Journal of Informatics Engineering and Software Applications

Performance Analysis Comparison of Load Balancing Using Round Robin and Least Connection Methods on E-Commerce Web Networks

Adinda Awalia[1], Husaini[2]*, Safriadi[3]
[1,2,3] Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA
*Penulis Korespondensi : husaini@pnl.ac.id

INFORMASI ARTIKEL

Riwayat artikel:
Diajukan pada 00 Maret 00
Direvisi pada 00 April 00
Publikasi pada 00 Juni 00

Kata kunci:
Web *E-Commerce*
Load balancing
Apache jmeter
Haproxy
Least connection
Round robin

Keywords:
Web E-Commerce
Load balancing
Apache jmeter
Haproxy
Least connection
Round robin

ABSTRAK

Peningkatan permintaan menyebabkan *web E-commerce* sibuk, hal tersebut dapat mengakibatkan *server web* menjadi *overload* dan akhirnya *server web* menjadi *down*. Oleh karena itu penerapan teknologi *Load balancing* menjadi penting untuk menyeimbangkan beban *server* secara efisien, meminimalkan risiko *overload*, dan meningkatkan kinerja *server web*. Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja dua metode *Load balancing*, yaitu *Round robin* dan *Least connection*, menggunakan parameter perhitungan CPU pada jaringan *web E-Commerce*. Hasil dari penelitian ini adalah sebuah *web E-Commerce* untuk melakukan perbandingan *Load balancing* menggunakan metode *Round robin* dan *Least connection* menggunakan aplikasi *jmeter* sebagai pemberi beban *user* dengan jumlah 10, 100, 250, 500, 1000, 1500, 2000 *user* dalam satu waktu rata-rata nilai CPU 520% pada pengujian tanpa metode, 275% pada pengujian menggunakan metode *least connection* serta 380% pada pengujian menggunakan metode *round robin*.

ABSTRACT

Increased demand causes the E-Commerce web to be busy, this can cause the web server to become overloaded and eventually the web server to go down. Therefore, the application of Load balancing technology is important to balance the server load efficiently, minimize the risk of overload, and improve web server performance. This study aims to analyze and compare the performance of two Load balancing methods, namely Round robin and Least connection, using CPU calculation parameters on the E-Commerce web network. The results of this study are an E-Commerce web to compare Load balancing using the Round robin and Least connection methods using the jmeter application as a user load provider with a number of 10, 100, 250, 500, 1000, 1500, 2000 users at one time an average CPU value of 520% in testing without a method, 275% in testing using the least connection method and 380% in testing using the round robin method.

1. Pendahuluan

Seiring pesatnya perkembangan teknologi digital, *E-commerce* telah mengubah pola perdagangan menjadi transaksi *online* melalui *platform marketplace*, yang meningkatkan permintaan akses *web* secara signifikan. Namun, tingginya permintaan ini dapat menyebabkan *server* mengalami kelebihan beban (*overload*), memperlambat kinerja, atau bahkan membuat *server* tidak dapat diakses (*down*). Kegagalan ini dapat merugikan reputasi dan finansial perusahaan [1]. Solusi tradisional seperti peningkatan *hardware*

seringkali hanya bersifat sementara dan mahal, serta tidak mengatasi masalah utama yakni manajemen fluktuasi beban *server* yang terus meningkat [2].

Teknologi *load balancing* hadir sebagai solusi fundamental untuk mengatasi masalah ini. *Load balancing* bekerja dengan menyeimbangkan beban pada *server*, mempercepat waktu respon, dan mencegah kemacetan dengan mendistribusikan permintaan pengguna secara efisien ke beberapa *server* yang tersedia [3]. Dalam implementasinya, terdapat berbagai metode *load balancing*, di antaranya yang paling umum adalah *Round Robin* dan *Least Connection*. Kajian literatur menunjukkan bahwa metode *Round Robin* (RR) adalah metode statis yang membagi beban secara bergilir dan berurutan ke setiap *server* [6]. Metode ini dikenal sederhana dan efektif untuk *server-server* dengan kapasitas yang seragam [5]. Sementara itu, *Least Connection* (LC) adalah metode dinamis yang mengarahkan trafik baru ke *server* dengan jumlah koneksi aktif paling sedikit pada saat itu [7]. Studi sebelumnya oleh [6], [8], dan [10] telah membandingkan metode-metode ini, namun seringkali dalam konteks jaringan umum atau *web server* statis.

