

**SKRIPSI**

IMPLEMENTASI ALGORITMA *ROUND ROBIN* DALAM SISTEM *MULTI-AGENT* DAN *MULTI-CLIENT* UNTUK *LOAD BALANCING* DINAMIS PADA JARINGAN LOKAL

*IMPLEMENTATION OF ROUND ROBIN ALGORITHM IN MULTI-AGENT AND MULTI-CLIENT SYSTEM FOR DYNAMIC LOAD BALANCING IN LOCAL NETWORKS*



WIWI NOPIANA  
D0221018

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SULAWESI BARAT MAJENE

2025

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA *ROUND ROBIN* DALAM SISTEM  
*MULTI-AGEN* DAN *MULTI-CLIENT* UNTUK *LOAD BALANCING*  
DINAMIS PADA JARINGAN LOKAL

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

WIWI NOPIANA  
NIM. D0221018

Telah diperiksa dan disetujui oleh:

Pembimbing I

Muh. Fuad Marisyur, S.Kom., M.Kom.  
NIP. 199205022019031017

Pembimbing II

Wawan Firgiawan, S.T., M.Kom  
NIDK. 8948080023

Dekan Fakultas Teknik,

Ketua Program Studi Informatika,

Prof. Dr. Ir. Hafsah Nirwana, M.T  
NIP. 196404051990032002

Muh. Rafli Rasyid, S.Kom., M.T  
NIP. 198808182022031006

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi. Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundangundangan yang berlaku (**UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70**).

Majene, 04 November 2025



Wiwj Nopiana

D0221018

## ABSTRAK

***Wivi Nopiana. Implementasi Algoritma Round Robin dalam Sistem Multi-Agent dan Multi-Client untuk Load Balancing Dinamis pada Jaringan Lokal. (Dibimbing oleh Muh. Fuad Mansyur dan Wawan Firgiawan)***

Load balancing merupakan mekanisme penting dalam sistem layanan jaringan untuk memastikan pemerataan beban kerja dan menjaga kestabilan performa layanan. Penelitian ini bertujuan untuk mengimplementasikan algoritma Round Robin dalam arsitektur sistem multi-agent dan multi-client yang dijalankan pada jaringan lokal (LAN). Sistem dirancang menggunakan tiga komputer, di mana satu komputer berperan sebagai *agent controller* (load balancer) dan dua komputer lainnya sebagai *server backend*. Beberapa *client* mengirimkan permintaan secara simultan ke *controller*, yang kemudian mendistribusikannya secara bergiliran ke server menggunakan konfigurasi berbasis NGINX. Pengujian dilakukan pada tiga skenario beban, yaitu 50, 100, dan 200 permintaan. Hasil penelitian menunjukkan bahwa algoritma Round Robin mampu mendistribusikan beban secara merata dengan rasio ideal (50:50) dan menghasilkan waktu respons yang stabil pada beban ringan hingga sedang. Sistem yang dikembangkan bersifat ringan, modular, dan mudah diimplementasikan pada jaringan lokal, sehingga cocok digunakan sebagai model edukatif maupun solusi teknis untuk sistem berskala kecil hingga menengah.

Kata Kunci: *Load balancing, Round Robin, Multi-Agent System, Multi-Client, Jaringan Lokal, NGINX*

## **ABSTRACT**

**Wiwi Nopiana.** *Implementation of the Round Robin Algorithm in a Multi-Agent and Multi-Client System for Dynamic Load Balancing on a Local Network. (Supervised by Muh. Fuad Mansyur and Wawan Firgiawan)*

*Load balancing is a crucial mechanism in network service systems to ensure equitable workload distribution and maintain service performance stability. This study aims to implement the Round Robin algorithm within a multi-agent and multi-client system architecture operating in a local area network (LAN). The system was designed using three computers: one acting as an agent controller (load balancer) and two others serving as backend servers. Multiple clients simultaneously sent requests to the controller, which distributed them alternately to the servers using an NGINX-based configuration. Testing was conducted under three load scenarios—50, 100, and 200 requests. The results showed that the Round Robin algorithm effectively distributed the load evenly between the two backend servers with an ideal 50:50 ratio and maintained stable response times under light to moderate loads. The developed system is lightweight, modular, and easy to deploy within local networks, making it suitable as both an educational model and a practical solution for small to medium-scale systems.*

