Journal of System and Computer Engineering (JSCE) ISSN (online): 2723-1240
Vol. 5 No. 1 Januari 2024

Optimalisasi *Throughput* Pada Penerapan *Load Balancing* Dalam Jaringan *Cloud* Menggunakan *Round Robin* dan *Least Connection*

Moh. Erkamim[1], Titin Prihatin[2], Sandra Dewi Saraswati[3], Mursalim Tonggiroh[4]
[1]Sistem Informasi Kota Cerdas, Universitas Tunas Pembangunan Surakarta
[2,3]Informatika, Fakultas Teknologi Informasi, Universitas Nusa Mandiri
[4]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Yapis Papua
Email: [1]erkamim@lecture.utp.ac.id, [2]titin.tpn@nusamandiri.ac.id,
[3]sandra.sww@nusamandiri.ac.id, [4]mursalim.t@gmail.com

Abstrak

Seiring dengan meningkatnya kompleksitas infrastruktur *cloud*, salah satu tantangan utama yang dihadapi adalah bagaimana mengoptimalkan *throughput* dalam jaringan *cloud*. *Throughput* yang optimal menjadi krusial untuk menjamin kinerja aplikasi dan layanan yang dihosting di *cloud*. Penggunaan teknik *load balancing* dapat mengoptimalkan *throughput*, namun diperlukan analisis dan pengujian yang seksama mengenai pendekatan *load balancing* yang tepat dalam meningkatkan kinerja *throughput*. Penelitian ini bertujuan untuk menganalisis efektivitas strategi *load balancing* dengan *Round Robin* dan *Least Connection* dalam meningkatkan *throughput* dalam jaringan *cloud*. Melalui pengujian empiris, kajian ini membandingkan kedua metode berdasarkan *throughput* yang dihasilkan pada berbagai tingkat koneksi, mulai dari 1000/600 hingga 5000/1000. Hasil pengujian menunjukkan bahwa kedua strategi memberikan performa yang serupa pada tingkat koneksi rendah, dengan *Round Robin* mencatat throughput sedikit lebih tinggi pada 1000/600. Namun, perbedaan menjadi hampir tidak terlihat seiring peningkatan jumlah koneksi. Pada 2000/700 hingga 5000/1000, throughput yang dihasilkan oleh kedua strategi menunjukkan perbedaan yang marginal, dimana *Least Connection* sedikit unggul di tingkat koneksi tertinggi. Temuan ini menandakan bahwa baik *Round Robin* maupun *Least Connection* dapat diaplikasikan secara efektif untuk meningkatkan throughput dalam lingkungan *cloud*, dengan pemilihan strategi spesifik bergantung pada karakteristik beban kerja dan permintaan layanan. Penelitian ini memberikan wawasan berharga bagi akademisi maupun para praktisi TI yang ingin mengoptimalkan jaringan *cloud* untuk mencapai kinerja yang maksimal.

Kata Kunci: *throughput, load balancing, round robin, least connection*

Abstract

As the complexity of cloud infrastructure increases, one of the main challenges faced is how to optimize throughput in cloud networks. Optimal throughput is crucial to guarantee the performance of applications and services hosted in the cloud. Using load balancing techniques can optimize throughput, but careful analysis and testing is needed regarding the appropriate load balancing approach to improve throughput performance. This research aims to analyze the effectiveness of load balancing strategies with Round Robin and Least Connection in increasing throughput in cloud networks. Through empirical testing, this study compares the two methods based on the resulting throughput at various connection levels, ranging from 1000/600 to 5000/1000. Test results show that both strategies provide similar performance at low connection rates, with Round Robin recording slightly

higher throughput at 1000/600. However, the difference becomes almost imperceptible as the number of connections increases. At 2000/700 to 5000/1000, the throughput generated by both strategies shows marginal differences, with Least Connection slightly superior at the highest connection level. These findings indicate that both Round Robin and Least Connection can be applied effectively to increase throughput in cloud environments, with the selection of specific strategies depending on workload characteristics and service requests. This research provides valuable insights for academics and IT practitioners who want to optimize cloud networks to achieve maximum performance.

