Indonesian Journal of Computer Science
ISSN 2302-4364 (*print*) dan 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam, No. 1, Padang, Indonesia, Telp. (0751) 7056199, 7058325
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Analisis Kinerja Algoritme WRR dengan WLC pada Load Balancing Nginx

Rama Baeturohman[1], Banu Santoso[2*]
[1]ramabaeturohman61@students.amikom.ac.id, banu@amikom.ac.id[2*]
[1,2]Fakultas Ilmu Komputer, Program Studi Teknik Komputer, Universitas Amikom Yogyakarta

Informasi Artikel

Diterima : 7 Dec 2023
Direview : 22 Dec 2023
Disetujui : 20 Feb 2024

Kata Kunci

Load balancing, Nginx, Weighted Round Robin (WRR), Weighted Least Connection (WLC)

Abstrak

Semakin meningkatnya pengunjung website mengakibatkan web server menjadi sibuk merespons permintaan klien. Untuk mengatasi masalah kelebihan beban, penggunaan load balancing menjadi solusi yang efektif. Nginx merupakan aplikasi web server memiliki keunggulan sebagai perangkat lunak sumber terbuka, ringan, dan berkinerja tinggi sebagai HTTP dan reverse proxy. Dalam konteks ini, algoritma penjadwalan menjadi penting untuk menyeimbangkan beban Algoritma penjadwalan seperti Weighted Round Robin (WRR) dan Weighted Least Connection (WLC) menjadi pilihan yang relevan. Pengujian dilakukan dengan mengirimkan 1000, 2000, dan 5000 permintaan dengan bobot server 10:15:20, analisis throughput menunjukkan bahwa algoritma WLC lebih unggul daripada algoritma WRR. WLC mencapai waktu throughput terbaik dengan nilai 108.988 Kbits/sec. Parameter response time menunjukkan bahwa algoritma WLC memiliki kinerja lebih baik, dengan waktu rata-rata response sebesar 2.463ms. Untuk parameter packet loss, algoritma WRR juga menunjukkan hasil yang lebih baik, dengan rata-rata nilai packet loss sebesar 55%.

Keywords

Load balancing, Nginx, Weighted Round Robin (WRR), Weighted Least Connection (WLC)

Abstrak

The increasing number of website visitors causes the webserver to become busy responding to client requests. Using load balancing is an effective solution to overcome the problem of overload. Nginx is a web server application with the advantage of being open-source, lightweight, high-performance software as an HTTP and reverse proxy. In this context, scheduling algorithms become essential to balance the load. Scheduling algorithms such as Weighted Round Robin (WRR) and Weighted Least Connection (WLC) become relevant choices. Testing was carried out by sending 1000, 2000, and 5000 requests with a server weight of 10:15:20, and throughput analysis showed that the WLC algorithm was superior to the WRR algorithm. WLC achieved the best throughput time with a 108,988 Kbits/sec value. The response time parameter shows that the WLC algorithm performs better, with an average response time of 2.463ms. The WRR algorithm also delivers better results for packet loss parameters, with an average packet loss value of 55%.

Attribution-ShareAlike 4.0 International License

A. Pendahuluan

Di era teknologi digital yang semakin berkembang, untuk mengetahui segala informasi dengan sangat mudah dan menyelesaikan tugas dalam waktu singkat. Satu teknologi yang penggunaanya sangat berpengaruh untuk berbagai kalangan masyarakat adalah website. Layanan web telah memasuki semua aspek dalam kehidupan sehari-hari, seperti belanja online, jejaring social, keuangan, pendidikan dan institusi pemerintahan. Teknologi web dirancang untuk dapat diakses oleh banyak orang dan dalam hal ini kemacetan pada jaringan dan kelebihan beban pada web server sering terjadi dan tidak dapat dihilangkan[1][2].

Web server dalam hal ini mempunyai peran yang penting dalam menangani permintaan pengguna layanan web dalam jumlah yang banyak, permintaan yang diterima oleh server web,diolah, dan dijawab. Selain efektivitas dan efisiensi untuk organisasi dan pembuat web dalam menentukan daya tarik server web, seperti respons cepat dan akurat terhadap pelanggan, oleh karena itu merancang sistem yang efisien dapat menangani kelebihan beban pada server[3].

Selain itu meningkatnya permintaan pada suatu website, maka web server menjadi sibuk merespon permintaan dari klien, dan jika permintaan klien melebihi kapasitas maka kemungkinan besar web server tidak dapat melayani permintaan dari klien, kapasitas server adalah komponen yang sangat penting untuk dipertimbangkan, karena banyaknya pengguna yang mengakses layanan server, dapat menyebabkan kelebihan beban dan kelebihan kapasitas yang dapat menyebabkan kegagalan pada server[4][5].