**Keywords:** *Load balancing, Round Robin, Multi-Agent System, Multi-Client, Local Network, NGINX*

## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi informasi dan komunikasi telah memengaruhi hampir semua bidang kehidupan, terutama dalam pengelolaan layanan berbasis jaringan komputer. Model *client-server* menjadi dasar dari berbagai sistem modern seperti sistem informasi akademik, *e-commerce*, hingga komputasi awan (Telkom University, 2023). Namun, meningkatnya jumlah pengguna yang mengakses layanan secara bersamaan sering menyebabkan ketidakseimbangan beban kerja antar server. Akibatnya, sebagian server menjadi kelebihan beban (*overload*), sementara server lain menganggur, yang berdampak pada penurunan performa sistem secara keseluruhan (Arora & Sood, 2019).

Untuk mengatasi hal tersebut, diperlukan mekanisme *load balancing*, yaitu proses pendistribusian beban kerja dari *client* ke beberapa *server* agar kinerja sistem tetap stabil dan merata (Buyya et al., 2013). Dengan *load balancing*, sistem dapat mengoptimalkan penggunaan sumber daya, meningkatkan *availability*, serta menghindari terjadinya *bottleneck* (Mitzenmacher, 2001). Prinsip ini menjadi landasan utama dalam arsitektur sistem terdistribusi, baik di lingkungan *cloud computing* maupun jaringan lokal (Singh et al., 2024).

Salah satu algoritma yang paling sederhana dan sering digunakan dalam *load balancing* adalah Round Robin. Algoritma ini bekerja dengan cara mendistribusikan setiap permintaan secara bergantian ke server yang tersedia, tanpa memperhatikan beban aktual server (Utami & Haryanto, 2020). Meskipun bersifat statis, Round Robin terbukti efektif untuk sistem dengan sumber daya yang homogen dan beban relatif seimbang (Wijaya et al., 2021). Penelitian sebelumnya menunjukkan bahwa strategi ini mampu memberikan waktu respons yang stabil dan distribusi beban yang cukup merata dibandingkan metode acak (Utami & Haryanto, 2020).

Sementara itu, konsep sistem *multi-agent* memberikan pendekatan yang lebih modular dan terdistribusi dalam pengelolaan beban kerja jaringan. Dalam sistem ini, setiap agen (*agent*) bekerja secara otonom namun tetap berkoordinasi dengan agen lain untuk mencapai tujuan bersama (Arifin & Nugroho, 2021). Pendekatan *multi-agent* ini juga telah diterapkan dalam sistem berbasis IoT dan *web server*

untuk meningkatkan efisiensi melalui komunikasi antar agen (Liu et al., 2023; Alfian et al., 2022).

Namun, sebagian besar penelitian terdahulu masih berfokus pada implementasi *load balancing* dalam konteks *cloud computing* (Rahmika et al., 2023; Shafiq et al., 2022). Padahal, lingkungan jaringan lokal (LAN) di institusi pendidikan, laboratorium, atau usaha kecil menengah juga membutuhkan sistem distribusi beban yang ringan dan mudah diimplementasikan. Dalam konteks tersebut, algoritma Round Robin memiliki keunggulan karena bersifat sederhana, tidak memerlukan pengukuran beban dinamis, serta dapat diintegrasikan dengan perangkat lunak *open-source* seperti NGINX (Cloud Raya, 2022).

Penelitian ini mengusulkan pengembangan sistem *multi-agent dan multi-client* berbasis LAN yang menerapkan algoritma Round Robin untuk mendistribusikan beban secara dinamis antar server. Sistem terdiri dari tiga komputer, yaitu satu agent *controller* sebagai *load balancer* dan dua server backend. *Client* dalam jaringan mengirim permintaan secara simultan ke *controller*, kemudian permintaan tersebut didistribusikan bergantian melalui mekanisme Round Robin yang dikonfigurasi pada NGINX.

