

Evaluasi Kinerja Load Balancing Berbasis Nginx pada Lingkungan Virtualisasi

Indra Febri¹, Antoni Haikal²

¹Program Studi Rekayasa Keamanan Siber, Jurusan Teknik Informatika, Politeknik Negeri Batam

¹zakriindrafabri@gmail.com

Abstrak Lonjakan trafik pada layanan web modern, terutama pada aplikasi e-commerce dan streaming video, menuntut mekanisme distribusi beban yang andal agar server tidak mengalami overload. Nginx sebagai reverse proxy sekaligus load balancer menyediakan tiga algoritma populer untuk keperluan ini, yaitu round-robin, least connections, dan IP hash. Penelitian ini mengevaluasi kinerja ketiga algoritma tersebut pada lingkungan virtualisasi, dengan tiga skenario pengujian yaitu normal load, overload, dan failover, serta mengukur response time, throughput, utilisasi CPU/memori, dan waktu pemulihan sistem (recovery time). Metode yang digunakan adalah eksperimen kuantitatif dengan pengulangan tiga kali pada setiap kombinasi algoritma dan skenario, dianalisis secara deskriptif, komparatif, dan diuji menggunakan uji t dua sampel independen pada tingkat kepercayaan 95%. Hasil pengujian menunjukkan bahwa least connections secara konsisten unggul di seluruh metrik dengan response time rata-rata 141,0 ms (round-robin 179,0 ms; IP hash 167,7 ms), throughput 107,7 request/detik (round-robin 93,3; IP hash 98,0), utilisasi CPU 63% dan RAM 71% (paling rendah di antara ketiganya), serta recovery time tercepat yaitu 3,6 detik. Uji t mengonfirmasi bahwa perbedaan performa antara least connections dengan kedua algoritma lain signifikan secara statistik ($p = 0,008$ dan $p = 0,018$), sementara perbedaan antara round-robin dan IP hash tidak signifikan ($p = 0,103$). Temuan ini mengindikasikan bahwa mekanisme distribusi beban yang mempertimbangkan kondisi koneksi aktif pada tiap server backend lebih adaptif dibandingkan pendekatan statis, sehingga least connections dapat direkomendasikan sebagai konfigurasi optimal untuk load balancing Nginx pada lingkungan virtualisasi.

Kata Kunci: *load balancing, Nginx, least connections, virtualisasi*

Abstract The growing volume of web traffic on modern services, particularly e-commerce and video streaming platforms, calls for a reliable mechanism to distribute workload evenly across servers and prevent overload. Nginx, functioning as both a reverse proxy and load balancer, offers three widely used algorithms for this purpose: round-robin, least connections, and IP hash. This study evaluates the performance of these three algorithms within a virtualized environment across three test scenarios: normal load, overload, and failover, measuring response time, throughput, CPU/memory utilization, and system recovery time. A quantitative experimental method was used, with three repetitions for each algorithm-scenario combination, analyzed descriptively and comparatively, and tested using an independent two-sample t-test at a 95% confidence level. The results show that least connections consistently outperformed the other algorithms, with an average response time of 141.0 ms (versus 179.0 ms for round-robin and 167.7 ms for IP hash), a throughput of 107.7 requests/second (versus 93.3 and 98.0), the lowest resource usage at 63% CPU and 71% RAM, and the fastest recovery time of 3.6 seconds. The t-test confirmed that the performance differences between least connections and the other two algorithms were statistically significant ($p = 0.008$ and $p = 0.018$), while the difference between round-robin and IP hash was not ($p = 0.103$). These findings suggest that a load-balancing mechanism accounting for each backend server's active connection load adapts more effectively than a static distribution approach, making least connections the recommended configuration.

Keywords: *load balancing, Nginx, least connections, virtualization*

I. PENDAHULUAN

Perkembangan pesat aplikasi web dan layanan berbasis cloud menyebabkan lonjakan besar dalam volume lalu lintas data yang harus diolah oleh server. Perubahan signifikan akibat digitalisasi bisnis, pertumbuhan Internet of Things (IoT), dan penggunaan big data analytics semakin menambah beban ini, sementara server tunggal seringkali tidak mampu menanggung beban kerja yang berubah-ubah dan tinggi [1]. Load balancing menjadi solusi penting untuk membagi beban kerja secara merata ke beberapa server, sehingga menjamin sistem tetap tersedia, dapat diperluas, dan bekerja efisien. Nginx merupakan salah satu server web dan load balancer open-source terpopuler, dikenal lewat keunggulan reverse proxy dan algoritma load balancing yang efisien seperti round-robin, least connections, dan IP hash, yang dapat menangani hingga jutaan permintaan per detik [2].

