The Indonesian Journal of Computer Science
www.ijcs.net
Volume 13, Issue 4, August 2024
https://doi.org/10.33022/ijcs.v13i4.4118

Pengaruh Load Balancing Pada Serangan DDoS Menggunakan Nginx

Dimas Satrya Bhayangkara[1], Wahid Miftahul Ashari[2]
dimasatrya@students.amikom.ac.id, wahidashari@amikom.ac.id
[1, 2]Fakultas Ilmu Komputer, Program Studi Teknik Komputer, Universitas Amikom Yogyakarta

Informasi Artikel

Diterima : 11 Jun 2024
Direvisi : 20 Jun 2024
Disetujui : 25 Jul 2024

Kata Kunci

Serangan DDoS, Load balancing, Nginx, Algoritma load balancing

Abstrak

Serangan DDoS (Distributed Denial Of Service) merupakan salah satu serangan siber yang sering terjadi. Serangan ini dapat membuat suatu server mengalami error. Berbagai metode sudah di lakukan untuk mengatasi serangan ini salah satunya load balancing. Load balancing bertanggung jawab untuk membagi beban kerja ke berbagai server dengan merata. Pada penelitian ini menggunakan load balancing Nginx. Penelitian dilakukan dengan mengirimkan request 100.000, 300.000, 400.000, dan 500.000. Throughput setelah menggunakan load balancing menunjukan keunggulan dengan rata – rata 9.581 Kb/s dibandingkan dengan tidak menggunakan load balancing. Response time menggunakan load balancing juga lebih baik dibandingkan dengan tidak menggunakan load balancing dengan rata – rata 4507.23 ms. Namun, pada packet loss menunjukan tidak adanya packet loss yaitu 0% setelah menggunakan load balancing dan sebelum menggunakan load balancing. Pengaruh Load balancing pada Nginx dapat mencegah terjadinya serangan DDoS dengan algoritma load balancing yang masih cukup baik untuk digunakan.

Keywords

DDoS attack, Load balancing, Nginx, Load balancing algorithm

Abstract

DDoS (Distributed Denial of Service) attacks are one of the most common cyberattacks. This attack can make a server experience an error. Various methods have been used to overcome this attack, one of which is load balancing. Load balancing is responsible for dividing the workload among various servers evenly. In this study, we used Nginx load balancing. The research was conducted by sending 100.000, 300.000, 400.000, and 500.000 requests. Throughput after using load balancing shows superiority, with an average of 9,581 kb/s compared to not using load balancing. Response time using load balancing is also better than not using load balancing, with an average of 4507.23 ms. However, the packet loss shows no packet loss, which is 0% after using load balancing and before using load balancing. The effect of load balancing on Nginx can prevent DDoS attacks with a load balancing algorithm that is still good enough to use.

A. Pendahuluan

Saat ini, layanan internet memiliki peranan penting dalam kehidupan sehari-hari, baik individual maupun organisasi sangat memanfaatkan adanya layanan internet. Pada tahun 2018 pengguna internet sudah mencapai 3.98 miliar dan semakin bertambah tiap tahunnya[1]. Layanan internet memberikan manfaat dalam berbagai bidang seperti komunikasi, informasi, mempermudah pekerjaan, dll[2]. Disisi lain kekurangan dari internet adalah penyalahgunaan internet. Salah satu penyalahgunaan pada jaringan internet adalah bahaya kejahatan siber. Salah satu serangan siber yang berbahaya adalah serangan DOS atau DDoS (Distributed Denial Of Service). Serangan ini menjadi ancaman tertinggi pada internet maupun layanan web[3], [4].

DDoS sendiri merupakan perkembangan atau variasi dari DOS (Denial Of Service). Serangan ini menargetkan seperti server, layanan, dan jaringan [1]. Serangan ini mengirimkan request dalam jumlah yang banyak yang menciptakan gangguan lalu lintas dengan memberi beban berlebih pada server, sehingga suatu situs dapat mengalami kinerja yang lambat ketika diakses, bahkan serangan ini bisa menyebabkan kegagalan server[5], [6]. Akibat serangan ini, banyak metode sudah dilakukan untuk menangani serangan ini. Salah satu metode yang dapat dilakukan untuk menangani serangan ini adalah dengan menggunakan lebih dari satu server atau sistem kluster server. Dalam menggunakan kluster server dibutuhkan cara untuk membagi beban agar merata pada setiap server yaitu dengan menggunakan load balancing[7], [8].