Pendekatan ini memiliki nilai aplikatif karena selain menjadi solusi teknis untuk pemerataan beban, sistem ini juga dapat dijadikan sebagai modul pembelajaran praktikum jaringan komputer di lingkungan kampus. Oleh karena itu, penelitian ini diberi judul: “*Implementasi Algoritma Round Robin dalam Sistem Multi-Agent dan Multi-Client untuk Load Balancing Dinamis pada Jaringan Lokal.*”

## 1.2. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi algoritma Round Robin dalam sistem *multi-agent* dan *multi-client* berbasis jaringan lokal?
2. Bagaimana mekanisme distribusi beban yang dihasilkan oleh sistem terhadap dua *server backend*?
3. Bagaimana performa sistem terhadap variasi jumlah permintaan *client*?

### 1.3. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma Round Robin pada sistem *multi-agent* dan *multi-client* berbasis LAN.
2. Menganalisis pemerataan beban yang dihasilkan oleh sistem.
3. Mengevaluasi waktu respons sistem terhadap jumlah permintaan *client* yang bervariasi.

### 1.4. Manfaat Penelitian

Untuk manfaat penelitian ini adalah sebagai berikut:

#### 1. Manfaat Akademik:

Menambah referensi ilmiah di bidang jaringan komputer, khususnya terkait implementasi *load balancing* berbasis algoritma Round Robin dalam sistem terdistribusi (Arifin & Nugroho, 2021; Liu et al., 2023).

#### 2. Manfaat Praktis:

Menjadi alternatif solusi *load balancing* ringan dan mudah diterapkan untuk lingkungan dengan sumber daya terbatas seperti laboratorium atau UKM (Cloud Raya, 2022).

#### 3. Manfaat Edukatif:

Dapat digunakan sebagai bahan ajar praktikum dalam mata kuliah *Jaringan Komputer* atau *Sistem Terdistribusi* untuk memperkenalkan konsep *load balancing* menggunakan NGINX (Telkom University, 2023).

### 1.5. Batasan Penelitian

Adapun Batasan masalah pada penelitian ini:

1. Sistem terdiri atas tiga node (1 controller dan 2 server backend).
2. Algoritma yang digunakan adalah Round Robin statis bawaan NGINX.
3. Uji dilakukan pada jaringan lokal (LAN) dengan jumlah *client* terbatas.
4. Sistem tidak melakukan *health check* otomatis atau pembobotan dinamis.
5. Implementasi difokuskan pada performa waktu respons dan distribusi beban.

## BAB V PENUTUP

### 5.1. Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem yang telah dilakukan, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Implementasi Algoritma Round Robin Berhasil Mendistribusikan Beban Secara Merata: Sistem *load balancing* berbasis algoritma Round Robin yang diimplementasikan menggunakan NGINX dalam arsitektur *multi-agent* dan *multi-client* pada jaringan lokal (LAN) berhasil menjalankan distribusi beban dengan baik. Setiap *server backend* menerima jumlah permintaan yang sama (rasio 50:50) pada semua skenario pengujian, baik beban ringan (50 request), sedang (100 request), maupun tinggi (200 request). Hasil ini sejalan dengan karakteristik deterministik algoritma Round Robin sebagaimana dijelaskan oleh Wijaya et al. (2021) dan Utami & Haryanto (2020), yang menunjukkan kestabilan distribusi pada sistem dengan jumlah server homogen.
2. Sistem Menunjukkan Stabilitas Waktu Respons pada Beban Ringan hingga Sedang Pengujian waktu respons menunjukkan bahwa sistem mampu menjaga kinerja stabil pada beban ringan dan sedang, dengan waktu respons rata-rata di bawah 200 ms. Pada beban tinggi (200 permintaan), waktu respons meningkat menjadi sekitar 290 ms, namun masih dalam batas wajar untuk lingkungan LAN. Hasil ini menunjukkan bahwa sistem bekerja secara efisien dan responsif dalam kondisi normal, sebagaimana diungkapkan oleh Shafiq et al. (2022) dan Singh et al. (2024) bahwa performa sistem terdistribusi tergantung pada efisiensi distribusi beban dan konfigurasi jaringan.
3. Pendekatan *Multi-Agent* dan *Multi-Client* Efektif untuk Lingkungan Lokal Terbatas Penerapan arsitektur *multi-agent* dan *multi-client* menjadikan sistem lebih modular, mudah dikontrol, serta memungkinkan pengujian paralel di jaringan lokal. Model ini mendukung efisiensi pengelolaan beban tanpa memerlukan infrastruktur cloud (Arifin & Nugroho, 2021; Liu et al., 2023). Hasil penelitian membuktikan bahwa pendekatan ini dapat menjadi alternatif sederhana untuk sistem pendidikan, laboratorium, atau instansi kecil yang membutuhkan solusi *load balancing* ringan.

