

EVALUASI LOAD BALANCING FAILOVER PADA SERVER WEB PEMANTAUAN SUNGAI BERBASIS SENSOR DAN VISUAL MENGGUNAKAN ROUND ROBIN DAN LEAST CONNECTION

Ferry Fachrizal¹, Sukardi²

Universitas Negeri Padang, Padang

e-mail: ferryfachrizal@student.unp.ac.id

Abstract: *This study aims to evaluate the implementation of failover load balancing on a sensor-based and visual river monitoring web server using the Round Robin and Least Connection algorithms. The system is designed to distribute user requests across multiple backend servers to ensure the service remains stable, responsive, and available in real time. River monitoring data is obtained from sensors and visual cameras and then presented through a web service. Testing was conducted by comparing the performance of the two algorithms based on response time, throughput, availability, downtime, and failover recovery time. The results show that Round Robin is able to distribute requests evenly but is less adaptive to changes in server load. Least Connection provides better performance because it directs requests to the server with the fewest active connections. The failover mechanism also successfully maintains service continuity when one of the servers experiences a failure. Therefore, Least Connection is recommended for river monitoring systems that require high availability and low response time.*

Keywords: *load balancing, failover, Round Robin, Least Connection, web server, river monitoring.*

Abstrak: Penelitian ini bertujuan untuk mengevaluasi penerapan load balancing failover pada server web pemantauan sungai berbasis sensor dan visual menggunakan algoritma Round Robin dan Least Connection. Sistem dirancang untuk mendistribusikan request pengguna ke beberapa server backend agar layanan tetap stabil, responsif, dan tersedia secara real-time. Data pemantauan sungai diperoleh dari sensor dan kamera visual, kemudian disajikan melalui layanan web. Pengujian dilakukan dengan membandingkan kinerja kedua algoritma berdasarkan response time, throughput, availability, downtime, dan failover recovery time. Hasil penelitian menunjukkan bahwa Round Robin mampu membagi request secara merata, tetapi kurang adaptif terhadap perubahan beban server. Least Connection menghasilkan kinerja lebih baik karena mengarahkan request ke server dengan jumlah koneksi aktif paling sedikit. Mekanisme failover juga berhasil menjaga layanan tetap berjalan ketika salah satu server mengalami gangguan. Dengan demikian, Least Connection lebih direkomendasikan untuk sistem pemantauan sungai yang membutuhkan availability tinggi dan response time rendah.

Kata kunci: load balancing, failover, Round Robin, Least Connection, server web, pemantauan sungai.

PENDAHULUAN

Sungai memiliki fungsi strategis dalam kehidupan sosial, ekologis, dan ekonomi masyarakat. Keberadaan sungai tidak hanya berkaitan dengan aliran air, tetapi juga menyangkut kualitas

lingkungan, keselamatan permukiman, aktivitas pertanian, ketersediaan air baku, hingga mitigasi risiko bencana. Dalam banyak wilayah, kondisi sungai dapat berubah secara cepat akibat curah hujan, sedimentasi, pembuangan limbah, perubahan tata guna lahan, dan aktivitas

manusia di sekitar daerah aliran sungai. Perubahan tersebut perlu dipantau secara berkelanjutan agar informasi mengenai kondisi sungai dapat diperoleh secara cepat, akurat, dan mudah diakses. Pemantauan sungai secara konvensional yang mengandalkan pengamatan langsung masih memiliki keterbatasan, terutama dari segi kecepatan, kontinuitas, dan jangkauan pengawas(Sajid et al., 2023).n. Karena itu, pemanfaatan teknologi sensor dan informasi visual menjadi pendekatan yang semakin relevan dalam membangun sistem pemantauan sungai modern. Sensor dapat digunakan untuk membaca parameter tertentu, seperti ketinggian air, arus, suhu, kekeruhan, atau kualitas air, sedangkan informasi visual melalui kamera dapat memberikan gambaran nyata mengenai kondisi fisik sungai(Lubis et al., 2022). Integrasi data sensor dan visual memungkinkan pengguna memperoleh informasi yang lebih lengkap, karena angka pengukuran dapat diperkuat dengan tampilan kondisi lapangan. Sistem seperti ini umumnya bekerja melalui server web yang berfungsi sebagai pusat penerimaan, pengolahan, penyimpanan, dan penyajian data kepada pengguna. Server web menjadi komponen penting karena seluruh data pemantauan perlu ditampilkan secara real time atau mendekati real time melalui dashboard, halaman informasi, atau sistem peringatan(Arfah et al., 2024b). Apabila server web tidak stabil, maka informasi yang seharusnya diterima secara cepat dapat mengalami keterlambatan, bahkan tidak dapat diakses sama sekali. Dalam konteks pemantauan sungai, gangguan akses bukan hanya persoalan teknis, tetapi dapat memengaruhi kualitas pengambilan keputusan(Nugroho, 2020).

