Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

TESTING PERFORMA FRAMEWORK EXPRESS.JS, GIN, DAN FASTAPI MENGGUNAKAN API DAN JMETER

Performance Testing Of Express.Js, Gin, And Fastapi Frameworks Using Api And Jmeter

Santiago[1], Benisius[2*]

Program Studi Informatika[1 2]
Fakultas Teknik dan Ilmu Komputer [1 2]
Universitas Kristen Krida Wacana[1 2]

santiago.412020004@civitas.ukrida.ac.id[1], ehba@ukrida.ac.id[2*]

Received: June 20, 2025**. Revised:** July 5, 2025**. Accepted:** July 9, 2025 **. Issue Period:** Vol.9 No.3 (2025), Pp. 1219-1226

Abstrak: Application Programming Interface (API) adalah antarmuka software-to-software yang mendifinisikan kontrak untuk suatu aplikasi untuk berbicara satu sama lain melalui internet tanpa interaksi pengguna. Pemilihan bahasa pemrograman dan framework untuk mengembangkan sebuah API melibatkan banyak pertimbangan. Pada survei yang dilakukan oleh Stackoverflow pada tahun 2023, menunjukkan bahwa bahasa pemrograman terpopuler pada tahun 2023 merupakan JavaScript dengan persentase pemilihan 63.61% dari 87.585 responden, diikuti dengan Python yang menduduki posisi ke 3 dengan tingkat pemilihan 49.28% dari 87.585 responden dan Go dengan tingkat pemilihan 13.24% dari 87.585 responden. Salah satu pertimbangan penting dalam memilih bahasa pemrograman dan framework yang akan dipilih adalah performanya, performa dari suatu API dapat diuji dengan melakukan uji beban. Penelitian ini mengusulkan sebuah pengujian beban terhadap framework Express.js, FastAPI, dan Gin dengan menggunakan JMeter untuk mencari parameter (Num threads, dan Ramp time) optimalnya. Melalui pengujian Gin memiliki berperforma paling baik dibandingkan kedua framework yang diuji dengan utilisasi CPU 0.03% dan penggunaan memori sebesar 26.74 MB serta throughput yang selalu unggul dengan ramp time yang singkat, namun bilamana ramp time tidak begitu singkat framework Express dapat menyaingi Gin dengan troughput yang sama.

Kata kunci: Application Programming Interface (API), JMeter, load testing

Abstract: *An Application Programming Interface (API) serves as a software-to-software interface that establishes a communication contract between applications over the internet without requiring user interaction. Choosing the right programming language and framework for API development can be challenging due to various considerations. According to a 2023 Stack Overflow survey, JavaScript was the most popular programming language, selected by 63.61% of the 87,585 respondents, followed by Python in third place with a 49.28% selection rate, and Go with 13.24%. A key factor in selecting the language and framework is performance, which can be evaluated through load testing. This study proposes load testing of three frameworks—Express.js, FastAPI, and Gin—using JMeter to optimize its parameters (number of threads and ramp time). Gin outperformed the other frameworks, achieving only 0.03% CPU utilization, 26.74 MB memory usage, and consistently high throughput with a short ramp time. However, with longer ramp times, Express can match Gin in terms of throughput.*

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Keywords: *Application Programming Interface (API), JMeter, Load Testing*