4. Keterbatasan Sistem dan Peluang Pengembangan Meskipun efektif, sistem ini masih memiliki keterbatasan karena menggunakan algoritma Round Robin statis tanpa mekanisme *health check* atau monitoring dinamis. Jika salah satu server mengalami gangguan, sistem tidak dapat secara otomatis mengalihkan beban ke server lain. Oleh karena itu, diperlukan integrasi mekanisme adaptif seperti Least Connection, Weighted Round Robin, atau sistem pemantauan otomatis (Patel & Thakkar, 2020; Raghav et al., 2022).

Secara keseluruhan, hasil penelitian ini memperlihatkan bahwa algoritma Round Robin masih sangat relevan untuk diterapkan dalam sistem *load balancing* berskala kecil hingga menengah. Pendekatan ini efisien, ringan, dan mudah diimplementasikan dalam jaringan lokal yang memiliki sumber daya terbatas namun membutuhkan stabilitas layanan.

## 5.2. Saran

Berdasarkan hasil dan temuan penelitian, beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Integrasi Algoritma Adaptif: Pengembangan selanjutnya dapat menggunakan algoritma dinamis seperti *Least Connection* atau *Weighted Round Robin* agar sistem dapat menyesuaikan distribusi beban berdasarkan performa aktual setiap server (Buyya et al., 2013; Raghav et al., 2022).
2. Penambahan Mekanisme *Health Check*: Untuk meningkatkan keandalan sistem, perlu ditambahkan fitur *health monitoring* pada NGINX agar *load balancer* dapat mendeteksi status server dan menghindari pengiriman permintaan ke server yang tidak aktif (Utami & Haryanto, 2020).
3. Uji Skalabilitas dengan Jumlah Node Lebih Banyak: Penelitian lanjutan dapat memperluas jumlah *server backend* dan *client* untuk menguji skalabilitas sistem dalam kondisi beban tinggi, sesuai dengan pendekatan *horizontal scaling* yang diusulkan oleh Mitzenmacher (2001) dan Singh et al. (2024).
4. Integrasi Pemantauan Berbasis *Multi-Agent*: Sistem *multi-agent* dapat dikembangkan menjadi lebih cerdas dengan komunikasi antar agen untuk melakukan redistribusi beban secara otonom (Liu et al., 2023; Alfian et al.,

2022). Pendekatan ini membuka peluang penerapan *intelligent load balancing* di masa depan.

5. Implementasi pada Lingkungan *Cloud* atau Hybrid Network: Untuk pengujian tingkat lanjut, sistem dapat diadaptasi ke arsitektur *cloud* seperti AWS atau Azure untuk melihat performa algoritma dalam lingkungan virtual terdistribusi yang lebih kompleks (Arora & Sood, 2019; Shafiq et al., 2022).

Penelitian ini memiliki implikasi penting baik secara akademik maupun praktis. Secara akademik, hasil penelitian dapat dijadikan bahan ajar atau modul praktikum untuk mata kuliah yang berkaitan dengan jaringan komputer dan sistem terdistribusi. Melalui konfigurasi sistem yang sederhana, mahasiswa maupun praktisi dapat memahami konsep dasar *load balancing* secara langsung tanpa memerlukan infrastruktur besar. Sementara dari sisi praktis dan teknologi, penerapan algoritma Round Robin berbasis NGINX membuktikan bahwa teknologi *open-source* dapat dimanfaatkan untuk meningkatkan kinerja sistem jaringan lokal secara efisien. Solusi ini dapat menjadi alternatif bagi lembaga pendidikan atau organisasi kecil dalam membangun sistem internal yang stabil, ringan, dan mudah diimplementasikan.