Load balancing menjadi salah satu solusi dalam menangani masalah kelebihan beban yang mengakibatkan kegagalan pada server, load balancing dapat membagi beban dengan tepat pada setiap node cluster dengan demikian tidak mengalami kegagalan pada server. Mendistribusikan ke beberapa server sesuai kapasitasnya, memastikan tidak ada satupun yang kelebihan beban adalah tugas dari load balancing[6][7]. Web server Nginx adalah salah satu software web server yang memiliki kelebihan yaitu open source, ringan dan berkinerja tinggi sebagai HTTP dan reverse proxy. Dengan menggunakan sumber daya sistem yang efisien, Nginx dapat menyediakan konten statis dengan cepat dan dapat menyebarkan konten HTTP secara dinamis di jaringan[8][9]. Selain itu dengan mengubah profilnya, Nginx dapat mengirimkan permintaan untuk menghubungkan server yang paling mampu pada node cluster yang digunakan[10].

Dalam teknik load balancing Nginx, algoritma penjadwalan diperlukan untuk menyeimbangkan beban, sehingga algoritma penjadwalan yang digunakan pada penyeimbang beban dapat mengirimkan permintaan pada server[11]. Weighted round robin (WRR) adalah salah satu algoritma penjadwalan yang mempertimbangkan kapasitas server untuk menentukan beban. Algoritma ini secara manual memasukkan parameter atau bobot ke masing-masing node cluster berdasarkan sumberdaya[12]. Sedangkan Weighted Least Connection (WLC), mengalokasikan bobot kinerja masing-masing server untuk membagi beban trafik. Server dengan kapasitas terkecil akan digunakan untuk koneksi baru [13].

Berdasarkan penelitian sebelumnya yang dilakukan oleh M.Rizqi dan I Dwi Nuryana (2022), mebahas perbandingan algoritma Weighted Round Robin dan Weighted Least Connection pada load balancing Nginx. Perbandingan penelitian yang akan dilakukan menggunakan load balancing Nginx dengan bobot masing-

masing dari server 10:15:20 dengan menggunakan 3 server web dengan membandingkan 2 metode algoritma penjadwalan, sedangkan penelitian ini menggunakan 3 server web dengan bobot 2:3:5 menggunakan 2 metode algoritma penjadwalan. Kesimpulan yang diperoleh dari penelitian sebelumnya bahwa algoritma WLC telah berhasil membagi beban dengan baik secara merata dan memiliki keunggulan pada pengujian throughput, response time, maupun error request dengan nilai rata-rata hasil pengujian throughput 150.9 thread/sec, response time 31.77 sec, dan error request 149.85 %, sedangkan pada WRR memiliki nilai rata-rata throughput 38.31 thread/sec, response time 261.98 sec dan error request 39.66%. Dari hasil dari penelitian sebelumnya menunjukan WLC lebih unggul dibandingkan algoritma WRR[14].

Penelitian sebelumnya yang dilakukan oleh S. Panjaitan (2019), membahas perbandingan performansi algoritma Weighted Round Robin dan Weighted Least Connection. Perbandingan yang dilakukan menggunakan load balancing Nginx dengan bobot masing-masing dari server 10:15:20 dengan menggunakan 3 server web dengan membandingkan 2 metode algoritma penjadwalan, sedangkan penelitian ini menggunakan 3 server web dengan menggunakan bobot 1:4:6 menggunakan 2 metode algoritma penjadwalan. Kesimpulan dari penelitian sebelumnya pengujian algoritma load balancing WRR dan WLC telah diimplementasikan dengan hasil pengujian response time algoritma WRR lebih baik dibandingkan dengan algoritma WLC, dengan memiliki rata-rata yang dimiliki WRR 53.2 msec. dan pengujian throughput WLC lebih baik dengan nilai sebesar 57.1 Mbits/sec[13].

Penelitian yang dilakukan oleh B. arif widodo (2021), membahas tentang kinerja load balancing algoritma Weighted Round Robin. Perbandingan yang dilakukan menggunakan load balancing Nginx dengan bobot masing-masing dari server 10:15:20 dengan menggunakan 3 server web dengan membandingkan 2 metode algoritma penjadwalan, sedangkan penelitian ini menggunakan 2 server web dengan menggunakan bobot 2:1, 3:1, 4:1, 5:1 menggunakan 1 metode algoritma penjadwalan. Kesimpulan dari penelitian sebelumnya implementasi algoritma penjadwalan Weighted Round Robin menunjukan bahwa server dengan beban 2 banding 1 lebih baik dalam membagi beban kerja dengan kriteria server yang berbeda. Nilai throughput tertinggi 6.717 Mbit/s dan delay 0,505 ms dan packet loss 0%[11].

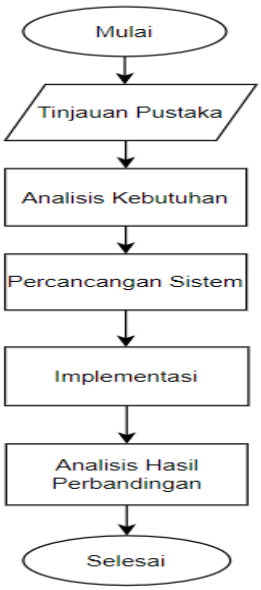
Penelitian yang dilakukan oleh M.Pradana (2020), membahas tentang performa load balancing algoritma Weighted Round Robin pada BPBD provinsi jawa timur. Perbandingan yang dilakukan menggunakan load balancing Nginx dengan bobot masing-masing dari server 10:15:20 dengan menggunakan 3 server web dengan membandingkan 2 metode algoritma penjadwalan, sedangkan penelitian ini menggunakan 2 server web dengan menggunakan bobot 2:1, 3:1, 2:3 menggunakan 1 metode algoritma penjadwalan. Kesimpulan yang didapat dari penelitian sebelumnya throughput pada load balancing algoritma Weighted Round Robin 2:3 pada 7500 request dan 10000 request lebih tinggi dibandingkan scenario lainya. Pada pengujian 7500 request dan 10000 request memiliki nilai error request yang kecil dibandingkan dengan algoritma WRR dengan bobot 2:3 mendapatkan nilai request lebih sedikit dibandingkan dengan skenario lainya.