Permasalahan utama dalam pengelolaan server web pemantauan sungai terletak pada kemampuan sistem dalam menangani permintaan akses dan menjaga layanan tetap berjalan ketika terjadi gangguan. Data sensor yang

dikirim secara periodik, tampilan visual yang membutuhkan bandwidth lebih besar, serta akses pengguna melalui dashboard dapat menimbulkan beban kerja yang tidak merata pada server. Jika seluruh permintaan hanya diarahkan ke satu server, maka server tersebut berpotensi mengalami kelebihan beban, penurunan performa, waktu respons yang lambat, atau bahkan kegagalan layanan. Kondisi ini semakin berisiko ketika sistem digunakan untuk pemantauan sungai secara terus-menerus, karena data yang masuk tidak berhenti pada jam tertentu(Rahmatika, Ahmad, okvi nugroho, alkhwarizmi, 2024). Server harus tetap bekerja selama perangkat sensor dan visual aktif mengirimkan data. Selain itu, sistem pemantauan sungai sering kali membutuhkan akses cepat pada saat kondisi darurat, misalnya ketika tinggi muka air meningkat secara tiba-tiba atau ketika tampilan visual menunjukkan potensi luapan. Pada situasi seperti itu, banyak pengguna dapat mengakses server dalam waktu bersamaan, baik petugas lapangan, pengelola sistem, maupun pihak lain yang berkepentingan(Lubis et al., 2023). Apabila server tidak mampu membagi beban secara efektif, maka sistem dapat mengalami bottleneck(Arfah et al., 2024a). Permasalahan lain muncul ketika salah satu server mengalami kegagalan. Tanpa mekanisme failover, kegagalan satu server dapat menyebabkan layanan tidak dapat diakses, meskipun data sensor dan visual tetap tersedia di sisi perangkat. Kegagalan ini dapat terjadi karena gangguan jaringan, kerusakan perangkat keras, kesalahan konfigurasi, atau kelebihan beban proses. Dalam sistem yang berkaitan dengan pemantauan lingkungan, waktu henti layanan perlu diminimalkan karena informasi yang terlambat dapat mengurangi efektivitas pemantauan. Oleh karena itu, dibutuhkan pendekatan yang tidak hanya mampu membagi beban akses, tetapi juga mampu mengalihkan layanan secara otomatis ketika salah satu server bermasalah(AI

Khowitzmi, Rahmad Syah, Mahyuddin K. M. Nasution, 2021).

Sebagai solusi, penerapan load balancing berbasis failover dapat digunakan untuk meningkatkan ketersediaan server web pada sistem pemantauan sungai berbasis sensor dan visual. Load balancing berperan dalam mendistribusikan permintaan pengguna atau aliran data ke beberapa server agar beban kerja tidak terpusat pada satu server saja. Dengan adanya pembagian beban, sistem dapat mempertahankan performa layanan meskipun jumlah permintaan meningkat. Sementara itu, failover berfungsi sebagai mekanisme pengalihan otomatis apabila salah satu server mengalami gangguan (Fadli et al., 2020). Kombinasi keduanya dapat membentuk sistem server yang lebih tangguh, karena layanan tetap dapat berjalan melalui server lain ketika terjadi kegagalan pada salah satu node. Dalam penelitian ini, algoritma Round Robin dan Least Connection menjadi fokus evaluasi karena keduanya sering digunakan dalam sistem load balancing dan memiliki pendekatan distribusi beban yang berbeda. Round Robin relatif sederhana karena membagi permintaan secara berurutan tanpa memperhatikan kondisi koneksi aktif pada server. Algoritma ini dapat bekerja efektif jika seluruh server memiliki spesifikasi dan beban kerja yang relatif sama. Namun, dalam sistem pemantauan sungai, jenis permintaan tidak selalu seragam (Widiarto et al., 2024). Akses data sensor dapat berukuran kecil tetapi berlangsung terus-menerus, sedangkan akses visual dapat lebih berat karena melibatkan gambar atau video. Pada kondisi seperti ini, Least Connection berpotensi memberikan distribusi yang lebih adaptif karena mempertimbangkan jumlah koneksi aktif pada masing-masing server (Putra & Purnomo, 2021). Server yang sedang menangani lebih sedikit koneksi akan memperoleh prioritas untuk menerima permintaan baru (Mardiyati & Cholifah, 2022). Evaluasi terhadap kedua