Keywords: *throughput, load balancing, round robin, least connection*

1. Pendahuluan

Dalam era digital yang terus berkembang, penggunaan teknologi *cloud computing* telah menjadi salah satu solusi utama bagi organisasi untuk menyimpan, mengelola, dan menyajikan data mereka. Seiring dengan meningkatnya kompleksitas infrastruktur *cloud*, salah satu tantangan utama yang dihadapi adalah bagaimana mengoptimalkan *throughput* dalam jaringan *cloud*. *Throughput* yang optimal menjadi krusial untuk menjamin kinerja aplikasi dan layanan yang dihosting di *cloud*. Efisiensi pengelolaan sumber daya jaringan ini sangat bergantung pada distribusi beban kerja yang optimal antara server yang tersedia (Golightly et al., 2022). *Load balancing* merupakan komponen yang penting dalam arsitektur komputasi awan yang menjamin distribusi permintaan yang seimbang ke berbagai Server, untuk menghindari *overloading* dan memaksimalkan *throughput* (Afzal & Kavitha, 2019). *Load balancing* dapat dikatakan sebagai mekanisme yang digunakan dalam sistem komputer dan jaringan untuk mendistribusikan beban kerja secara merata di antara berbagai sumber daya komputasi, seperti server atau node dalam jaringan (Hakim et al., 2019). Tujuan utama dari *load balancing* adalah untuk meningkatkan kinerja, ketersediaan, dan kehandalan sistem dengan mencegah terjadinya kelebihan beban pada satu atau beberapa sumber daya, sementara sumber daya lainnya mungkin berada di bawah beban minimum atau tidak digunakan sepenuhnya (Malau, 2022). *Round Robin* dan *Least Connection* adalah dua strategi *load balancing* yang paling umum digunakan dalam jaringan *cloud*, di mana keduanya memiliki keunikan dalam penanganan permintaan layanan. *Round Robin* mendistribusikan permintaan secara merata berdasarkan urutan, tanpa mempertimbangkan beban sebenarnya pada Server, sedangkan *Least Connection* memprioritaskan Server dengan jumlah koneksi aktif terendah, potensial memberikan respons lebih cepat pada permintaan baru (Wira Harjanti et al., 2022). Dalam konteks yang semakin menuntut kecepatan dan efisiensi, penting untuk mengevaluasi kinerja kedua strategi ini secara komparatif untuk memahami kondisi di mana masing-masing strategi lebih efektif.

Terdapat beberapa penelitian terdahulu yang memanfaatkan pendekatan *load balancing* dalam meningkatkan efisiensi pada jaringan *cloud*. Penelitian pertama mengenai penerapan teknik *load balancing* pada lingkungan *cloud* untuk pengikatan performa web server (Riskiono & Darwis, 2020). Penelitian ini memperlihatkan bahwa teknik *load balancing* memberikan performa yang lebih cepat dalam response time dan dapat mendistribusikan secara merata beban kerjanya. Penelitian ini selaras dengan penelitian mengenai pengembangan model *load balancing* untuk meningkatkan kehandalan dan kinerja sistem *cloud* (Riskiono & Pasha, 2020). Penelitian tersebut

menunjukkan bahwa teknik *load balancing* dapat mengurai distribusi beban kerja jaringan sehingga meningkatkan *response time*. Penelitian lain, mengkaji bahwa tidak hanya response time yang dapat ditingkatkan melalui terkini *load balancing* namun juga berpengaruh pada *throughput* (Wira Harjanti et al., 2022). Hal ini didukung juga oleh penelitian lainnya yang telah melakukan pengujian secara empiris melalui serangkaian yang mengukur hasil rata-rata *throughput* pada berbagai tingkat koneksi (Al-Said Ahmad & Andras, 2019). Penelitian tersebut berusaha untuk menentukan strategi mana yang menawarkan performa terbaik dalam lingkungan *cloud* yang berbeda. Hal ini memberikan indikasi bahwa pemilihan strategi *load balancing* harus dipertimbangkan berdasarkan skenario penggunaan spesifik dan distribusi beban kerja yang diperkirakan (Nugrahadi et al., 2019). Analisis ini berkontribusi pada literatur yang ada dengan menyediakan wawasan baru tentang implementasi strategi *load balancing* dalam jaringan *cloud*. Hasil dari penelitian ini diharapkan dapat dijadikan referensi bagi para praktisi TI dalam merancang dan mengoptimalkan infrastruktur *cloud* untuk mendapatkan kinerja yang lebih baik dan lebih efisien.

