

# Pengujian Stress Testing API Sistem Pelayanan dengan Apache JMeter

Ni Luh Ayu Sonia Ginasari<sup>a1</sup>, Kadek Suar Wibawa<sup>a2</sup>, Ni Kadek Ayu Wirdiani<sup>a3</sup>

<sup>a</sup>Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana  
Bukit Jimbaran, Bali, Indonesia

e-mail: <sup>1</sup>[soniaginasari67@gmail.com](mailto:soniaginasari67@gmail.com), <sup>2</sup>[suar\\_wibawa@unud.ac.id](mailto:suar_wibawa@unud.ac.id), <sup>3</sup>[ayuwirdiani@unud.ac.id](mailto:ayuwirdiani@unud.ac.id)

## Abstrak

Sistem yang baik tentunya harus memiliki kualitas API yang mumpuni. Salah satu sistem yang menggunakan API adalah sistem pelayanan laboratorium berbasis Android, sehingga mengakibatkan kualitas API untuk pengaksesan data menjadi salah satu hal yang perlu diperhatikan dalam pembuatan sistem mobile. Salah satu cara untuk memastikan kualitas dari API pada sistem dibutuhkan pengujian sistem seperti Stress testing. Stress testing yang dilakukan dengan menggunakan tools Apache JMeter untuk lebih mengoptimalkan sistem dan memeriksa ketahanan dari sistem. Skenario stress testing ini menggunakan 25, 50, dan 75 sampel dengan periode ramp up adalah 10 detik dan loop count 1 dengan menguji beberapa bidang API seperti laboratorium, bidang, layanan laboratorium, berita, dan pengujian. Hasil yang didapatkan pada pengujian adalah pada bagian rata-rata waktu pengujian (average) mendapatkan nilai yang konstan, pada bagian throughput mendapatkan nilai yang meningkat, dan pada bagian deviation mendapatkan nilai yang menurun sehingga ini membuktikan bahwa sistem mempunyai ketahanan yang lumayan baik.

**Kata kunci:** API, Apache JMeter, Ketahanan sistem, Pengujian, Stress testing

## Abstract

Good system must have a qualified API quality. One system that uses an API is an Android-based laboratory service system, Accessing data to be one of the things that needs to be considered in making a mobile system. One way to ensure the quality of the API on the system requires system testing such as Stress testing. Stress testing is done using Apache JMeter tools to further optimize the system and check the robustness of the system. Stress testing scenario uses 25, 50, and 75 samples with a ramp up period of 10 seconds and a loop count of 1 by testing several API fields such as laboratories, fields, laboratory services, news, and testing. The results obtained in the test are that the average test time gets a constant value, in the throughput section it gets an increasing value, and in the deviation section it gets a decreasing value, so this proves that the system has fairly good resilience.

**Keywords :** API, Apache JMeter, Stress testing, System robustness, System testing

## 1. Introduction

Perkembangan teknologi pada bagian pengolahan informasi terus meningkat seiring dengan perkembangan zaman, sehingga mengakibatkan kualitas API (*Application Programming Interface*) untuk pengaksesan data menjadi salah satu hal yang perlu diperhatikan dalam pembuatan sistem *mobile*. Salah satu sistem yang menggunakan API adalah sistem pelayanan laboratorium berbasis Android. Sistem ini merupakan sistem pelayanan uji sampel laboratorium pengukuran agar pengguna yang menggunakannya dapat menguji sampel dengan lebih efektif, praktis, dan efisien. Sistem ini menggunakan API yang dihubungkan dari 000webhost.com untuk mengakses data menuju database yang disimpan pada 000webhost.com.

