commun. Syst. Netw. Technol.*, pp. 519–523, 2023.
- [25] M. A. Umar and C. Zhanfang, "A study of automated software testing: Automation tools and frameworks," *Int. J. Comput. Sci. Eng.*, pp. 217–225, 2019.
- [26] J. Wang, X. Bai, L. Li, Z. Ji, and H. Ma, "A model-based framework for cloud API testing," in *Proc. Annu. Comput. Softw. Appl. Conf.*, pp. 60–65, 2017.
- [27] U. Zahroh, R. Andreswari, and E. N. Alam, "Back-end design and development on Rekaruang application with microservices architecture," *J. Teknik Informatika dan Sistem Informasi*, pp. 86–96, 2022.
- [28] A. A. Syahri, "Jurnal pengertian statistika," *Statistika Pendidikan*, vol. 6, pp. 121–129, 2015.
- [29] BAPPENAS RI, *Undang-Undang Nomor 7 Tahun 1960 Tentang Statistik, Demographic Research*, vol. 1, pp. 4–7, 1960.

- [30] A. K. Dwivedi, I. Mallawaarachchi, and L. A. Alvarado, "Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method," *Statistics in Medicine*, vol. 36, no. 14, pp. 2187–2205, 2017.
- [31] S. B. Fatima and D. K. Anupama, "Non-parametric test in pharmaceutical statistical calculations," *Int. J. Pharmaceutical Sci. Rev. Res.*, vol. 77, no. 2, 2022.
- [32] T. Harris and J. W. Hardin, "Exact Wilcoxon signed-rank and Wilcoxon Mann-Whitney ranksum tests," *Stata Journal*, vol. 13, no. 2, pp. 337–343, 2013.
- [33] H.-Y. Kim, "Nonparametric statistical methods: 2. Nonparametric methods for comparing three or more groups and repeated measures," *Restor. Dent. Endod.*, vol. 39, no. 4, pp. 329–335, 2014.
- [34] S. I. Maiyanti *et al.*, "Implementasi Wilcoxon signed-rank test univariat dan multivariat untuk menguji perbedaan derajat nyeri pasien endometriosis sebelum dan sesudah tindakan operasi," vol. 2, no. 1, pp. 1–7, 2023.
- [35] S. Padmasari, F. N. Azizah, and N. Larasati, "Edukasi home pharmacy care terhadap kepatuhan dan kontrol glukosa darah pada pasien diabetes melitus," *J. Sains Farmasi & Klinis*, vol. 8, no. 2, pp. 182–189, 2021.
- [36] F. A. Pratiwi, H. Sholih, and R. Z. Dalimunthe, "Meningkatkan kemandirian belajar siswa dengan layanan bimbingan klasikal dengan teknik STAD," *J. Fokus Konseling*, vol. 7, no. 1, pp. 1–8, 2021.
- [37] D. Rudianto *et al.*, "Pengaruh hubungan e-learning dalam mata kuliah MAFIKI di Institut Teknologi Sumatera menggunakan metode Wilcoxon," *Indonesian J. Appl. Math.*, vol. 1, no. 1, pp. 1–5, 2020.
- [38] Sabarudin *et al.*, "Efektivitas pemberian edukasi secara online melalui media video dan leaflet terhadap tingkat pengetahuan pencegahan Covid-19 di Kota Baubau," *J. Farmasi Galenika*, vol. 6, no. 2, pp. 309–318, 2020.
- [39] S. T. Setianingsih and Nelmiawati, "Media informasi Dinas Kominfo Kota Batam," *J. Appl. Multimedia Netw.*, vol. 4, no. 1, pp. 1–9, 2020.
- [40] S. Siregar, *Statistik Parametrik Untuk Penelitian Kuantitatif*, PT Bumi Aksara, 2013.
- [41] V. W. Sujarweni, *SPSS untuk Penelitian*, Pustaka Baru Press, 2015.
- [42] D. Sunyoto, *Teori, Kuesioner dan Analisis Data untuk Pemasaran*, Graha Ilmu, 2013.
- [43] D. T. Untari, *Buku Ajar Statistik 1*, 2019.
- [44] S. Wahyuning, *Dasar-dasar Statistika*, Yayasan Prima Agus, 2021.
- [45] W. A. Windi, M. Taufiq, and T. Muhammad, "Implementasi Wilcoxon signed-rank test untuk mengukur efektivitas pemberian video tutorial dan PPT," *Produktif: J. Ilm. Pendidik. Teknol. Inform.*, vol. 5, no. 1, pp. 405–410, 2022.

- [46] Zulkipli, Zulfachmi, and A. Rahmad, "Alasan peneliti menggunakan analisis statistik Wilcoxon (non-parametrik)," *Prosiding SNISTEK*, vol. 6, pp. 119–125, 2024.
- [47] S. Goericke, *The Future of Software Quality Assurance*, iSQI, 2019.
- [48] M. J. Hossain, *Software Quality Assurance (Project)*, 2020.
- [49] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed., 2010.
- [50] (2024). Apache Software Foundation, "Apache JMeter" [online]. Available: <https://www.apache.org/>
- [51] (2023). AWS, "What is APM (Application Performance Monitoring)?" [online]. Available: <https://aws.amazon.com/what-is/application-performance-monitoring/>
- [52] (2024). Locust.io, "Locust" [online]. Available: <https://locust.io/>
- [53] (2024). J. Davids, "API fundamentals," IBM Developer [online]. Available: <https://developer.ibm.com/articles/api-fundamentals/>
- [54] (2024). Odown.com, "API response time standards: What's good, bad, and unacceptable," [online]. Available: <https://odown.com/>
- [55] D. A. Fansha, M. Y. H. Setyawan, and M. N. Fauzan, "Load Test pada Microservice yang menerapkan CQRS dan Event Sourcing," *Jurnal Buana Informatika*, pp. 126-134, 2021.
- [56] M. Neelapu, "The Effectiveness of Load and Performance Testing on Application Scalability," *ESP Journal of Engineering & Technology Advancements*, pp. 171-180, 2024.