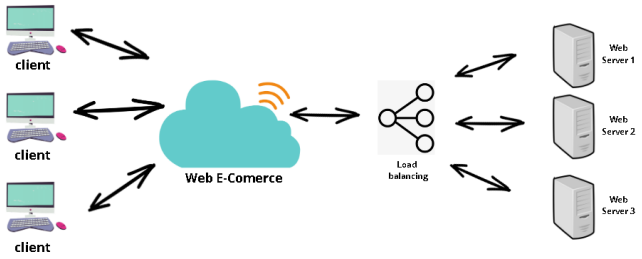
Meskipun perbandingan RR dan LC telah banyak dilakukan, kebaruan ilmiah (*novelty*) penelitian ini terletak pada analisis kinerja spesifik dalam konteks *web E-commerce* yang transaksional. Aplikasi *E-commerce* memiliki karakteristik beban yang sangat fluktuatif dan seringkali melibatkan durasi koneksi yang bervariasi (ada yang hanya melihat-lihat, ada yang melakukan transaksi kompleks). Terdapat kesenjangan (*gap*) dalam literatur yang membandingkan secara langsung dampak kedua metode ini terhadap utilisasi sumber daya *server* yang paling krusial, yaitu beban CPU, dalam skenario simulasi *E-commerce* tersebut.

Berdasarkan latar belakang dan kesenjangan literatur yang telah diuraikan, tujuan penelitian ini adalah untuk menganalisis dan membandingkan secara kuantitatif kinerja metode *load balancing Round Robin* dan *Least Connection*. Kinerja ini diukur secara spesifik berdasarkan parameter penggunaan CPU pada *web server* dalam sebuah arsitektur jaringan *E-commerce* yang disimulasikan menerima beban pengguna (*user*) yang tinggi dan bervariasi (dari 10 hingga 2000 *user*) menggunakan *JMeter*.

2. Metode

2.1 Rancangan Arsitektur Jaringan

Berdasarkan Gambar 2. Secara singkat penjelasan mengenai cara kerja dari sistem ini yaitu dimulai dari *client* yang mengirim permintaan akses ke situs *web server* dengan melalui *web E-commerce*.



Gambar 2. Rancang Bangun *Load Balancing* pada *web E-commerce*

Permintaan dari *client* akan masuk ke *Load balancing* terlebih dahulu, kemudian dari *Load balancing* permintaan *client* akan diteruskan ke beberapa *web server* yang kemudian akan dilakukan pembagian beban

terhadap beberapa *web server*. Maka *web server* akan memproses permintaan *client*. Selanjutnya akan dilakukan pengamatan pembagian beban *Load balancing* ke *web server* sudah bekerja dengan baik atau tidak.

2.2 Rancangan Arsitektur Jaringan

Pengujian yang telah dilakukan adalah membuat sebuah situs *web* sederhana dan klien akan mencoba mengakses, situs *web* yang ada pada *server*. Pengamatan dilakukan berdasarkan pada pembagian beban *web server* secara merata. Adapun pengujian yang dilakukan seperti pada Tabel I.

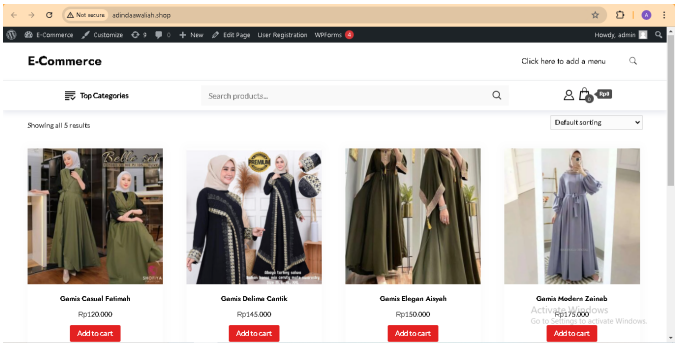
Tabel 1. Skenario *Load Balancing*

Web server	Metode	Jumlah User	Keterangan
Web E-Comerce	*Round Robin*	10 *user*	CPU
		100 *user*	
		250 *user*	
		500 *user*	
		1000 *user*	
		1500 *user*	
		2000 *user*	
	Least Connection	10 *user*	
		100 *user*	
		250 *user*	
		500 *user*	
		1000 *user*	
		1500 *user*	
		2000 *user*	