Menjadikan algoritma WRR 2 :3 menjadi pilihan terbaik untuk mengatasi request error[15].

B. Metode Penelitian

1. Tahapan Penelitian

   Tahapan penelitian merupakan rencana penelitian untuk menerapkan sistem yang dibuat oleh peneliti, Pada penelitian ini melakukan dalam 5 tahapan untuk melakukan penelitian dapat dilihat pada Gambar 1. berupa diagram penelitian yang meliputi :



Gambar 1. Diagram Penelitian

a. Tinjauan Pustaka dimana peneliti melakukan tinjauan dan mempelajari jurnal-jurnal atau paper yang berkaitan dengan tema penelitian, yaitu adalah tentang load balancing algoritma WRR dan WLC menggunakan Nginx sebagai load balancing yang berkemampuan tinggi dalam menangani beban traffic.
b. Analisis Kebutuhan merupakan proses identifikasi, pemahaman dan dokumentasi kebutuhan yang digunakan dalam menyelesaikan penelitian, kebutuhan hardware dan software untuk mengembangkan system dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Hadware

No	Kategori	*Spesifikasi*
1	OS	Windows 11
2	RAM	8 GB
3	Processor	Intel Core i5 Gen 8

Tabel 2. Software

No	Software	*Fungsi*
1	VirtualBox	Aplikasi yang digunakan untuk membuat Server dan load balancing
2	Putty SSH	Aplikasi yang digunakan untuk akses Server menggunakan SSH
3	Apache Benchmark	Aplikasi yangdigunakan untuk pengujian load balancing

c. Perancangan Sistem pada tahap ini peneliti membuat rancangan infrastruktur yang akan digunakan dalam perancangan sistem dengan tujuan memberikan gambaran tentang sistem yang akan dibuat. Perancangan sistem meliputi load balancing Nginx, algoritma Weighted Round Robin(WRR) dan algoritma Weighted Least Connection(WLC).
d. Implementasi merupakan penerapan dari hasil tahap analisis data, tahap desain dan tahap simulasi yang akan diimplementasikan dengan load balancing Nginx menggunakan metode algoritma WRR dan WLC.
e. Analisis Hasil Perbandingan pada tahap ini merupakan pengujian sistem yang telah dibuat menggunakan apache Benchmark dan melakukan perhitungan menggunakan parameter quality of service (QoS).

2. Metode Pengumpulan Data

Metode pengumpulan data dari penelitian ini berasal dari dua sumber

a. Study Literatur

Dalam penelitian ini, informasi diperoleh dari berbagai sumber dan referensi yang mendukung analisis perbandingan algoritma Weighted Round Robin (WRR) dan Weighted Least Connection (WLC).

b. Observasi

Metode observasi merupakan pendekatan yang digunakan untuk mengamati secara langsung aktivitas dalam penelitian. Pengumpulan data dilaksanakan dengan mengirimkan simulasi pengujian ke server load balancing.

3. Load balancing

Load balancing merupakan teknologi yang membagi beban kerja ke beberapa server guna mencegah beban berlebih pada satu web server tertentu. Tujuan dari load balancing adalah meningkatkan throughput, mengurangi waktu respons, dan mencegah terjadinya overload pada satu server. Dengan menerapkan load balancing, pembagian beban kerja dapat mencapai efisiensi optimal dalam penggunaan sumber daya dan mengurangi waktu analisis data secara minimal[16][17].

4. Nginx

Nginx merupakan software web server yang open source dan memiliki kinerja tinggi, Nginx dirancang untuk menangani lalu lintas web dengan optimal, ringan, dan mudah ditingkatkan. Kelebihan utama Nginx

meliputi kemampuan untuk menangani banyak koneksi bersamaan, selain itu Nginx juga sering digunakan sebagai reverse proxy server, dan load balancer Kemampuannya untuk menangani permintaan HTTP dengan cepat. Selain itu, Nginx dapat digunakan untuk menyajikan konten statis dan dinamis, membuatnya sangat fleksibel dan dapat diintegrasikan dengan berbagai jenis aplikasi web[18].

5. Algoritma Weighted Round Robin

Algoritma penjadwalan Weighted Round Robin (WRR) merupakan suatu metode penjadwalan yang digunakan untuk meratakan lalu lintas atau beban kerja di antara sejumlah server dengan memberikan bobot pada masing-masing server. Bobot ini mencerminkan kontribusi atau kapasitas relatif dari setiap server terhadap sistem. Setiap server diberikan bobot berupa nilai integer yang mencerminkan kapasitas pemrosesannya. Server dengan bobot yang lebih tinggi mendapatkan prioritas untuk menerima koneksi baru daripada server dengan bobot yang lebih rendah. Selain itu, server dengan bobot yang setara akan menangani lebih banyak koneksi dibandingkan dengan server yang memiliki bobot lebih rendah[13].