LAMPIRAN

Lampiran 1 Tabel Hasil Pengujian JMeter

Test	API Request	Error	Response Time					Throughput (req/s)
			Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	
1	Login	0	2253,30	224,00	7995,00	3683,55	7735,93	6,64
	Status Login	0	1889,90	38,00	7603,00	4382,15	6472,94	6,98
	List Jadwal Ujian	0	2095,53	131,00	8021,00	3702,95	7800,23	7,43
	Mulai Ujian	0	1276,08	38,00	7333,00	3137,70	3832,65	7,76
	Daftar Soal Ujian	0	1267,31	36,00	9867,00	3082,00	3756,23	7,88
	Soal Ujian	0	1236,01	31,00	7605,00	3195,85	7462,55	8,26
	Submit Jawaban	4	1159,73	43,00	7587,00	2870,20	7219,93	8,50
2	Login	4	2671,70	728,00	21039,00	6211,80	8737,78	5,62
	Status Login	4	1752,65	114,00	8421,00	3560,70	6308,80	5,77
	List Jadwal Ujian	4	2401,87	230,00	9358,00	6203,40	8356,24	5,86
	Mulai Ujian	4	1561,24	51,00	8532,00	4080,30	7885,63	6,14
	Daftar Soal Ujian	4	1740,94	32,00	8818,00	4450,85	8131,18	6,25
	Soal Ujian	4	1847,89	35,00	12641,00	6015,85	8982,02	6,40
	Submit Jawaban	4	1572,06	44,00	9232,00	4308,25	7700,31	6,68
3	Login	0	3180,41	979,00	17317,00	5577,50	10865,54	4,07
	Status Login	0	2704,41	181,00	5790,00	5164,80	5487,95	4,13
	List Jadwal Ujian	0	3028,32	999,00	7775,00	4630,00	6263,94	4,10
	Mulai Ujian	0	2777,22	166,00	5700,00	4885,00	5629,38	4,24

	Daftar Soal Ujian	0	2611,50	141,00	5668,00	4620,90	5522,76	4,27
	Soal Ujian	0	2660,97	55,00	8346,00	4670,75	5671,80	4,34
	Submit Jawaban	0	2527,42	50,00	5833,00	4389,65	5585,00	4,35
4	Login	0	1536,08	147,00	3984,00	3260,25	3657,90	9,59
	Status Login	0	1176,35	35,00	4000,00	3552,50	3873,44	10,22
	List Jadwal Ujian	0	1386,26	125,00	7193,00	3084,05	3625,34	11,05
	Mulai Ujian	0	1056,53	34,00	10051,00	2961,70	9973,76	11,82
	Daftar Soal Ujian	0	806,61	31,00	3628,00	2553,75	3405,58	12,08
	Soal Ujian	0	702,20	30,00	10043,00	2644,10	3469,33	12,29
	Submit Jawaban	4	592,01	37,00	3862,00	2410,65	2951,89	12,99
	5	Login	0	2911,32	459,00	10537,00	5955,95	8776,33
Status Login		0	1971,99	46,00	10091,00	4521,60	7224,13	5,77
List Jadwal Ujian		0	2108,14	138,00	7146,00	4202,80	4939,93	5,89
Mulai Ujian		0	1555,41	36,00	6596,00	4660,15	5943,40	6,07
Daftar Soal Ujian		0	1808,75	30,00	10173,00	4925,95	8686,87	6,09
Soal Ujian		0	1907,53	31,00	10143,00	5610,15	10014,48	6,30
Submit Jawaban		0	2029,63	41,00	11299,00	7502,55	10056,26	6,59
6	Login	0	1678,00	154,00	8924,00	2767,00	4938,52	6,16
	Status Login	0	724,77	47,00	3636,00	2228,70	2890,27	6,30
	List Jadwal Ujian	1	1766,47	133,00	21037,00	2915,65	4965,98	6,30
	Mulai Ujian	1	807,59	34,00	3974,00	2228,85	2468,86	6,50
	Daftar Soal Ujian	1	1008,77	36,00	4714,00	2233,95	2979,15	6,54
	Soal Ujian	1	981,02	33,00	4549,00	2341,85	3420,30	6,78

	Submit Jawaban	1	910,26	41,00	4970,00	2333,75	3482,96	6,96
7	Login	0	1782,30	237,00	4532,00	3054,65	4025,55	5,56
	Status Login	1	941,91	40,00	2107,00	2635,50	3046,85	5,66
	List Jadwal Ujian	0	1700,22	152,00	3958,00	2929,75	3473,74	5,66
	Mulai Ujian	0	825,07	35,00	4197,00	2395,00	3102,60	5,79
	Daftar Soal Ujian	0	906,45	41,00	4394,00	2267,95	3833,98	5,81
	Soal Ujian	0	1164,72	39,00	4122,00	2847,75	3781,91	5,83
	Submit Jawaban	1	1245,32	46,00	5015,00	2856,60	3468,99	5,84
8	Login	0	1122,05	137,00	15570,00	2460,50	7757,23	6,51
	Status Login	0	308,34	36,00	6351,00	858,70	1811,16	6,62
	List Jadwal Ujian	0	1409,25	121,00	15546,00	4824,90	7896,79	6,62
	Mulai Ujian	0	773,53	32,00	8916,00	6223,30	7367,11	6,79
	Daftar Soal Ujian	0	678,51	35,00	8028,00	3178,85	6966,67	6,85
	Soal Ujian	0	498,83	35,00	3620,00	2422,15	2902,70	6,89
	Submit Jawaban	0	310,21	38,00	2913,00	954,65	1853,25	6,91
9	Login	4	1526,81	154,00	16831,00	3951,05	8855,73	5,65
	Status Login	4	611,70	39,00	6472,00	2523,50	5118,93	5,78
	List Jadwal Ujian	4	1549,77	127,00	9653,00	6130,85	8555,24	5,79
	Mulai Ujian	4	758,96	42,00	9194,00	4205,90	9039,45	5,88
	Daftar Soal Ujian	4	606,89	38,00	8646,00	2012,80	7979,91	5,89
	Soal Ujian	4	749,69	43,00	16825,00	2361,90	8745,84	5,90
	Submit Jawaban	4	737,53	46,00	8918,00	2673,55	8743,63	5,97
10	Login	0	2462,02	965,00	9298,00	5278,50	7932,36	4,29
	Status Login	0	1720,31	102,00	8179,00	4960,80	6988,63	4,37

