Infotekmesin
Vol.15, No.01, Januari 2024
p-ISSN: 2087-1627, e-ISSN: 2685-9858
DOI: 10.35970/infotekmesin.v15i1.2176, pp.211-215

Perbandingan Kinerja Antara *Gatling* dan *Apache JMeter* pada Uji Beban RESTful API

Prih Diantono Abda'u[1*], Agus Susanto[2], Abdul Rohman Supriyono[3], Dwi Novia Prasetyanti[4]
[1, 2,3,4]Program Studi Teknik Informatika, Politeknik Negeri Cilacap
[1,2,3,4]Jln. Dr. Soetomo No.1 Karangcengis Sidakaya, Kabupaten Cilacap, 53212, Indonesia
E-mail: abdau@pnc.ac.id[1], agussusanto@pnc.ac.id[2], a.rohman.sy@pnc.ac.id[3], dnprasetyanti@pnc.ac.id[4]

Info Naskah:
Naskah masuk: 14 Desember 2023
Direvisi: 23 Januari 2024
Diterima: 24 Januari 2024

Abstrak

Penelitian ini mengeksplorasi dan membandingkan kinerja dua alat pengujian beban populer, yaitu Gatling dan Apache JMeter, dengan fokus pada pengujian kinerja API. Pertumbuhan pesat dalam pengembangan aplikasi web dan mobile menyoroti kebutuhan mendesak untuk memastikan kinerja API yang optimal. Penelitian ini dilakukan untuk memberikan wawasan mendalam mengenai kelebihan dan kekurangan kedua alat pengujian ini melalui penggunaan skenario pengujian serupa. Metode eksperimental melibatkan penerapan skenario pengujian yang mencakup variasi beban dan permintaan tinggi pada kedua alat. Parameter utama yang diamati meliputi waktu respons API, throughput, dan latensi. Analisis mendalam dilakukan terhadap data yang diperoleh untuk mengevaluasi keandalan dan efisiensi masing-masing alat. Hasil penelitian ini memberikan pemahaman komprehensif mengenai kinerja Gatling dan Apache JMeter dalam konteks pengujian kinerja API. Temuan ini dapat memberikan panduan praktis bagi pengembang perangkat lunak dan praktisi pengujian dalam memilih alat pengujian beban yang sesuai dengan kebutuhan proyek. Rekomendasi saran untuk penelitian kedepan mencakup pemperluas eksplorasi alat pengujian beban lainnya, pembandingan dengan skenario uji yang lebih kompleks, dan integrasi dengan alat monitoring kinerja untuk analisis yang lebih holistik. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap pemahaman dan pemilihan alat pengujian beban yang efektif dalam pengembangan aplikasi web dan mobile.

Keywords:
API performances;
load testing tools;
gatling and jmeter comparison;

Abstract

This research explores and compares the performance of two popular load testing tools, namely Gatling and Apache JMeter, with a focus on API performance testing. The rapid growth in web and mobile application development highlights the urgent need to ensure optimal API performance. This research was conducted to provide in-depth insight into the advantages and disadvantages of these two testing tools through the use of similar testing scenarios. The experimental method involves implementing test scenarios that include load variations and high demands on both devices. The main parameters observed include API response time, throughput, and latency. In-depth analysis was carried out on the data obtained to evaluate the reliability and efficiency of each tool. The results of this research provide a comprehensive understanding of the performance of Gatling and Apache JMeter in the context of API performance testing. These findings can provide practical guidance for software developers and testing practitioners in selecting load testing tools that suit their project needs. Recommendations for future research include expanding exploration of other load testing tools, comparison with more complex test scenarios, and integration with performance monitoring tools for more holistic analysis. Thus, this research is expected to make a significant contribution to the understanding and selection of effective load testing tools in web and mobile application development.

***Penulis korespondensi:**
Prih Diantono Abda'u
E-mail: abdau@pnc.ac.id

1. Pendahuluan

Pentingnya pengujian perangkat lunak yang kuat dalam pengembangan perangkat lunak dinamis tidak dapat diabaikan. Jaminan kualitas menjadi kunci untuk memastikan bahwa aplikasi tidak hanya memenuhi ekspektasi kinerja tetapi juga memberikan pengalaman pengguna yang mulus. Metodologi pengujian yang ketat memegang peran penting dalam mengidentifikasi dan mengatasi potensi masalah sebelum berdampak pada pengguna akhir, menjadikan pengujian perangkat lunak sebagai komponen yang sangat penting dalam siklus hidup pengembangan[1].