Pertama klien akan mengakses situs *web E-commerce* adindaawaliah.shop kemudian permintaan dari klien akan masuk ke *load balancer* Apache, dari *load balancer* Apache permintaan klien akan diteruskan ke beberapa *server* dengan melakukan pembagian beban terhadap *server* lain. Pada tahap selanjutnya akan dilakukan pengamatan apakah pembagian beban di *load* balancer Apache bekerja atau tidak. Dalam proses pengujian yang dilakukan pada penelitian ini ialah memasukkan dari 10, 100, 250, 500, 1000, 1500 sampai 2000 permintaan akses *user*. Pengujian tersebut meliputi hasil perhitungan CPU.

3. Hasil Dan Pembahasan

3.1 Web E-Commerce

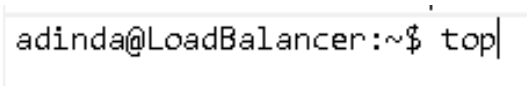


Gambar 3. Tampilan *web E-commerce*

Web E-commerce menjadi penyedia layanan untuk menerima permintaan klien, untuk menerapkan sistem *load balancing* dari metode *Round Robin* dan *Least Connection*. Tampilan *web E-commerce* dapat dilihat pada Gambar 3.

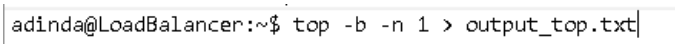
3.2 Pengujian

Pengujian dilakukan dengan menghitung beban CPU yang digunakan pada *web server* pada durasi waktu yang sama yaitu 120 (detik), ketika simulasi *user* Apache *JMeter* dijalankan, pada *console web server* menggunakan perintah "top" untuk menampilkan jumlah CPU yang terpakai secara *real-time*, dapat dilihat pada Gambar 4.



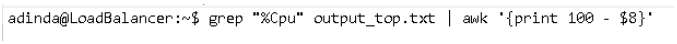
Gambar 4. Perintah Top

Untuk menyimpan *output* dari perintah top untuk analisa lebih lanjut dapat dilihat pada Gambar 5.



Gambar 5. Menyimpan data CPU ke file

Selanjutnya menghitung total penggunaan CPU dapat dilihat pada Gambar 6.



Gambar 6. Mengolah data CPU

3.3 Pengujian pada web E-Commerce tanpa Load Balancing

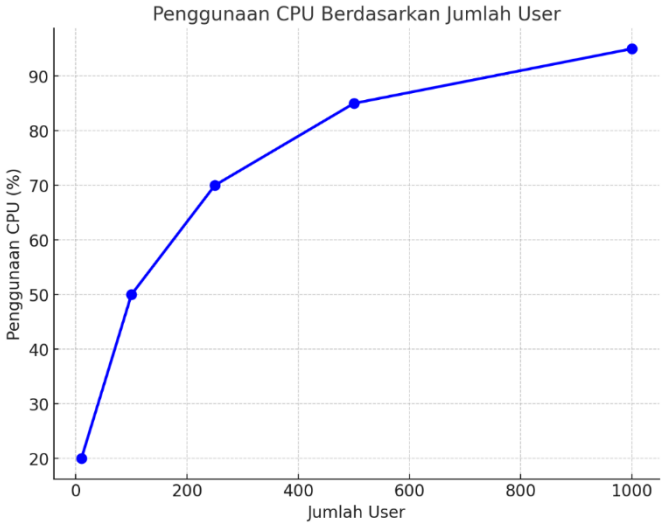
Pada pengujian ini beban ditempatkan pada satu server tanpa pembagian server lain. Hasil dapat dilihat pada Tabel II.