Load balancing bertanggung jawab untuk membagi beban kerja ke berbagai server dengan merata, sehingga ketika suatu server diserang dapat mengurangi resiko kegagalan pada server. Selain itu, beban kerja menjadi lebih ringan karena beban tersebut dibagi diantara berbagai server. Walaupun load balancing pada awalnya tidak dibuat untuk melawan serangan DDoS, namun load balancing menjadi lapisan keamanan yang efektif untuk mencegah serangan DDoS dengan konfigurasi yang tepat[9], [10]. Dalam melakukan pembagian kerja pada load balancing dibutuhkan algoritma yang berguna untuk melakukan penjadwalan. Beberapa algoritma yang sering digunakan dan cukup bagus adalah Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), Ip Hash [11], [12].

Pada penelitian – penelitian sebelumnya haproxy digunakan sebagai load balancing. Haproxy merupakan software open source penyedia high availability dan load balancing yang berguna dalam mengelola situs web dengan lalu lintas yang tinggi. Namun, Haproxy hanya berfungsi sebagai proksi aplikasi load balancing dan tidak bisa membuat layanan web sendiri atau web server sendiri. Namun, dalam penelitian ini menggunakan penyedia load balancing sekaligus sebagai web server. Nginx jadi pilihan karena Nginx menyediakan web server sekaligus load balancing dan reverse proxy, dengan tingkat ketergantungan yang minim terhadap stabilitas jaringan, ini cocok digunakan dalam lingkungan jaringan yang kompleks. Nginx memiliki keunggulan dibandingkan dengan Haproxy seperti fungsionalitas, kemudahan dalam penggunaan karena proses instalasinya yang mudah, konfigurasi yang fleksibel, dan lainnya[12], [13].

Pada penelitian ini membahas tentang metode load balancing untuk mengatasi serangan DDoS pada web server Nginx dengan menggunakan algoritma

penjadwalan Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), dan IP Hash (IH). Serangan DDoS dilakukan dengan menggunakan tool apache benchmark dengan mengirimkan total request 100.000, 300.000, 400.000, dan 500.000. Perhitungan throughput, response time, packet loss menggunakan parameter Quality Of Service (QOS). Menggunakan total 4 virtual machines sebagai web server dengan 1 sebagai server load balancing.

B. Tinjauan Pustaka

Penelitian yang berjudul "Distributed Denial of Service Attack Mitigation using High Availability Proxy and Network Load Balancing" membahas tentang pencegahan serangan DDoS dengan menggunakan High Availability Proxy and Network Load Balancing. Penelitian ini membandingkan performa server yang tidak menggunakan load balancing dan yang menggunakan load balancing. Serangan DDoS dilakukan menggunakan tool apache jmeter dengan mengirimkan total request HTTP 50000, 100000, 150000, 200000, 250000, and 300000. Hasil dari penelitian ini High Availability Proxy and Network Load Balancing dapat membagi beban atau lalu lintas yang masuk ke server secara merata ke dalam sebuah kluster server dan memastikan tidak ada satu server yang menjadi hambatan, sehingga mengurangi dampak dari serangan DDoS, mendeteksi dan memblokir lalu lintas berbahaya, menjaga ketersediaan layanan, dan memastikan server tetap performa yang tinggi[3].

Berdasarkan penelitian yang berjudul "Mitigating denial of service attacks with load balancing", membahas tentang pentingnya mitigasi terhadap serangan DDoS dengan load balancing dan efektivitas penggunaan HAProxy sebagai mekanisme pertahanan. Cara HAProxy dalam mendeteksi dan mencegah serangan DDoS menggunakan beberapa mekanisme seperti mekanisme dynamic scheduling, connection limit, real time monitoring, configuration flexibility, dan load balancing. Pada penelitian ini menggunakan goldeneye tool untuk membuat serangan DDoS dan apache sebagai web server. Menggunakan load balancing HAProxy tidak hanya membantu mengoptimalkan pemanfaatan sumber daya yang ada tetapi juga mencegah server utama mengalami kegagalan ketika terjadi serangan DDoS karena load balancing membagi lalu lintas atau beban dari banyak request masuk ke beberapa server berdasarkan algoritma yang dikonfigurasi[4].

Pada penelitian yang berjudul "Optimization of Load Balancing Algorithms to Deal with Ddos Attacks Using Whale optimization Algorithm", membahas tentang metode load balancing dengan menggunakan Whale Optimization Algorithm (WOA) untuk mengatasi serangan DDoS dan dicoba dibandingkan dengan beberapa algoritma load balancing seperti Round-Robin (RR), Particle Swarm Optimization (PSO), and Genetic Algorithms (GA). Menggunakan load balancing dengan algoritma Whale Optimization Algorithm (WOA) mampu berkerja lebih baik dari pada beberapa algoritma sebelumnya untuk menangani request dari klien. Kesimpulannya dengan load balancing mampu membantu dalam mengurangi serangan DDoS dengan cara membagi beban secara adil pada server dengan algoritma penjadwalan[9].