Pengelolaan beban kerja secara efisien dalam jaringan *cloud* merupakan tantangan kritis dalam memastikan performa dan keandalan layanan yang tinggi (Riskiono & Pasha, 2020). *Load balancing* adalah teknik kunci dalam menangani tantangan ini, memungkinkan distribusi permintaan layanan secara dinamis di antara berbagai Server, sehingga mengoptimalkan pemanfaatan sumber daya, meminimalkan waktu tanggap, dan memaksimalkan *throughput* (Hendrana & Suartana, 2022). Strategi *load balancing* seperti *Round Robin* dan *Least Connection* telah banyak dipelajari dalam literatur akademik, masing-masing dengan kelebihan dan kekurangan mereka. *Round Robin* mendistribusikan permintaan secara bergiliran yang distribusi beban yang masuk secara merata (Hidayat et al., 2020). Sementara itu, *Least Connection* memprioritaskan Server dengan jumlah koneksi aktif terendah, potensial menawarkan performa yang lebih baik dalam skenario beban kerja yang sangat bervariasi (Sujarwo et al., 2023). Dalam konteks tersebut, penelitian ini berfokus pada analisis komparatif antara *Round Robin* dan *Least Connection* dalam upaya untuk mengoptimalkan *throughput*, yang merupakan indikator penting dari efisiensi jaringan dalam *cloud* computing. Melalui eksperimen yang dirancang secara sistematis dan implementasi dalam lingkungan *cloud* yang terkontrol, penelitian ini mengevaluasi performa kedua strategi di berbagai tingkat koneksi, memberikan pemahaman yang lebih mendalam tentang bagaimana masing-masing strategi dapat berkontribusi terhadap peningkatan *throughput*. Penelitian ini juga bertujuan untuk mengisi kesenjangan dalam literatur dengan menyediakan data empiris yang dapat dijadikan referensi bagi praktisi TI dalam memilih strategi *load balancing* yang paling sesuai dengan kebutuhan spesifik infrastruktur *cloud* mereka. Dengan berfokus pada *throughput* sebagai metrik kinerja utama, temuan dari studi ini diharapkan dapat memberikan kontribusi pada pengambilan keputusan yang lebih tepat dan efektif terkait optimasi jaringan *cloud*.

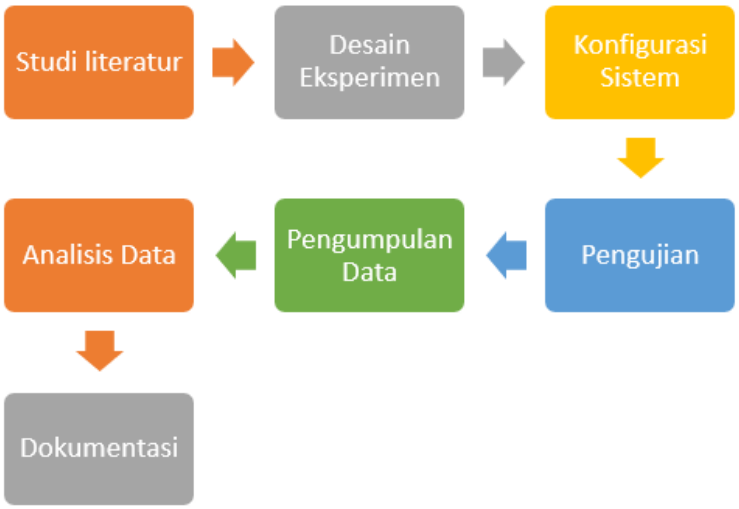
Penelitian ini bertujuan untuk memberikan analisis komparatif dari strategi *Round Robin* dan *Least Connection* dalam konteks peningkatan *throughput* dalam jaringan *cloud*. Penelitian ini menguji kinerja kedua strategi ini pada berbagai tingkat koneksi dan mengukur *throughput* yang dihasilkan. Dengan berfokus pada *throughput* sebagai metrik kinerja utama, temuan dari studi ini diharapkan dapat memberikan kontribusi pada pengambilan keputusan yang lebih tepat dan efektif terkait optimasi jaringan *cloud*. Penelitian ini bertujuan untuk memberikan analisis komparatif dari strategi *Round Robin* dan *Least Connection* dalam konteks peningkatan *throughput*

dalam jaringan *cloud*. Kami menguji kinerja kedua strategi ini pada berbagai tingkat koneksi dan mengukur *throughput* yang dihasilkan.

