



## Analisis Perbandingan *Load Balancing* Menggunakan Algoritma *Round Robin* dan *Weighted Round Robin* pada Mikrotik

Muhammad Lailul Karim <sup>1\*</sup>, Muhammad Imron Rosadi <sup>1</sup>

<sup>1</sup> Program Studi Teknik Informatika, Universitas Yudharta Pasuruan, Indonesia

\* Korespondensi: [ilularekK.05@gmail.com](mailto:ilularekK.05@gmail.com)

**Sitasi:** M. L. Karim and M. I. Rosadi, "Analisis Perbandingan Load balancing Menggunakan Algoritma Round Robin dan Weighted Round Robin pada Mikrotik", Jurnal Teknologi Informasi Dan Multimedia, vol. 8, no. 2, pp. 245-256, 2026. <https://doi.org/10.35746/jtim.v8i2.844>

Diterima: 26-07-2025

Direvisi: 21-09-2025

Disetujui: 28-03-2026



**Copyright:** © 2026 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

**Abstract:** This study analyzes and compares the performance of *Round Robin* (RR) and *Weighted Round Robin* (WRR) *load balancing* algorithms in a multi-WAN network environment using MikroTik devices. The research aims to evaluate the effectiveness of both algorithms in distributing network traffic across internet links with unequal bandwidth capacities. The experimental setup consisted of two ISP connections with bandwidths of 10 Mbps and 20 Mbps, configured on a MikroTik RB750Gr3 router. Performance evaluation was conducted using key network parameters, including packet loss, *Latency*, bandwidth utilization, and CPU load. The results show that the *Round Robin* algorithm, which distributes traffic evenly without considering link capacity, leads to high packet loss and unstable *Latency*, particularly on the lower-bandwidth link. In contrast, the *Weighted Round Robin* algorithm, which allocates traffic proportionally based on bandwidth capacity using a 1:2 ratio, demonstrates significantly better performance. WRR reduces packet loss by up to 4–6 times, provides more stable *Latency*, optimizes bandwidth utilization, and results in lower CPU load on the router. Although WRR requires more complex configuration, the findings indicate that it is more suitable and efficient for multi-WAN networks with significant bandwidth differences between ISP links.

**Keywords:** mikrotik; *load balancing*; *Round Robin*; *weighted Round Robin*.

**Abstrak:** Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja algoritma *Load balancing Round Robin* (RR) dan *Weighted Round Robin* (WRR) pada jaringan multi-WAN menggunakan perangkat MikroTik. Fokus penelitian ini adalah mengevaluasi efektivitas kedua algoritma dalam mendistribusikan trafik jaringan pada jalur Internet dengan kapasitas bandwidth yang tidak seimbang. Pengujian dilakukan melalui simulasi jaringan menggunakan satu router MikroTik RB750Gr3 yang terhubung ke dua ISP dengan kapasitas bandwidth masing-masing sebesar 10 Mbps dan 20 Mbps. Evaluasi kinerja dilakukan berdasarkan parameter packet loss, *Latency*, pemanfaatan bandwidth, dan CPU load. Hasil penelitian menunjukkan bahwa algoritma *Round Robin*, yang membagi trafik secara merata tanpa mempertimbangkan kapasitas jalur, menghasilkan packet loss yang tinggi dan *Latency* yang tidak stabil, terutama pada jalur dengan bandwidth lebih kecil. Sebaliknya, algoritma *Weighted Round Robin* yang menerapkan pembobotan distribusi trafik dengan rasio 1:2 sesuai kapasitas bandwidth mampu menurunkan packet loss hingga 4–6 kali, menghasilkan *Latency* yang lebih stabil, pemanfaatan bandwidth yang lebih optimal, serta penggunaan CPU router yang lebih efisien. Meskipun konfigurasi *Weighted Round Robin* lebih kompleks, algoritma ini terbukti lebih efektif untuk diterapkan pada jaringan multi-WAN dengan perbedaan kapasitas bandwidth yang signifikan antar ISP.