Pada penelitian yang berjudul "DDOS Mitigation In Cloud Computing Environment By Dynamic Resource Scaling With Elastic Load Balancing",

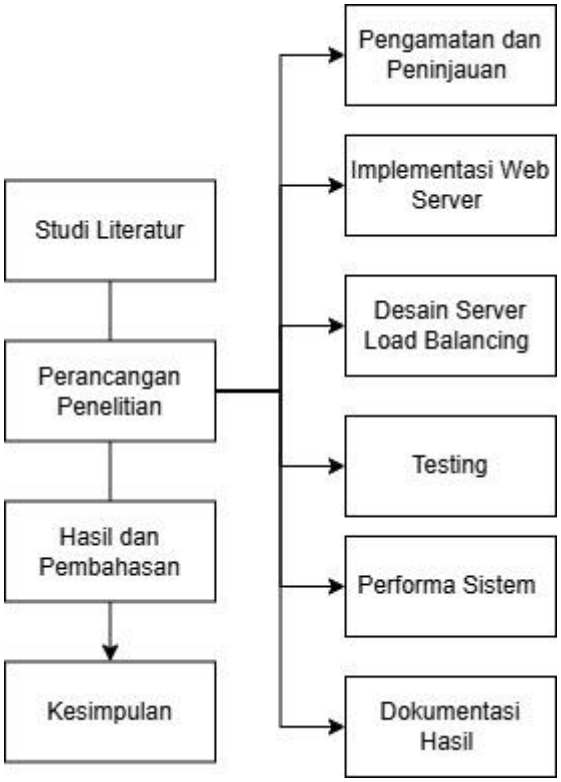
membahas tentang metode baru untuk menilai penggunaan sumber daya dengan mengurangi sumber daya menggunakan elastic load balancing, dengan melakukan analisis data request atau lalu lintas yang masuk dan dari kondisi server. Serangan DDoS yang terus terjadi menyebabkan penggunaan sumber daya yang tinggi yang menargetkan server korban, sehingga berdampak pada konflik sumber daya, yang menghambat layanan klien yang sah. Metode elastic load balancing mengurangi sumber daya dan layanan dengan mengorbankan sumber daya korban selama waktu serangan untuk mengurangi dampaknya dan memfasilitasi pemulihan pada server. Hasil dari penelitian ini metode elastic load balancing efisien dalam mengurangi serangan DDoS dengan mengurangi penggunaan sumber daya, dengan ini mengurangi waktu layanan pengguna[14].

Pada penelitian yang berjudul "Optimized load balance scheduling algorithm", membahas tentang fungsi dari load balancing untuk mengatasi serangan DDoS dengan algoritma penjadwalan yang sudah ditentukan. Pada penelitian ini dilakukan testing melakukan serangan DDoS pada server yang belum menggunakan load balancing dan yang menggunakan load balancing dengan menggunakan optimalisasi hybrid algoritma BAT dan Cuckoo. Hasil dari penelitan ini load balancing dengan optimasi dari algoritma yang diusulkan mampu meminimalisasi serangan DDoS yang tujuannya memberi beban tak berujung pada server. Perbandingan performa dari server sebelum dan sesudah menggunakan load balancing adalah total 60%, dengan adanya load balancing serangan DDoS dapat dikurangi dengan melakukan pembagian beban secara merata[15].

Setelah tinjuan pustaka dilakukan perbedaan dari penelitian sebelumnya beban serangan DDoS yang dilakukan berbeda, pada penelitian ini beban yang diberikan lebih besar dari penelitian – penelitian sebelumnya dengan total requests 100.000, 300.000, 400.000, dan 500.000. Perbedaan perbandingan algoritma yang digunakan pada penelitian ini menggunakan 5 algoritma yaitu penjadwalan Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), dan IP Hash (IH). Pada penelitian ini menggabungkan beberapa metode dari penelitian sebelumnya, metode yang dilakukan akan dijelaskan pada Bab C Metode Penelitian. Perhitungan parameter dilakukan menggunakan metode Quality Of Service (QOS) dengan menghitung parameter throughput, response time, packet loss. Penelitian ini bertujuan untuk memberikan pemahaman bagaimana pengaruh dari load balancing menggunakan Nginx pada server terhadap serangan DDoS dengan melihat perbandingan dari perhitungan menggunakan parameter QOS pada throughput, response time, dan packet loss dan kekurangan dari server ketika terkena serangan DDoS tanpa menggunakan load balancing.