	List Jadwal Ujian	0	2434,86	191,00	8671,00	4968,25	7279,28	4,38
	Mulai Ujian	0	1315,53	34,00	7668,00	3775,70	6889,71	4,55
	Daftar Soal Ujian	0	1198,48	35,00	7092,00	3715,05	5202,02	4,62
	Soal Ujian	0	1093,25	39,00	5712,00	3350,70	5313,79	4,63
	Submit Jawaban	0	1256,51	44,00	7088,00	3742,40	5648,17	4,64
11	Login	0	2001,49	456,00	7375,00	3625,50	6955,03	4,96
	Status Login	0	1328,63	50,00	9648,00	3989,45	9353,23	5,04
	List Jadwal Ujian	0	2160,47	154,00	10114,00	5821,45	8706,46	5,09
	Mulai Ujian	0	1095,75	44,00	5011,00	3279,95	4494,16	5,52
	Daftar Soal Ujian	0	951,19	45,00	4678,00	2564,30	3047,62	5,56
	Soal Ujian	0	969,72	53,00	3639,00	2525,00	2979,99	5,87
	Submit Jawaban	0	1008,25	44,00	4100,00	2807,55	3447,11	6,01
12	Login	0	2069,21	403,00	3960,00	2879,80	3493,78	6,79
	Status Login	0	1438,36	47,00	9495,00	3080,35	4548,34	6,98
	List Jadwal Ujian	0	2128,53	118,00	9059,00	3220,95	3551,06	7,01
	Mulai Ujian	0	1544,26	32,00	4337,00	3375,85	3950,91	7,32
	Daftar Soal Ujian	0	1583,94	33,00	9135,00	3211,95	3640,74	7,37
	Soal Ujian	0	1383,24	35,00	5192,00	2523,80	3457,46	7,43
	Submit Jawaban	0	1316,91	45,00	3838,00	2654,50	3518,60	7,90
13	Login	0	3801,98	1358,00	10650,00	6994,15	9479,31	4,51
	Status Login	0	3389,40	267,00	9405,00	5048,80	6289,28	4,49
	List Jadwal Ujian	0	3154,50	1484,00	7699,00	4922,95	5910,14	4,38
	Mulai Ujian	0	3427,16	452,00	8228,00	5546,10	6296,92	4,48

	Daftar Soal Ujian	0	3900,70	2223,00	12955,00	7012,95	8770,32	4,38
	Soal Ujian	0	3606,91	834,00	9705,00	5646,30	7673,04	4,49
	Submit Jawaban	0	3655,94	102,00	13028,00	5256,35	6408,81	4,63
14	Login	0	2740,94	1126,00	6330,00	4360,85	5425,97	4,59
	Status Login	0	2397,10	273,00	6334,00	4884,00	5481,19	4,61
	List Jadwal Ujian	0	2802,31	543,00	10314,00	4488,60	5275,59	4,62
	Mulai Ujian	0	2350,11	50,00	9576,00	3884,30	5578,76	4,78
	Daftar Soal Ujian	0	2310,99	46,00	9842,00	4338,50	5415,50	4,92
	Soal Ujian	0	2319,76	37,00	5694,00	3977,10	4540,58	4,98
	Submit Jawaban	0	2353,34	60,00	6488,00	4090,05	4991,88	5,02
	15	Login	0	3109,22	932,00	9095,00	5320,30	8558,67
Status Login		0	2732,96	164,00	6264,00	4681,50	5731,76	4,46
List Jadwal Ujian		0	2701,35	930,00	5264,00	4329,45	4978,86	4,37
Mulai Ujian		0	2217,00	159,00	4901,00	3672,95	4590,69	4,51
Daftar Soal Ujian		0	2144,87	47,00	4898,00	3466,05	4173,24	4,60
Soal Ujian		0	2561,16	38,00	7938,00	4843,05	6874,06	4,62
Submit Jawaban		0	2886,46	46,00	9087,00	6377,35	9000,58	4,68
16		Login	0	2740,94	1126,00	6330,00	4360,85	5425,97
	Status Login	0	2397,10	273,00	6334,00	4884,00	5481,19	4,61
	List Jadwal Ujian	0	2802,31	543,00	10314,00	4488,60	5275,59	4,62
	Mulai Ujian	0	2350,11	50,00	9576,00	3884,30	5578,76	4,78
	Daftar Soal Ujian	0	2310,99	46,00	9842,00	4338,50	5415,50	4,92
	Soal Ujian	0	2319,76	37,00	5694,00	3977,10	4540,58	4,98