6. Algoritma Weighted Least Connection

Algoritma Weighted Least Connections (WLC) merupakan suatu pendekatan penjadwalan yang membagi beban lalu lintas berdasarkan bobot kinerja individu dari setiap server. Tiap server diberi bobot yang menentukan seberapa banyak request yang dapat ditangani oleh server tersebut. Semakin tinggi bobot server, semakin banyak permintaan yang dapat diterima. Sejalan dengan algoritma Weighted Round Robin, jika dua server memiliki jumlah koneksi yang sama, server yang memiliki bobot lebih tinggi akan dipilih untuk menangani permintaan yang baru. Keunggulannya terletak pada kemampuannya untuk mengelola kinerja server berdasarkan jumlah koneksi aktif di setiap server, dan jumlah koneksi aktif ini dapat diatur oleh administrator.[19].

7. Quality of Service

Parameter yang digunakan pada Quality of Service[20] sebagai berikut :

a. Throughput

Throughput merupakan total jumlah paket yang berhasil tiba di tujuan selama suatu periode waktu tertentu, dibagi oleh durasi interval waktu tersebut. Istilah "Throughput" seringkali terkait dengan bandwidth karena mencerminkan kemampuan jaringan untuk mentransmisikan data dalam situasi nyata dapat dilihat pada Nomor 1[21].

Throughput = (Jumlah data yang dikirim) / (Kecepatan yang disediakan) (1)

b. Response Time

Waktu Response Time menunjukkan seberapa cepat server memproses permintaan yang diterima dari klien. Waktu Response Time yang diukur menunjukkan berapa lama waktu yang dibutuhkan untuk menyelesaikan setiap permintaan pengujian dan berapa permintaan yang terkirim[14].

c. Packet loss

Packet loss adalah keadaan yang menunjukkan seberapa banyak paket yang hilang secara keseluruhan. Hal ini dapat terjadi akibat tabrakan data dan faktor penghambat lainnya, seperti jumlah pengguna yang tinggi pada suatu waktu, sehingga dapat meningkatkan tingkat packet loss dapat dilihat pada Nomor 3 [22].

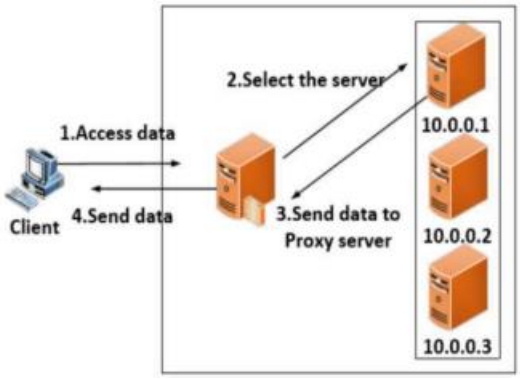
Packet Loss = (Jumlah packet hilang)/(Jumlah paket diterima) x 100% (2)

8. Perancangan Topologi

Perancangan topologi yang akan dilakukan adalah membentuk desain dan perancangan sistem. Dalam penelitian ini, ada 3 tahap dalam perancangan topologi yaitu perancangan sistem load balancing menggunakan Nginx, perancangan sistem load balancing Nginx algoritma WRR, perancangan sistem load balancing Nginx algoritma WLC.

a. Load Balancing Nginx

Proxy terbalik adalah fondasi dari penyeimbang beban Nginx. Dalam load balancing, perhitungan dilakukan untuk menentukan server yang paling sesuai untuk membagi beban di antara server. Prinsip dasar ini dapat ditemukan dalam ilustrasi pada Gambar 2.

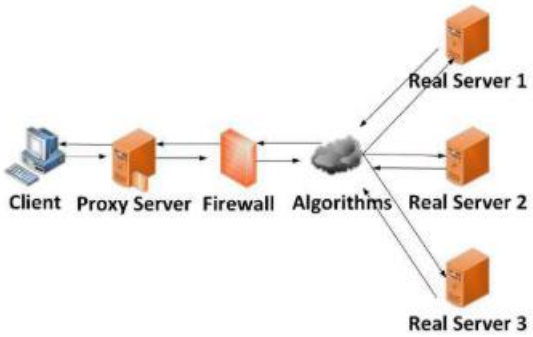


Gambar 2. Prinsip Load balancing [10]

Seperti yang ditunjukan pada Gambar 2 klien meminta akses data terlebih dahulu pada server proxy dan kemudian server proxy akan memilih server back-end mana yang sesuai kapasitasnya. Kemudian server back-end akan meneruskan data yang diminta oleh pengguna kembali kepada klien.