I. PENDAHULUAN

Setiap tahunnya jumlah pengguna internet terus mengalami pertumbuhan dan pada 2029 diperkirakan akan mencapai 7.9 milyar orang [1]. Pada Juli 2024, terdapat 5,45 miliar pengguna internet di seluruh dunia, yang berarti 67,1 persen dari populasi global [2]. Di Indonesia sendiri jumlah pengguna internet juga terus meningkat. Hal ini terindikasi dari data penggunaan internet yang naik sekitar 78.8% pada 2020. Selain itu kenaikan jumlah kepemilikan komputer rumah di tahun 2020 juga mengalami pertumbuhan 18.83% [3]. Salah satu layanan internet yang paling populer yang telah menjadi alat penting bagi bisnis untuk mempromosikan layanan, mengotomatisasi proses bisnis, serta menyimpan informasi penting adalah web. Aplikasi web bekerja dengan cara mengambil dari data dari server. Seiring meningkatnya permintaan data lintas berbagai sistem, pengembang dihadapkan pada tantangan untuk dapat menyediakan dan mengintegrasikannya dengan mudah ke dalam aplikasi lain. Salah satu pendekatan umum adalah dengan menggunakan API (Application Programming Interface) [4]. API adalah sebuah antarmuka yang berperan sebagai titik masuk ke entitas perangkat lunak yang dapat digunakan secara berulang [5]. API bukanlah entitas perangkat lunak tunggal yang independen, melainkan dikemas dan ditawarkan oleh software libraries [6], frameworks [7], atau web services [8]. Melaluinya pertukaran data, fitur dan fungsional antara aplikasi software dapat terfasilitasi serta menghemat waktu dan tenaga [9]. Oleh karena itu tidak mengherankan jika API banyak digunakan oleh pengembang perangkat lunak dan penggunaannya sangat dianjurkan untuk meningkatkan kualitas perangkat lunak sekaligus menjadikan pengembangan menjadi lebih mudah [10].

API dapat dikembangkan menggunakan berbagai bahasa pemrograman seperti JavaScript, Go, dan Python. Menurut survei Stack Overflow 2023, JavaScript adalah bahasa pemrograman paling populer, dipilih oleh 63,61% dari 87.585 responden, diikuti oleh Python sebesar 49,28%, dan Go sebesar 13,24% [11]. Saat ini seiring dengan kebutuhan bisnis untuk bertukar data dalam jumlah besar maka diperlukan sebuah API yang andal dan efisien. Kedua hal tadi dapat dinilai melalui pengujian kinerja dan pengujian beban. Analisis komparatif sangat penting untuk memastikan bahasa pemrograman yang optimal dipilih untuk pengembangan API.

Ueda et. all mengevaluasi kinerja JavaScript, Go, dan Java menggunakan tool JMeter. Mereka mengukur throughput dan waktu eksekusi untuk menilai efisiensi dan kecepatan respons API yang dikembangkan dalam bahasa-bahasa tersebut. Studi tersebut menemukan bahwa Go, dengan penyetelan sederhana, mengungguli Node.js dan Java [12]. Sedangkan studi lain oleh Lei et. all menilai kinerja PHP, Python, dan Node.js menggunakan ApacheBench dan LoadRunner, mengevaluasi parameter seperti permintaan per detik, waktu antar permintaan, throughput, hits per detik, dan rata-rata waktu respons transaksi menunjukkan bahwa performa Node.js secara signifikan mengungguli bahasa-bahasa lain yang diuji [13].

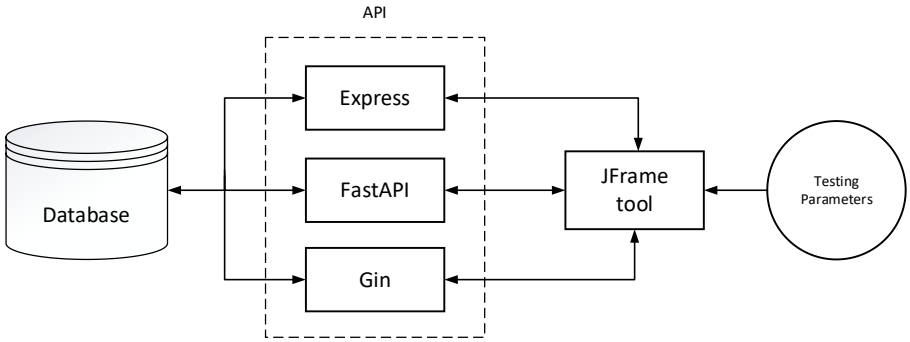
Menemukan parameter optimal dalam pengujian membutuhkan beberapa iterasi pengujian dengan JMeter adalah salah satu tools yang dapat digunakan. Penelitian ini akan menggunakan JMeter untuk menguji kinerja tiga API framework: Express.js, Gin, dan FastAPI untuk mendapatkan parameter pengujian yang optimal. Melaluinya dapat diketahui performa dari masing-masing API framework.