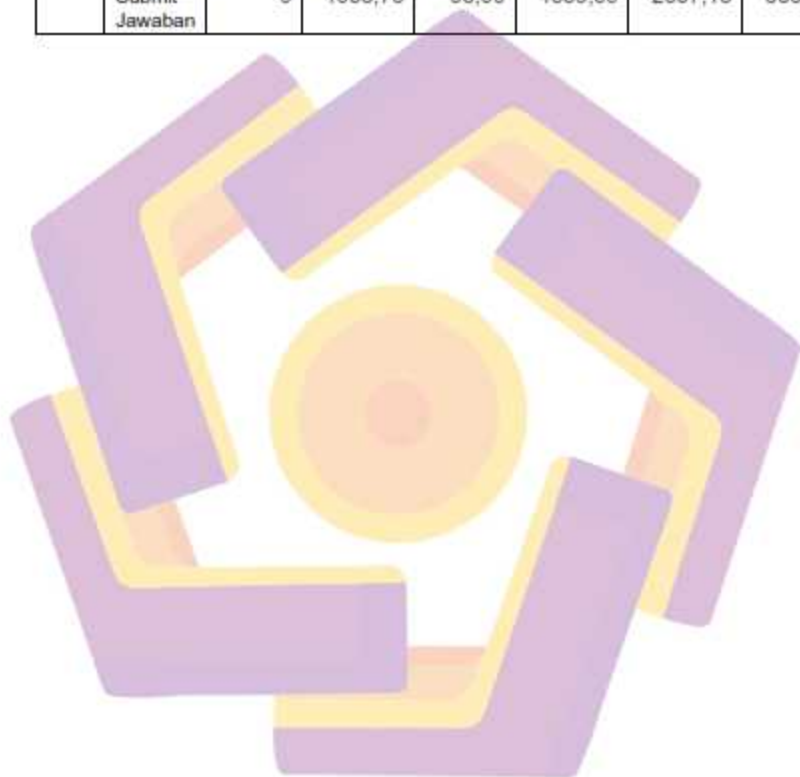
	Submit Jawaban	0	2353,34	60,00	6488,00	4090,05	4991,88	5,02
17	Login	0	1571,97	228,00	2920,00	2372,30	2632,81	8,23
	Status Login	0	659,55	47,00	2960,00	2171,85	2220,96	8,57
	List Jadwal Ujian	0	1385,86	190,00	2561,00	2062,60	2401,97	8,61
	Mulai Ujian	0	399,26	37,00	2713,00	1204,70	1394,90	9,01
	Daftar Soal Ujian	0	539,01	36,00	2666,00	1625,40	1927,54	9,05
	Soal Ujian	0	684,55	40,00	2867,00	1729,35	2343,55	9,10
	Submit Jawaban	0	720,17	46,00	2977,00	2155,45	2490,50	9,13
	18	Login	0	2763,06	658,00	6934,00	4393,60	5281,42
Status Login		0	2314,42	94,00	4982,00	3931,15	4891,30	4,12
List Jadwal Ujian		0	3059,88	687,00	6238,00	5430,70	5880,97	4,09
Mulai Ujian		0	3171,30	111,00	6945,00	6060,10	6741,92	4,21
Daftar Soal Ujian		0	2762,11	119,00	10501,00	4545,85	5409,98	4,47
Soal Ujian		0	2778,89	110,00	10328,00	4557,10	10020,69	4,61
Submit Jawaban		0	2748,68	52,00	10872,00	4609,50	5025,67	4,74
19		Login	0	2203,18	282,00	10341,00	3997,40	7309,07
	Status Login	0	1869,31	51,00	7857,00	5455,10	7468,71	6,34
	List Jadwal Ujian	0	2384,57	218,00	10614,00	6313,10	7057,19	6,34
	Mulai Ujian	0	1890,94	30,00	8059,00	6185,65	7048,94	6,70
	Daftar Soal Ujian	0	1543,12	36,00	7896,00	4916,30	5765,51	6,98
	Soal Ujian	0	1389,85	34,00	7757,00	3616,15	6504,79	7,19
	Submit Jawaban	0	1545,61	38,00	7969,00	3717,40	7253,05	7,23
	20	Login	0	1652,07	202,00	4148,00	2542,70	3417,08
Status Login		0	666,08	54,00	2964,00	2396,10	2831,99	7,00

	List Jadwal Ujian	0	1663,72	164,00	4352,00	2779,80	3727,75	7,03
	Mulai Ujian	0	674,27	35,00	2791,00	1818,75	2557,95	7,34
	Daftar Soal Ujian	0	561,60	43,00	3700,00	1762,50	2449,45	7,38
	Soal Ujian	0	995,09	36,00	3910,00	2946,15	3735,99	7,61
	Submit Jawaban	0	941,88	43,00	3901,00	2585,15	3298,80	7,63
21	Login	0	1970,90	528,00	4846,00	3022,00	3663,64	5,43
	Status Login	0	1130,42	114,00	3212,00	2387,10	2960,77	5,53
	List Jadwal Ujian	0	1723,24	164,00	3888,00	2611,75	3013,77	5,55
	Mulai Ujian	0	774,55	63,00	3318,00	2196,85	2984,55	5,70
	Daftar Soal Ujian	0	982,81	36,00	3775,00	2524,80	3109,64	5,73
	Soal Ujian	0	1221,42	42,00	4452,00	2614,25	3198,89	5,75
	Submit Jawaban	0	1130,88	47,00	3602,00	2400,70	3187,03	5,79
22	Login	0	1370,65	272,00	3245,00	2458,80	3049,31	8,78
	Status Login	0	588,56	48,00	2659,00	2372,70	2637,79	9,21
	List Jadwal Ujian	0	1312,30	151,00	4272,00	2331,55	3171,15	9,33
	Mulai Ujian	10	515,49	40,00	2680,00	1917,80	2576,83	9,97
	Daftar Soal Ujian	10	509,46	47,00	2374,00	1729,65	2117,93	10,22
	Soal Ujian	10	559,02	45,00	2594,00	1892,95	2514,51	10,72
	Submit Jawaban	10	570,21	45,00	2651,00	1666,70	2545,06	10,78
23	Login	0	1481,83	46,00	16155,00	4770,10	8811,54	2,68
	Status Login	0	2512,93	58,00	13053,00	7772,45	11828,44	2,64
	List Jadwal Ujian	0	2387,67	221,00	14269,00	5491,80	9835,54	2,65
	Mulai Ujian	0	1481,83	46,00	16155,00	4770,10	8811,54	2,68

	Daftar Soal Ujian	0	1536,87	45,00	13056,00	4980,80	9976,47	2,69
	Soal Ujian	0	1691,39	43,00	16585,00	5560,40	9219,04	2,69
	Submit Jawaban	0	2216,03	44,00	14111,00	7087,55	13264,51	2,70
24	Login	0	6482,02	1169,00	33418,00	16672,95	23489,28	2,16
	Status Login	0	6036,80	168,00	36904,00	17407,05	28057,80	2,21
	List Jadwal Ujian	0	5606,00	440,00	36126,00	16461,00	29472,44	2,27
	Mulai Ujian	0	5912,05	77,00	36380,00	16708,65	32630,01	2,32
	Daftar Soal Ujian	0	6653,24	45,00	36925,00	18246,15	36212,95	2,36
	Soal Ujian	0	6502,09	60,00	37229,00	16787,95	32483,15	2,38
	Submit Jawaban	0	5524,41	45,00	37664,00	16159,60	28056,58	2,44
25	Login	0	5660,85	1429,00	19247,00	11634,50	14956,02	2,26
	Status Login	0	5357,75	168,00	22180,00	10750,85	12687,98	2,29
	List Jadwal Ujian	0	4985,53	1499,00	18734,00	10557,20	13633,76	2,32
	Mulai Ujian	0	4887,04	330,00	14945,00	10635,90	13371,63	2,34
	Daftar Soal Ujian	0	4671,50	65,00	13754,00	10485,85	11684,42	2,39
	Soal Ujian	0	4997,66	46,00	16097,00	10108,65	13503,65	2,43
	Submit Jawaban	0	4825,42	42,00	13767,00	10021,90	12447,71	2,47
26	Login	0	2843,79	721,00	8015,00	5145,70	6329,88	2,89
	Status Login	0	2572,63	245,00	10225,00	4395,65	6355,95	2,91
	List Jadwal Ujian	0	2784,14	597,00	6171,00	4674,40	5720,63	2,91
	Mulai Ujian	0	2310,38	102,00	5380,00	4010,85	4654,48	2,99
	Daftar Soal Ujian	0	2336,94	68,00	5397,00	4452,90	5355,01	3,02
	Soal Ujian	0	2295,45	63,00	6503,00	4406,55	4831,51	3,04