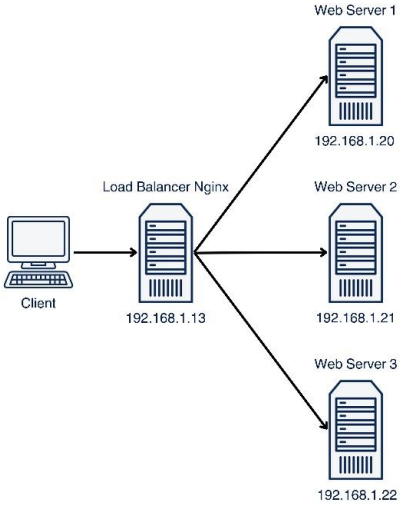
C. Metode Penelitian

Metode yang digunakan terdapat 4 tahapan yang akan dijelaskan pada gambar 1, sebagai berikut :



Gambar 1. Alur Metode Penelitian

1. **Metode Studi Literatur**
   Pada tahap metode studi literatur peneliti melakukan identifikasi dan analisis dari dokumentasi penelitian sebelumnya untuk mendapat referensi dan informasi yang sesuai dengan tujuan dan tema penelitian ini yaitu load balancing untuk mengatasi serangan DDoS. Dokumentasi ini berupa jurnal atau papper dari penelitan sebelumnya.
2. **Perancangan Penelitian**
   Pada tahap metode perancangan penelitian, peneliti membuat langkah – langkah yang digunakan pada penelitian. Pada tahap ini terdapat 6 langkah yang digunakan sebagai berikut :
   A. Pengamatan dan Peninjauan merupakan tahap penelitian yang dilakukan studi literatur untuk mendapat informasi dari jurnal dan referensi yang sesuai dengan tujuan penelitian ini, yaitu DDoS (Distributed Denial Of Service), Load balancing, dan QOS (Quality Of Service).
   B. Implementasi Web Server merupakan tahap peneliti membuat web server dilakukan perancangan sistem spesifikasi dari web server dan kemudian dilakukan intalasi pada web server tersebut dengan melakukan instalasi web server Nginx.
   C. Desain Server Load Balancing merupakan tahap pembuatan topologi dan rancangan sistem server load balancing menggunakan load balancing Nginx. Pada tahap ini juga dilakukan konfigurasi load balancing.



Gambar 2. Topologi Load Balancing Nginx

D. Testing dilakukan oleh peneliti melakukan testing pada web server load balancing dengan mengirimkan serangan DDoS. Serangan DDoS dilakukan dengan tool testing web yaitu Apache benchmark dengan mengirimkan requests 100.000, 300.000, 400.000, dan 500.000.
E. Performa Sistem dimana peneliti melakukan pengamatan pada performa saat dan setelah serangan DDoS dilakukan. Pengamatan dilakukan pada server sebelum menggunakan load balancing dan sesudah menggunakan.
F. Dokumentasi Hasil merupakan tahap peneliti melakukan analisis hasil dari pengamatan pada performa sistem dengan melakukan perhitungan menggunakan parameter dari QOS yaitu throughput, response time, dan packet loss dan kemudian hasil perhitungan dilakukan dokumentasi.

3. Hasil dan Pembahasan
Pada tahap ini hasil dan pembahasan akan dijelaskan secara detail pada BAB D. Hasil dan Pembahasan.

4. Simpulan
Pada tahap ini akan dijelaskan pada BAB E. Kesimpulan

Hasil dan pembahasan akan diukur menggunakan parameter yang sudah ditentukan sebelumnya pada tahapan studi literatur. Adapun penjelasan parameter penelitian yang akan digunakan akan dijelaskan dibawah ini :

1. Load balancing
load balancing merupakan salah satu metode yang sering digunakan dalam mengelola banyak server atau kluster server. Fitur utamanya adalah antara setiap server mempunyai peran yang sama pentingnya, dan berkerja sama memberikan layanan kepada sekelompok pengguna sesuai dengan aturan algoritma tertentu. Load balancing membagi beban secara merata pada setiap server. Tujuannya adalah memaksimalkan penggunaan sumber daya

dengan mengoptimalkan throughput, response time, dan mencegah terjadinya kelebihan beban[17], [18].