---

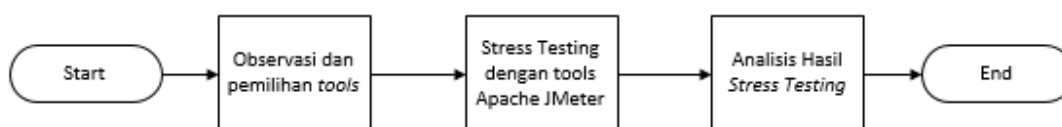
Sistem yang baik tentunya harus memiliki kualitas API yang mumpuni, untuk memastikan kualitas dari API pada sistem dibutuhkan pengujian seperti *Stress testing*. *Stress testing* adalah salah satu jenis pengujian yang digunakan untuk menguji stabilitas dan keandalan sistem. Pengujian ini terutama menentukan ketahanan sistem dan penanganan kesalahan dalam kondisi beban yang sangat berat. *Stress testing* dilakukan untuk memastikan bahwa sistem tidak akan *crash* di bawah situasi krisis. Tujuan dari *stress testing* adalah untuk memastikan kegagalan sistem dan untuk memantau bagaimana sistem pulih kembali dengan baik, jika sistem dapat memulihkannya tanpa kehilangan data dan tanpa menimbulkan kebocoran keamanan berarti telah berhasil melewati *stress testing*.

*Stress testing* dapat dilakukan melalui *load testing tools*, seperti salah satunya adalah Apache JMeter. Apache JMeter adalah perangkat lunak *open source*, aplikasi Java murni 100% yang dirancang untuk memuat perilaku fungsional pengujian dan mengukur kinerja. *Tools* ini dapat digunakan untuk menguji kinerja baik pada sumber daya statis dan dinamis. Apache JMeter mengirim permintaan ke server target dengan mensimulasikan sekelompok pengguna kemudian mengumpulkan data untuk menghitung statistik dan menampilkan metrik kinerja melalui berbagai format.

Berdasarkan hal tersebut maka pengujian *stress testing* sistem pelayanan laboratorium perlu dilakukan untuk mengoptimalkan sistem menjadi lebih efektif dan efisien dengan *tools* yang digunakan untuk melakukan *stress testing* adalah Apache JMeter.

## 2. Metodologi Penelitian

Metodologi penelitian ini menjelaskan mengenai alur tahapan *stress testing* yang akan dilakukan dengan menggunakan Apache JMeter untuk lebih mengoptimalkan sistem dan memeriksa ketahanan dari sistem. Alur tahapan dari pengujian ini adalah sebagai berikut.



Gambar 1. Alur Tahapan Penelitian

Gambar 1 merupakan alur tahapan penelitian *stress testing* API sistem pelayanan laboratorium berbasis Android. Penelitian ini dimulai dengan observasi dan pemilihan *tools* yang sesuai dengan sistem, *tools* yang digunakan adalah Apache JMeter. Selanjutnya dilanjutkan dengan dimulainya *stress testing* dan diakhiri dengan Analisa hasil pengujian.

## 3. Kajian Pustaka

Terdapat beberapa teori yang digunakan pada penelitian ini. Teori-teori yang digunakan dalam *Stress testing* API Sistem pelayanan ini adalah sebagai berikut.

### 3.1. API (Application Programming Interface)

*Application programming interface* (API) merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. API dapat dikatakan sebagai penghubung suatu sistem dengan sistem lainnya yang memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi [1].

### 3.2. Pengujian Sistem

Pengujian sistem diperlukan untuk memastikan apakah sistem dapat berjalan dengan baik sesuai dengan fungsionalitas yang diharapkan. Pengujian merupakan aspek penting dalam menentukan kualitas sistem. Pengujian ini tidak terpisah dari siklus hidup pengembangan sistem [2].

### 3.3. Stress Testing

*Stress testing* adalah cara untuk melihat bagaimana dampak dari suatu situasi yang berpotensi menimbulkan *error* pada tingkat ketahanan sistem. Pengujian akan dilakukan dengan jumlah pengguna yang berubah-ubah dalam suatu waktu dan akan diuji bagaimana respon yang diberikan oleh sistem saat diakses dengan jumlah pengguna yang banyak dan dalam waktu tertentu [3].