Peran strategis *Application Programming Interfaces* (API) sebagai tulang punggung pengembangan perangkat lunak modern semakin meningkat. API memfasilitasi komunikasi yang efisien antara berbagai komponen perangkat lunak, mendukung pertumbuhan aplikasi web dan mobile. Efektivitas serta efisiensi API langsung memengaruhi kinerja aplikasi secara keseluruhan, menjadikan pengujian menyeluruh sebagai prasyarat[2].

Dalam konteks pengujian perangkat lunak, pengujian beban memiliki peran krusial. Pendekatan ini melibatkan penilaian kinerja aplikasi di bawah berbagai tingkat beban dan permintaan, penting untuk mengukur ketahanan, skalabilitas, dan respons aplikasi dalam skenario dunia nyata. Penelitian ini secara khusus mendalami domain krusial pengujian beban, fokusnya terletak pada evaluasi dua alat terkemuka: *Gatling* dan *Apache JMeter*, untuk menguji kinerja API[3].

Gatling dan Apache JMeter muncul sebagai alat pengujian beban terdepan dengan keunggulan dan kekurangan masing-masing. Dalam upaya mengoptimalkan kinerja API, pemilihan alat pengujian menjadi kritis. Penelitian ini bertujuan untuk memberikan perbandingan mendalam antara *Gatling* dan *Apache JMeter*, menyoroti keunggulan dan kelemahan keduanya dalam konteks pengujian API[4].

Untuk menilai kinerja API secara komprehensif, penelitian ini memasukkan metrik utama seperti waktu respons, throughput, dan latensi. Metrik ini memberikan pemahaman yang holistik tentang perilaku API dalam berbagai skenario beban dan permintaan. Dengan mengeksplorasi pengukuran ini, penelitian bertujuan memberikan evaluasi menyeluruh terhadap kemampuan *Gatling* dan *Apache JMeter* dalam memberikan hasil pengujian API yang akurat dan dapat diandalkan[5].

Perbandingan antara dua alat pengujian beban terkemuka, yaitu *Gatling* dan *Apache JMeter*, memberikan kontribusi berharga dalam memahami perbedaan kunci di antara keduanya. Penggunaan metrik utama seperti waktu *respons*, *throughput*, dan latensi memberikan pandangan holistik terhadap kinerja API, yang dapat menjadi panduan berharga bagi praktisi pengujian dan pengembang perangkat lunak. Selain itu, rekomendasi untuk menggunakan lebih dari satu alat pengujian beban dalam pengujian API menunjukkan pemahaman mendalam terhadap kompleksitas interaksi API dan menawarkan solusi praktis untuk pengambilan keputusan yang lebih baik dalam pemilihan alat pengujian. Inisiatif ini memberikan kontribusi yang berarti dalam mendukung praktik pengujian yang lebih efektif dan terinformasi di dalam industri pengembangan perangkat lunak.

Pada penelitiannya yang berjudul *Comparative Analysis of Web Platform Assessment Tools*, penulis membandingkan 4 platform pengujian sekaligus yaitu Apache JMeter, *Apache Flood*, *The Grinder*, dan *Gatling* dengan metode kualitatif dan kuantitatif[6]. Namun pengujian berfokus pada pengiriman data didalam *Search Engine*, sedangkan penelitian ini berfokus pada pengujian API pada sebuah aplikasi *Cross-Platform* yaitu E-Patrol.

2. Metode

Melakukan studi perbandingan kinerja API menggunakan alat uji beban seperti *Gatling* dan *Apache JMeter* memerlukan pendekatan sistematis untuk memastikan hasil yang akurat dan bermakna[7]. Metodologi pada penelitian ini seperti pada gambar 1.