2. Round Robin (RR)
   Algoritma round robin beroperasi dengan asumsi bahwa semua node yang terlibat memiliki sumber daya yang setara. Tugas yang baru masuk akan dialokasikan ke node secara berurutan sesuai dengan suatu pola rotasi. Meskipun sederhana, algoritma ini tidak mempertimbangkan perbedaan dalam kemampuan komputasi antara setiap node[11].
3. Least Connection (LC)
   Algoritma ini secara cepat mengarahkan mengarahkan pengunjung ke server yang memiliki jumlah koneksi aktif paling sedikit pada saat pengunjung tersebut meminta layanan, terutama ketika ada lonjakan penggunaan. Algoritma ini bertujuan untuk menjaga keseimbangan yang adil diantara semua server yang tersedia[19].
4. Weighted Round Robin (WRR)
   Algoritma ini untuk setiap server yang memiliki bobot digunakan secara bergantian. Algoritma ini sama dengan algoritma round robin, perbedaannya algoritma ini bersifat static, sehingga tidak mengubah bobot server setiap operasinya. Selain itu, CPU yang digunakannya lebih kecil untuk menjalankannya. Setiap bobot diberikan pada setiap server untuk menangani setiap beban yang masuk ke server. Server dengan bobot tinggi mendapat prioritas menangani beban paling banyak sebaliknya, yang memiliki bobot kecil mendapat pritoritas terakhir[19].
5. Weighted Least Connection (WLC)
   Algoritma WLC merupakan algoritma perkembangan dari least connection. Algoritma ini juga hampir sama seperti algoritma WRR. Setiap server diberikan bobot berdasarkan kemampuannya. Karena itu, beban server ditentukan oleh jumlah koneksi yang diterimanya. Saat beban baru diberikan, perbandingan antara jumlah koneksi dan bobot dari setiap server dihitung, kemudian beban dari tugas tersebut diberikan dari server bobot besar ke server dengan bobot rendah[11].
6. Ip Hash
   Algoritma Ip Hash mengarahkan permintaan dari alamat IP ke sistem bagian web back-end yang sama, memungkinkan para klien dengan alamat Ip yang sama dan sistem web back-end untuk menjaga sesi yang konsisten. Algoritma ini meneruskan paket dari pengirim yang sama ke server yang sama dengan memanajemen hash dari alamat Ip pengirim dan tujuan[12].
7. Quality Of Service
   Parameter Quality Of Service yang akan digunakan pada pengujian ini[20], yaitu :
   a. Throughput
      Rumus yang digunakan pada parameter throughput [21] ditunjukan pada Nomor 1 sebagai berikut:

      Throughput = jumlah data yang dikirim/waktu pengiriman data *(1)*

   b. Packet Loss

Rumus yang digunakan pada parameter packet loss [22] ditunjukan pada Nomor 2 sebagai berikut :

Packet loss = (paket data dikirim – paket data diterima)/paket data dikirim x 100% (2)

c. Response Time
Response time adalah waktu yang dibutuhkan server untuk menerima jumlah requests dari client [16]. Rumus response time ditunjukan pada nomor 3 sebagai berikut :

Response time = waktu paket dikirim – paket diterima (3)

8. Skenario Pengujian
Pengujian dilakukan dengan tool testing apache benchmark yang mengirimkan total request 100.000, 300.000, 400.0000, dan 500.000 pada server load balancing Nginx dengan algoritma yang sudah ditentukan. Pengujian pertama dilakukan pada server sebelum menggunakan load balancing. pada pengujian ke dua dilakukan pada server sesudah menggunakan load balancing. Kemudian, setelah testing dilakukan perhitungan pada throughput, response time, dan packet loss menggunakan parameter Quality Of Service (QOS). Pengujian ini bertujuan untuk mengetahui kinerja dari load balancing Nginx dengan algoritma yang ditentukan untuk mengatasi serangan DDoS dan membandingkan algoritma yang digunakan pada load balancing.