	Submit Jawaban	0	2145,66	61,00	6134,00	4402,90	5701,77	3,09
27	Login	1	2555,60	270,00	21053,00	4131,15	5693,54	2,48
	Status Login	1	2073,29	71,00	7467,00	4121,60	5245,63	2,50
	List Jadwal Ujian	1	2603,92	191,00	8595,00	4230,60	7609,66	2,46
	Mulai Ujian	1	2085,44	34,00	8521,00	4166,15	5381,89	2,51
	Daftar Soal Ujian	2	2227,23	41,00	21055,00	4014,75	7334,86	2,55
	Soal Ujian	2	2227,23	41,00	21055,00	4014,75	7334,86	2,55
	Submit Jawaban	1	2075,51	46,00	8587,00	3800,25	5314,61	2,57
28	Login	0	2105,76	1113,00	7038,00	3236,20	5048,10	3,40
	Status Login	0	1386,28	76,00	6761,00	3161,55	5683,31	3,43
	List Jadwal Ujian	0	2253,61	325,00	5459,00	4704,75	5235,24	3,43
	Mulai Ujian	0	1351,86	49,00	6028,00	3311,95	5090,21	3,57
	Daftar Soal Ujian	0	1441,69	109,00	5264,00	3546,90	4424,96	3,67
	Soal Ujian	0	1512,30	55,00	5882,00	3170,20	5472,64	3,74
	Submit Jawaban	0	1505,87	56,00	5794,00	3143,20	4515,24	3,82
29	Login	0	2170,17	469,00	7678,00	3509,85	4441,82	3,62
	Status Login	0	1421,46	152,00	8325,00	3232,25	4980,28	3,68
	List Jadwal Ujian	0	2177,14	484,00	9348,00	3506,65	4336,36	3,67
	Mulai Ujian	0	1239,65	96,00	4039,00	3134,00	3806,92	3,76
	Daftar Soal Ujian	0	1204,43	40,00	6549,00	3142,30	3701,50	3,82
	Soal Ujian	0	1281,44	50,00	5252,00	3108,85	3797,46	3,84
	Submit Jawaban	0	1386,63	41,00	4409,00	2970,85	3507,73	3,85
30	Login	0	1871,77	704,00	4837,00	3079,55	3670,59	4,39
	Status Login	0	913,30	136,00	3358,00	2218,35	2605,95	4,41

List Jadwal Ujian	0	1779,77	283,00	4856,00	2918,95	3758,36	4,42
Mulai Ujian	0	889,04	47,00	4226,00	2283,90	2934,90	4,51
Daftar Soal Ujian	0	990,77	45,00	4243,00	2315,80	2836,94	4,53
Soal Ujian	0	1176,56	94,00	4929,00	2784,65	3278,98	4,59
Submit Jawaban	0	1053,76	98,00	4859,00	2557,15	3583,91	4,64



Lampiran 2 Tabel Hasil Pengujian Locust

Test	API Request	Error	Response Time					Throughput (req/s)
			Avg Response Time (ms)	Min (ms)	Max (ms)	Percentile ke-95 (ms)	Percentile ke-99 (ms)	
1	Login	0,00	212,15	118,11	760,20	550,00	700,00	0,97
	Status Login	0,00	47,90	32,05	341,13	71,00	110,00	14,77
	List Jadwal Ujian	0,00	56,39	29,54	4203,79	130,00	320,00	15,63
	Mulai Ujian	0,00	46,50	28,73	348,13	76,14	83,39	15,50
	Daftar Soal Ujian	0,00	49,34	28,73	865,09	83,10	124,56	15,33
	Soal Ujian	0,00	46,13	29,33	343,67	73,17	80,44	14,21
	Submit Jawaban	0,00	53,50	34,87	558,39	84,00	130,00	14,06
2	Login	0,00	2424,64	130,79	34879,52	24000,00	32000,00	0,81
	Status Login	0,00	705,47	33,55	29831,73	4700,00	8100,00	13,11
	List Jadwal Ujian	0,00	767,48	31,74	32421,05	4900,00	8300,00	13,80
	Mulai Ujian	0,00	846,19	30,05	10042,71	4451,60	5182,10	13,69
	Daftar Soal Ujian	0,00	49,34	28,73	865,09	83,10	124,56	15,33
	Soal Ujian	0,00	888,83	31,75	28758,54	4571,48	5427,13	12,68
	Submit Jawaban	0,00	670,38	37,42	10149,09	4500,00	7700,00	12,56
3	Login	0,00	6823,59	129,68	39701,38	29000,00	34000,00	0,98
	Status Login	0,00	1624,60	30,58	48307,47	7000,00	11000,00	17,26
	List Jadwal Ujian	0,00	1684,34	30,17	40928,33	7100,00	17000,00	18,05
	Mulai Ujian	0,00	2043,86	28,62	28858,39	6048,93	6625,62	17,92
	Daftar Soal Ujian	0,00	2078,96	31,02	52206,07	6086,98	8387,74	17,78