b. Algoritma Weighted Round Robin

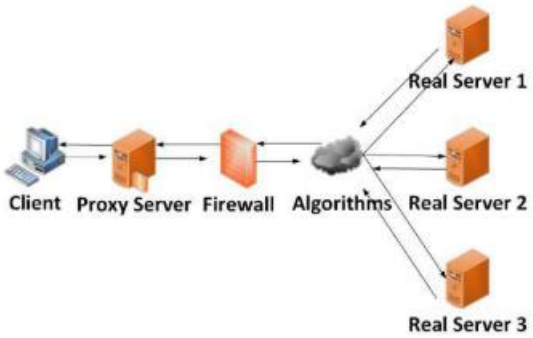
Sistem yang dirancang akan menjadi pedoman dalam pengembangan sistem load balancing dengan menggunakan algoritma Weighted Round Robin. Saat pengguna mengakses server, penyeimbang beban dapat menerapkan algoritma untuk menentukan bobot pada setiap server, sehingga membantu mengurangi risiko kelebihan beban. Ilustrasi rancangan sistem dapat ditemukan pada Gambar 3.



Gambar 3. Metode Algoritma Weighted Round Robin [10]

c. Algoritma Weighted Least Connection

Perancangan sistem yang dilakukan adalah akan digunakan sebagai panduan dalam pembuatan sistem load balancing algoritma Weighted Least Connection. Sama seperti rancangan sebelumnya Ketika pengguna mengakses server, penyeimbang beban dapat menggunakan algoritma untuk menentukan bobot pada masing-masing server yang akan membantu mengurangi terjadinya kelebihan beban. Yang membedakan adalah cara kerja pada algoritma itu sendiri. Rancangan sistem dapat dilihat pada Gambar 4.



Gambar 4. Metode Algoritma Weighted Least Connection[10]

9. Skenario pengujian

Penelitian ini akan mengadakan tiga rangkaian pengujian. Pada Gambar 3 dan Gambar 4 terlihat topologi logika, dimana keduanya memiliki fungsi yang sama, yaitu dari klien sebagai pengirim dan penerima data dari server. Pengujian bertujuan untuk menguji kinerja algoritma Weighted Round Robin (WRR) dan Weighted Least Connection (WLC) menggunakan Apache Benchmark. Pengujian awal melibatkan pengiriman 1000 permintaan dengan bobot server 10:15:20, pengujian kedua dilakukan dengan mengirimkan 2000 permintaan dengan bobot server yang sama, dan pengujian ketiga melibatkan pengiriman 5000 permintaan dengan bobot server 10:15:20.

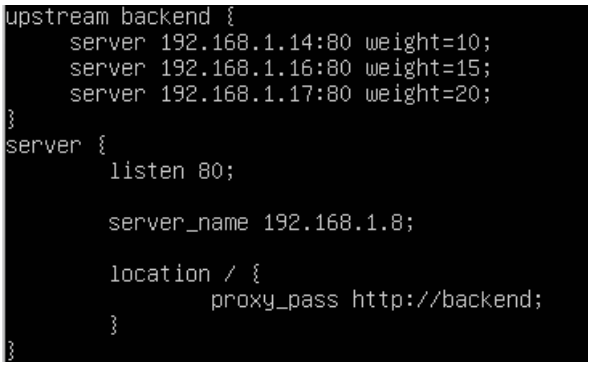
C. Hasil dan Pembahasan

Setelah menyelesaikan implementasi analisis kebutuhan dan perancangan sistem, langkah selanjutnya adalah mengambil hasil dan melakukan pembahasan.

Hasil dan pembahasan ini berfokus pada analisis perbandingan kinerja antara Weighted Round Robin (WRR) dan Weighted Least Connection (WLC) menggunakan Load Balancing Nginx.

1. Konfigurasi Algoritma Weighted Round Robin

   Setelah install Nginx dan web server apache2 selanjutnya mengatur ip pada pengaturan Nginx agar ip web server yang telah ditentukan dapat diakses oleh load balancing Nginx. Untuk mengatur konfigurasi Nginx buka "nano /etc/Nginx/sites-available/default/". Untuk konfigurasi bisa dilihat pada Gambar 5.



Gambar 5. Konfigurasi Weighted Round Robin

2. Konfigurasi Algoritma Least Connection

   Sama seperti pada Gambar 5. Yang membedakan pada Weighted Least Connection hanya ada pada awalan konfigurasi. Untuk konfigurasi terlihat pada Gambar 6.