II. METODE DAN MATERI

Rancangan arsitektur testing adalah seperti pada Gambar 1.

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)



Gambar 1. Arsitektur system

Penjelasan singkat mengenai tahapan testing adalah sebagai berikut:

(1) Database PostgreSQL di-hosting menggunakan layanan virtual private server (VPS) dari Niaga Hoster dengan OS Ubuntu. Paket yang dipilih untuk VPS adalah KVM 2, Niaga Hoster menjanjikan 2vCPU core, memory sebanyak 8 GB (gigabyte), penyimpanan sebesar 100 GB dengan tipe NVMe serta 2 TB (terabyte) bandwidth. Tabel yang digunakan untuk pengujian atau eksperimen adalah table tt_sm dan tt_sm_detail yang masing-masing berisi 117.072 dan 133.923 data, karena data yang banyak maka saat pengambilan data SM akan di limit sebanyak 10, 50 dan 100 untuk pengujian nantinya.

(2) Pengembangan API dilakukan menggunakan code editor Visual Studio Code dengan spesifikasi komputer I5 10400F dan RAM 16 GB. selain itu ekstensi Visual Studio Code juga digunakan untuk memudahkan proses pengembangan API, ekstensi yang digunakan pada penelitian ini diantaranya adalah Go, Pylance, Python, dan ESLint.

(3) Tujuan utama dari load testing adalah mengetahui batasan-batasan sistem, seperti kapasitas maksimal, waktu respon dan apakah system dapat beroperasi secara konsisten dalam situasi beban yang tinggi. Pada eksperimen ini JMeter akan digunakan untuk menguji ketiga framework API, Dalam konteks ini, pengujian akan dilakukan dengan menemukan parameter optimal bagi JMeter. Proses dimulai dengan menentukan parameter awal untuk JMeter, yaitu num_threads, ramp_time. Selanjutnya, Python digunakan untuk mengubah test plan awal sesuai dengan parameter yang ditentukan. Modifikasi ini meliputi perubahan pada num_threads, dan ramp_time dalam file test plan JMeter. Setelah test plan dimodifikasi, load testing akan dilakukan menggunakan JMeter, di mana proses ini juga diotomasi dengan bantuan Python untuk mengelola eksekusi dan pengumpulan hasil testing.

III. PEMBAHASA DAN HASIL

Pada tahap pertama yang dilakukan adalah mengkoneksikan database dengan API yang dikembangkan menggunakan tiga frameworks: Express, FastAPI dan Gin. Masing-masing API dari ketiga frameworks akan memiliki tiga endpoint dengan tujuan pengujian yang dapat dilihat pada Tabel I.

Tabel I. *Endpoint functions*

Endpoint	Respon	Fungsi Pengujian
/fibonacci	Mengembalikan kode status dan hasil Fibonacci dari angka	Menguji performa perhitungan sederhana dari bahasa pemrograman yang dipilih dan kapabilitas rekursif
/login	Mengembalikan kode status dan data berupa token dan refresh token jwt	Menguji performa bahasa pemrograman untuk melakukan query ke database menggunakan clausa where dan mevalidasi hash password serta pembuatan token jwt
/sm	Mengembalikan kode status dan data dari transaksi dan detail transaksi yang	Menguji performa bahasa pemrograman dalam melakukan query ke database

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

terpaut dengan transaksi	dengan menggunakan clausa JOIN dan seberapa cepat bahasa pemrograman melakukan perombakan pada data yang telah didapat.