2. Metodologi Penelitian

2.1 Tahapan Penelitian

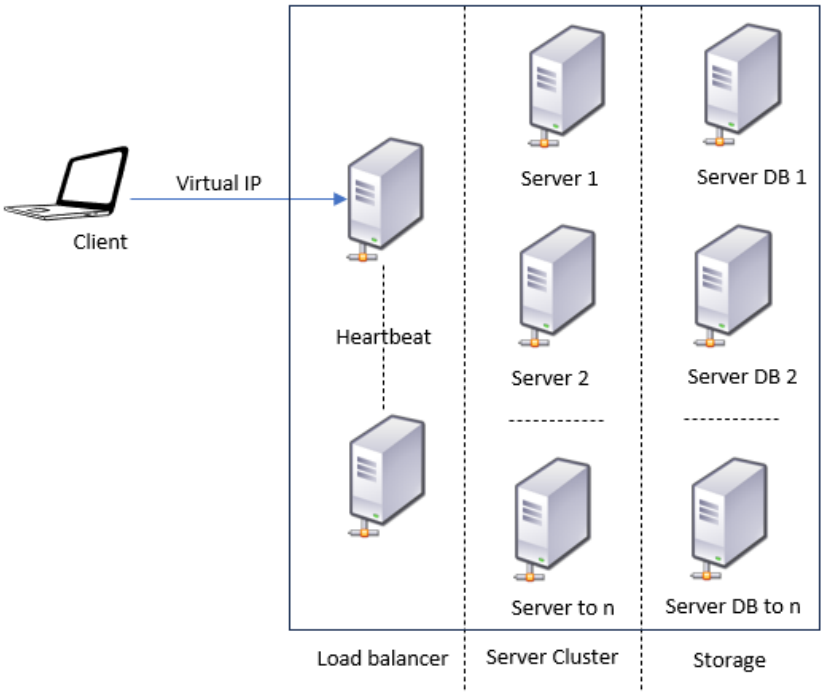
Tahapan penelitian ini dirancang untuk secara sistematis menguji dan membandingkan pengaruh strategi *load balancing* dengan *Round Robin* dan *Least Connection* terhadap *throughput* dalam jaringan *cloud*. Dalam penelitian ini, model jaringan *cloud* dibangun di atas mesin virtual. Penelitian ini dibagi menjadi beberapa tahap utama seperti yang diperlihatkan pada Gambar 1, yang masing-masing dirinci sebagai berikut:



Gambar 1. Tahapan dalam penelitian

2.2 Persiapan Lingkungan Penelitian

Load balancing merupakan suatu mekanisme yang digunakan dalam sistem komputer dan jaringan untuk mendistribusikan beban kerja atau lalu lintas secara merata di antara beberapa sumber daya atau server (Azmi et al., 2022). Dengan adanya *load balancing*, sistem dapat menangani lebih efisien permintaan pengguna atau pekerjaan yang masuk, mengoptimalkan penggunaan sumber daya, dan mengurangi risiko kegagalan akibat beban yang tidak seimbang (Saiputra & Mukti, 2023). Pada tahap ini, akan disiapkan lingkungan virtual yang memodelkan infrastruktur *cloud* dengan menggunakan mesin virtual. Lingkungan ini termasuk konfigurasi server penyeimbang beban, server web, dan basis data. Setiap komponen dikonfigurasi dengan spesifikasi teknis yang telah ditentukan untuk menjamin bahwa lingkungan penelitian konsisten dengan model yang ada. Arsitektur yang digunakan divisualisasikan pada Gambar 2.



Gambar 2. Arsitektur Server *Load Balancer*

Gambar 2 menggambarkan arsitektur *load balancing* dalam jaringan komputer. Client terhubung ke sistem melalui Virtual IP, yang merupakan alamat IP yang digunakan untuk menyembunyikan keberadaan dan jumlah Server yang sebenarnya dalam cluster. *Load balancer* bertugas mendistribusikan beban kerja yang masuk dari klien ke berbagai Server yang tersedia dalam cluster Server (Harefa et al., 2021). Komunikasi '*heartbeat*' terlihat digambarkan sebagai koneksi titik putus-putus antara *load balancer* dan Server, mengindikasikan mekanisme pemantauan kesehatan Server secara berkala. Server dalam cluster ini diberi label dari 'Server 1' hingga 'Server ke n', menandakan skalabilitas dan kemampuan untuk menambah Server tambahan sesuai kebutuhan. Di sisi kanan cluster Server, terdapat Server basis data yang juga diberi label dari 'Server DB 1' hingga 'Server DB ke n', menunjukkan infrastruktur penyimpanan data yang terdistribusi. Keseluruhan sistem ini bertujuan untuk memastikan pengalokasian sumber daya yang efisien dan ketersediaan layanan yang tinggi untuk klien.

2.3 Pemilihan dan Konfigurasi Alat Uji

Alat uji yang dipilih untuk penelitian ini harus mampu mensimulasikan permintaan klien ke Server dalam berbagai tingkat beban. Parameter uji yang digunakan yaitu *throughput*. *Throughput* pada *load balancing* merujuk pada jumlah data atau transaksi yang dapat diproses oleh sistem atau jaringan dalam suatu periode waktu tertentu (Nuraini, 2022). Secara sederhana, *throughput* mengukur seberapa efisien dan cepat suatu jaringan dapat menangani lalu lintas atau permintaan dari pengguna (Rafli et al., 2022). Pada penelitian ini digunakan sebuah alat uji yang dapat mengirimkan permintaan HTTP ke Server penyeimbang beban yang kemudian mendistribusikan beban ke Server web sesuai dengan algoritma *load balancing* yang diuji.

