

# PERBANDINGAN DAN IMPLEMENTASI LOAD BALANCER PADA WEB SERVER MENGGUNAKAN NGINX PADA GG SUSPENSION

Irlon.

Program Studi Teknik Informatika, FTI, Institut Teknologi Budi Utomo Jakarta,  
[dahil.irlon@gmail.com](mailto:dahil.irlon@gmail.com)

## Abstrak

Penelitian ini membahas penerapan load balancer pada server web menggunakan Nginx, dengan fokus pada perbandingan tiga algoritma: Round Robin, IP Hash, dan Least Connection. Di era digital yang ditandai dengan peningkatan penggunaan internet, server web sering mengalami kelebihan beban, yang mengakibatkan penurunan kinerja dan waktu henti. Load balancer bertindak sebagai perantara untuk mendistribusikan permintaan pengguna secara efisien, sehingga meningkatkan stabilitas dan kecepatan respons. Metodologi penelitian yang digunakan adalah kuantitatif, dengan pengukuran berdasarkan metrik seperti waktu respons, throughput, dan pemanfaatan CPU. Hasil penelitian menunjukkan bahwa setiap algoritma memiliki kelebihan dan kekurangan yang berbeda. Dari analisis yang dilakukan, algoritma Least Connection terbukti menjadi pilihan terbaik dalam situasi dengan lalu lintas yang berfluktuasi karena kemampuannya untuk mengoptimalkan penggunaan sumber daya dan menjaga kinerja server di bawah beban tinggi. Pemilihan algoritma yang tepat sangat penting untuk memastikan kinerja server web yang optimal dalam menangani lalu lintas yang tinggi.

Kata kunci: Load Balancer, Nginx, Round Robin, IP Hash, Least Connection, kinerja server, distribusi beban.

## 1. PENDAHULUAN

Kemajuan teknologi digital mendorong banyak organisasi untuk menggunakan layanan berbasis web dalam berbagai aktivitas, seperti komunikasi dan transaksi keuangan. Namun, pertumbuhan ini memunculkan tantangan baru, terutama dalam menangani beban lalu lintas yang tinggi dan fluktuatif pada server. Ketika jumlah pengguna meningkat, server sering kali kewalahan, menyebabkan waktu respon melambat hingga terjadi downtime. Untuk mengatasi masalah ini, diperlukan teknologi manajemen beban yang mampu mendistribusikan lalu lintas secara efisien ke beberapa server, sehingga kualitas layanan tetap terjaga [1].

Salah satu solusi yang banyak digunakan adalah load balancing. Load balancer berfungsi sebagai perantara antara klien dan server mendistribusikan permintaan secara merata untuk menegakkan kelebihan secara merata untuk mencegah kelebihan beban pada server tertentu. Nginx menjadi salah satu perangkat lunak populer yang menawarkan fungsi load balancing, dengan dukungan algoritma seperti Round Robin, IP Hash, dan Least Connection. Setiap

algoritma memiliki cara kerja yang berbeda dalam mendistribusikan beban, yang membuatnya penting untuk diuji agar sesuai dengan kebutuhan sistem [2]. mengarahkan permintaan berdasarkan alamat IP klien, memastikan koneksi konsisten. Least Connection memilih server dengan koneksi aktif paling sedikit, ideal untuk beban yang tidak merata. Meskipun ketiga algoritma dirancang untuk meningkatkan efisiensi, efektivitasnya bergantung pada jenis lalu lintas yang dihadapi, sehingga pemilihan algoritma yang tepat menjadi hal krusial [3].

Penelitian ini bertujuan mengimplementasikan load balancer pada web server menggunakan Nginx dan membandingkan performa algoritma Round Robin, IP Hash, dan Least Connection. Pengujian dilakukan berdasarkan metrik seperti waktu respon, throughput, dan pemanfaatan CPU. Dengan penelitian ini, diharapkan diperoleh gambaran menyeluruh mengenai kelebihan dan kekurangan setiap algoritma, serta rekomendasi konfigurasi terbaik untuk menghadapi lalu lintas tinggi dan dinamis pada web server [4].