Tabel II Pengujian Tanpa *Load Balancing*

Jumlah user	Penggunaan CPU %
10	20
100	50
250	70
500	85
1000	95
1500	100
2000	100

Berdasarkan Tabel II dapat dilihat pembahasan dari perhitungan CPU pada Gambar 7. Berdasarkan Gambar 7 penggunaan CPU dengan jumlah *user* yang bervariasi, dapat disimpulkan bahwa peningkatan jumlah pengguna secara langsung mempengaruhi persentase penggunaan CPU di *server*. Pada awalnya, untuk jumlah pengguna kecil (10 *user*), penggunaan CPU berada pada tingkat yang rendah, yaitu 20%. Ini menandakan bahwa *server* masih memiliki banyak kapasitas pemrosesan yang tersisa untuk menangani lebih banyak pengguna. Seiring dengan peningkatan jumlah pengguna menjadi 100, 250, dan 500, penggunaan CPU meningkat tajam hingga mencapai 85%. Kenaikan ini menunjukkan bahwa *server* harus bekerja lebih keras untuk memproses lebih banyak permintaan pengguna, menyebabkan peningkatan pemanfaatan sumber daya komputasi. Ketika jumlah pengguna mencapai 1000, penggunaan CPU sudah mendekati batas maksimal, yaitu 95%, yang berarti *server* hampir mencapai kapasitas pemrosesan maksimalnya. Pada jumlah pengguna 1500 dan 2000, penggunaan CPU mencapai 100%, yang menunjukkan bahwa *server* sudah kelebihan beban. Pada titik ini, *server* tidak dapat lagi menangani tambahan beban dengan baik, sehingga dapat menyebabkan penurunan performa yang signifikan seperti

waktu respons yang lambat, *throughput* yang menurun, dan potensi terjadinya kegagalan sistem. Hal ini mengindikasikan bahwa untuk jumlah pengguna di atas 1000, sistem perlu menggunakan mekanisme distribusi beban seperti *load balancing* atau menambah kapasitas *server* untuk menjaga performa tetap stabil.



Gambar 7. Grafik CPU tanpa *load balancing*

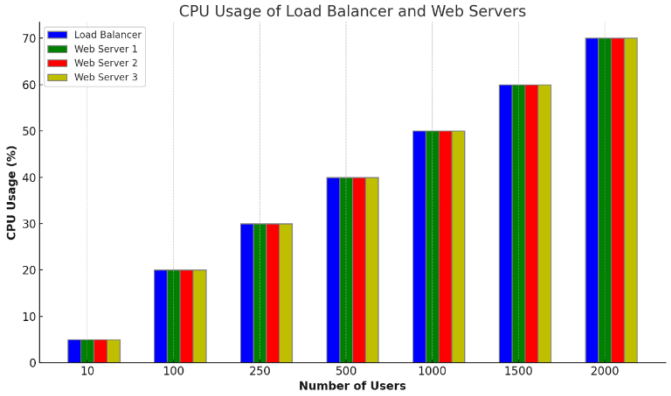
3.4 Pengujian Pada Web E-Commerce Menggunakan Metode Least Connection

Pengujian ini dilakukan dengan membagi beban kepada *server* yang memiliki jumlah koneksi paling sedikit. Hasil dapat dilihat pada Tabel III.

Tabel IV Pengujian Metode *Least Connection*

User	**CPU %**			
	Loadbalancer	**Web server 1**	**Web server 2**	**Web server 3**
10	5	5	5	5
100	20	20	20	20
250	30	30	30	30
500	40	40	40	40
1000	50	50	50	50
1500	60	60	60	60
2000	70	70	70	70

Berdasarkan Tabel IV dapat dilihat pembahasan dari perhitungan CPU pada Gambar 8.



Gambar 8. Grafik CPU metode *Least Connection*

Berdasarkan Gambar 8 data penggunaan CPU, dapat dilihat bahwa seiring dengan bertambahnya jumlah pengguna, penggunaan CPU pada *load balancer* dan ketiga *web server* meningkat secara proporsional dan merata. Pada 10 pengguna pertama, penggunaan CPU baik pada *load balancer* maupun *web server* berada pada tingkat yang sangat rendah, yakni 5%. Ini menunjukkan bahwa sistem berfungsi dengan sangat efisien untuk beban yang ringan, di mana setiap *server* menerima porsi beban yang sama. Saat jumlah pengguna meningkat menjadi 100, penggunaan CPU juga meningkat secara linear, masing-masing *server* menunjukkan peningkatan hingga 20%, lalu terus bertambah secara stabil pada 250 dan 500 pengguna, masing-masing mencapai 30% dan 40%. Pada titik ini, pembagian beban masih berjalan efisien tanpa ada *server* yang kelebihan beban, yang menunjukkan bahwa *load balancer* berhasil mendistribusikan beban dengan baik. Ketika jumlah pengguna mencapai 1000, penggunaan CPU di semua komponen sistem meningkat menjadi 50%, dan ini terus bertambah hingga mencapai 70% untuk 2000 pengguna. Kenaikan ini konsisten dan merata, yang menunjukkan bahwa *load balancer* mampu menjaga distribusi beban yang seimbang di antara *server-server*. Tidak ada indikasi adanya ketidakmerataan distribusi beban yang besar, yang menunjukkan bahwa metode *load balancing* yang digunakan berjalan efektif hingga skala pengguna yang lebih besar.