	Soal Ujian	0,00	2208,51	31,32	37471,53	6455,01	7211,73	16,67
	Submit Jawaban	2,00	1407,51	36,12	17255,89	6400,00	9200,00	16,52
4	Login	0,00	6823,59	129,68	39701,38	29000,00	34000,00	0,98
	Status Login	0,00	1624,60	30,58	48307,47	7000,00	11000,00	17,26
	List Jadwal Ujian	0,00	1684,34	30,17	40928,33	7100,00	17000,00	18,05
	Mulai Ujian	0,00	2043,86	28,62	28858,39	6048,93	6625,62	17,92
	Daftar Soal Ujian	0,00	2078,96	31,02	52206,07	6086,98	8387,74	17,78
	Soal Ujian	0,00	2208,51	31,32	37471,53	6455,01	7211,73	16,67
	Submit Jawaban	2,00	1407,51	36,12	17255,89	6400,00	9200,00	16,52
	Login	39,00	14660,05	120,42	75973,69	57000,00	71000,00	0,97
5	Status Login	28,00	2456,55	32,86	57319,27	12000,00	19000,00	9,46
	List Jadwal Ujian	28,00	3130,50	30,35	70814,40	13000,00	39000,00	10,20
	Mulai Ujian	17,00	3814,39	29,05	44212,73	10659,27	11608,64	10,04
	Daftar Soal Ujian	25,00	4213,41	31,14	65486,06	11323,21	14932,79	9,87
	Soal Ujian	21,00	4309,37	30,06	45031,60	12359,18	13543,72	8,96
	Submit Jawaban	21,00	2294,10	37,71	44054,37	12000,00	15000,00	8,84
	Login	0,00	4431,55	129,71	35803,22	19000,00	31000,00	0,93
	Status Login	0,00	1259,88	34,96	13003,47	6300,00	9400,00	9,63
6	List Jadwal Ujian	0,00	1343,65	32,39	30411,84	6700,00	11000,00	10,40
	Mulai Ujian	0,00	1621,21	30,60	12789,31	5661,07	5885,67	10,26
	Daftar Soal Ujian	0,00	1606,19	31,42	35401,02	5522,25	6362,73	10,08
	Soal Ujian	0,00	1762,94	30,85	18805,45	5447,50	5584,44	9,04
	Submit Jawaban	1,00	1199,40	37,40	12467,95	5900,00	9400,00	8,91
	Login	0,00	4694,64	133,53	23417,50	17000,00	23000,00	0,97
7								

	Status Login	0,00	1381,48	36,37	14524,36	4700,00	6800,00	9,57
	List Jadwal Ujian	0,00	1481,49	33,48	25777,31	5000,00	9600,00	10,34
	Mulai Ujian	0,00	1441,80	32,45	8295,55	3955,70	4034,47	10,25
	Daftar Soal Ujian	0,00	1450,38	34,01	15485,73	4056,13	4338,51	10,06
	Soal Ujian	0,00	1525,14	34,12	11402,91	3884,46	3944,11	8,97
	Submit Jawaban	0,00	1328,67	38,96	9616,61	4400,00	6200,00	8,81
8	Login	9,00	10132,36	903,74	33674,80	25000,00	30000,00	0,97
	Status Login	0,00	2728,38	78,03	33521,52	6900,00	15000,00	6,40
	List Jadwal Ujian	0,00	3495,46	70,12	33179,28	12000,00	18000,00	7,20
	Mulai Ujian	0,00	2763,08	64,39	28994,26	6416,13	6416,13	7,06
	Daftar Soal Ujian	0,00	2825,15	49,69	25053,99	6670,74	6670,74	6,85
	Soal Ujian	0,00	3244,67	54,93	43572,65	7173,41	7173,41	5,78
	Submit Jawaban	1,00	2827,11	57,39	33502,59	7600,00	17000,00	5,68
	Login	18,00	19208,36	299,83	62183,41	43000,00	54000,00	0,95
9	Status Login	1,00	5106,01	52,00	46428,97	10000,00	18000,00	7,58
	List Jadwal Ujian	15,00	6256,28	50,67	58494,82	22000,00	36000,00	8,31
	Mulai Ujian	0,00	5627,09	45,98	35546,73	9574,28	9679,18	8,13
	Daftar Soal Ujian	2,00	5693,87	49,30	54116,64	9636,01	10969,08	7,99
	Soal Ujian	1,00	6060,15	52,21	53751,12	10566,43	10661,52	7,01
	Submit Jawaban	0,00	4842,81	63,20	35757,41	9900,00	13000,00	6,86
	Login	0,00	5197,85	132,22	44716,19	31000,00	39000,00	0,91
10	Status Login	0,00	1374,95	34,05	17675,05	7800,00	11000,00	9,27
	List Jadwal Ujian	0,00	1443,35	32,39	46698,79	7800,00	12000,00	9,95

	Mulai Ujian	0,00	1781,63	29,57	12349,54	6786,44	7192,49	9,83
	Daftar Soal Ujian	0,00	1854,09	31,41	45444,45	7026,86	8244,10	9,69
	Soal Ujian	0,00	2309,80	30,44	24176,83	7158,48	7408,05	8,72
	Submit Jawaban	1,00	1331,30	36,62	13434,53	7700,00	10000,00	8,60
11	Login	8,00	4116,46	136,23	36343,59	17000,00	32000,00	1,69
	Status Login	0,00	1344,83	36,92	18460,56	5700,00	15000,00	8,76
	List Jadwal Ujian	2,00	1496,45	31,61	32711,27	6100,00	16000,00	10,07
	Mulai Ujian	0,00	1485,73	31,36	16364,71	5197,32	5197,32	9,78
	Daftar Soal Ujian	0,00	1535,61	32,87	16122,63	6010,10	6010,10	9,61
	Soal Ujian	0,00	1796,67	31,11	16136,47	5725,89	5725,89	7,82
	Submit Jawaban	0,00	1451,25	39,38	16155,58	5900,00	15000,00	7,55
	12	Login	3,00	8422,48	130,19	47635,47	26000,00	44000,00
Status Login		15,00	2767,54	36,44	31359,12	9300,00	20000,00	6,52
List Jadwal Ujian		18,00	2949,85	33,73	35626,07	13000,00	22000,00	7,95
Mulai Ujian		12,00	2470,17	32,53	24116,43	6042,54	6042,54	7,60
Daftar Soal Ujian		5,00	2515,93	34,26	30786,84	5979,13	5979,13	7,32
Soal Ujian		20,00	3098,81	33,63	27252,56	6695,09	6695,09	5,54
Submit Jawaban		20,00	2811,45	40,14	36990,67	8400,00	19000,00	5,27
13		Login	19,00	9510,04	131,63	74542,37	42000,00	63000,00
	Status Login	47,00	1637,85	34,32	38616,05	9400,00	19000,00	7,48
	List Jadwal Ujian	78,00	2266,12	2,21	48890,65	13000,00	38000,00	8,13
	Mulai Ujian	62,00	2888,95	2,40	82429,45	12677,34	12940,86	7,94
	Daftar Soal Ujian	57,00	2567,06	3,97	45287,93	12951,56	13613,15	7,77