2.4 Pengujian dan Pengambilan Data

Setelah lingkungan dan alat uji siap, kami melakukan serangkaian pengujian. Pengujian ini melibatkan simulasi beban klien dengan jumlah koneksi yang beragam, mulai dari 1000 hingga 5000 koneksi simultan. Setiap uji coba dilakukan secara terpisah untuk strategi *Round Robin* dan *Least Connection*, dengan pemantauan *throughput* sebagai indikator kinerja utama. Pengujian dilakukan dengan mensimulasikan permintaan klien pada berbagai tingkat koneksi, mulai dari 1000/600 hingga 5000/1000. Tingkat koneksi tersebut mencerminkan jumlah permintaan simultan dan sesi aktif yang ditangani oleh Server. Proses ini memastikan pengujian kinerja *load balancing* dalam berbagai skenario beban kerja.

2.5 Pengumpulan Data

Data *throughput* dikumpulkan dari sistem penyeimbang beban menggunakan alat monitoring kinerja yang telah terintegrasi. Pengumpulan data ini mencakup waktu respons dan jumlah data yang berhasil diproses per satuan waktu.

2.6 Analisis Data dan Dokumentasi Hasil

Hasil pengujian dianalisis untuk menentukan pengaruh dari masing-masing strategi *load balancing* terhadap *throughput* jaringan *cloud*. Perbandingan dilakukan berdasarkan rata-rata *throughput* yang dihasilkan dari masing-masing metode pada tingkat koneksi yang berbeda. Kemudian dilanjutkan dengan dokumentasi dari hasil analisis yang didapatkan untuk selanjutnya dituliskan ke dalam artikel.

3. Hasil Dan Pembahasan

Untuk melakukan pengujian *load balancing* dengan menerapkan pendekatan *Least Connection* dan *Round Robin* maka terlebih dahulu dilakukan perancangan arsitekturnya. Fungsi utama dari *load balancing* adalah untuk mencegah terjadinya ketidakseimbangan beban pada jaringan komputer, sehingga meningkatkan kinerja dan ketersediaan sistem secara keseluruhan. Arsitektur tersebut diterapkan pada lingkungan virtual yang memodelkan infrastruktur *cloud* dengan menggunakan mesin virtual. Lingkungan ini termasuk konfigurasi server penyeimbang beban, server web, dan basis data. Setelah merancang dan menerapkan teknik *load balancing* menggunakan algoritma *Least Connection* dan *Round Robin* di lingkungan simulasi *cloud*, kemudian beralih ke fase pengujian. Pada penelitian ini dikembangkan sejumlah skenario beban kerja yang beragam untuk menguji dan membandingkan kinerja kedua metode tersebut di bawah berbagai kondisi. Pengujian *throughput* pada server dengan teknik *load balancing* menggunakan algoritma *Round Robin* dan *Least Connection* yang dilakukan sebanyak 5 kali. Hal ini dilakukan karena untuk mengevaluasi konsistensi dan stabilitas performa sistem dengan tetap mempertimbangkan berbagai macam faktor yang terkait, sehingga kebenaran hasil pengujian *throughput* tetap dapat dipertanggung jawabkan. Maka, hasil pengujian yang telah dilakukan dapat mewakili kinerja masing-masing algoritma *Round Robin* dan *Least Connection* yang dapat dijadikan acuan untuk penilaian. Hasil pengujian terhadap *throughput* pada *load balancing* menggunakan pendekatan *Round Robin* tersaji pada Tabel 3.

Tabel 3. Hasil pengujian *throughput* (KB/s) Server dengan *Round Robin*

Response time (ms) *Load Balancing* dengan *Round Robin*							
No	Tingkat Koneksi	Pengujian					Hasil Rata-rata Pengujian
		ke-1	ke-2	ke-3	ke-4	ke-5	
1	1000/600	306.5	306.6	306.3	306.7	306.1	306.44
2	2000/700	357.7	357.8	357.8	357.8	357.2	357.66
3	3000/800	408.8	409	408.9	409	408.6	408.86
4	4000/900	460.2	460.2	460.2	460.2	460	460.16
5	5000/1000	511	511.2	511.3	511.3	511.2	511.2

Pada Tabel 3 ini, menunjukkan data *throughput* dalam *kilobyte* per detik (KB/s) dari sistem *load balancing* yang menggunakan algoritma *Round Robin*. Terdapat lima tingkat koneksi yang berbeda, mulai dari 1000/600 hingga 5000/1000. Untuk setiap tingkat koneksi, dilakukan lima kali pengujian (ke-1 hingga ke-5) untuk menentukan *throughput*. Hasil pengujian menunjukkan peningkatan *throughput* yang konsisten seiring dengan peningkatan tingkat koneksi, dimulai dari rata-rata 306.44 KB/s pada tingkat koneksi terendah hingga 511.2 KB/s pada tingkat koneksi tertinggi. Variabilitas *throughput* antar pengujian pada setiap tingkat koneksi tergolong rendah, menunjukkan bahwa sistem *load balancing* memberikan kinerja yang stabil. Rata-rata hasil pengujian pada setiap tingkat koneksi memberikan gambaran umum tentang performa sistem yang tampaknya meningkat proporsional dengan jumlah koneksi yang diatur.