3.5 Pengujian Pada Web E-Commerce Menggunakan Metode Round Robin

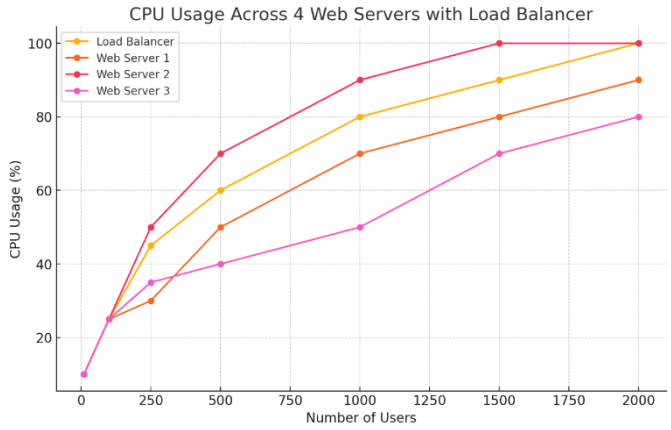
Pengujian menggunakan metode ini dilakukan dengan membagi beban secara berurutan kepada 4 *web server*. Hasil dapat dilihat pada Tabel IV.

Tabel IV Pengujian Metode *Round Robin*

User	**CPU %**			
	Loadbalancer	**Web server 1**	**Web server 2**	**Web server 3**
10	10	10	10	10
100	25	25	25	25
250	45	30	50	35
500	60	50	70	40
1000	80	70	90	50
1500	90	80	100	70
2000	100	90	100	80

Berdasarkan Tabel IV dapat dilihat pembahasan dari perhitungan CPU pada Gambar 9. Berdasarkan Gambar 9 Analisis dari tabel penggunaan CPU menunjukkan bagaimana beban pengguna berdampak pada kinerja *load balancer* dan tiga *web server*. Pada jumlah pengguna yang rendah, yaitu 10 dan 100, penggunaan CPU pada semua sistem cukup seimbang, masing-masing sekitar 10% dan 25%. Ini menunjukkan bahwa baik *load balancer* maupun *web server* dapat dengan mudah menangani beban rendah tanpa masalah performa. Namun, seiring dengan meningkatnya jumlah pengguna hingga 250, terdapat peningkatan penggunaan CPU yang signifikan, terutama pada *web server* 2 yang mencapai 50%, sedangkan *web server* 1 dan 3 tetap relatif stabil. Saat beban meningkat menjadi 500 pengguna, penggunaan CPU di seluruh sistem mulai menunjukkan variasi yang lebih jelas. *Load balancer* mencatat penggunaan CPU sebesar 60%, sementara *web server* 3 mengalami lonjakan hingga 70%, menandakan bahwa *server* ini mungkin sedang menghadapi beban yang lebih berat dibandingkan yang lain. Pada tingkat pengguna yang lebih tinggi, yaitu 1000 dan 1500, penggunaan CPU semakin mendekati batas maksimum, dengan *load balancer* mencapai 80% dan 90%, serta *server* 3 memuncak di 100% pada 1500 pengguna. Ini menandakan bahwa *server*

tersebut sudah berada di ambang kapasitas, berpotensi menyebabkan penurunan performa atau keterlambatan respons. Ketika jumlah pengguna mencapai 2000, seluruh sistem mengalami penggunaan CPU maksimum, dengan *load balancer* dan beberapa *server* mendekati atau mencapai 100%. Hal ini mengindikasikan bahwa baik *load balancer* maupun *server* tidak dapat lagi menangani beban tambahan dengan efektif, yang dapat berdampak buruk pada pengalaman pengguna, termasuk waktu respons yang lebih lambat atau bahkan kegagalan dalam melayani permintaan.