Virtualisasi menjadi pondasi utama infrastruktur modern karena memungkinkan pengelolaan sumber daya komputasi secara fleksibel dan efisien dari sisi biaya, serta relevan dengan tren seperti hybrid cloud dan edge computing. Namun demikian, virtualisasi dapat membawa overhead tambahan, seperti peningkatan latensi antar-VM dan konsumsi sumber daya CPU atau memori, yang perlu diukur agar performa tetap maksimal. Penelitian sebelumnya mengidentifikasi bahwa ketidakseimbangan beban pada VM cloud dapat menyebabkan overload dan latensi tinggi, sehingga evaluasi load balancing Nginx diusulkan sebagai solusi praktis untuk mencegah masalah serupa melalui distribusi trafik adaptif dan monitoring real-time [3].

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengevaluasi kinerja tiga algoritma load balancing Nginx: round-robin, least connections, dan IP hash pada lingkungan virtualisasi melalui simulasi yang menggambarkan kondisi nyata seperti lonjakan trafik

pada aplikasi e-commerce atau layanan streaming video. Rumusan masalah penelitian meliputi: (1) bagaimana performa Nginx dalam menangani lonjakan trafik pada skenario normal load dan overload; (2) faktor apa yang menyebabkan penurunan performa akibat overhead virtualisasi; dan (3) seberapa efektif fault tolerance Nginx terhadap kegagalan server virtual serta berapa waktu pemulihan sistemnya.

II. METODE PENELITIAN

A. Jenis dan Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif untuk mengukur dan menganalisis data numerik terkait kinerja load balancing Nginx pada lingkungan virtualisasi. Desain penelitian berupa eksperimen simulasi, dengan pembagian fase persiapan infrastruktur, implementasi load balancing, pengujian performa, dan analisis hasil.

B. Variabel Penelitian

Variabel independen meliputi algoritma load balancing (round-robin, least connections, IP hash) dan skenario beban kerja (normal load, overload, failover). Variabel dependen meliputi response time, throughput, utilisasi CPU/memori, dan fault tolerance (waktu pemulihan sistem). Variabel kontrol meliputi spesifikasi hardware server fisik, versi software, serta infrastruktur jaringan dasar yang dijaga stabil.

C. Topologi dan Prosedur Pengujian

Topologi penelitian terdiri atas satu server fisik sebagai host virtualisasi, satu unit Nginx sebagai reverse proxy dan load balancer, minimal tiga unit backend server, klien simulasi trafik (Apache Bench/JMeter), serta jaringan virtual yang menghubungkan seluruh komponen secara internal. Pengujian dilakukan melalui tiga skenario: (1) normal load, dengan trafik 1.000–5.000 request per detik; (2) overload, dengan trafik ditingkatkan bertahap hingga 5.000–10.000 request per detik atau lebih; dan (3) failover, dengan mematikan salah satu server backend secara manual untuk mengukur waktu pemulihan sistem. Setiap skenario diukur selama 30–60 menit dan diulang sebanyak tiga kali untuk setiap algoritma guna memastikan keakuratan data.

D. Teknik Analisis Data

Data dianalisis secara deskriptif (rata-rata, standar deviasi, koefisien variasi) dan komparatif antar algoritma berdasarkan metrik response time, throughput, utilisasi sumber daya, dan recovery time. Signifikansi perbedaan performa antar algoritma diuji menggunakan uji t dua sampel independen pada tingkat kepercayaan 95%, dengan hipotesis nol berupa tidak terdapat perbedaan performa antar algoritma yang dibandingkan.

III. HASIL DAN PEMBAHASAN

A. Response Time

Pengujian response time menunjukkan bahwa least connections secara konsisten memberikan waktu respons terendah, sebagaimana disajikan pada Tabel 1.

TABEL 1. HASIL PENGUJIAN RESPONSE TIME

Algoritma	Uji 1	Uji 2	Uji 3	Rerata
Round Robin	182	176	179	179,0
Least Conn.	141	138	144	141,0
IP Hash	165	170	168	167,7

Least connections tercatat 38 ms lebih cepat dibandingkan round-robin dan 26,7 ms lebih cepat dibandingkan IP hash. Pada round-robin, request dibagi secara bergiliran tanpa mempertimbangkan kondisi beban masing-masing backend. Least connections lebih adaptif karena mengarahkan request ke server dengan koneksi aktif paling rendah, sedangkan IP hash sangat dipengaruhi oleh sebaran alamat IP klien. Koefisien variasi seluruh algoritma berada di bawah 3%, yang mengindikasikan performa stabil pada setiap pengulangan (Tabel 2).