Gambar 6. Konfigurasi Weighted Least Connection

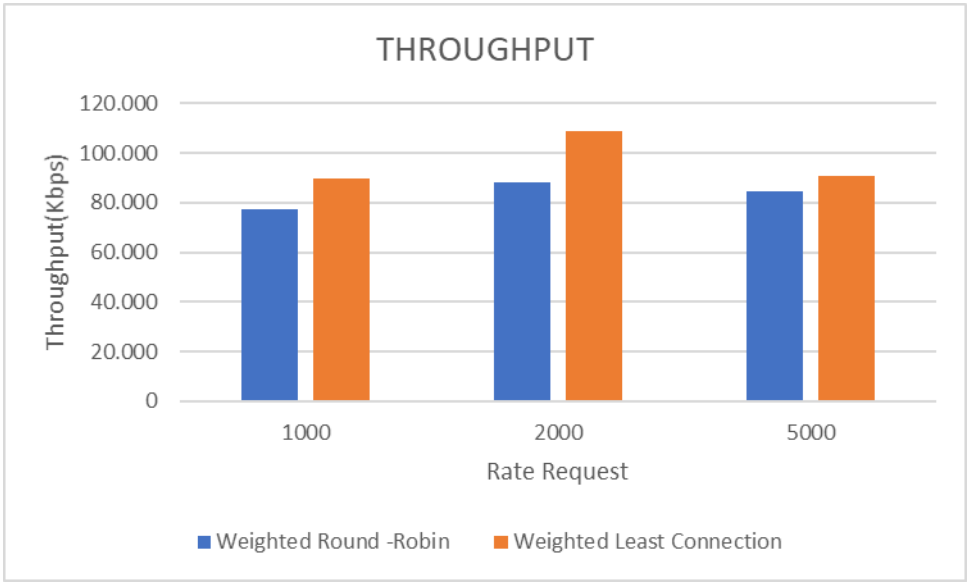
Backend upstream merupakan kelompok server yang difungsikan sebagai server upstream, mencakup alamat IP serta bobot yang telah ditentukan pada server tersebut. Bobot berperan dalam menentukan seberapa sering setiap server menerima permintaan, dimana bobot yang lebih tinggi mengindikasikan bahwa server akan menerima lebih banyak permintaan.

Untuk bagian server blok ini mendefinisikan bagaimana Nginx akan menangani permintaan untuk server. Pada dasarnya Nginx akan mendengarkan permintaan pada port 80 dan akan meneruskan pada salah satu server yang berada pada grup backend yang telah didefinisikan

sebelumnya. Location berguna untuk menentukan cara Nginx menangani permintaan yang diterimanya. Dalam hal ini permintaan akan diteruskan pada grup server backend yang telah didefinisikan sebelumnya menggunakan Proxy_pass http://backend. Untuk bagian server Block ini mendefinisikan bagaimana Nginx akan menangani permintaan untuk server. Jadi, konfigurasi pada Nginx berguna sebagai reverse proxy untuk menerima permintaan pada IP 192.168.1.8:80 dan meneruskannya ke salah satu dari tiga server backend dengan bobot tertentu sesuai dengan distribusi beban yang didefinisikan.

3. Analisis Hasil Throughput

Pengujian throughput dilaksanakan dengan tujuan untuk menilai kecepatan pengiriman paket data dalam sistem jaringan yang telah dikembangkan. Berdasarkan Gambar 7, rata-rata throughput pada load balancing Nginx dengan algoritma penjadwalan Weighted Least Connection menunjukkan performa yang lebih unggul. Data secara keseluruhan menunjukkan bahwa algoritma Weighted Least Connection mencapai kinerja 15% lebih baik dibandingkan dengan algoritma penjadwalan Weighted Round Robin. Dari hasil pengujian ini, nilai rata-rata throughput untuk kedua algoritma penjadwalan tersebut ketika request berjumlah 2000, algoritma Weighted Round Robin mendapatkan 88.070 Kbits/sec dan algoritma Weighted Least Connection mendapat throughput sebesar 108.988 Kbits/sec.



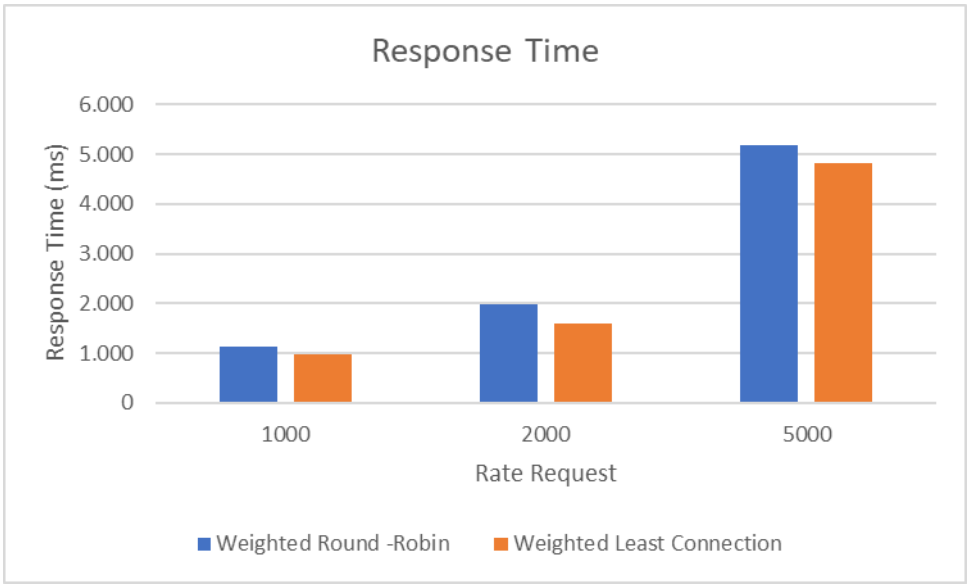
Gambar 7. Throughput

4. Analisis Hasil Response Time

Pengujian Response time dilakukan untuk mengetahui berapa kecepatan Response server dalam menanggapi request klien. Pengujian request dilakukan menggunakan apache benchmark dengan jumlah koneksi dan request 1000, 2000, dan 5000 secara bersamaan.