Pengujian logika endpoint dilakukan menggunakan aplikasi Postman. Tujuannya untuk memverifikasi bahwa logika yang diimplementasikan pada setiap endpoint dalam aplikasi web sesuai dengan ekspektasi. Proses pengujian melibatkan pengirimkan HTTP request ke endpoints yang telah dibuat dengan menggunakan kombinasi parameter dan payload data yang berbeda. Setiap request dievaluasi untuk memastikan bahwa respons adalah sesuai, baik untuk data kembalian maupun HTTP status code. Lebih lanjut, pengujian juga mencakup penanganan kasus-kasus eksepsi dan error yang mungkin terjadi selama eksekusi endpoint. Hasil dari pengujian ini digunakan untuk memvalidasi dan memastikan bahwa aplikasi berfungsi dengan baik serta memenuhi spesifikasi fungsional yang telah ditetapkan.

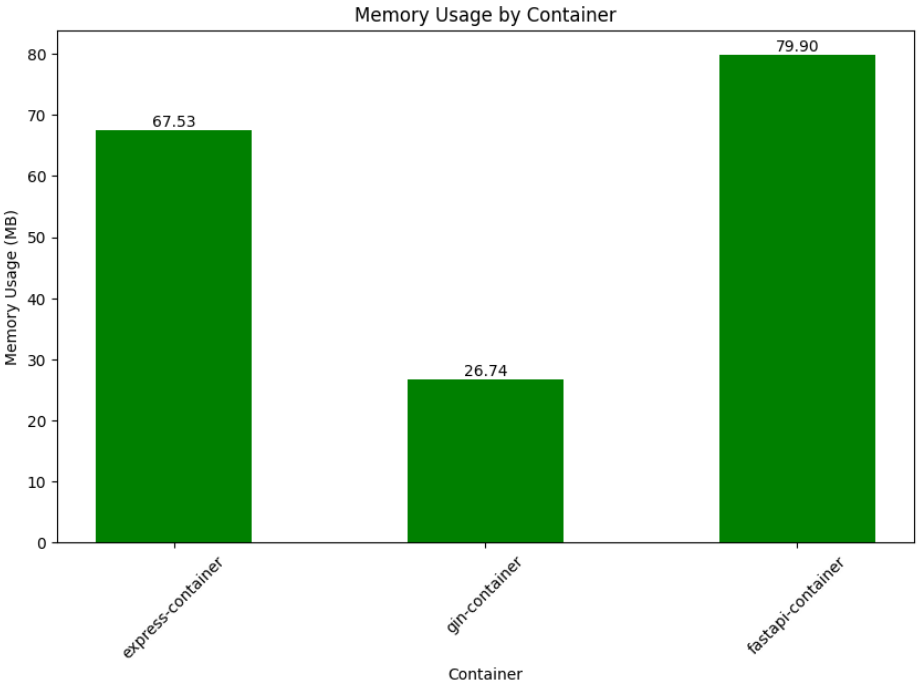
Selanjutnya, untuk penentuan parameter JMeter, diperlukan fungsi untuk memodifikasi file dengan format Java Management Extensions (JMX), fungsi tersebut bertujuan untuk mengubah nomor port dan mengubah angka num_threads dan ramp time yang akan dihasilkan lalu digunakan modul subprocess untuk menjalankan JMeter menggunakan Python.