algoritma tersebut penting untuk mengetahui algoritma mana yang lebih sesuai dalam mendukung ketersediaan dan stabilitas server web pemantauan sungai. Pengujian dapat dilakukan dengan memperhatikan beberapa parameter kinerja, seperti response time, throughput, jumlah koneksi yang berhasil dilayani, tingkat kegagalan akses, serta kemampuan sistem dalam tetap berjalan ketika salah satu server mengalami gangguan. Melalui evaluasi tersebut, penelitian tidak hanya membahas implementasi teknologi load balancing, tetapi juga menilai efektivitas algoritma dalam konteks kebutuhan sistem pemantauan sungai yang menuntut kontinuitas layanan, kecepatan akses, dan ketahanan terhadap kegagalan (Parinduri & Hutagalung, 2019).

Berdasarkan uraian tersebut, penelitian mengenai Evaluasi Load Balancing Failover pada Server Web Pemantauan Sungai Berbasis Sensor dan Visual Menggunakan Round Robin dan Least Connection memiliki relevansi teknis dan praktis. Dari sisi teknis, penelitian ini dapat memberikan gambaran mengenai perbandingan kinerja dua algoritma load balancing dalam menjaga kestabilan layanan server web. Hasil evaluasi dapat menunjukkan apakah pembagian beban secara bergiliran melalui Round Robin sudah cukup memadai, atau apakah pendekatan Least Connection lebih efektif ketika sistem menghadapi variasi beban dan jumlah koneksi yang dinamis. Dari sisi praktis, penelitian ini dapat mendukung pengembangan sistem pemantauan sungai yang lebih andal, terutama pada aspek penyediaan informasi secara berkelanjutan. Sistem yang mengintegrasikan data sensor dan informasi visual membutuhkan infrastruktur yang tidak hanya mampu menerima data, tetapi juga mampu menyajikan informasi secara konsisten kepada pengguna. Apabila server memiliki ketersediaan tinggi, maka proses

pemantauan dapat berlangsung lebih stabil dan risiko kehilangan akses informasi dapat ditekan.

METODE

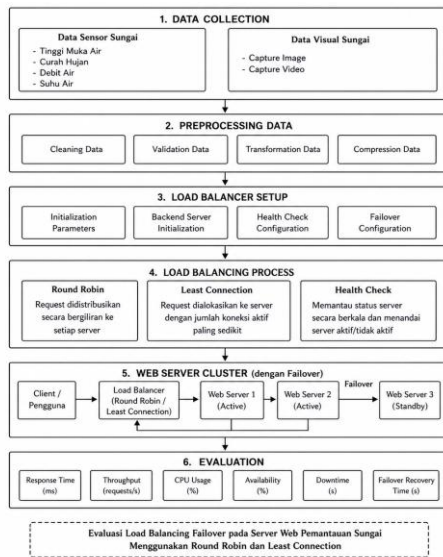
Penelitian ini berfokus pada evaluasi load balancing failover pada server web pemantauan sungai berbasis sensor dan visual menggunakan algoritma Round Robin dan Least Connection. Tujuan utama penelitian ini adalah menganalisis kinerja kedua algoritma tersebut dalam mendistribusikan beban permintaan layanan web serta menilai kemampuan sistem dalam mempertahankan ketersediaan layanan ketika terjadi gangguan pada salah satu server. Sistem yang dievaluasi merupakan server web pemantauan sungai yang menerima data dari perangkat sensor dan informasi visual sebagai sumber utama pemantauan. Data sensor yang digunakan dapat berupa data ketinggian air, arus, suhu, kekeruhan, atau parameter lain yang merepresentasikan kondisi sungai, sedangkan informasi visual diperoleh melalui kamera atau perangkat dokumentasi visual yang menampilkan kondisi sungai secara langsung. Data tersebut kemudian dikirimkan ke server web untuk diproses, disimpan, dan ditampilkan melalui halaman pemantauan atau dashboard. Dalam penelitian ini, server web tidak hanya berfungsi sebagai media penyajian informasi, tetapi juga sebagai komponen utama yang harus mampu menangani permintaan akses pengguna secara stabil dan berkelanjutan.