Gambar 1. Metodologi Penelitian

Berdasarkan pada Gambar 1, untuk tahapan kegiatan penelitian yang dilakukan diantaranya yaitu:

1) Menentukan Skenario Uji sesuai dengan pola penggunaan pada dunia nyata. Mempertimbangkan jenis permintaan API seperti GET atau POST serta ukuran *payload* yang berbeda[8].
2) Eksekusi Uji, menjalankan skenario uji yang telah ditentukan menggunakan Gatling dan Apache JMeter. Memantau sumber daya sistem selama pengujian untuk mengidentifikasi potensi *bottleneck*. Uji dijalankan beberapa kali untuk memperhitungkan variabilitas dan memastikan keandalan[9].
3) Mengumpulkan metrik kinerja selama eksekusi uji. Mencakup waktu respons, *throughput*, dan tingkat kesalahan (*errors*).
4) Analisis dan Perbandingan, mengumpulkan data yang dikumpulkan untuk setiap alat pengujian. Bandingkan metrik kinerja dan identifikasi perbedaan atau pola yang signifikan[10].
5) Mendokumentasikan seluruh proses, termasuk skenario uji dan hasil uji[11].
6) Meringkas temuan dan menarik kesimpulan berdasarkan perbandingan kinerja. Memberikan rekomendasi untuk pemilihan Gatling atau Apache JMeter berdasarkan hasil studi.

3. Hasil dan Pembahasan

3.1 Menentukan Skenario Uji

Skenario atau tahapan yang akan dilakukan dalam pengujian ini antara lain:

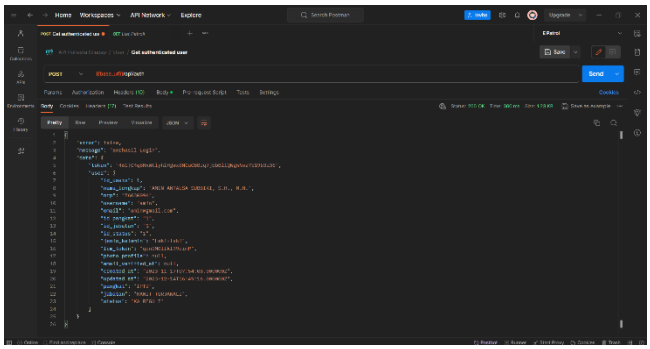
a) Pengguna melakukan *login*.
b) Memastikan *response code* yang didapatkan sesuai dengan yang diharapkan.

c) Pengguna melakukan proses memuat data tugas patrol.
d) Memastikan *response code* yang didapatkan sesuai dengan yang diharapkan.
e) Langkah a-d diulang hingga 300 kali
f) Menunggu jeda waktu 1 detik (1000ms)

3.2 Eksekusi Uji

Menjalankan eksekusi uji berdasarkan skenario yang telah disusun.

a. Proses login E-Patrol

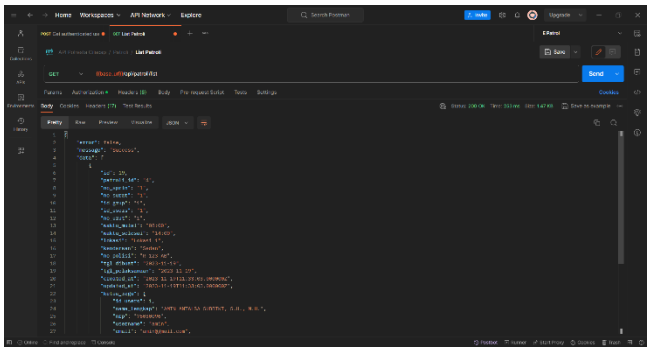


Gambar 2. API Login

b. Menggunakan *assertion* dan memastikan *response code* 200

Gambar 3. *Response Code* 200

c. Memuat data tugas patroli



Gambar 4. API list data patroli

d. Memastikan kembali agar *response code* yang muncul adalah 200 (sukses) bukan 500 (gagal)
e. Menunggu jeda waktu 1 detik (1000ms) yang diatur pada konfigurasi pengujian.