Selanjutnya, dengan skenario yang sama akan diujikan pada arsitektur *load balancing* menggunakan pendekatan *Least Connection*. Hasil pengujian terhadap *throughput* pada *load balancing* menggunakan pendekatan *Round Robin* tersaji pada Tabel 4.

Tabel 4. Hasil pengujian *throughput* (KB/s) Server dengan *Least Connection*

Response time (ms) *Load Balancing* dengan *Least Connection*							
No	Tingkat Koneksi	Pengujian					Hasil Rata-rata Pengujian
		ke-1	ke-2	ke-3	ke-4	ke-5	
1	1000/600	306.1	306.2	306.1	306.2	306.3	306.18
2	2000/700	357.9	357.8	357.6	357.9	357.9	357.82
3	3000/800	409.1	408.9	408.8	409.1	409.1	409
4	4000/900	460.1	460.3	460.1	460.4	460.3	460.24
5	5000/1000	511.1	511.3	511.4	511.4	511.3	511.3

Pada Tabel 4, memaparkan hasil *throughput* dalam kilobyte per detik (KB/s) dari sistem *load balancing* yang menggunakan strategi *Least Connection*. Data mencakup lima tingkat koneksi yang berbeda, mulai dari 1000/600 hingga 5000/1000, dengan lima pengujian dilakukan pada masing-masing tingkat. Hasil pengujian menunjukkan peningkatan *throughput* yang konsisten dengan peningkatan tingkat koneksi. Pada tingkat koneksi terendah, *throughput* rata-rata adalah 306.18 KB/s, sementara pada tingkat koneksi tertinggi, nilai rata-rata meningkat menjadi 511.3 KB/s.

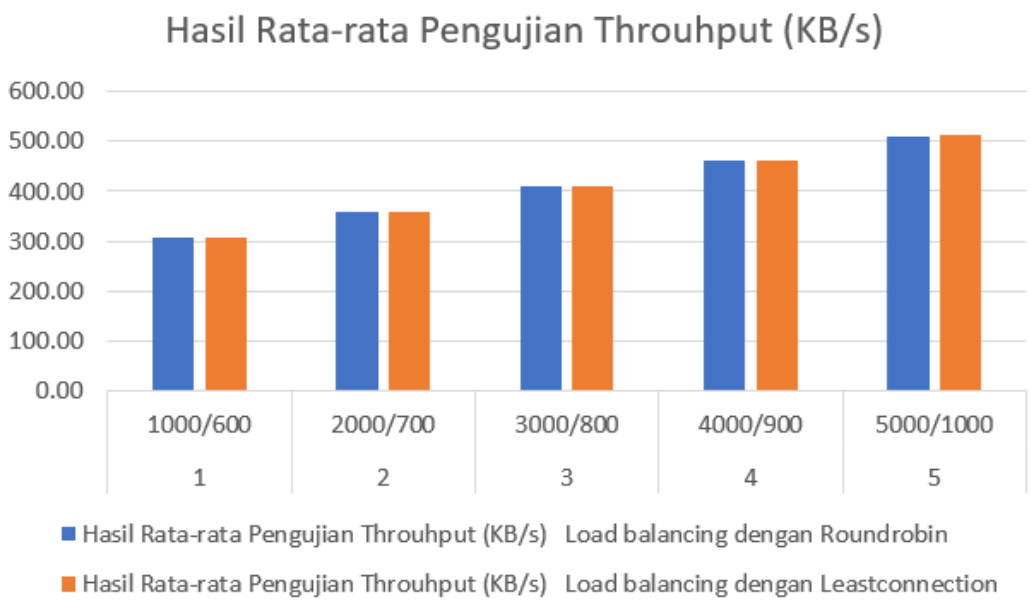
Fluktuasi antara hasil pengujian untuk setiap tingkat koneksi sangat minim, menandakan bahwa performa *load balancing* cukup stabil. Data ini menunjukkan bahwa sistem mampu menangani lebih banyak beban kerja dengan efisiensi yang relatif konstan, yang terlihat dari peningkatan linear *throughput* seiring dengan peningkatan tingkat koneksi. Rata-rata hasil pengujian memberikan indikasi kinerja yang konsisten pada setiap tingkat koneksi, yang mencerminkan efektivitas metode *Least Connection* dalam skenario yang diuji.