Metode yang digunakan dalam penelitian ini adalah metode eksperimen dengan pendekatan komparatif. Eksperimen dilakukan dengan merancang arsitektur server yang terdiri atas load balancer, beberapa server web, basis data, serta perangkat pemantauan berbasis sensor dan visual. Load balancer berfungsi sebagai pengatur distribusi permintaan akses menuju server web,

sedangkan server web bertugas menyediakan layanan pemantauan sungai kepada pengguna. Pada tahap perancangan, sistem dikonfigurasi menggunakan dua algoritma load balancing, yaitu Round Robin dan Least Connection. Algoritma Round Robin bekerja dengan membagi permintaan akses secara bergiliran ke setiap server yang tersedia, sedangkan Least Connection mengarahkan permintaan ke server dengan jumlah koneksi aktif paling sedikit. Kedua algoritma tersebut diuji pada kondisi yang sama agar hasil evaluasi dapat dibandingkan secara objektif. Selain itu, mekanisme failover diterapkan untuk memastikan bahwa layanan tetap dapat berjalan ketika salah satu server mengalami gangguan, mati, atau tidak dapat merespons permintaan pengguna.

Data yang digunakan dalam penelitian ini terdiri atas data pengujian sistem, data performa server, dan data respons layanan web. Data pengujian diperoleh melalui simulasi akses pengguna terhadap server web pemantauan sungai. Simulasi dilakukan dengan memberikan sejumlah permintaan akses dalam beberapa skenario beban, mulai dari beban ringan, sedang, hingga berat. Setiap skenario bertujuan untuk mengetahui kemampuan algoritma dalam mendistribusikan permintaan ke beberapa server. Data performa server yang diamati meliputi penggunaan CPU, penggunaan memori, jumlah koneksi aktif, distribusi beban pada setiap server, dan kestabilan server selama pengujian berlangsung. Sementara itu, data respons layanan web meliputi response time, throughput, jumlah permintaan yang berhasil dilayani, jumlah permintaan yang gagal, error rate, serta waktu yang dibutuhkan sistem untuk melakukan failover ketika salah satu server mengalami gangguan. Seluruh data tersebut digunakan untuk menilai efektivitas Round Robin dan Least Connection dalam meningkatkan ketersediaan server web pemantauan

sungai. Berikut ini arsitektur sistem yang terdapat pada gambar 1



Gambar 1 Arsitektur Sistem

Formulasi metode dalam penelitian ini berfokus pada evaluasi kinerja load balancing failover pada server web pemantauan sungai. Sistem menerima permintaan dari pengguna, lalu load balancer mendistribusikan permintaan tersebut ke beberapa server web. Dua metode yang digunakan adalah Round Robin dan Least Connection. Pada metode Round Robin, setiap permintaan didistribusikan secara bergiliran ke server yang tersedia. Jika terdapat n server, maka permintaan ke- i akan diarahkan ke server dengan rumus $S = i \bmod n$. Metode ini sederhana karena tidak mempertimbangkan beban aktual server, tetapi efektif untuk server dengan spesifikasi dan kapasitas yang relatif sama. Pada metode Least Connection, load balancer memilih server yang memiliki jumlah koneksi aktif paling sedikit. Formulasinya dapat dinyatakan sebagai $S = \min(C_1, C_2, \dots, C_n)$, dengan C_n sebagai jumlah koneksi aktif pada server ke- n . Server dengan nilai koneksi aktif terkecil akan menerima permintaan berikutnya. Metode ini lebih

adaptif dibanding Round Robin karena mempertimbangkan kondisi beban server secara real time. Oleh karena itu, Least Connection lebih sesuai ketika jumlah permintaan pengguna tidak merata atau ketika server memiliki waktu pemrosesan yang berbeda.