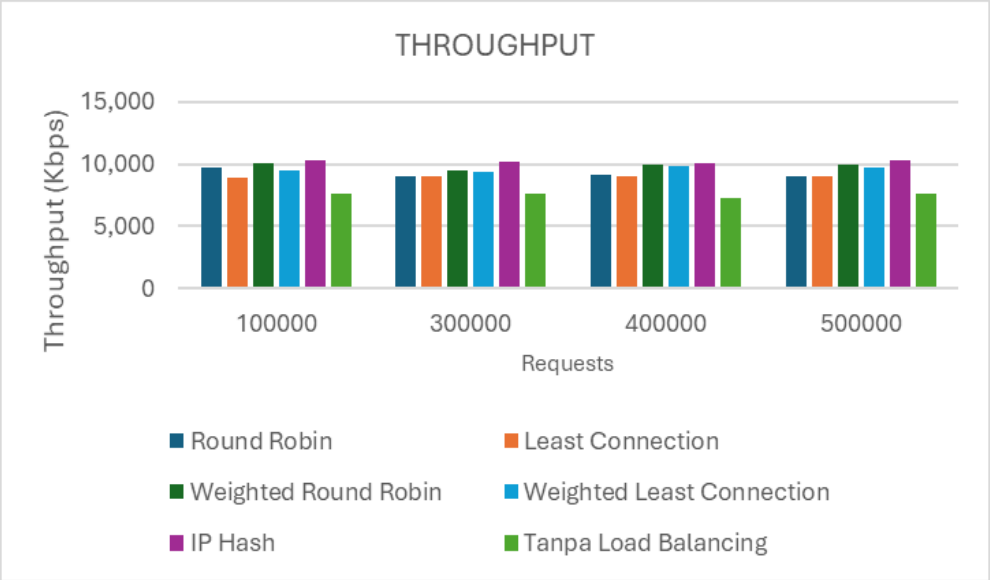
D. Hasil dan Pembahasan

A. Analisa Hasil Throughput
Setelah dilakukan pengujian pada throughput sebelum menggunakan load balancing throughput lebih kecil dibandingkan setelah menggunakan load balancing perbandingan ini dapat dilihat pada gambar 3, rata – rata throughput sebelum menggunakan load balancing adalah 7.518 Kb/s, sedangkan setelah menggunakan load balancing menunjukan rata – rata throughputnya 9.581 Kb/s. Penggunaan load balancing menunjukan keunggulan dibandingkan dengan sebelum menggunakan load balancing.

Tabel 1. Hasil Perhitungan Throughput

No	Algoritma	Requests			
		100000	300000	400000	500000
1	Round Robin	9,768 Kb/s	8,999 Kb/s	9,093 Kb/s	9,070 Kb/s
2	Least Connection	8,938 Kb/s	9,012 Kb/s	9,029 Kb/s	8,984 Kb/s
3	Weighted Round Robin	10,021 Kb/s	9,532 Kb/s	9,900 Kb/s	9,918 Kb/s
4	Weighted Least Connection	9,515 Kb/s	9,408 Kb/s	9,851 Kb/s	9,749 Kb/s
5	IP Hash	10,323 Kb/s	10,205 Kb/s	10,043 Kb/s	10,254 Kb/s
6	Tanpa Load Balancing	7,574 Kb/s	7,609 Kb/s	7,222 Kb/s	7,668 Kb/s

Pada tabel 1 menampilkan informasi terkait hasil perhitungan throughput menggunakan rumus Quality Of Service (QOS) dari pengujian serangan DDoS pada server sebelum dan sesudah menggunakan load balancing dengan algoritma Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), Ip Hash. Dari hasil pengujian ini untuk melihat pengaruh dari load balancing pada serangan DDoS dilakukan perbandingan pada server sebelum dan sesudah menggunakan load balancing untuk lebih jelasnya dapat dilihat pada grafik perbandingan pada gambar 3.



Gambar 3. Grafik Perbandingan Throughput

Gambar 3 merupakan grafik perbandingan dari hasil pengujian pada throughput. Dapat terlihat dengan menggunakan load balancing dengan 5 algoritma Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), Ip Hash menunjukan nilai throughput yang lebih unggul dibandingkan tanpa load balancing. rata – rata algoritma yang digunakan pada load balancing berada lebih tinggi dibandingkan tanpa menggunakan load balancing.

B. Analisa Hasil Response Time

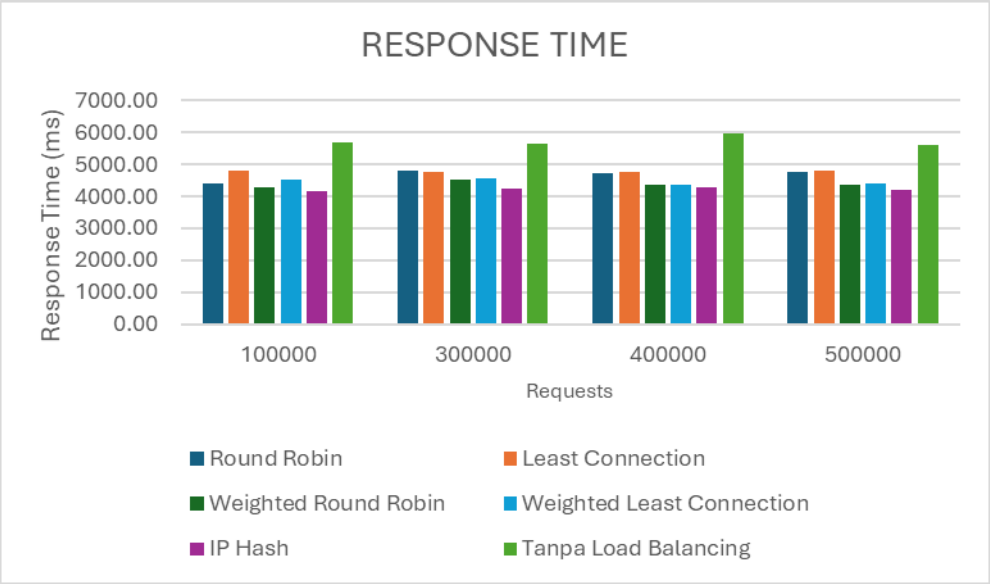
Pada hasil response time setelah dilakukan pengujian menggunakan load balancing untuk mengatasi serangan DDoS menunjukan hasil yang lebih unggul dibandingkan sebelum menggunakan load balancing. Response time sebelum menggunakan load balancing rata – rata mendapat 5732.81 ms lebih tinggi dibandingkan setelah menggunakan load balancing dengan rata – rata 4507.23 ms. Untuk perbandingan response time dapat dilihat pada gambar 4.