Hasil pengujian beban API akan disajikan melalui tabel dan grafik dari masing-masing test plan. Ada total lima test plan yang akan dievaluasi, meliputi fibonacci(10), login, sm dengan batasan 10, sm dengan batasan 50, dan sm dengan batasan 100. Setiap test plan akan melalui tiga tahapan. Pada tahap pertama, num_threads akan dipilih secara acak antara 1 hingga 20 dengan ramp time antara 1 hingga 10. Pada tahap kedua, num_threads akan diacak antara 31 hingga 60 dengan ramp time antara 1 hingga 30. Sedangkan pada tahap terakhir, num_threads akan diacak antara 51 hingga 100 dengan ramp time antara 1 hingga 50. Tiap tahap akan terbatas pada 5 generasi dengan jumlah populasi yang terbatas yaitu 6 populasi. Pemilihan rentang angka untuk num_threads dan ramp time didasarkan pertimbangan untuk mencakup variasi yang cukup, namun tetap memperhatikan keterbatasan jumlah populasi dan generasi agar proses pencarian parameter dan pengujian beban tidak menjadi terlalu kompleks dan memakan waktu yang berlebihan.

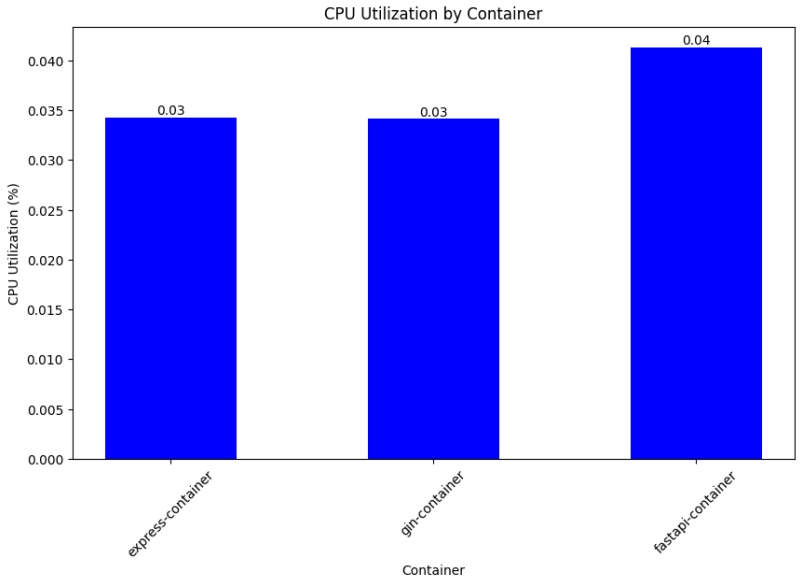
Selama pengujian berlangsung, penggunaan memori dan utilisasi CPU dapat diobservasi menggunakan docker, pengambilan data penggunaan memori dan utilisasi CPU akan menggunakan python, dimana tiap 20 detik python akan menulis data tersebut ke excel file, grafik dari penggunaan memori dan utilisasi CPU dapat dilihat pada gambar 2 dan gambar 3. Hasil throughput dari tiap framework dan test plan adalah seperti pada tabel II.

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)



Gambar 2. Penggunaan memori tiap API



Gambar 3. Utilisasi CPU tiap API

Tabel II. Hasil *throughput* dari setiap *framework* dan *test plan*

Scenario	*Num threads*	*Ramp time*	*Throughput*		
			Express.js	**Gin**	**FastAPI**

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

	18	1	**18.5**	**18.5**	18.4
Fibonacci 10	47	10	**4.8**	**4.8**	**4.8**
	88	1	89.7	**89.9**	89.8
	15	1	11.8	**13.7**	3.3
Login	59	13	**4.6**	**4.6**	3.4
	88	2	18.3	**40.7**	0.1
	17	1	6.5	**8.7**	2.9
SM Limit 10	52	12	4.2	**4.3**	3
	52	4	7.2	**10.9**	2.7
	17	2	5.8	**7.6**	2.9
SM Limit 50	56	14	**3.9**	**3.9**	2.9
	94	2	7.6	**13.5**	0.2
	20	3	5.7	**6.3**	2.3
SM Limit 100	54	17	**3.1**	**3.1**	2.4
	98	1	6.7	**11.4**	0.2

Hasil pengujian untuk endpoint fibonacci, pada tahap pertama, num threads optimal terukur sebesar 18 dengan ramp time 1. Hasilnya, sebesar 18.5, konsisten pada semua framework. Pada tahap kedua, num threads terbaik adalah 47 dengan ramp time 10, menghasilkan nilai 4.8 yang seragam di seluruh framework. Sedangkan pada tahap ketiga, num threads optimal adalah 88 dengan ramp time 1. Hasil tiap framework pada tahap ketiga sedikit berbeda, throughput tertinggi dicapai oleh framework gin (89.9), diikuti oleh fastAPI (89.8), sementara express memiliki throughput terendah (89.7).