Gambar 9. Grafik CPU metode *Round Robin*

3.6 Perbandingan Setelah dan Sebelum Penggunaan Metode Round Robin & Least Connection

Perbandingan hasil sebelum dan sesudah penggunaan *load balancing* dengan metode *Round Robin* dan *Least Connection* seperti pada Tabel V. Dari hasil Tabel V terlihat bahwa saat tidak menggunakan metode *load balancing*, seluruh beban ditangani oleh satu *load balancer*. Hal ini menyebabkan peningkatan yang sangat cepat dalam penggunaan CPU seiring bertambahnya pengguna. Pada 1000 pengguna, *load balancer* hampir mencapai kapasitas penuh (95%), dan pada 1500 hingga 2000 pengguna, *load balancer* mencapai 100% penggunaan CPU. Ini menunjukkan bahwa tanpa *load balancing*, sistem tidak mampu menangani beban yang lebih besar secara efisien, menyebabkan potensi kegagalan sistem. Dengan metode *Least Connection*, beban didistribusikan secara merata ke beberapa *web server* berdasarkan jumlah koneksi yang aktif. Penggunaan CPU di antara *server* tetap seimbang dengan rata-rata sekitar 275% di setiap *server*. Bahkan dengan 2000 pengguna, *load balancer* dan *web server* hanya mencapai 70% penggunaan CPU. Ini menunjukkan bahwa *Least Connection* sangat efektif dalam menghindari penumpukan beban pada satu *server*, menghasilkan distribusi yang stabil dan efisien, serta menjaga penggunaan CPU dalam batas yang aman bahkan pada beban tinggi. Pada metode *Round Robin* menunjukkan performa yang cukup baik dalam membagi beban di antara *web server*, namun penggunaan CPU sedikit lebih bervariasi dibandingkan *Least Connection*. Pada 2000 pengguna, beberapa *server* mencapai penggunaan CPU yang sangat tinggi, dengan *server* 2 dan 3 mencapai 100%, sementara *server* lainnya lebih rendah. Rata-rata penggunaan CPU adalah sekitar 380%, lebih tinggi dari *Least Connection* tetapi lebih rendah dari pada tanpa metode *load balancing*. Meskipun metode ini masih efisien dalam membagi beban, perbedaan dalam penggunaan CPU di antara *server* menunjukkan bahwa *Round Robin* mungkin kurang optimal dalam menangani skenario di mana *server* memiliki kemampuan berbeda atau beban yang tidak merata.

Tabel V Perbandingan *Load Balancing*

Tanpa metode				
User	***Loadbalancer* %**	**Web server 1 %**	**Web server 2 %**	**Web server 3 %**
10	20	-	-	-
100	50	-	-	-
250	70	-	-	-
500	85	-	-	-
1000	95	-	-	-
1500	100	-	-	-
2000	100	-	-	-
Rata-rata	**520 %**			
Least connection				
User	***Loadbalancer* %**	**Web server 1 %**	**Web server 2 %**	**Web server 3 %**
10	5	5	5	5
100	20	20	20	20
250	30	30	30	30
500	40	40	40	40
1000	50	50	50	50
1500	60	60	60	60
2000	70	70	70	70
Rata-rata	275 %	275 %	275 %	275 %
	275 %			
Round robin				
User	***Loadbalancer* %**	**Web server 1 %**	**Web server 2 %**	**Web server 3 %**
10	10	10	10	10
100	25	25	25	25
250	45	30	50	35
500	60	50	70	40
1000	80	70	90	50
1500	90	80	100	70
2000	100	90	100	80
Rata-rata	410 %	355 %	445 %	310 %
	380 %			

Rata-rata persentase CPU (%): *Load balancing* tanpa metode: 520%, *Load balancing* metode *Least Connection*: 275%, dan *Load balancing* metode *Round Robin*: 380%. *Load balancing* dengan metode *Least Connection* terbukti sebagai metode yang paling efisien, dengan distribusi beban yang merata dan penggunaan CPU yang stabil, bahkan pada jumlah pengguna yang tinggi. Pada metode *Round Robin* menujukkan performa yang baik, namun kurang konsisten dalam mendistribusikan beban dibandingkan dengan metode *Least Connection*. Sedangkan tanpa *load balancing*, sistem menjadi sangat tidak efisien, dengan satu *server* yang kelebihan beban saat jumlah pengguna meningkat, yang dapat menyebabkan kegagalan sistem di bawah beban berat.