## 2. METODOLOGI

Pada metode pengumpulan data berdasarkan banyak atau kuantitatif digunakan untuk mengumpulkan data berdasarkan pengamatan langsung yang dapat diukur secara numerik. Penelitian

ini dapat mengumpulkan data melalui penelitian dan analisis statistik metrik pada performa dan kecepatan sebuah web aplikasi atau website pada saat di akses, tingkat kecepatan, kesetabilan dan jumlah kesalahan atau gagal response yang ditemukan. Kemudian menganalisis data yang dikumpulkan secara statistik untuk mengidentifikasi perbedaan yang signifikan antara menggunakan metode round robin, least connections dan ip hash.



Gambar 1. Diagram Metode Penelitian.  
Sumber: Olehan Data Mandiri

### 1. Studi Literatur

Penelitian diawali dengan studi literatur tentang konsep load balancing, keunggulan Nginx sebagai load balancer, serta praktik implementasinya pada web server. Literatur yang digunakan mencakup jurnal ilmiah, buku, dan dokumentasi resmi untuk memahami algoritma Round Robin, IP Hash, dan Least Connection. Tahap ini bertujuan membangun landasan teori yang mendukung perancangan pengujian dan analisis hasil.

### 2. Perancangan Skenario Pengujian

Peneliti merancang skenario pengujian dengan mengatur konfigurasi web server dan Nginx sebagai load balancer menggunakan algoritma Round Robin, IP Hash, dan Least

Connection. Parameter kinerja yang diukur meliputi waktu respons, throughput, penggunaan CPU, dan tingkat kesalahan. Perancangan ini memastikan skenario pengujian mencakup aspek penting untuk membandingkan performa web server.

### 3. Implementasi Lingkungan Pengujian

Lingkungan pengujian mencakup server aplikasi web, Nginx sebagai load balancer, dan Apache JMeter sebagai alat uji kinerja. Setiap komponen dikonfigurasi secara optimal untuk memastikan integrasi berjalan lancar dan menghasilkan data yang akurat. Lingkungan ini dirancang menyerupai lalu lintas server nyata guna memperoleh hasil relevan.

### 4. Pengujian Kinerja

Pengujian dilakukan dengan JMeter untuk mengirimkan beban kerja realistis pada web server, baik dengan maupun tanpa Nginx sebagai load balancer. Setiap algoritma diuji (Round Robin, IP Hash, dan Least Connection), dengan pengukuran meliputi waktu respons, throughput, penggunaan CPU, latensi, dan tingkat kesalahan. Data dihasilkan untuk membandingkan performa secara akurat.

### 5. Analisis dan Evaluasi

Data pengujian dianalisis untuk mengevaluasi efektivitas algoritma Round Robin, IP Hash, dan Least Connection dalam meningkatkan kinerja web server. Perbedaan performa tiap algoritma dianalisis berdasarkan parameter kinerja, sehingga dapat ditentukan manfaat implementasi Nginx terhadap web server.

### 6. Penarikan Kesimpulan

Hasil analisis dirangkum untuk menyimpulkan dampak implementasi Nginx dan efektivitas algoritma yang diuji. Kesimpulan mencakup rekomendasi algoritma terbaik, implikasi praktis, dan saran untuk penelitian lanjutan. Penelitian ini diharapkan memberikan wawasan praktis terkait solusi load balancing berbasis Nginx

## 3. HASIL DAN PEMBAHASAN

Pada bagian ini Peneliti menjelaskan bagaimana penerapan pada tiga metode yang dilakukan pada server yang digunakan website ggsuspension.com.