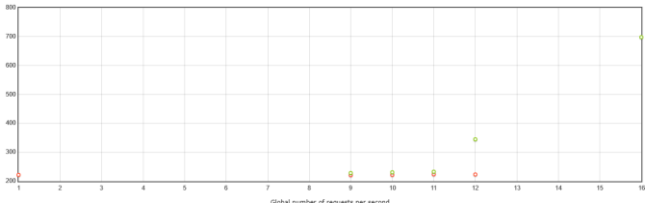
3.3 Pengumpulan Data

Hasil pengujian menggunakan Apache JMeter:

a. *Response Time*

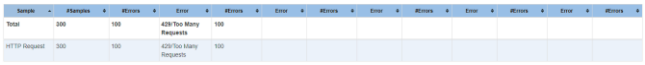
Gambar 5. *Response time* Apache JMeter

b. *Throughput*



Gambar 6. Throughput JMeter

c. *Errors*



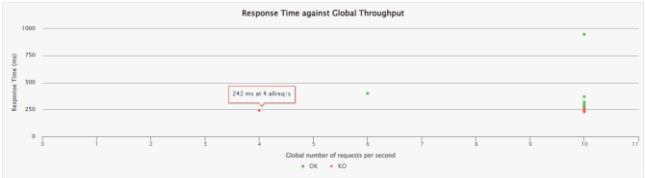
Gambar 7. *Errors* JMeter

3.3 Hasil pengujian menggunakan *Gatling*

a. *Response time*

Gambar 8. *Response time* Gatling

b. *Throughput*



Gambar 9. *Throughput* Gatling

c. *Errors*

Gambar 10. *Errors* Gatling

3.4 Analisis dan Perbandingan

Pada bagian ini penulis akan menampilkan perbandingan antara Gatling dengan Apache Jmeter secara kualitatif dan secara kuantitatif.

a) Analisis Kualiatif

Analisis kualitatif hanya dilakukan pada karakteristik yang paling relevan. Tabel 1 menunjukkan analisis kualitatif dari Gatling dan Apache Jmeter sebagai alat pengujian beban pada API.

Tabel 1. Analisis kualitatif

No	Karakteristik	Alat	
		Gatling	Jmeter
1	Bahasa Pemrograman	Scala	Java
2	Arsitektur dan Eksekusi	Asinkron berbasis Akka	Sinkron dengan thread per pengguna
3	Konfigurasi	DSL (*Domain-Specific Language*)	Antarmuka pengguna grafis
4	Integrasi dan Ekosistem	Terbatas	Luas, banyak plugin yang tersedia

Berdasarkan hasil analisis pada tabel 1, maka dapat disimpulkan bahwa Gatling memiliki lebih banyak keunggulan pada sisi arsitektur, eksekusi dan konfigurasi. Namun Apache JMeter lebih unggul pada sisi luasnya komunitas sehingga terdapat banyak plugin untuk mempermudah penggunaan sesuai dengan kebutuhan[12]. Akan tetapi karena penelitian ini berfokus pada performa pengujian beban, maka hasil akhir dari perbandingan antara Gatling dan Apache JMeter akan diperkuat dengan analisis kuantitatif[13] seperti yang akan dijelaskan pada bagian selanjutnya.

b) Analisis Kuantitatif

Analisis kuantitatif dilakukan terhadap karakteristik yang relevan seperti *response time*, *throughput,* dan *errors.* Dari berbagai karakteristik yang dihasilkan oleh Apache JMeter[14], tiga karakteristik tersebutlah yang memiliki kesamaan dengan output yang dihasilkan oleh Gatling. Tabel 2 menunjukkan perbandingan dari pengujian beban pada REST API yang dieksekusi sebanyak 300 kali seperti yang disajikan pada tahap pengumpulan data.

Tabel 2. Analisis kuantitatif

No	Karakteristik	Alat	
		Gatling	Jmeter
1	*Response time (average)*	251 ms	262 ms
2	*Throughput*	9,67 transactions/s	9,99 transactions/s
3	Errors	33%	33%

Berdasarkan hasil pada tabel 2, maka dapat disimpulkan bahwa *Gatling* menghasil waktu respons yang lebih rendah dibandingan JMeter, menunjukkan bahwa Gatling lebih efisien dalam menangani beban uji tertentu. Meskipun waktu response Gatling lebih rendah, tetapi *throughput* Apache JMeter lebih tinggi. *Throughput* mengukur jumlah transaksi dalam satu detik, hal ini menunjukkan bahwa JMeter dapat menangani lebih banyak transaksi per detik dibandingkan Gatling.