4. Kesimpulan

Berdasarkan hasil dan pembahasan pada penelitian yang telah dilakukan, maka dapat disimpulkan bahwa berdasarkan hasil pengujian *Load balancing* menggunakan aplikasi *JMeter* sebagai pemberi beban *user* dengan jumlah 10, 100, 250, 500, 1000, 1500, 2000 *user* dalam satu waktu memiliki rata-rata nilai CPU 520% pada pengujian tanpa metode, 275% pada pengujian menggunakan metode *Least Connection* serta 380% pada pengujian menggunakan metode *Round Robin*. *Load* balancing dengan metode *Least Connection* terbukti sebagai metode yang paling efisien, dengan distribusi beban yang merata dan penggunaan CPU

yang stabil, bahkan pada jumlah pengguna yang tinggi. Pada metode *Round Robin* menunjukkan performa yang baik, namun kurang konsisten dalam mendistribusikan beban dibandingkan dengan metode *Least Connection*. Sedangkan Tanpa *load balancing*, sistem menjadi sangat tidak efisien, dengan satu *server* yang kelebihan beban saat jumlah pengguna meningkat, yang dapat menyebabkan kegagalan sistem di bawah beban berat.

Referensi

[1] Universitas Internasional Batam, M. D. Firmansyah, H. Herman, dan Universitas Internasional Batam, "Perancangan Web E- Commerce Berbasis Website pada Toko Ida Shoes," JOINT, vol. 4, no. 1, hlm. 361–372, Mei 2023, doi: 10.37253/joint.v4i1.6330.

[2] M. Arman, N. Wijaya, dan H. Irsyad, "Analisis Kinerja Web Server Menggunakan Algoritma Round Robin dan Least Connection," SISFOKOM, vol. 8, no. 2, hlm. 55–59, Mar 2017, doi: 10.32736/sisfokom.v6i1.143.

[3] M. F. Darmawan dan S. Risnanto, "Implementasi Failover Gateway Recursive Dan Load Balancing Menggunakan Metode Per Connection Classifier," infotronik, vol. 8, no. 2, hlm. 56, Des 2023, doi: 10.32897/infotronik.2023.8.2.1887.

[4] I. Adiyasa dan R. Munadi, "Perancangan Dan Implementasi Web Dengan Load Balancer Untuk Membantu Manajemen Trans Metro Bandung".

[5] H. Nasser dan T. Witono, "Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancng Menggunakan Opnet Modeler," vol. 12, no. 1, 2016.

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

[7] A. Fadila, M. Nasir, dan S. Safriadi, "Implementasi Sistem Load Balancing Web Server Pada Jaringan public Cloud Computing Menggunakan Least Connection," jaise, vol. 3, no. 2, hlm. 50, Okt 2023, doi: 10.30811/jaise.v3i2.4578.

[8] M. A. Waluyo, F. Antony, dan C. Setiawan, "Implementasi Load Balancing Web Server Dengan Haproxy Menggunakan Algoritma Round Robin," Jinig, vol. 1, no. 1, hlm. 46–52, Jul 2023, doi: 10.36982/jinig.v1i1.3074.

[9] M. F. Zulkarnaen dan M. I. Isnaini, "Implementasi Load Balancing Dengan Metode Equal Cost Multi-Path," JIRE, vol. 1, no. 1, hlm. 13, Mei 2018, doi: 10.36595/jire.v1i1.26.

[10] H. Setiawan, "Instalasi Serta Konfigurasi HAproxy Sebagai Load Balancing Web Server," vol. 2, no. 1, 2023.

[11] G. Triono, "Implementasi Load Balancing Dengan Menggunakan Algoritma Round Robin Pada Kasus Pendaftaran Siswa Baru Sekolah Menengah Pertama Labschool Unesa Surabaya," 2015.

[12] D. A. R. Ananta, P. H. Trisnawan, dan K. Amron, "Implementasi Load Balancing Web Server pada Software Defined Networking menggunakan Metode K-Nearest Neighbor (K-NN)".

[13] S. A. Rahman dan T. Y. Hadiwandra, "Perbandingan Algoritma Weighted Least Connection dan Weighted Round Robin pada Load Balancing Berbasis Docker Swarm," ISI, vol. 8, no. 2, hlm. 228, Nov 2023, doi: 10.35314/isi.v8i2.3395.