### 3.1 Metode Round Robin

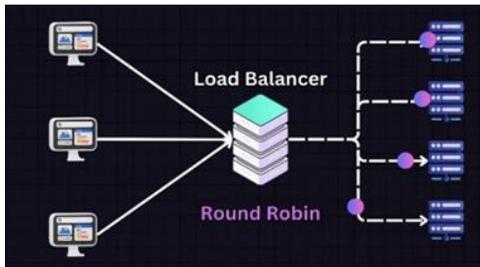
Konfigurasi pada round robin tidak menambahkan keyword pada upstream reserve proxy yang di setting seperti pada Gambar 3.1

```
upstream backend {
    # no load balancing method is specified for Round Robin
    server backend1.example.com;
    server backend2.example.com;
}
```

Gambar 3.1 Konfigurasi round robin  
Sumber: Olehan Data Mandiri

### Prosedur Request client

- Setiap request akan di kirimkan ke satu server load balancer pada server virtual.
- Traffict request akan di distribusikan ke semua server secara merata seperti pada Gambar 3.2



Gambar 3.2 Perancangan Skenario Pengujian Load Balancer Round Robin  
Sumber: Olehan Data Mandiri

Setiap request baik melalui laptop dan komputer (PC) maupun smartphone akan di tangani oleh server load balancer dan akan di distribusikan ke semua server yang sudah di daftarkan di load balancer secara merata

### 3.2 Metode Least Connections

Konfigurasi pada least connections menambahkan keyword `least_conn` pada upstream reserve proxy yang di setting seperti pada Gambar 3.3

```
upstream backend {
    least_conn;
    server backend1.example.com;
    server backend2.example.com;
}
```

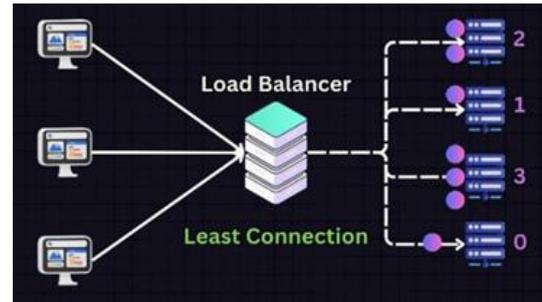
Gambar 3.3 Konfigurasi Least Connections  
Sumber: Olehan Data Mandiri

### Prosedur Request client

- Setiap request akan di kirimkan ke d

webservice load balancer dan akan didistribusikan ke beberapa server yang sudah di konfigurasi.

- Traffict request di hendling oleh server load balancer dan akan mendistribusikan request ke server yang koneksi aktifnya paling sedikit atau server yang mendapat request paling sedikit



Gambar 3.4 Perancangan Skenario Pengujian Load Balancer Least Connection  
Sumber: Olehan Data Mandiri

Setiap request baik melalui laptop maupun smartphone akan di hendling satu server load balancer dan akan di distribusikan ke server yang request nya paling sedikit.

### 3.3 Metode IP Hash

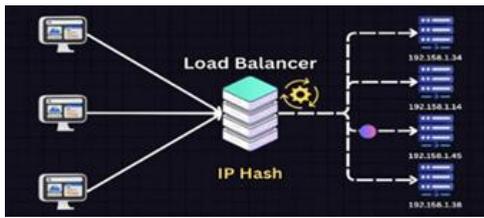
Konfigurasi pada IP Hash menambahkan keyword `ip_hash` pada upstream reserve proxy yang di setting seperti pada Gambar 3.5.

```
upstream backend {
    ip_hash;
    server backend1.example.com;
    server backend2.example.com;
}
```

Gambar 3.5 Konfigurasi IP Hash  
Sumber: Olehan Data Mandiri

### Prosedur Request client

- Alamat IP client request akan di ambil oleh nginx
- Traffict request dari alamat IP client yang sama akan selalu didistribusikan ke server yang sama atau server yang pernah menangani request dari IP client tersebut

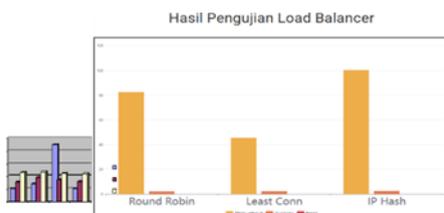


Gambar 3. 6 Perancangan Skenario Pengujian Load Balancer IP Hash  
Sumber: Olehan Data Mandiri