Selanjutnya, hasil pengujian dari kedua metode tersebut akan dibandingkan untuk mengetahui kinerja dari masing-masing metode *load balancing* tersebut. Perbandingan hasil rata-rata pengujian *throughput* dalam kilobyte per detik (KB/s) untuk dua model *load balancing*, yaitu dengan *Round Robin* dan *Least Connection*, pada lima tingkat koneksi yang berbeda tersaji Pada Tabel 5.

Tabel 5. Hasil rata-rata pengujian waktu respon (ms) Server pada kedua model

No	Tingkat Koneksi	Hasil Rata-rata Pengujian *throughput* (KB/s)	
		Load Balancing dengan *Round Robin*	*Load Balancing* dengan *Least Connection*
1	1000/600	306.44	306.18
2	2000/700	357.66	357.82
3	3000/800	408.86	409.00
4	4000/900	460.16	460.24
5	5000/1000	511.20	511.30

Pada Tabel 5, menampilkan perbandingan hasil rata-rata pengujian *throughput* dalam kilobyte per detik (KB/s) untuk dua model *load balancing*, yaitu dengan *Round Robin* dan *Least Connection*, pada lima tingkat koneksi yang berbeda. Data menunjukkan perbedaan yang sangat kecil antara kedua model untuk tingkat koneksi yang sama, yang menandakan bahwa kedua model memiliki efisiensi yang serupa dalam penanganan beban kerja. Pada tingkat koneksi 1000/600, hasil rata-rata pengujian untuk *Round Robin* adalah 306.44 KB/s, sedangkan untuk *Least Connection* adalah 306.18 KB/s. Pola serupa terlihat pada tingkat koneksi yang lebih tinggi, dengan perbedaan rata-rata pengujian yang tetap minimal, tidak melebihi 0.24 KB/s. Terakhir, untuk tingkat koneksi 5000/1000, kedua model hampir identik dengan *throughput* rata-rata 511.20 KB/s untuk *Round Robin* dan 511.30 KB/s untuk *Least Connection*. Kesimpulannya, kedua metode *load balancing* tersebut menunjukkan performa yang sebanding pada berbagai tingkat koneksi dalam hal *throughput*. Untuk memudahkan dalam menganalisis hasil pengujian, maka hasil kinerja dua metode tersebut divisualisasikan dalam bentuk grafik pada Gambar 3.



Gambar 3. Hasil nilai waktu respon antara metode Server tunggal dan *Load Balancing*

Grafik batang pada Gambar 3 menampilkan hasil rata-rata pengujian *throughput* dalam kilobyte per detik (KB/s) untuk dua metode *load balancing*: *Round Robin* dan *Least Connection*, diukur pada lima tingkat koneksi yang berbeda. Batang-batang berwarna oranye mewakili hasil untuk *Round Robin*, sedangkan biru mewakili *Least Connection*. Pada setiap tingkat koneksi, dari 1000/600 hingga 5000/1000, nilai *throughput* cenderung meningkat, yang mencerminkan kinerja sistem yang semakin baik seiring dengan peningkatan jumlah koneksi. Selisih antara hasil pengujian kedua metode sangat tipis, menunjukkan bahwa tidak ada perbedaan signifikan dalam efektivitas kedua metode tersebut dalam penanganan beban kerja. Ini terlihat dari ketinggian batang yang hampir sama pada setiap pasangan tingkat koneksi. Pada tingkat koneksi tertinggi, kedua metode hampir memberikan hasil yang identik, dengan nilai *throughput* yang sangat dekat satu sama lain. Keseluruhan grafik menunjukkan bahwa kedua strategi *load balancing* berperforma dengan baik, dengan perbedaan yang minim antara keduanya dalam kondisi pengujian yang telah dilakukan.

4. Kesimpulan

Dalam konteks *cloud*, metode *Round Robin* efektif untuk mengatur trafik ringan dan memberikan waktu respons yang stabil, cocok untuk aplikasi dengan trafik yang dapat diprediksi. Sebaliknya, *Least Connection* lebih unggul dalam mengelola trafik padat, yang penting untuk aplikasi yang mengalami lonjakan permintaan. Kesimpulan yang dapat ditarik adalah bahwa *load balancing* dengan *Least Connection* mampu mengelola permintaan dengan efektif hingga tingkat tertentu sebelum performanya mulai terdegradasi dengan peningkatan beban. Waktu respons yang masih berada di bawah ambang 4 ms hingga tingkat koneksi 5000/1000 menunjukkan bahwa metode ini cukup efektif, namun ada ruang untuk optimasi lebih lanjut untuk menangani peningkatan beban dengan lebih efisien. Perbandingan langsung dengan data dari

metode *Round Robin* dapat memberikan wawasan tentang kelebihan dan keterbatasan kedua metode dalam berbagai skenario beban kerja. Pilihan strategi *load balancing* harus disesuaikan dengan mempertimbangkan pola trafik dan kebutuhan spesifik layanan *cloud* untuk mencapai keseimbangan antara efisiensi sumber daya dan keandalan layanan.