Tabel 2. Hasil Perhitungan Response Time

No	Algoritma	Requests			
		100000	300000	400000	500000
1	Round Robin	4410.14 ms	4786.74 ms	4737.67 ms	4749.52 ms

2	Least Connection	4819.50 ms	4780.24 ms	4770.96 ms	4795.01 ms
3	Weighted Round Robin	4298.74 ms	4519.14 ms	4351.20 ms	4343.62 ms
4	Weighted Least Connection	4527.11 ms	4578.65 ms	4373.09 ms	4418.70 ms
5	IP Hash	4173.02 ms	4221.17 ms	4289.37 ms	4200.94 ms
6	Tanpa Load Balancing	5687.89 ms	5661.38 ms	5964.44 ms	5617.54 ms

Pada tabel 2 menampilkan informasi terkait hasil perhitungan Response time dari pengujian serangan DDoS pada server sebelum dan sesudah menggunakan load balancing dengan algoritma Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), Ip Hash. Berdasarkan hasil pengujian ini untuk melihat pengaruh dari load balancing pada serangan DDoS dilakukan perbandingan pada server sebelum dan sesudah menggunakan load balancing untuk lebih jelasnya dapat dilihat pada grafik perbandingan pada gambar 4.



Gambar 4. Grafik Perbandingan Response Time

Gambar 4 merupakan grafik perbandingan dari hasil pengujian pada response time. Dapat terlihat dengan menggunakan load balancing dengan 5 algoritma Round Robin (RR), Least Connection (LC), Weighted Round Robin (WRR), Weighted Least Connection (WLC), Ip Hash menunjukan waktu response time yang lebih cepat dibandingkan tanpa load balancing. rata – rata algoritma yang digunakan pada load balancing berada dibawah response time 5000.00 ms dibandingkan tanpa menggunakan load balancing yang rata – rata diatas response time 5000.00 ms.

C. Analisa Hasil Packet Loss

Analisis paket loss bertujuan untuk mendapatkan jumlah paket yang hilang ketika request dikirimkan ke server. Pada hasil analisis paket loss tanpa menggunakan load balancing menunjukan tidak adanya paket loss, yaitu dengan jumlah 0%, semua request yang dikirimkan sukses dikirimkan. Pada

packet loss menggunakan load balancing juga sama semua paket yang dikirimkan terkirim semua dengan jumlah 0%. Dari hasil pengujian perhitungan packet loss perbedaan jumlah packet loss antara menggunakan load balancing dan tanpa load balancing menunjukan tidak adanya perbedaan. Kedua kondisi menunjukan hasil yang sama yaitu tidak adanya packet loss semua packet yang dikirimkan terkirim dengan sempurna.

E. Simpulan

Penggunaan load balancing memiliki kinerja yang cukup baik untuk mengatasi serangan DDoS dibandingkan dengan tidak menggunakan load balancing. Hasil pengujian throughput menunjukan ketika menggunakan load balancing menunjukan rata – rata 9.581 Kb/s lebih tinggi dibandingkan dengan tidak menggunakan load balancing dengan rata – rata 7.518 Kb/s. Response time ketika menggunakan load balancing juga lebih cepat dengan rata – rata 4507.23 ms, sedangkan tanpa load balancing mendapat rata – rata 5732.81 ms. Namun, pada hasil pengujian packet loss ketika menggunakan load balancing dan tanpa menggunakan load balancing menunjukan hasil yang sama yaitu 0% semua paket yang dikirimkan terkirim semua tidak ada paket yang hilang. Kesimpulannya penggunaan load balancing dengan algoritma yang ditentukan mampu memberikan dampak yang cukup signifikan untuk mengatasi serangan DDoS dengan melihat dari aspek parameter QOS throughput, response time, dan packet loss.

F. Referensi

[1] S. R. M. Zeebaree, K. Jacksi, and R. R. Zebari, "Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 505–512, 2020, doi: 10.11591/ijeecs.v19.i1.pp505-512.

[2] A. Abhishta, W. van Heeswijk, M. Junger, L. J. M. Nieuwenhuis, and R. Joosten, "Why would we get attacked? An analysis of attacker's aims behind DDoS attacks," *J Wirel Mob Netw Ubiquitous Comput Dependable Appl*, vol. 11, no. 2, pp. 3–22, Jul. 2020, doi: 10.22667/JOWUA.2020.06.30.003.