Hasil pengujian untuk endpoint login, pada tahap pertama, num threads optimal terukur sebesar 15 dengan ramp time 1, hasil dari ketiga framework berbeda-beda gin menempati urutan pertama dengan hasil 13.7, diikuti oleh express dengan hasil 11.8 dan fastapi di urutan terakhir dengan hasil 3.3. Pada tahap kedua, num threads optimal terukur sebesar 15 dengan ramp time 13, hasil dari framework express dan gin sebanding yaitu dengan angka 4.6, sedangkan fastapi menghasilkan throughput sebesar 3.4. Hasil ketiga, num threads yang ditemukan paling optimal terukur sebesar 88 dengan ramp time 2, gin menempati urutan pertama pada tahap ketiga dengan throughput sebesar 40.7, diikuti dengan expess sebesar 18.3 dan fastapi dengan perolehan throughput sebesar 0.1.

Hasil pengujian untuk endpoint SM dengan limit 10, pada tahap pertama, num threads optimal terukur sebesar 17 dengan ramp time 1, pada tahap pertama ini gin menempati urutan pertama dengan throughput sebesar 8.7 diikuti oleh express sebesar 6.5 dan fastapi 2.9. Pada tahap kedua num threads optimal bernilai 52 dan ramp time 12, pada tahap kedua gin tetap menempati urutan pertama dengan perolehan hasil 4.3 diikuti oleh express sebesar 4.2 dan fastapi sebesar 3.0. Tahap terakhir ditemukan nilai optimal num threads sebesar 52 dengan ramp time sebesar 4, gin menempati urutan pertama dengan perolehan hasil throughput sebesar 10.9, diikuti oleh express pada angka 7.2 dan fastapi sebesar 2.7.

DOI: 10.52362/jisamar.v9i3.1986

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 9 No.3 (August 2025)

Hasil pengujian untuk endpoint SM dengan limit 50, pada tahap pertama menemukan num threads optimal sebesar 17 dan ramp time sebesar 2, gin menempati urutan pertama dengan throughput sebesar 7.6, diikuti oleh express sebesar 5.8 dan fastapi sebesar 2.9. Pada tahap kedua, ditemukan num threads optimal sebesar 56 dengan ramp time sebesar 14, gin dan express memiliki throughput sebesar 3.9 dan fastapi sebesar 2.9. Pada tahap ketiga, ditemukan masing-masing num threads dan ramp time sebesar 94 dan 2, gin menempati urutan pertama dengan perolehan hasil 13.5, diikuti oleh express dengan hasil 7.6 dan fastapi sebesar 0.2.