Mekanisme failover digunakan untuk menjaga ketersediaan layanan ketika salah satu server mengalami gangguan. Load balancer melakukan health check secara berkala untuk memeriksa status server. Jika server tidak merespons dalam batas waktu tertentu, maka server tersebut ditandai tidak aktif dan permintaan dialihkan ke server lain yang masih aktif. Status server dapat dirumuskan sebagai $H(S_i) = 1$ jika server aktif dan $H(S_i) = 0$ jika server gagal. Dengan formulasi ini, load balancer hanya mengirimkan permintaan ke server yang memiliki status aktif.

Evaluasi sistem dilakukan dengan beberapa parameter, yaitu response time, throughput, CPU usage, availability, downtime, dan failover recovery time. Response time dihitung dari selisih waktu respons diterima dan waktu permintaan dikirim, yaitu $RT = T_{response} - T_{request}$. Throughput dihitung dari jumlah permintaan yang berhasil diproses dalam satuan waktu, yaitu $TP = \frac{N}{T}$.

HASIL DAN PEMBAHASAN

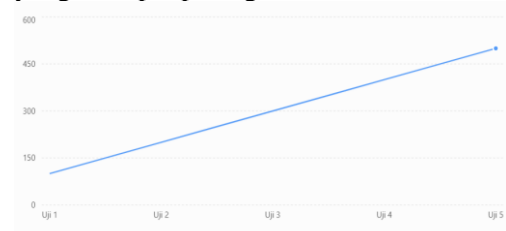
Implementasi dilakukan dengan memanfaatkan data sensor sungai dan data visual yang diperoleh dari kamera pemantauan sebagai sumber data utama yang diakses melalui layanan web. Sistem dirancang untuk mendistribusikan permintaan pengguna ke beberapa server backend menggunakan mekanisme load balancing guna menjaga kinerja, stabilitas, dan ketersediaan layanan. Algoritma Round Robin digunakan untuk mendistribusikan permintaan secara

bergiliran ke setiap server yang tersedia, sedangkan algoritma Least Connection mengalokasikan permintaan ke server yang memiliki jumlah koneksi aktif paling sedikit. Selain itu, sistem dilengkapi dengan mekanisme health check dan failover yang berfungsi mendeteksi kegagalan server secara otomatis serta mengalihkan trafik ke server lain yang masih aktif tanpa mengganggu akses pengguna. Kinerja kedua algoritma dievaluasi berdasarkan beberapa parameter, yaitu response time, throughput, penggunaan CPU, availability, downtime, dan failover recovery time. Melalui implementasi ini, layanan web pemantauan sungai diharapkan mampu memberikan akses informasi kondisi sungai secara real-time dengan tingkat keandalan yang tinggi meskipun terjadi gangguan pada salah satu server. Selain meningkatkan ketersediaan layanan, penerapan load balancing failover juga diharapkan dapat mengoptimalkan distribusi beban kerja antar server, mengurangi risiko terjadinya bottleneck, serta mendukung pengambilan keputusan yang lebih cepat dan akurat dalam kegiatan pemantauan dan mitigasi potensi bencana terkait kondisi sungai.

Hasil Penerapan Load Balancing Round Robin

Penerapan algoritma Round Robin pada sistem load balancing dilakukan dengan mendistribusikan setiap permintaan pengguna secara bergiliran ke seluruh server backend yang terdaftar. Pada penelitian ini, load balancer mengalokasikan request secara berurutan ke Server 1, Server 2, dan Server 3 tanpa mempertimbangkan jumlah koneksi aktif atau tingkat penggunaan sumber daya pada masing-masing server. Tujuan penerapan metode ini adalah untuk memastikan distribusi beban yang merata pada seluruh server sehingga tidak terjadi penumpukan permintaan pada satu server tertentu. Hasil pengujian menunjukkan bahwa algoritma Round Robin mampu

mendistribusikan request secara seimbang ke setiap server. Dari total 900 request yang diberikan selama pengujian, masing-masing server menerima sekitar 300 request atau sebesar 33,3% dari total permintaan. Distribusi yang merata ini menunjukkan bahwa algoritma Round Robin bekerja sesuai dengan prinsip dasar pembagian beban secara bergiliran. Berikut ini hasil pengujian round robin yang terdapat pada gambar 2

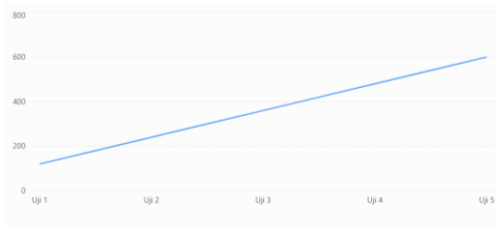


Gambar 2 Grafik Hasil Pengujian Round Robin

Hasil Penerapan Load Balancing Least Connection

Penerapan algoritma Least Connection dilakukan dengan mengarahkan setiap request pengguna ke server yang memiliki jumlah koneksi aktif paling sedikit. Berbeda dengan Round Robin, metode ini tidak hanya membagi request secara bergiliran, tetapi juga mempertimbangkan kondisi koneksi pada masing-masing server. Dengan cara ini, server yang sedang memiliki beban lebih rendah akan menerima request baru terlebih dahulu.