### 3.4. Apache JMeter

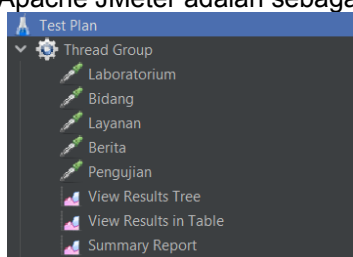
Apache JMeter adalah proyek *open source* yang digunakan dengan menggunakan bahasa Java yang digunakan untuk *load* dan *performance testing tool*. Apache JMeter akan membuat beberapa simulasi pengguna yang akan mengakses server dengan jumlah user dan jeda *request* yang berbeda-beda tergantung konfigurasi yang dilakukan pada Apache JMeter [4].

## 4. Hasil dan Pembahasan

Hasil dan pembahasan penelitian *stress testing* ini menggunakan *tools* Apache JMeter dan menganalisa hasil pengujian yang telah dilakukan. Hasil dan pembahasan dari penelitian ini adalah sebagai berikut.

### 4.1. Pengujian Stress Testing

Pengujian *stress testing* dilakukan dengan memasukkan alamat API sistem pelayanan pada *tools* Apache JMeter. Pengujian ini menguji beberapa bidang API seperti laboratorium, bidang laboratorium, layanan laboratorium, berita laboratorium, dan pengujian uji sampel laboratorium. Skenario *testing* pada Apache JMeter adalah sebagai berikut.



Gambar 2. Skenario *testing* pada Apache JMeter

Gambar 2 merupakan skenario *testing* pada Apache JMeter. Skenario *testing* ini menggunakan 25, 50, dan 75 sampel dengan periode *ramp up* adalah 10 detik dan *loop count* adalah 1. Hasil dari pengujian skenario *testing* terhadap 25 sampel adalah sebagai berikut.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
Laboratoriu...	5	1138	782	2038	487.18	0.00%	34.0/min	1.49	0.08	2691.0
Bidang	5	498	481	533	18.25	0.00%	37.2/min	0.82	0.09	1354.0
Layanan	5	493	472	511	12.44	0.00%	37.2/min	1.22	0.09	2015.0
Berita	5	498	475	524	16.68	0.00%	37.3/min	0.57	0.08	935.0
Pengujian	5	502	479	524	18.54	0.00%	37.2/min	0.68	0.10	1128.0
TOTAL	25	626	472	2038	336.43	0.00%	2.3/sec	3.66	0.34	1624.6

Gambar 3. Hasil Pengujian 25 sampel pada Apache JMeter

Gambar 3 merupakan Hasil pengujian 25 sampel pada Apache JMeter. Hasil dari pengujian menyatakan bahwa pengujian berhasil dengan 0 *error*. Hasil dari pengujian skenario *testing* terhadap 50 sampel adalah sebagai berikut.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
Laboratoriu...	10	879	739	1531	233.00	0.00%	1.1/sec	2.84	0.15	2691.0
Bidang	10	500	463	528	19.04	0.00%	1.2/sec	1.61	0.18	1354.0
Layanan	10	493	467	521	15.07	0.00%	1.2/sec	2.40	0.18	2015.0
Berita	10	503	471	540	20.18	0.00%	1.2/sec	1.12	0.16	935.0
Pengujian	10	503	468	550	23.45	0.00%	1.2/sec	1.35	0.21	1128.0
TOTAL	50	576	463	1531	184.85	0.00%	4.5/sec	7.08	0.66	1624.6

Gambar 4. Hasil Pengujian 50 sampel pada Apache JMeter

Gambar 4 merupakan Hasil pengujian 50 sampel pada Apache JMeter. Hasil dari pengujian menyatakan bahwa pengujian berhasil dengan 0 error. Hasil dari pengujian scenario *testing* terhadap 75 sampel adalah sebagai berikut.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
Laboratoriu...	15	823	743	1349	143.01	6.67%	1.5/sec	4.35	0.21	3006.7
Bidang	15	529	468	895	100.48	13.33%	1.6/sec	3.40	0.24	2163.9
Layanan	15	491	464	517	19.14	20.00%	1.6/sec	4.85	0.23	3097.7
Berita	15	503	457	582	29.83	26.67%	1.6/sec	4.14	0.21	2666.6
Pengujian	15	499	473	547	20.65	33.33%	1.6/sec	4.98	0.27	3228.3
TOTAL	75	569	457	1349	150.93	20.00%	6.1/sec	16.90	0.90	2832.6