TABEL 2. ANALISIS KESTABILAN RESPONSE TIME

Algoritma	Rerata	SD	CV (%)
Round Robin	179,0	3,00	1,68
Least Conn.	141,0	3,00	2,13
IP Hash	167,7	2,52	1,50

B. Throughput

Hasil pengujian throughput pada Tabel 3 menunjukkan least connections menghasilkan throughput tertinggi, unggul 14,4 req/s dibandingkan round-robin dan 9,7 req/s dibandingkan IP hash.

TABEL 3. HASIL PENGUJIAN THROUGHPUT (REQ/S)

Algoritma	Uji 1	Uji 2	Uji 3	Rerata
Round Robin	92	95	93	93,3
Least Conn.	106	109	108	107,7
IP Hash	98	97	99	98,0

Pembagian beban yang mempertimbangkan koneksi aktif secara real-time memungkinkan backend server dimanfaatkan lebih merata, sementara pembagian statis pada round-robin dapat menyebabkan sebagian server lebih cepat penuh, dan pemetaan tetap pada IP hash membuat distribusi trafik kurang merata pada kondisi tertentu.

C. Utilisasi CPU dan Memori

TABEL 4. UTILISASI CPU DAN MEMORI (%)

Algoritma	CPU	RAM
Round Robin	68	74
Least Conn.	63	71
IP Hash	66	73

Least connections menggunakan CPU 5% lebih rendah dibandingkan round-robin dan 3% lebih rendah dibandingkan IP hash, dengan pola serupa pada RAM. Efisiensi ini terjadi karena request dialokasikan ke server yang masih memiliki kapasitas koneksi lebih banyak, sehingga mencegah penumpukan beban pada satu backend tertentu.

D. Fault Tolerance

TABEL 5. HASIL PENGUJIAN FAULT TOLERANCE

Algoritma	Recovery Time (detik)
Round Robin	4,8
Least Connections	3,6
IP Hash	4,2

Least connections memulihkan layanan 1,2 detik lebih cepat dibandingkan round-robin dan 0,6 detik lebih cepat dibandingkan IP hash. Pada least connections, begitu satu backend gagal, sisa request otomatis diarahkan ke backend dengan koneksi paling sedikit tanpa jeda distribusi ulang. Round-robin tetap membagi request secara bergiliran dalam proporsi yang sama seperti sebelum kegagalan, sehingga penyesuaian ulang beban berjalan lebih lambat. Pada IP hash, kegagalan satu backend memicu proses pemetaan ulang (rehashing) yang tidak mempertimbangkan jumlah koneksi aktif pada backend tujuan.

E. Uji Statistik

TABEL 6. HASIL UJI T-TEST

Perbandingan	t	p	Ket.
RR vs LC	-7,35	0,008	Signifikan
RR vs IP Hash	-2,14	0,103	Tdk Sig.
LC vs IP Hash	-5,21	0,018	Signifikan

Hasil uji menunjukkan hipotesis nol ditolak pada perbandingan round-robin vs least connections ($p = 0,008$) dan least connections vs IP hash ($p = 0,018$), sedangkan perbandingan round-robin vs IP hash tidak menunjukkan perbedaan signifikan ($p = 0,103$). Hasil ini memperkuat bahwa least connections memberikan performa yang benar-benar lebih baik secara statistik dibandingkan dua algoritma lainnya.

F. Pembahasan

Secara keseluruhan, least connections merupakan algoritma dengan performa terbaik pada seluruh skenario pengujian, memberikan response time terendah, throughput tertinggi, utilisasi sumber daya paling efisien, dan recovery time tercepat, sebagaimana dirangkum pada Tabel 7.

TABEL 7. PERBANDINGAN ROUND ROBIN (BASELINE) DAN LEAST CONNECTIONS

Metrik	RR	LC	Peningkatan
Response Time (ms)	179,0	141,0	21,2%
Throughput (req/s)	93,3	107,7	15,4%
CPU (%)	68	63	7,4%
RAM (%)	74	71	4,1%
Recovery (s)	4,8	3,6	25,0%

Pada round-robin, request dibagi secara bergiliran tanpa memperhatikan kondisi server backend; metode ini cukup baik untuk beban yang merata tetapi kurang adaptif saat salah satu server lebih sibuk. IP hash lebih cocok untuk menjaga konsistensi sesi pengguna, namun distribusi bebannya tidak selalu paling efisien. Temuan ini mendukung hipotesis penelitian bahwa penerapan Nginx pada lingkungan virtualisasi mampu meningkatkan performa layanan web dan menjaga ketersediaan sistem,

dengan least connections sebagai konfigurasi paling optimal.