Jika dilihat dari Gambar 8 algoritma WRR lebih unggul secara keseluruhan dibandingkan dengan metode algoritma WLC. Weighted Least

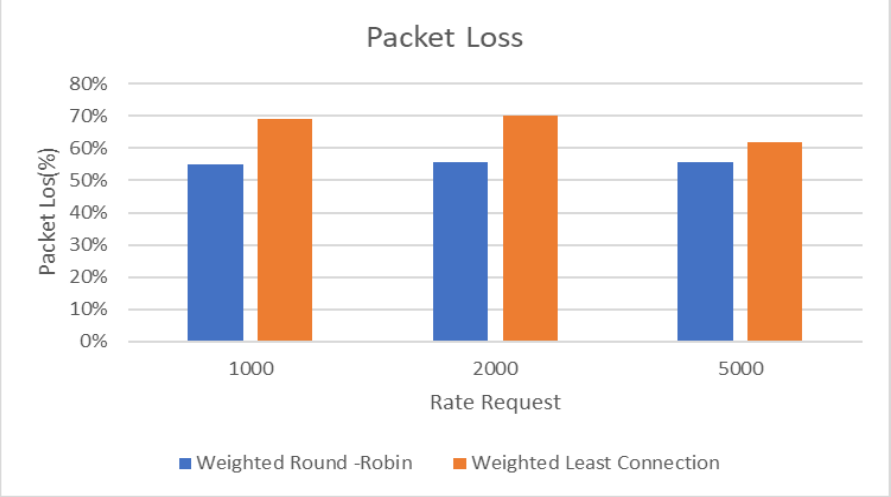
Connection mendapatkan hasil yang kurang optimal dibandingkan dengan algoritma Weighted Round Robin, dilihat dari seluruh rata-rata Response time pada grafik algoritma WLC mendapatkan 2.463ms dan algoritma WRR 2.761ms. dari data yang dilihat Weighted Least Connection memiliki Response time lebih baik karena request dapat diteruskan ke server dengan optimal sesuai bobot masing-masing server sehingga beban kerja server lebih merata.



Gambar 8. Response Time

5. Analisis Hasil Packet Loss

Pengujian kehilangan paket dilakukan untuk menentukan jumlah paket yang hilang saat permintaan dilakukan. Berdasarkan data yang terlihat pada Gambar 9, algoritma Weighted Round Robin mengalami jumlah kehilangan paket yang lebih rendah dibandingkan dengan algoritma Weighted Least Connection. Jika dilihat dari grafik tabel algoritma Weighted Round Robin lebih baik dalam menangani packet loss yang memiliki rata-rata packet loss 55% dan algoritma Weighted Least Connection memiliki rata-rata packet loss 67%.



Gambar 9. Packet Loss

D. Simpulan

Melalui pengujian analisis kinerja algoritma Weighted Round Robin dan Weighted Least Connection pada load balancing Nginx, berhasil dilakukan evaluasi kinerja keduanya. Hasil analisis menunjukkan bahwa dalam uji throughput, algoritma Weighted Least Connection unggul dengan perbandingan 15% lebih baik daripada algoritma Weighted Round Robin. Algoritma Weighted Least Connection (WLC) mencapai waktu throughput terbaik sebesar 108.988 Kbits/sec, sedangkan algoritma Weighted Round Robin (WRR) memiliki throughput sebesar 88.070 Kbits/sec. Sementara itu, untuk parameter response time, algoritma Weighted Least Connection menunjukkan kinerja lebih baik dengan rata-rata waktu response 2.463ms dibandingkan dengan algoritma Weighted Round Robin. Dalam hal packet loss, algoritma Weighted Round Robin juga menunjukkan kinerja superior karena mengalami lebih sedikit kehilangan paket saat meneruskan request ke server dibandingkan dengan algoritma Weighted Least Connection. Algoritma Weighted Round Robin memiliki nilai rata-rata packet loss sebesar 55%. Dari hasil pengujian, dapat disimpulkan bahwa konfigurasi bobot pada setiap server berdampak pada jumlah koneksi yang diterima oleh masing-masing server, dengan server berbobot lebih tinggi akan lebih banyak menangani permintaan dibandingkan dengan server berbobot lebih rendah.

E. Referensi

[1] Y. Cao, "Load balancing design of web cluster based on Nginx under novel virtualization platform," *2021 Int. Conf. Comput. Commun. Artif. Intell. CCAI 2021*, pp. 166–170, 2021, doi: 10.1109/CCAI50917.2021.9447535.

[2] K. Chandravanshi, G. Soni, and D. K. Mishra, "Design and Analysis of an Energy-Efficient Load Balancing and Bandwidth Aware Adaptive Multipath N-Channel Routing Approach in MANET," *IEEE Access*, vol. 10, no. October, pp. 110003–110025, 2022, doi: 10.1109/ACCESS.2022.3213051.

[3] O. H. Jader, S. R. M. Zeebaree, and R. R. Zebari, "A state of art survey for web server performance measurement and load balancing mechanisms," *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 535–543, 2019.