Gambar 5. Hasil Pengujian 75 sampel pada Apache JMeter

Gambar 5 merupakan Hasil pengujian 75 sampel pada Apache JMeter. Hasil dari pengujian menyatakan bahwa pengujian mendapat total *error* sebanyak 20% dengan pesan *too many request* pada server. Rangkuman hasil dari pengujian 25, 50 dan 75 sampel pada Apache Jmeter adalah sebagai berikut.

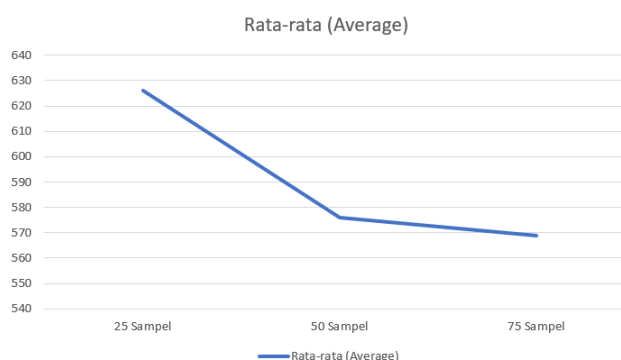
Tabel 1. Rangkuman Hasil Pengujian

Parameter	25 Sampel	50 Sampel	75 Sampel
Average	626	576	569
Deviation	336.43	184.85	150.93
Throughput	2.3/sec	4.5/sec	6.1/sec
Error	0%	0%	20%
Byte Send	0.34	0.66	0.9
Byte Received	3.66	7.08	16.90

Tabel 1 merupakan rangkuman dari hasil pengujian 25, 50 dan 75 sampel pada Apache JMeter. Hasil pengujian 75 sampel pada Apache JMeter. *Throughput* dihitung sebagai permintaan/satuan waktu. Waktu dihitung dari awal sampel pertama hingga akhir sampel terakhir. Ini termasuk interval antar sampel, karena seharusnya mewakili beban di *server*. *Deviation* menunjukkan penyimpangan dari rata-rata [5].

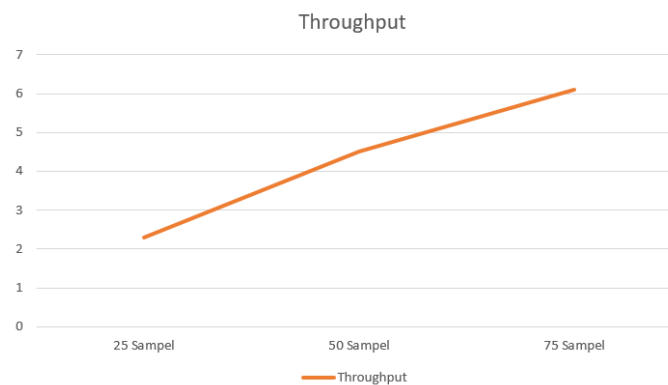
#### 4.2. Analisa Hasil Pengujian

Pengujian yang telah dilakukan terhadap 25, 50, dan 75 sampel menghasilkan beberapa poin penting. Poin penting yang ditunjukkan terdapat pada rata-rata, *throughput*, dan *error*. Berikut ini merupakan hasil grafik dari rata-rata waktu pengujian.