[3] R. R. Zebari, S. R. M. Zeebaree, A. B. Sallow, H. M. Shukur, O. M. Ahmad, and K. Jacksi, "Distributed Denial of Service Attack Mitigation using High Availability Proxy and Network Load Balancing," in *3rd International Conference on Advanced Science and Engineering, ICOASE 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 174–179. doi: 10.1109/ICOASE51841.2020.9436545.

[4] A. Ezenwe, E. Furey, and K. Curran, "Mitigating denial of service attacks with load balancing," *Journal of Robotics and Control (JRC)*, vol. 1, no. 4, pp. 129–135, Jul. 2020, doi: 10.18196/jrc.1427.

[5] M. Kiruthika, J. J. Charivukalayil, S. Chavan, J. J. Mathew, and C. Cardoza, "Enhancement of detection mechanisms for HTTP based DoS/DDoS attacks," 2023.

[6] R. R. Zebari, K. Jacksi, and S. R. M. Zeebaree, "Performance analysis of IIS10.0 and Apache2 Cluster-based Web Servers under SYN DDoS Attack", [Online]. Available: https://www.researchgate.net/publication/340341694

[7] Z. P. Putro and R. A. Supono, "Comparison Analysis of Apache and Nginx Webserver Load Balancing on Proxmox VE in Supporting Server Performance," *International Research Journal of Advanced Engineering and Science*, vol. 7, no. 3, pp. 144–151, 2022.

[8] E. Qin, Y. Wang, L. Yuan, and Y. Zhong, "Research on nginx dynamic load balancing algorithm," in *Proceedings - 2020 12th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020, pp. 620–624. doi: 10.1109/ICMTMA50254.2020.00138.

[9] N. M. Abdulkareem and S. R. M. Zeebaree, "Optimization of Load Balancing Algorithms to Deal with Ddos Attacks Using Whale optimization Algorithm," *the journal of duhok university*, vol. 25, no. 2, pp. 65–85, Nov. 2022, doi: 10.26682/sjuod.2022.25.2.7.

[10] Institute of Electrical and Electronics Engineers, *2020 6th International Conference on Web Research (ICWR).*

[11] N. V. Jungum, N. Mohamudally, and N. Nissanke, "A Dynamic Load Balancing Algorithm for Distributing Mobile Codes in Multi-Applications and Multi-Hosts Environment," *International Journal of Computer Science Issues*, vol. 17, 2020, doi: 10.5281/zenodo.3991567.

[12] 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.

[13] C. Gao and H. Wu, "An Improved Dynamic Smooth Weighted Round-robin Load-balancing Algorithm," in *Journal of Physics: Conference Series*, Institute of Physics, 2022. doi: 10.1088/1742-6596/2404/1/012047.

[14] A. Somasundaram and V. S. Meenakshi, "DDOS Mitigation In Cloud Computing Environment By Dynamic Resource Scaling With Elastic Load Balancing," 2021.

[15] R. M. Abdul-Hussein and A. H. Mohammed, "Optimized load balance scheduling algorithm," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, pp. 81–88, Feb. 2022, doi: 10.12928/TELKOMNIKA.v20i1.22464.

[16] T. Wira 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, Dec. 2022, doi: 10.33086/atcsj.v5i2.3743.

[17] Y. Cao, "Load Balancing Design of Web Cluster Based on Nginx under Novel Virtualization Platform," *International Conference on Computer Communication and Artificial Intelligence*, 2021.

[18] X. Jiang, H. Yang, Y. Yang, and Z. Chen, "Cluster load balancing algorithm based on dynamic consistent hash," *Journal of Intelligent and Fuzzy Systems*, vol. 41, no. 3, pp. 4461–4468, 2021, doi: 10.3233/JIFS-189706.

[19] B. Alankar, G. Sharma, H. Kaur, R. Valverde, and V. Chang, "Experimental setup for investigating the efficient load balancing algorithms on virtual cloud," *Sensors (Switzerland)*, vol. 20, no. 24, pp. 1–26, Dec. 2020, doi: 10.3390/s20247342.

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

[21] W. W. Dale, F. Hariadi, R. Mikaela, and I. Malo, "The Effect Of Queue Tree On Packet Loss In Bandwitch Management Online Based School Exam Pengaruh Queue Tree pada paket loss dalam management Bandwitch Ujian Sekolah Berbasis Online," 2021.

[22] N. Nurmiati, L. Surimi, and S. Subardin, "Analisis Kinerja Load Balancing Terhadap Jaringan Internet Menggunakan Metode Equal Cost Multi Path (ECMP)," *Digital Transformation Technology*, vol. 2, no. 2, pp. 52–62, Nov. 2022, doi: 10.47709/digitech.v2i2.1779.