	Soal Ujian	70,00	2706,47	2,78	45064,48	13276,38	13276,38	7,01
	Submit Jawaban	87,00	1823,98	2,33	44385,95	9400,00	24000,00	6,96
14	Login	6,00	14446,09	131,24	56157,67	44000,00	52000,00	0,89
	Status Login	1,00	3355,19	36,28	37810,19	11000,00	15000,00	5,32
	List Jadwal Ujian	10,00	4238,28	33,21	52933,31	15000,00	39000,00	6,03
	Mulai Ujian	0,00	4182,31	33,20	17159,78	9095,25	9095,25	5,86
	Daftar Soal Ujian	0,00	4231,69	34,38	16860,44	9060,45	9060,45	5,72
	Soal Ujian	0,00	4807,37	33,01	26905,60	10025,99	10025,99	4,87
	Submit Jawaban	1,00	3341,17	33,01	23935,98	11000,00	14000,00	4,72
	Login	3,00	12968,14	134,12	62625,64	44000,00	54000,00	0,94
15	Status Login	0,00	2668,64	36,72	23756,02	9800,00	14000,00	6,21
	List Jadwal Ujian	1,00	3526,51	33,17	60654,98	13000,00	33000,00	6,93
	Mulai Ujian	0,00	3340,43	32,28	17476,05	7865,53	7865,53	6,83
	Daftar Soal Ujian	0,00	3304,23	33,47	17424,89	7704,05	7704,05	6,68
	Soal Ujian	1,00	3569,85	30,73	27179,36	8307,57	8307,57	5,71
	Submit Jawaban	1,00	2624,02	41,30	17348,50	9600,00	14000,00	5,54
	Login	0,00	5715,93	126,18	42922,63	32000,00	39000,00	0,94
16	Status Login	0,00	1066,82	32,56	20240,19	6600,00	10000,00	9,67
	List Jadwal Ujian	0,00	1249,81	31,36	35487,95	7700,00	12000,00	10,40
	Mulai Ujian	0,00	1555,78	29,77	13356,41	6695,09	7307,18	10,22
	Daftar Soal Ujian	0,00	1462,98	31,06	38509,39	6567,76	8332,54	10,02
	Soal Ujian	0,00	1415,72	30,61	24393,31	7072,62	7324,61	9,10
	Submit Jawaban	1,00	1017,89	37,49	12450,29	6300,00	10000,00	9,00
	17	Login	0,00	5349,12	127,61	31752,39	25000,00	31000,00

	Status Login	0,00	1293,19	33,09	17534,89	6200,00	9200,00	9,30
	List Jadwal Ujian	0,00	1340,43	31,09	30580,19	6200,00	9600,00	10,10
	Mulai Ujian	0,00	1305,97	31,51	11671,46	5528,75	5656,60	9,94
	Daftar Soal Ujian	0,00	1325,83	32,32	27004,11	5393,61	5686,53	9,79
	Soal Ujian	0,00	1756,73	31,03	19108,30	5753,25	5782,95	8,72
	Submit Jawaban	0,00	1267,35	38,54	11515,22	6100,00	8900,00	8,57
18	Login	1,00	13212,12	131,51	53970,65	43000,00	52000,00	0,91
	Status Login	14,00	2653,98	36,44	39526,00	11000,00	19000,00	6,12
	List Jadwal Ujian	16,00	3722,67	33,82	59652,91	19000,00	44000,00	6,84
	Mulai Ujian	17,00	4128,40	34,19	35445,46	10499,40	10499,40	6,67
	Daftar Soal Ujian	13,00	4048,48	34,14	33681,04	10673,27	10673,27	6,52
	Soal Ujian	14,00	4015,90	33,49	35826,11	10931,25	10931,25	5,63
	Submit Jawaban	7,00	2545,54	40,98	35125,02	10000,00	16000,00	5,52
	Login	2,00	13550,57	147,21	49214,81	40000,00	46000,00	0,91
19	Status Login	8,00	2592,44	37,47	25549,52	9700,00	17000,00	6,30
	List Jadwal Ujian	4,00	3350,88	35,42	47856,03	11000,00	38000,00	6,97
	Mulai Ujian	6,00	3774,38	34,31	23713,05	9193,60	9193,60	6,88
	Daftar Soal Ujian	4,00	3655,08	33,55	19257,48	9224,13	9224,13	6,71
	Soal Ujian	2,00	3418,37	34,67	22827,75	9422,55	9422,55	5,81
	Submit Jawaban	5,00	2501,93	42,34	19275,08	9800,00	12000,00	5,68
	Login	17,00	15727,59	130,19	66375,39	47000,00	58000,00	0,90
20	Status Login	5,00	4073,27	37,52	43918,35	12000,00	19000,00	4,96
	List Jadwal Ujian	17,00	5285,01	34,97	59416,66	21000,00	43000,00	5,65

	Mulai Ujian	8,00	5605,03	31,89	33164,73	11067,95	11067,95	5,47
	Daftar Soal Ujian	9,00	5529,27	34,17	35837,42	10777,54	10777,54	5,34
	Soal Ujian	7,00	6256,80	33,29	32212,20	11374,57	11374,57	4,45
	Submit Jawaban	4,00	3960,23	40,06	31194,17	11000,00	19000,00	4,32
21	Login	0,00	4572,45	124,98	30736,59	20000,00	27000,00	0,97
	Status Login	0,00	1126,85	36,38	18506,56	5200,00	10000,00	9,82
	List Jadwal Ujian	0,00	1249,62	33,45	32708,83	5700,00	11000,00	10,65
	Mulai Ujian	0,00	1033,93	31,65	11969,70	4467,72	4736,75	10,48
	Daftar Soal Ujian	0,00	991,98	33,06	12876,89	4310,64	4525,54	10,29
	Soal Ujian	0,00	1122,13	30,90	18766,48	4529,95	4630,21	9,20
	Submit Jawaban	0,00	1057,33	38,81	15802,67	5100,00	8800,00	9,07
	Login	0,00	11774,19	198,13	53586,00	35000,00	45000,00	0,95
22	Status Login	0,00	3135,91	43,36	32455,28	9100,00	15000,00	5,69
	List Jadwal Ujian	0,00	4207,11	36,22	43841,49	15000,00	33000,00	6,51
	Mulai Ujian	0,00	3715,93	36,42	20983,42	7125,67	7125,67	6,35
	Daftar Soal Ujian	0,00	3611,96	37,38	26918,72	7084,65	7084,65	6,19
	Soal Ujian	0,00	4000,96	33,20	29196,95	7767,97	7767,97	5,16
	Submit Jawaban	0,00	3034,68	46,27	23242,44	8800,00	12000,00	5,01
	Login	0,00	11380,46	275,32	45119,13	36000,00	42000,00	0,93
23	Status Login	0,00	3056,41	45,80	22768,38	9000,00	11000,00	5,82
	List Jadwal Ujian	0,00	3927,17	64,92	42471,82	10000,00	33000,00	6,56
	Mulai Ujian	0,00	3666,24	49,40	12437,94	6960,34	6960,34	6,45
	Daftar Soal Ujian	0,00	3731,03	48,27	17674,78	7049,45	7049,45	6,27