Nginx akan mengambil alamat IP dari permintaan yang datang NGINX kemudian akan menghitung hash dari alamat IP tersebut. Hash ini akan menghasilkan nilai numerik .Nilai hash ini dibagi dengan jumlah server yang tersedia untuk menentukan server mana yang akan menangani permintaan tersebut. Dengan cara ini, permintaan dari alamat IP yang sama akan selalu diarahkan ke server yang sama

### 3.4 Hasil Perbandingan

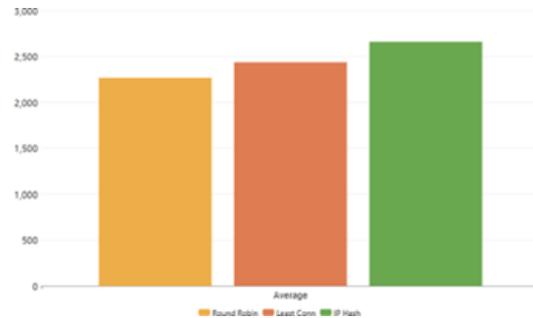
Pada pembahasan ini Peneliti akan bandingkan dalam bentuk grafik perbedaan dari penggunaan tiga metode. Namun sebelumnya, akan menghitung throughput, average, error rate dari aspek waktu dari ketiga metode ini, dalam melakukan perhitungan menggunakan grafik berikut ini :



Gambar 3.7 Grafik Throughput  
Sumber: Olehan Data Mandiri

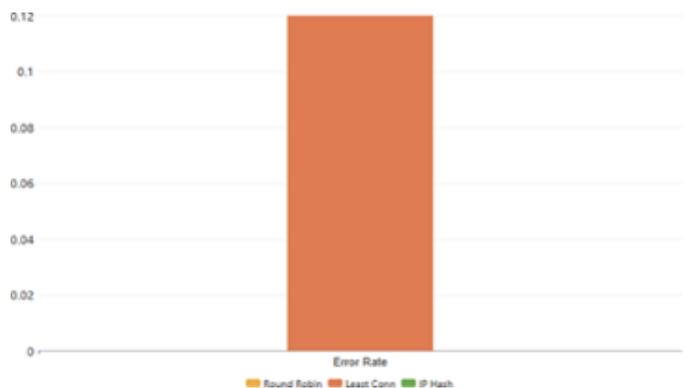
Pada grafik di atas, terlihat hasil pengujian performa tiga algoritma load balancer: Round Robin, Least Connections, dan IP Hash. Algoritma IP Hash menunjukkan performa terbaik dengan throughput tertinggi, memberikan stabilitas dan efisiensi dalam distribusi beban. Round Robin, meskipun sederhana dan mudah diterapkan, menunjukkan kinerja yang kurang optimal dalam situasi dengan beban tidak merata, sedangkan Least Connections berfungsi baik dalam menyeimbangkan beban, tetapi memiliki throughput yang lebih rendah dibandingkan IP Hash. Grafik ini

menggambarkan perbandingan yang jelas antara ketiga algoritma, menekankan pentingnya pemilihan algoritma yang tepat berdasarkan kebutuhan spesifik pada server.



Gambar3.8 Grafik Average  
Sumber: Olehan Data Mandiri

Grafik di atas menunjukkan rata-rata waktu respon dari tiga algoritma load balancer: Round Robin, Least Connections, dan IP Hash. Dari hasil pengujian, terlihat bahwa IP Hash memiliki waktu respon terendah, menunjukkan efisiensi yang lebih baik dalam menangani permintaan pengguna. Round Robin dan Least Connections memiliki waktu respon yang sedikit lebih tinggi, tetapi keduanya masih dalam rentang yang dapat diterima. Perbedaan waktu respon ini menggambarkan efektivitas masing-masing algoritma dalam mendistribusikan beban, dengan IP Hash menonjol sebagai pilihan yang lebih optimal untuk menjaga kecepatan dan kinerja server di bawah beban yang fluktuatif