[4] D. P. Kuswandono and A. Saepuloh, "Analisis Troughput Dan Waktu Response Web Server Menggunakan Load Balance Dengan Algoritma Round Robin," *… Gerbang STMIK Bani …*, vol. 11, no. 2, pp. 84–93, 2021, [Online]. Available: http://jurnal.stmik.banisaleh.ac.id/index.php/JIST/article/view/86

[5] M. I. Vol, "Media Informatika Vol.22 No.1 (2023) 35," vol. 22, no. 1, pp. 35–43, 2023.

[6] A. I. 3 Fahmi Apriliansyah 1, Iskandar Fitri 2 and Fakultas, "Implementasi Load Balancing Pada Web Server Menggunakan Nginx," *17th Int. Conf. Geoinformatics - Theor. Appl. Asp.*, vol. 6, no. 1, 2018, doi: 10.3997/2214-4609.201801770.

[7] A. Chyrvon, K. Lisovskyi, and N. Kyryndas, "the Main Methods of Load Balancing on the Nginx Web Server," pp. 146–151, 2023, doi: 10.36074/logos-26.05.2023.040.

[8] D. K. Hakim, D. Y. Yulianto, and A. Fauzan, "Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains*

dan Teknol., vol. 3, no. 2, p. 85, 2019, doi: 10.30595/jrst.v3i2.5165.

[9] E. Qin, Y. Wang, L. Yuan, and Y. Zhong, "Research on Nginx dynamic load balancing algorithm," *Proc. - 2020 12th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2020*, pp. 620–624, 2020, doi: 10.1109/ICMTMA50254.2020.00138.

[10] C. Ma and Y. Chi, "Evaluation Test and Improvement of Load Balancing Algorithms of Nginx," *IEEE Access*, vol. 10, pp. 14311–14324, 2022, doi: 10.1109/ACCESS.2022.3146422.

[11] B. Arifwidodo, V. Metayasha, and S. Ikhwan, "Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE," *J. Telekomun. dan Komput.*, vol. 11, no. 3, p. 210, 2021, doi: 10.22441/incomtech.v11i3.11775.

[12] A. Hanafiah, "Implementasi Load Balancing Dengan Algoritma Penjadwalan Weighted Round Robin Dalam Mengatasi Beban Webserver," *IT J. Res. Dev.*, vol. 5, no. 2, pp. 226–233, 2021, doi: 10.25299/itjrd.2021.vol5(2).5795.

[13] S. L. P. Panjaitan, "Analisis Perbandingan Performansi Load Balancing Menggunakan Algoritma Weighted Round Robin dengan Weighted Least Connection pada Software Defined Network," *Fak. Inform. Univ. Telkom*, 2019.

[14] M. N. A. Rizqi and I. K. Dwi Nuryana, "Analisis Perbandingan Kinerja Algoritma Weighted Round Robin dan Weighted Least Connection Menggunakan Load Balancing Nginx Pada Virtual Private Server(VPS)," *J. Informatics Comput. Sci.*, vol. 4, no. 01, pp. 67–75, 2022, doi: 10.26740/jinacs.v4n01.p67-75.

[15] M. S. Pradana and A. Prapanca, "Analisis Performa Load Balancing Algoritma Weighted Round Robin di Infrastruktur BPBD Provinsi Jawa Timur," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 109–114, 2020, doi: 10.26740/jinacs.v1n02.p109-114.

[16] F. Mardi Nurzaman, F. Chahyadi, and M. Radzi Rathomi, "Analisis Perbandingan Performa Load Balancer Nginx Dan Haproxy Pada Docker," vol. 11, no. 01, pp. 16–25, 2022.

[17] M. Claudia and M. Rifqi, "Analisa Perbandingan Performansi Hot Standby Router Protocol (HSRP) dengan Gateway Load Balancing Protocol (GLBP) Pada Router Spoke DMVPN," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 504, 2021, doi: 10.30865/mib.v5i2.2846.

[18] A. Schwartz, "Bringing Innovative Load Balancing to NGINX," *Int. J. Res. …*, vol. 4, no. 3, pp. 121–126, 2021, [Online]. Available: http://www.journals.resaim.com/ijresm/article/view/586

[19] H. Karim, R. Primananda, and W. Yahya, "Implementasi Load Balancing Web Server Dengan Algoritme Weighted Least Connection Pada Software Defined Network," vol. 3, no. 1, pp. 109–118, 2019.

[20] A. Tantoni, M. T. A. Zaen, and L. Mutawalli, "Komparasi QoS Load Balancing Pada 4 Line Internet dengan Metode PCC, ECMP dan NTH," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 110, 2022, doi: 10.30865/mib.v6i1.3436.

[21] P. Risnaldy and I. Neforawati, "Analisa QOS (Quality of Service) Zeroshell pada Mekanisme Load Balancing dan Failover," *Multinetics*, vol. 6, no. 1, pp. 8–14, 2020, doi: 10.32722/multinetics.v6i1.2750.

[22] Aprianto Budiman, M. Ficky Duskarnaen, and Hamidillah Ajie, "Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 4, no. 2, pp. 32–36, 2020, doi: 10.21009/pinter.4.2.6.