4. Kesimpulan dan Rekomendasi

Berdasarkan hasil analisis kualitatif dan analisis kuantitatif yang telah dilakukan, diperoleh kesimpulan bahwa Gatling dan Apache JMeter menunjukkan performa yang sebanding. Dalam penilaian ini, waktu respons menjadi keunggulan utama Gatling, sementara Apache JMeter mencatat keunggulan dalam jumlah transaksi per detik. Rekomendasi untuk memberikan penilaian lebih lanjut adalah dengan merujuk pada standar penilaian yang telah diakui dalam literatur ilmiah terkait analisis komparatif alat penilaian platform web[15].

Berdasarkan hasil pada tabel 2, pemilihan antara *Gatling* dan JMeter sebaiknya didasarkan pada kebutuhan, krakteristik proyek, dan skenario uji yang sesuai. Jika kecepatan waktu respon (*response time*) menjadi fokus utama, Gatling merupakan pilihan yang baik. Akan tetapi jika *throughput*, fitur umum serta ketersediaan plugin lebih penting, maka Apache JMeter bisa menjadi pilihan yang sesuai.

Ucapan Terimakasih

Penelitian ini dibiayai oleh dana DIPA No: 072/PL43/HK.07/2023 Pusat Penelitian dan Pengabdian Kepada Masyarakat Politeknik Negeri Cilacap, sesuai perjanjian pelaksanaan Penelitian Nomor: 535/PL43.PO1/LL/2023.

Daftar Pustaka

[1] M. S. Lamada, A. S. Miru, and R.- Amalia, “Pengujian Aplikasi Sistem Monitoring Perkuliahan Menggunakan Standar ISO 25010,” *Jurnal MediaTIK*, vol. 3, no. 3, 2020, doi: 10.26858/jmtik.v3i3.15172.

[2] G. H. Setiawan, I. M. B. Adnyana, and K. Budiarta, “Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing,” pp. 539–542.

[3] D. I. Permatasari, “Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian,” *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.

[4] A. Amarulloh, “ANALISIS PERBANDINGAN PERFORMA WEB SERVICE REST MENGGUNAKAN FRAMEWORK LARAVEL, DJANGO, DAN Node JS PADA APLIKASI BERBASIS WEBSITE,” vol. 09, no. 01, pp. 12–17, 2023.

[5] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, “Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website,” *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 11, no. 1, p. 48, 2021, doi: 10.36448/expert.v11i1.2012.

[6] S. Paz and J. Bernardino, “Comparative analysis of web platform assessment tools,” *WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies*, no. Webist, pp. 116–125, 2017, doi: 10.5220/0006308101160125.

[7] A. H. Rakhmah and H. Purwoko, “Efektivitas Web Api Dalam Integrasi Bahasa Pemrograman Multi Platform,” *Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi)*, vol. 5, no. 1, pp. 18–22, 2021.

[8] N. A. Rahman, M. M., & Abdullah, “A Survey on Software Load Testing Techniques.,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 78–85, 2019.

[9] R. Hidayanto and P. Sawitri, "Performance Testing of e-Payment Website Using JMeter," *International Research Journal of Advanced Engineering and Science*, vol. 4, no. 3, 2019.

[10] S. Menon, "Performance Testing: Concepts, Techniques, and Tools.," *ournal of Software Engineering and Applications*, vol. 11, no. 09, pp. 433–449, 2018.

[11] R. T. Fielding *et al.*, "Reflections on the REST architectural style and 'principled design of the modern web architecture' (impact paper award," in *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 2017. doi: 10.1145/3106237.3121282.

[12] J. Wang and J. Wu, "Research on performance automation testing technology based on JMeter," in *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, 2019. doi: 10.1109/ICRIS.2019.00023.

[13] R. K. Lenka, S. Mamgain, S. Kumar, and R. K. Barik, "Performance Analysis of Automated Testing Tools: JMeter and TestComplete," in *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 2018. doi: 10.1109/ICACCCN.2018.8748521.

[14] V. Tiwari, S. Upadhyay, J. K. Goswami, and S. Agrawal, "Analytical Evaluation of Web Performance Testing Tools: Apache JMeter and SoapUI," in *Proceedings - 2023 12th IEEE International Conference on Communication Systems and Network Technologies, CSNT 2023*, 2023. doi: 10.1109/CSNT57126.2023.10134699.

[15] S. Paz and J. Bernardino, "Comparative analysis of web platform assessment tools," *WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies*, no. Webist, pp. 116–125, 2017, doi: 10.5220/0006308101160125.