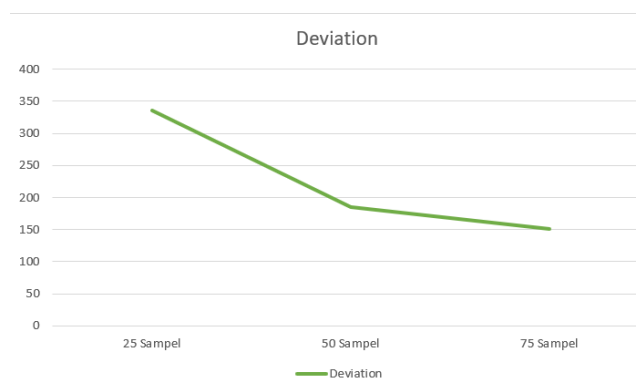
Gambar 6. Grafik Hasil Pengujian Rata-rata (Average) Waktu

Gambar 6 merupakan tampilan grafik hasil pengujian rata-rata (*average*) waktu pada ms. Bisa dilihat pada grafik bahwa rata-rata waktu yang didapatkan tidak berbeda jauh sehingga dapat berjalan secara konsisten. Berikut ini merupakan hasil grafik dari *throughput* pengujian.



Gambar 7. Grafik Hasil Pengujian *Throughput*

Gambar 7 merupakan tampilan grafik hasil pengujian dari *throughput* per detiknya. Bisa dilihat pada grafik bahwa *throughput* yang didapatkan meningkat sebanding dengan jumlah request sampel. Hal ini menunjukkan bahwa kinerja sistem dalam menangani beban dapat berjalan dengan baik dikarenakan semakin meningkatnya *throughput* sebanding dengan jumlah request sampel maka semakin baik kinerja server. Berikut ini merupakan hasil grafik dari pengujian *deviation*.



Gambar 8. Grafik Hasil Pengujian *Deviation*

Gambar 8 merupakan tampilan grafik hasil pengujian *deviation* pada ms. Bisa dilihat pada grafik bahwa *deviation* yang didapatkan hasilnya menurun yang berarti bahwa penyimpangan dari rata-rata semakin kecil sehingga semakin kecil penyimpangan maka semakin baik hasilnya.

## 5. Kesimpulan

Pengujian yang dilakukan pada penelitian ini yaitu *stress testing* API sistem pelayanan dengan tools Apache JMeter. *Stress testing* yang dilakukan dengan menguji 5 bidang API yaitu laboratorium, bidang laboratorium, layanan laboratorium, berita laboratorium, dan pengujian uji sampel laboratorium. *Stress testing* ini menggunakan 25, 50, dan 75 sampel dengan periode ramp up 10 detik dan count loop adalah 1. Hasil yang didapatkan pada pengujian adalah pada bagian rata-rata waktu pengujian (*average*) mendapatkan nilai yang konstan, pada bagian *throughput* mendapatkan nilai yang meningkat sebanding dengan jumlah request sampelnya, dan pada bagian *deviation* mendapatkan nilai yang menurun sehingga ini membuktikan bahwa sistem mempunyai ketahanan yang lumayan baik dimana hasil grafik nilai *average*, *throughput*, dan *deviation* mendapat nilai yang baik.

## References

- [1] M. F. Ramadhani, "Pembangunan Aplikasi Informasi, Pengaduan, Kritik, Dan Saran Seputar Kota Cimahi Pada Platform Android," *J. Ilm. Komput. dan Inform.*, p. 9, 2015.
- [2] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan

- Apache JMeter pada Sistem Informasi Pertanian,” *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.
- [3] M. Arman, “Analisa Kinerja Web Server E-learning Menggunakan Apache Benchmark dan Httperf,” *J. Integr.*, vol. 8, no. 2, pp. 93–100, 2016.
- [4] S. A. Kamarudin, Kusrini, “Uji kinerja sistem web service pembayaran mahasiswa menggunakan apache jmeter (studi kasus: Universitas amikom yogyakarta),” *Teknol. Inf.*, vol. XIII, no. 1, pp. 44–52, 2018.
- [5] Dr. Y R Ghodasara, D. K. C. Kamani, D. G. J. Kamani, M. P. S. Parsania, and 1Professor, “STRESS TESTING OF WEBSITES USING,” *IJCRT*, vol. 6, no. 2, pp. 157–160, 2018.
-