Hasil pengujian menunjukkan bahwa Least Connection mampu menyesuaikan distribusi beban ketika jumlah request meningkat. Pada kondisi trafik rendah, distribusi request masih relatif merata. Namun, ketika trafik meningkat, load balancer mulai mengarahkan request ke server yang memiliki koneksi aktif lebih sedikit. Mekanisme ini membantu mengurangi penumpukan koneksi pada server tertentu dan menjaga kestabilan layanan web pemantauan sungai. Berikut ini hasil pengujian algoritma Least Connection yang terdapat pada gambar 3

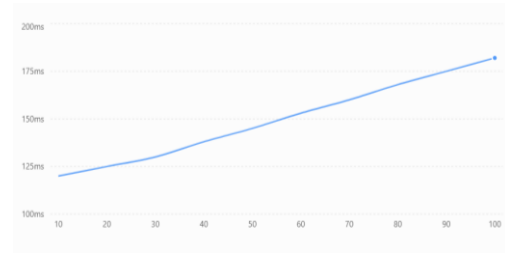


Gambar 3 Hasil Pengujian Algoritma Least Connection

Hasil Pengujian Failover Server

Pengujian failover server dilakukan untuk mengetahui kemampuan sistem dalam mempertahankan ketersediaan layanan ketika terjadi kegagalan pada salah satu server backend. Pengujian dilakukan dengan mensimulasikan kondisi server tidak aktif (down) saat sistem sedang menerima request dari pengguna. Load balancer kemudian melakukan proses health check secara berkala untuk mendeteksi status setiap server yang terhubung dalam cluster.

Hasil pengujian menunjukkan bahwa ketika salah satu server mengalami kegagalan, load balancer berhasil mendeteksi kondisi tersebut dan secara otomatis menghentikan pengiriman request ke server yang bermasalah. Selanjutnya, seluruh request dialihkan ke server lain yang masih aktif sehingga layanan web pemantauan sungai tetap dapat diakses tanpa mengalami gangguan yang signifikan. Proses pengalihan trafik berlangsung secara otomatis tanpa memerlukan intervensi administrator sistem. Berdasarkan hasil pengamatan, waktu pemulihan layanan (failover recovery time) relatif singkat karena proses perpindahan beban kerja dilakukan segera setelah server gagal terdeteksi oleh load balancer. Dengan demikian, penerapan mekanisme failover pada sistem load balancing terbukti efektif dalam menjaga stabilitas layanan, meminimalkan downtime, serta mendukung operasional sistem pemantauan sungai yang membutuhkan ketersediaan layanan secara berkelanjutan. Berikut ini hasil pengujian file over yang terdapat pada tabel 1



Gambar 4 Pengujian File Over

Pada pengujian ini, jumlah request dinaikkan secara bertahap dari 10 sampai 100 request. Hasil pengujian menunjukkan bahwa mekanisme failover tetap berjalan stabil meskipun jumlah request meningkat. Pada 10 request, failover recovery time tercatat sebesar 120 ms. Nilai ini meningkat menjadi 182 ms pada 100 request. Peningkatan tersebut masih berada dalam batas wajar karena load balancer membutuhkan waktu tambahan untuk mendeteksi server gagal, menjalankan health check, dan mengalihkan trafik ke server aktif.

Secara umum, hasil pengujian menunjukkan bahwa sistem mampu mempertahankan layanan web pemantauan sungai ketika salah satu server mengalami gangguan. Tidak terjadi kegagalan akses total selama proses pengujian. Dengan demikian, mekanisme failover dapat dinilai efektif dalam meminimalkan downtime dan menjaga availability sistem.