	Soal Ujian	0,00	3661,03	92,25	18649,67	7187,52	7187,52	5,29
	Submit Jawaban	0,00	2946,38	66,72	11382,48	8700,00	10000,00	5,15
24	Login	10,00	11596,96	243,76	55142,45	36000,00	51000,00	0,91
	Status Login	0,00	3454,38	51,51	24223,93	11000,00	15000,00	5,41
	List Jadwal Ujian	1,00	4338,64	53,93	59377,40	14000,00	35000,00	6,10
	Mulai Ujian	0,00	4424,58	47,86	22512,81	8596,84	8596,84	5,96
	Daftar Soal Ujian	0,00	4411,13	45,33	21081,82	8551,68	8551,68	5,81
	Soal Ujian	0,00	4530,26	53,12	20830,71	8503,14	8503,14	4,91
	Submit Jawaban	1,00	3163,81	60,87	18142,06	10000,00	14000,00	4,78
	Login	13,00	14641,49	317,52	72868,82	52000,00	65000,00	0,76
25	Status Login	0,00	4572,11	51,21	26889,89	11000,00	15000,00	5,59
	List Jadwal Ujian	3,00	5569,80	62,40	60803,84	14000,00	43000,00	6,21
	Mulai Ujian	0,00	5165,76	58,74	28391,21	9747,14	9747,14	6,07
	Daftar Soal Ujian	0,00	5318,97	41,16	29332,74	9942,71	9942,71	5,98
	Soal Ujian	5,00	5903,18	62,10	27134,84	10161,17	10161,17	5,14
	Submit Jawaban	5,00	4591,71	72,18	18294,94	11000,00	14000,00	5,03
	Login	6,00	8478,61	293,33	53782,20	28000,00	48000,00	0,91
	Status Login	2,00	2642,07	62,95	44472,51	9600,00	19000,00	6,60
26	List Jadwal Ujian	3,00	2907,72	56,21	45472,14	11000,00	23000,00	7,30
	Mulai Ujian	0,00	3304,08	51,95	41988,48	9039,40	9039,40	7,14
	Daftar Soal Ujian	0,00	3193,20	52,78	37705,51	8512,95	8512,95	7,01
	Soal Ujian	0,00	3482,17	56,82	36205,85	8793,55	8793,55	6,05
	Submit Jawaban	0,00	2503,04	40,49	38073,53	8900,00	15000,00	5,94
	Login	0,00	7854,01	347,86	42714,07	28000,00	35000,00	0,94
27								

	Status Login	0,00	1957,98	55,00	29697,47	7500,00	11000,00	8,02
	List Jadwal Ujian	0,00	2416,69	60,87	36064,45	8100,00	22000,00	8,78
	Mulai Ujian	0,00	2626,61	54,51	14222,52	6153,33	6158,13	8,60
	Daftar Soal Ujian	0,00	2563,64	44,69	17789,77	6107,51	6107,51	8,44
	Soal Ujian	0,00	2592,93	43,98	19207,86	6536,50	6536,50	7,44
	Submit Jawaban	1,00	1922,58	57,75	14865,95	7200,00	9900,00	7,31
28	Login	0,00	7564,10	332,04	32646,24	22000,00	27000,00	0,96
	Status Login	0,00	1796,32	40,88	17073,77	5200,00	7200,00	7,97
	List Jadwal Ujian	1,00	2201,78	38,19	29169,89	6500,00	17000,00	8,73
	Mulai Ujian	0,00	1664,18	33,81	15125,91	4452,07	4452,07	8,58
	Daftar Soal Ujian	0,00	1676,16	33,52	8181,44	4230,48	4230,48	8,44
	Soal Ujian	1,00	1687,41	35,47	12696,66	4225,35	4225,35	7,40
	Submit Jawaban	1,00	1776,89	42,91	8652,06	4900,00	6700,00	7,25
	Login	0,00	1729,06	263,99	16700,29	4600,00	12000,00	0,97
29	Status Login	0,00	562,51	36,01	7594,07	1800,00	4700,00	11,81
	List Jadwal Ujian	0,00	606,99	38,71	8986,93	1900,00	4300,00	12,59
	Mulai Ujian	0,00	605,13	40,94	7221,06	1799,43	1973,43	12,43
	Daftar Soal Ujian	0,00	655,34	39,94	15787,63	1972,00	2310,75	12,29
	Soal Ujian	0,00	659,38	40,32	7596,70	1922,15	2043,80	11,21
	Submit Jawaban	0,00	571,13	50,30	7512,60	1600,00	4100,00	11,09
	Login	0,00	1692,89	158,13	11078,41	4700,00	9100,00	0,99
30	Status Login	0,00	433,04	38,96	5053,24	1300,00	2600,00	12,34
	List Jadwal Ujian	0,00	509,73	35,90	10439,22	1500,00	3000,00	13,16

Mulai Ujian	0,00	431,09	36,94	4581,81	1226,57	1369,37	12,98
Daftar Soal Ujian	0,00	440,92	33,25	6115,48	1250,15	1442,02	12,84
Soal Ujian	0,00	436,39	33,17	3899,27	1218,34	1316,22	11,71
Submit Jawaban	0,00	451,44	46,21	4970,37	1200,00	2300,00	11,57