IV. KESIMPULAN

Berdasarkan hasil pengujian dan analisis terhadap kinerja load balancing berbasis Nginx pada lingkungan virtualisasi dengan tiga algoritma, round-robin, least connections, dan IP hash pada skenario normal load, overload, dan failover, dapat disimpulkan bahwa Nginx mampu menangani lonjakan trafik pada skenario yang menggambarkan kondisi e-commerce dan streaming video dengan cukup baik. Least connections menunjukkan performa terbaik pada seluruh metrik yang diuji: response time rata-rata 141,0 ms, throughput rata-rata 107,7 request/detik, utilisasi CPU 63% dan RAM 71%, serta recovery time tercepat 3,6 detik. Uji t-test dengan tingkat kepercayaan 95% mengonfirmasi bahwa perbedaan performa antara least connections dengan round-robin ($p = 0,008$) dan dengan IP hash ($p = 0,018$) signifikan secara statistik, sedangkan perbedaan antara round-robin dan IP hash tidak signifikan ($p = 0,103$). Berdasarkan temuan tersebut, least connections direkomendasikan sebagai konfigurasi optimal untuk load balancing Nginx pada lingkungan virtualisasi, khususnya untuk aplikasi dengan pola trafik yang tidak menentu seperti e-commerce dan layanan streaming video. Penelitian selanjutnya disarankan untuk membandingkan hasil ini pada platform virtualisasi lain, memperpanjang durasi pengujian, serta mengkaji parameter tuning Nginx dan aspek keamanan jaringan secara lebih mendalam.

DAFTAR PUSTAKA

- [1] J. Amankwah-Amoah, Z. Khan, G. Wood, and G. Knight, "COVID-19 and digitalization: The great acceleration," *Journal of Business Research*, vol. 136, pp. 602–611, 2021.
- [2] F. Apriliansyah, I. Fitri, and A. Iskandar, "Implementasi load balancing pada web server menggunakan Nginx," *Indonesian Journal of Computer Science*, 2020.
- [3] M. A. Shahid, M. M. Alam, and M. M. Su'ud, "Performance evaluation of load-balancing algorithms with different service broker policies for cloud computing," *Applied Sciences*, vol. 13, no. 3, 2023.
- [4] B. Arifwido, V. Metayasha, and S. Ikhwan, "Analisis kinerja load balancing pada server web menggunakan algoritma Weighted Round Robin pada Proxmox VE," *Jurnal Telekomunikasi dan Komputer*, 2021.
- [5] Z. Bustomi, M. Syahiruddin, M. I. Afandi, and K. F. H. Holle, "Load balancing web server menggunakan Nginx pada lingkungan virtual," *Jurnal Penelitian Informatika dan Telematika (JPIT)*, vol. 2, no. 1, p. 32, 2019.
- [6] R. Baeturohman and B. Santoso, "Analisis kinerja algoritme WRR dengan WLC pada load balancing Nginx," *Indonesian Journal of Computer Science*, 2024.
- [7] T. W. Harjanti, H. Setiyani, and J. Trianto, "Load balancing analysis using round-robin and least-connection algorithms for server service response time," *Applied Technology and Computing Science Journal*, vol. 5, no. 2, pp. 40–49, 2022.
- [8] C. Ma and Y. Chi, "Evaluation test and improvement of load balancing algorithms of Nginx," *IEEE Access*, vol. 10, pp. 14311–14324, 2022.
- [9] A. A. J. Sinlae, M. Bagir, and M. H. Prayitno, "Analisis perbandingan algoritma round-robin dengan least-connection terhadap peningkatan nilai throughput pada layanan web server," *JURIKOM*, vol. 9, no. 5, 2022.

- [10] A. F. Zahir, H. Wijaya, M. Sanwasih, and A. Arisantoso, "Analisis efektivitas metode round-robin dan least-connection dalam load balancing terhadap throughput server web," *JIMA-ILKOM*, vol. 4, no. 1, pp. 24–33, 2025.
- [11] H. A. Mustaqhim, P. P. Anwas, and L. N. L. Badratul, "Performance comparison of NGINX, Apache, and Lighttpd using WRK on a Debian," *Bit-Tech*, vol. 8, no. 1, pp. 670–680, 2025.
- [12] A. A. Nizar, S. A. Karimah, and E. M. Jadied, "Analysis of virtualization performance on resource efficiency using containers and unikernel," in *Proc. ICoABCD*, 2024, pp. 125–130.
- [13] M. H. Prayitno and J. Trianto, "Analisis performa load balancing terhadap throughput pada kluster server untuk mendukung smart city," *Jurnal Teknoinfo*, vol. 18, no. 1, pp. 173–181, 2024.
- [14] D. Silva, J. Rafael, and A. Fonte, "Toward optimal virtualization: An updated comparative analysis of Docker and LXD container technologies," *Computers*, vol. 13, no. 4, p. 94, 2024.
- [15] C. Zaelani and G. M. Suranegara, "Performance evaluation of multi-cloud failover using domain name system," *Journal of Applied Informatics and Computing*, vol. 10, no. 1, 2026.