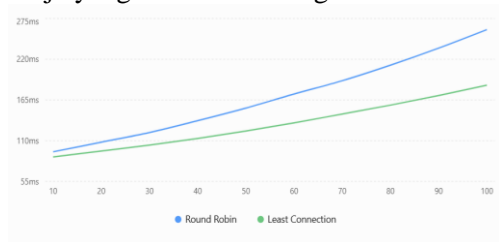
Hasil Perbandingan Round Robin dan Least Connection

Hasil perbandingan menunjukkan bahwa algoritma Round Robin dan Least Connection sama-sama mampu mendistribusikan request ke beberapa server backend. Namun, keduanya memiliki karakteristik kinerja yang berbeda. Round Robin membagi request secara bergiliran ke setiap server sehingga distribusi beban terlihat merata pada kondisi normal. Metode ini lebih sederhana dan mudah diterapkan, tetapi tidak mempertimbangkan jumlah koneksi aktif atau kondisi beban aktual pada

masing-masing server.

Least Connection menunjukkan hasil yang lebih adaptif karena load balancer memilih server dengan jumlah koneksi aktif paling sedikit. Ketika jumlah request meningkat, metode ini mampu mengurangi penumpukan koneksi pada server tertentu. Hal ini membuat response time lebih stabil dan penggunaan sumber daya server lebih seimbang dibandingkan Round Robin.

Pada pengujian failover, kedua metode tetap dapat menjaga layanan web pemantauan sungai agar tetap berjalan ketika salah satu server mengalami gangguan. Namun, Least Connection memberikan performa yang lebih baik pada kondisi trafik dinamis karena proses distribusi request menyesuaikan keadaan server secara real-time. Dengan demikian, Least Connection lebih sesuai untuk sistem pemantauan sungai yang menerima request tidak stabil, sedangkan Round Robin lebih sesuai untuk kondisi server yang memiliki kapasitas sama dan beban kerja yang relatif seimbang.



Gambar 6 Hasil perbandingan

Pada gambar 6 menggambarkan hasil perbandingan response time antara algoritma Round Robin dan Least Connection pada pengujian 10 sampai 100 request. Sumbu horizontal menunjukkan jumlah request yang diberikan ke sistem. Sumbu vertikal menunjukkan waktu respons dalam satuan milidetik. Semakin rendah nilai response time, semakin baik kinerja load balancing.

Pada grafik, response time Round Robin meningkat dari 95 ms pada 10 request menjadi 260 ms pada 100 request. Kenaikan ini menunjukkan bahwa Round Robin kurang adaptif ketika beban request

semakin besar karena distribusi trafik tetap dilakukan secara bergiliran tanpa melihat kondisi koneksi aktif pada server.

Sementara itu, Least Connection meningkat dari 88 ms pada 10 request menjadi 185 ms pada 100 request. Nilai ini lebih rendah dibandingkan Round Robin pada setiap skenario pengujian. Hal ini menunjukkan bahwa Least Connection lebih stabil karena request diarahkan ke server dengan koneksi aktif paling sedikit.

Berdasarkan diagram tersebut, Least Connection memiliki performa lebih baik dibandingkan Round Robin. Metode ini mampu menjaga response time tetap lebih rendah saat jumlah request meningkat. Oleh karena itu, Least Connection lebih sesuai untuk sistem web pemantauan sungai yang memiliki trafik dinamis dan membutuhkan layanan real-time.

SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan load balancing failover pada server web pemantauan sungai berbasis sensor dan visual menggunakan algoritma Round Robin dan Least Connection berhasil dirancang dan diimplementasikan untuk mendukung ketersediaan layanan pemantauan sungai secara real-time. Sistem ini mengintegrasikan data sensor, data visual, load balancer, server backend, health check, dan mekanisme failover sehingga mampu menjaga layanan tetap berjalan meskipun terjadi peningkatan request atau gangguan pada salah satu server. Implementasi sistem menunjukkan bahwa algoritma Round Robin mampu mendistribusikan request secara bergiliran ke setiap server backend dengan pola pembagian yang merata. Namun, metode ini belum mempertimbangkan jumlah koneksi aktif dan kondisi beban aktual server. Sementara itu, algoritma Least Connection menunjukkan kinerja yang lebih adaptif karena mengarahkan request

ke server dengan jumlah koneksi aktif paling sedikit. Hasil pengujian menunjukkan bahwa Least Connection menghasilkan response time yang lebih rendah dan lebih stabil dibandingkan Round Robin ketika jumlah request meningkat. Hasil pengujian failover menunjukkan bahwa load balancer mampu mendeteksi server yang mengalami gangguan melalui mekanisme health check dan secara otomatis mengalihkan trafik ke server lain yang masih aktif. Dengan demikian, layanan web pemantauan sungai tetap dapat diakses tanpa gangguan signifikan. Secara keseluruhan, Least Connection lebih direkomendasikan untuk sistem pemantauan sungai yang memiliki trafik dinamis, membutuhkan response time rendah, serta menuntut availability tinggi.