Gambar 3.9 Grafik Error  
Sumber: Olehan Data Mandiri

Grafik di atas menunjukkan tingkat kesalahan (error rate) dari tiga algoritma load balancer: Round Robin, Least Connections, dan IP Hash. Terlihat bahwa algoritma Least Connections memiliki tingkat kesalahan yang lebih tinggi dibandingkan dengan yang lainnya, mencapai sekitar 0.12, sementara algoritma Round Robin dan IP Hash menunjukkan

hasil yang lebih rendah. Tingginya tingkat kesalahan pada Least Connections mengindikasikan bahwa algoritma ini mungkin kurang efektif dalam menangani lonjakan beban atau tidak mampu mendistribusikan permintaan dengan baik dalam situasi tertentu. Hal ini menunjukkan perlunya evaluasi lebih lanjut dan mungkin optimasi untuk algoritma ini agar dapat meningkatkan kinerja dan keandalan sistem saat menghadapi tekanan yang lebih besar.

*Tabel 3.1 Hasil Perbandingan*

Aspek	Round Robin	Least Connections	Ip Hash
Throughput	85/detik	40/detik	100/detik
Average	2,43 detik	2,43 detik	2,65 detik
Error	0,0%	0,12%	0,0%

Hasil perbandingan pada Tabel 3.1 bahwa menggunakan load balancer dengan metode IP Hash mampu menangani hingga 100 request perdetik dan 40 request perdetik pada metode least connections dan 85 request perdetik pada metode round robin, pada webserver dengan nginx dengan load balancer IP Hash lebih cepat dan setabil. Dari hasil penelitian dan analisa ini maka webserver pada GG Suspension akan menggunakan load balancer dengan metode IP Hash

### KESIMPULAN

Berdasarkan hasil pengujian dan analisis performa algoritma distribusi beban pada web server berbasis Nginx, dapat disimpulkan sebagai berikut:

- a. Hasil pengujian bervariasi tergantung jaringan internet di sisi klien dan server.
- b. Performa algoritma Round Robin, IP Hash, dan Least Connections menunjukkan variasi dalam efisiensi distribusi beban. Round Robin mendistribusikan beban secara merata dengan rata-rata waktu respon 2,43 detik dan throughput 85 permintaan per detik, namun kurang optimal saat beban tidak merata. Least Connections juga memiliki waktu respon 2,43 detik, tetapi throughput lebih rendah (40 permintaan per detik). Kelebihannya adalah menyeimbangkan beban pada server dengan kapasitas

berbeda. IP Hash, dengan waktu respon 2,65 detik dan throughput 100 permintaan per detik, lebih baik dalam menjaga konsistensi koneksi untuk aplikasi yang memerlukan sesi berkelanjutan.

- c. Hasil pengujian menunjukkan bahwa algoritma IP Hash memiliki performa paling optimal dalam menjaga stabilitas waktu respon, throughput, dan pemanfaatan CPU pada server dengan lalu lintas fluktuatif. Dengan throughput tertinggi (100 permintaan per detik) dan tingkat kesalahan 0%, algoritma ini efisien untuk aplikasi yang memerlukan sesi konsisten. Round Robin kurang optimal saat beban tidak merata, sementara Least Connections lebih baik untuk skenario keseimbangan koneksi, meskipun throughputnya lebih rendah dibandingkan IP Hash.

### DAFTAR PUSTAKA

- [1] Prakoso, R. D., & Asmunin. (2018). Implementasi dan perbandingan performa proxmox dalam virtualisasi dengan tiga virtual server ( Studi Kasus: Information Technology of UNESA ). *Jurnal Manajemen Informatika*, 8, 79–85.
- [2] Herdian, R., Munadi, R., & Riza, T. A. (2015). Implementation and Performance Analysis of Load Balancing on Virtual Servers Using Zen Load Balancer. *E-Proceeding of Engineering*, 2(1), 202–208.
- [3] safira Amara P "Pengertian Load Balancing dan Manfaatnya untuk Server" December 23,2020  
<https://www.goldenfast.net/blog/pengertian-loadbalancing/> (Di akses tanggal 11 Desember 2021)
- [4] Aziz, A., & Tampati, T. (2015). Analisis Web Server untuk Pengembangan Hosting Server Institusi: Pembandingan Kinerja Web Server Apache dengan Nginx. *Multinetics*, 1(2), 12.