Daftar Pustaka

Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing – A hierarchical taxonomical classification. *Journal of Cloud Computing*, *8*(1). https://doi.org/10.1186/s13677-019-0146-7

Al-Said Ahmad, A., & Andras, P. (2019). Scalability analysis comparisons of cloud-based software services. *Journal of Cloud Computing*, *8*(1). https://doi.org/10.1186/s13677-019-0134-y

Azmi, K., Syamsul, S., & Razi, F. (2022). Studi Penggunaan Dua ISP Dengan Load Balancing dan Failover Untuk Meningkatkan Kinerja Jaringan Berbasis Router Mikrotik. *JURNAL TEKTRO*, *06*(02), 176–183.

Golightly, L., Chang, V., Xu, Q. A., Gao, X., & Liu, B. S. C. (2022). Adoption of cloud computing as innovation in the organization. *International Journal of Engineering Business Management*, *14*, 1–17. https://doi.org/10.1177/18479790221093992

Hakim, D. K., Yulianto, D. Y., & Fauzan, A. (2019). Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX. *Jurnal Riset Sains Dan Teknologi*, *3*(2), 85–92. https://doi.org/10.30595/jrst.v3i2.5165

Harefa, H. S., Triyono, J., & Raharjo, S. (2021). Implementasi Load Balancing Web Server Untuk Optimalisasi Kinerja Web Server Dan Database. *Jurnal Jarkom*, *09*(01), 10–20.

Hendrana, T. S., & Suartana, I. M. (2022). Penerapan Container Load Balancing untuk Manajemen Trafik pada Learning Management System. *JINACS: Journal of Informatics and Computer Science*, *04*(02), 169–182.

Hidayat, T., Azzery, Y., & Mahardiko, R. (2020). Load Balancing Network by using Round Robin Algorithm: A Systematic Literature Review. *Jurnal Online Informatika*, *4*(2), 85. https://doi.org/10.15575/join.v4i2.446

Malau, B. G. (2022). Implementasi Load Balancing Mikrotik Jaringan Internet Di Pardamean Sibisa, Ajibata, Toba Samosir, Sumatra Utara. *JCS-TECH: Journal of Computer Science and Technology*, *2*(1), 20–29. https://doi.org/10.54840/jcstech.v2i1.23

Nugrahadi, D. T., Herteno, R., & Anshari, M. (2019). Pengaruh Implementasi Load Balancing Dan Tuning Web Server Pada Response Time Raspberry Pi. *Klik - Kumpulan Jurnal Ilmu Komputer*, *6*(2), 211. https://doi.org/10.20527/klik.v6i2.249

Nuraini, R. (2022). Implementasi Metode Load Balancing Untuk Peningkatan Nilai Troughput Pada Server. *Kumpulan JurnaL Ilmu Komputer (KLIK)*, *09*(03), 467–478.

Rafli, M., Fitri, I., & Andrianingsih, A. (2022). Pengujian Kinerja Load Balancing Web Server Menggunakan Nginx Riverse Proxy Berbasis OS Centos 7. *Jurnal Teknik Informatika Dan Sistem Informasi*, *9*(3), 1824–1840.

Riskiono, S. D., & Darwis, D. (2020). Peran Load Balancing Dalam Meningkatkan Kinerja Web Server Di Lingkungan Cloud. *Krea-TIF*, *8*(2), 1. https://doi.org/10.32832/kreatif.v8i2.3503

Riskiono, S. D., & Pasha, D. (2020). Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning. *Jurnal Teknoinfo*, *14*(1), 22–26. https://doi.org/10.33365/jti.v14i1.466

Saiputra, F. K., & Mukti, A. R. (2023). Implementasi Load Balancing Menggunakan Metode PCC (Per Connection Classifier) Pada Yayasan Bina Jaya Palembang. *Jurnal Jupiter*, *15*(1), 489–499.

Sujarwo, M. A. I. F. P., Istikmal, I., & Irawan, A. I. (2023). Analysis of Load Balancing Least Connection and Shortest Expected Delay Algorithm for Web Server Using Kube-Proxy on Kubernetes. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *11*(2), 439. https://doi.org/10.26760/elkomika.v11i2.439

Wira Harjanti, T., Setiyani, H., & Trianto, J. (2022). Load Balancing Analysis Using Round-Robin and Least-Connection Algorithms for Server Service Response Time. *Applied Technology and Computing Science Journal*, *5*(2), 40–49. https://doi.org/10.33086/atcsj.v5i2.3743