DAFTAR PUSTAKA

- Al Khowarizmi, Rahmad Syah, Mahyuddin K. M. Nasution, M. E. (2021). Sensitivity Of Mape Using Detection Rate For Big Data Forecasting Crude Palm Oil On K-Nearest Neighbor. *International Journal Of Electrical And Computer Engineering (Ijece)*, 11(3), 2697–2704. <https://doi.org/10.11591/ijece.V11i3.Pp2697-2704>
- Arfah, M., Fachrizal, F., & Nugroho, O. (2024a). Developing A Model Of Association Rules With Machine Learning In Determining User Habits On Social Media. *Eastern-European Journal Of Enterprise Technologies*, (2).
- Arfah, M., Fachrizal, F., & Nugroho, O. (2024b). *Developing A Model Of Association Rules With Machine Learning In Determining User Habits On Social Media*. 55–61. <https://doi.org/10.15587/1729-4061.2024.305116>
- Fadli, S., Ashari, M., & Imtihan, K. (2020). Sistem Penjadwalan Event Organizer Dengan Metode Round Robin (Rr). *Jurnal Manajemen Informatika Dan Sistem Informasi*, 3(2), 100–107.
- Lubis, A. R., Prayudani, S., Lubis, M., & Nugroho, O. (2022). Sentiment Analysis On Online Learning During The Covid-19 Pandemic Based On Opinions On Twitter Using Knn Method. *2022 1st International Conference On Information System & Information Technology (Icisit)*, 106–111.
- Lubis, A. R., Safitri, H. R., Lubis, M., & Nugroho, O. (2023). Implementation Of Preprocessing In Text Summarization Techniques For Indonesian Language Documents Using The Flax T5 Approach. *2023 11th International Conference On Cyber And It Service Management (Citsm)*, 1–6.
- Mardiyati, S., & Cholifah, W. N. (2022). Sistem Penjadwalan Bus Di Terminal Jatijajar Depok Menggunakan Algoritma Round Robin. *Jurnal Fasilkom*, 12(1), 48–55.
- Nugroho, O. (2020). Implementation Of Marker Based Tracking Method In The Interactive Media Of Traditional Clothes Knowledge-Based On Augmented Reality 360. *Journal Of Computer Science, Information Technology And Telecommunication Engineering*, 1(2), 37–43.
- Parinduri, I., & Hutagalung, S. N. (2019). Teknik Penjadwalan Prosesor Fifo, Sjf Non Preemptive, Round Robin. *Prosiding Seminar Nasional Riset Information Science (Senaris)*, 1(September), 864. <https://doi.org/10.30645/Senaris.V1i0.93>
- Putra, T. D., & Purnomo, R. (2021). Analisis Algoritma Round Robin Pada Penjadwalan Cpu. *Jurnal Ilmiah Teknologi Informasi Asia*, 15(2), 85.
- Rahmatika, Ahmad, Okvi Nugroho,

-
- Alkhowarizmi, Tri Andre Anu. (2024). *Using Relational Learning In Exploring The Effectiveness Of Using Hashtags In Future Topics And User Relations In X. 2*, 62–68. <https://doi.org/10.15587/1729-4061.2024.306726>
- Sajid, J., Hayawi, K., Malik, A. W., Anwar, Z., & Trabelsi, Z. (2023). A Fog Computing Framework For Intrusion Detection Of Energy-Based Attacks On Uav-Assisted Smart Farming. *Applied Sciences (Switzerland)*, 13(6). <https://doi.org/10.3390/app13063857>
- Widiarto, W., Maheswari, D., Sari, D. P., & Arianto, K. J. (2024). Implementasi Algoritma Round Robin Dan Priority Pada Sistem Antrian Rumah Sakit. *Jurnal Fasilkom*, 14(2), 507–513.