Selain itu, penelitian ini juga memiliki dampak sosial dan ekonomi karena sistem yang dikembangkan mudah direplikasi dan memiliki biaya implementasi rendah. Pendekatan ini berpotensi mendukung upaya digitalisasi dan efisiensi kerja di lingkungan pendidikan maupun pemerintahan lokal, serta mendorong kemandirian teknologi di wilayah dengan keterbatasan sumber daya. Secara keseluruhan, penelitian ini membuktikan bahwa algoritma Round Robin dapat diterapkan secara efektif dalam arsitektur sistem *multi-agent* dan *multi-client* pada jaringan lokal. Sistem yang dibangun menunjukkan kinerja stabil, pemerataan beban ideal, serta efisiensi tinggi, dan diharapkan dapat dikembangkan lebih lanjut dengan menambahkan kecerdasan adaptif serta pemantauan otomatis untuk mendukung *load balancing* cerdas pada lingkungan *cloud*, *edge computing*, dan *Internet of Things (IoT)*.

## DAFTAR PUSTAKA

- Alfian, F., Suryono, T. & Prasetyo, A. (2022) 'Smart Multi-agent System for Load Balancing in IoT Gateway', *Journal of Communications*, 17(3), pp. 170–178.
- Arifin, R. & Nugroho, A. (2021) 'Sistem Load Balancing Berbasis Rule pada Jaringan LAN Menggunakan Metode Agent-Based', *Jurnal Informatika Mulawarman*, 16(1), pp. 12–20.
- Arora, N. & Sood, M. (2019) 'Load Balancing Techniques in Cloud Computing: A Review', *International Journal of Advanced Research in Computer Science*, 10(5), pp. 78–83.
- Buyya, R., Vecchiola, C. & Selvi, S.T. (2013) *Mastering Cloud Computing: Foundations and Applications Programming*. San Francisco: Morgan Kaufmann.
- Cloud Raya (2022) 'Belajar Load Balancing Server: Pengertian, Jenis, dan Cara Kerja'. Available at: <https://cloudraya.com/blog/belajar-load-balancing-server-pengertian-jenis-dan-cara-kerja/> (Accessed: 2 November 2025).
- Liu, Q., Wu, C. & Yu, J. (2023) 'Cooperative Multi-agent Model for Dynamic Load Distribution in Web Servers', *Concurrency and Computation: Practice and Experience*, 35(1), e6863.
- Mitzenmacher, M. (2001) 'The Power of Two Choices in Randomized Load Balancing', *IEEE Transactions on Parallel and Distributed Systems*, 12(10), pp. 1094–1104.
- Patel, S. & Thakkar, V. (2020) 'Load Balancing Using DNS Round Robin Technique for Large Scale Web Applications', *International Journal of Engineering Research and Applications*, 10(1), pp. 45–49.
- Rahmika, A.R., Tahir, Z., Paundu, A.W. & Zainuddin, Z. (2023) 'Web Server Load Balancing Mechanism with Least Connection Algorithm and Multi-agent System', *CommIT Journal*, 17(2), pp. 245–258.
- Raghav, Y.Y., Vyas, V. & Rani, H. (2022) 'Load balancing using dynamic algorithms for cloud environment: A survey', *Materials Today: Proceedings*, 69(2), pp. 349–353.

- Shafiq, D.A., Jhanjhi, N.Z. & Abdullah, A. (2022) 'Load balancing techniques in cloud computing environment: A review', *Journal of King Saud University - Computer and Information Sciences*, 34(7), pp. 3910–3933.
- Singh, A., Sahu, A.K., Siddiqui, N.A. & Singh, S. (2024) 'Optimizing Cloud Performance: A Comprehensive Study of Load Balancing Strategies and Algorithms', *Smart Internet of Things*, 1(1), pp. 1–16.
- Telkom University (2023) 'Mengenal Jaringan Client-Server: Konsep dan Cara Kerja'. Available at: <https://jakarta.telkomuniversity.ac.id/mengenal-jaringan-client-server-konsep-dan-cara-kerja/> (Accessed: 2 November 2025).
- Utami, A. & Haryanto, R. (2020) 'Comparative Study of Round Robin and Random Strategy on HAProxy Load Balancing', *Jurnal Teknologi dan Sistem Komputer*, 8(2), pp. 147–153.
- Wijaya, A., Santosa, R. & Aditya, M. (2021) 'Implementasi Least Connection pada Load Balancer untuk Meningkatkan Kinerja Web Server', *Jurnal Teknologi Informasi dan Ilmu Komputer*, 6(3), pp. 277–284.