Hasil pengujian untuk endpoint SM dengan limit 100, pada tahap pertama, num threads optimal terukur sebesar 20 dengan ramp time 3, pada tahap pertama ini gin menempati urutan pertama dengan throughput sebesar 6.3 diikuti oleh express sebesar 5.7 dan fastapi 2.3. Pada tahap kedua num threads optimal bernilai 54 dan ramp time 17, pada tahap kedua gin dan express memiliki nilai yang sebanding dengan perolehan hasil 3.1 dan fastapi sebesar 2.4. Tahap terakhir ditemukan nilai optimal num threads sebesar 98 dengan ramp time sebesar 1, gin menempati urutan pertama dengan perolehan hasil throughput sebesar 11.4, diikuti oleh express pada angka 6.7 dan fastapi sebesar 0.2.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat dapat disimpulkan bahwa dari ketiga framework yang diuji, Express.js dan Gin memiliki nilai rata-rata utilisasi CPU yang sama yaitu 0.03%, sementara FastAPI memiliki rata-rata 0.04%. Dalam hal konsumsi memori, Gin menunjukkan efisiensi tertinggi dengan rata-rata 26.74 MB, diikuti Express.js (67.53 MB) dan FastAPI (79.90 MB). Pengujian beban pada setiap endpoint menunjukkan keunggulan Gin dengan ramp-time yang singkat, namun Express.js mampu menyaingi kinerjanya pada ramp-time yang lebih lama. Pemilihan parameter optimal adalah penting dalam pengujian JMeter untuk mencapai kinerja optimal sistem yang diuji, serta memberikan fleksibilitas dalam skalabilitas pengujian tanpa perubahan signifikan.

REFERENSI

[1] Lexie Pelchen, Internet Usage Statistics in 2024, 2024. https://www.forbes.com/home-improvement/internet/internet-statistics/

[2] Petrosyan, A., Worldwide Digital Population, 2024. https://www.statista.com/statistics/617136/digital-population-worldwide/

[3] BPS, Statistik Telekomunikasi Indonesia, vol. 13, no. 1. 2020.

[4] Rediana Koçi, Xavier Franch, Petar Jovanovic, Alberto Abelló. 2023. Web API evolution patterns: A usage-driven approach. The Journal of Systems & Software vol. 198 111609,

[5] Martin P Robillard, Eric Bodden, David Kawrykow, Mira Mezini, and Tristan Ratchford. 2013. Automated API Property Inference Techniques. IEEE Transactions on Software Engineering 39, 5 (may 2013), 613–637.

[6] Danny Dig. 2005. Using refactorings to automatically update component-based applications. In Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications- OOPSLA ’05. ACMPress, New York, New York, USA, 234.

[7] Daqing Hou and Xiaojia Yao. 2011. Exploring the Intent behind API Evolution: A Case Study. In 2011 18th Working Conference on Reverse Engineering. IEEE, USA, 131–140.

[8] S M Sohan,CraigAnslow, andFrank Maurer. 2015. SpyREST: Automated RESTful API Documentation Using an HTTP Proxy Server. In 30th IEEE/ACM International Conference on Automated Software Engineering. IEEE, USA, 271–276.

[9] Phuong T Nguyen, Juri Di Rocco, Davide Di Ruscio, Lina Ochoa, Thomas Degueule, and Massimiliano Di Penta. 2019. FOCUS: A Recommender System for Mining API Function Calls and Usage Patterns. In 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, USA, 1050–1060.

DOI: 10.52362/jisamar.v9i3.1986

[10] Maxime Lamothe, Yann-Gaël Guéhéneuc, and Weiyi Shang. 2021. A Systematic Review of API Evolution Literature. ACM Comput. Surv. 54, 8, Article 171 (November 2022), 36 pages. https://doi.org/10.1145/3470133

[11] Stackoverflow, "2020 Developer Survey," 2020. https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-all-respondents

[12] Y. Ueda and M. Ohara, "Performance competitiveness of a statically compiled language for server-side Web applications," ISPASS 2017 - IEEE Int. Symp. Perform. Anal. Syst. Softw., pp. 13–22, 2017, doi: 10.1109/ISPASS.2017.7975266.

[13] K. Lei, Y. Ma, and Z. Tan, "Performance comparison and evaluation of web development technologies in PHP, Python and Node.js," Proc. - 17th IEEE Int. Conf. Comput. Sci. Eng. CSE 2014, Jointly with 13th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2014, 13th Int. Symp. Pervasive Syst., pp. 661–668, 2015, doi: 10.1109/CSE.2014.142.

DOI: 10.52362/jisamar.v9i3.1986