**Kata kunci:** mikrotik; *load balancing*; *Round Robin*; *weighted round robin*;

## 1. Pendahuluan

Di era digital saat ini, kebutuhan akan koneksi internet yang stabil dan cepat semakin meningkat. Hal ini mendorong penggunaan berbagai solusi teknologi untuk memastikan kualitas layanan yang optimal, baik dalam sektor pendidikan, bisnis, maupun layanan publik[1]. Salah satu solusi yang sering diterapkan adalah penggunaan beberapa jalur koneksi internet (Wide Area Network) atau disebut sebagai multi-WAN yang bertujuan untuk menghindari keterlambatan akses, *downtime* dan kemacetan trafik yang dapat mengganggu aktivitas pengguna[1]. Namun, penerapan multi-WAN tanpa pengelolaan yang tepat dapat berujung pada pemborosan bandwidth dan distribusi beban yang tidak merata, yang justru memperburuk kinerja jaringan[2].

Untuk mengatasi masalah ini, konsep *Load balancing* menjadi penting dalam manajemen jaringan[3]. Teknik ini berfungsi untuk mendistribusikan trafik data ke beberapa jalur atau server secara merata, sehingga beban kerja dapat disesuaikan dengan kapasitas masing-masing jalur dan performa jaringan tetap optimal[4]. Dalam berbagai penelitian sebelumnya, beragam algoritma *Load balancing* telah dikembangkan untuk meningkatkan efisiensi distribusi trafik, di antaranya algoritma *Round Robin* (RR) yang sederhana dan sering digunakan, serta *Weighted Round Robin* (WRR) yang memberikan bobot pada jalur atau server sesuai dengan kapasitasnya[5]. Meskipun *Round Robin* cukup populer karena kemudahan implementasinya, algoritma ini sering kali menimbulkan masalah dalam situasi di mana jalur internet atau server memiliki kapasitas yang berbeda[6]. Dalam kondisi ini, *Weighted Round Robin* lebih unggul karena dapat menyesuaikan distribusi beban sesuai dengan kapasitas jalur, sehingga meningkatkan efisiensi penggunaan bandwidth[7], [5].

Penelitian ini berfokus pada penerapan dua algoritma tersebut, *Round Robin* dan *Weighted Round Robin*, pada perangkat Mikrotik sebagai *load balancer* dalam mendistribusikan trafik keluar ke dua jalur ISP yang berbeda. Pengujian dilakukan dengan menggunakan dua modem Tenda yang terhubung ke layanan internet berbeda[8]. Fokus utama penelitian ini adalah mengevaluasi efektivitas kedua algoritma dalam mendistribusikan trafik dengan menggunakan parameter kinerja nyata seperti *Latency*, *throughput*, *packet loss*, dan *CPU utilization*[9]. Penelitian ini juga membandingkan kinerja kedua algoritma untuk mengetahui mana yang lebih optimal dalam konteks pengelolaan trafik internet.

Saat ini, kebanyakan penelitian yang ada lebih sering berfokus pada pengujian satu algoritma saja atau hanya menjelaskan teori tanpa implementasi praktis[10]. Oleh karena itu, penelitian ini memberikan kontribusi yang berbeda dengan membandingkan *Round Robin* dan *Weighted Round Robin* secara langsung pada perangkat Mikrotik dan menganalisis performa keduanya dalam konteks penggunaan internet publik[11]. Temuan dalam penelitian ini diharapkan dapat memberikan wawasan baru bagi pengelola jaringan dalam memilih algoritma *Load balancing* yang lebih efisien dan sesuai dengan kebutuhan mereka[12].

Adapun tujuan dari penelitian ini adalah untuk menerapkan algoritma *Load balancing* dengan menggunakan *Round Robin* dan *Weighted Round Robin* pada perangkat Mikrotik untuk mendistribusikan trafik jaringan secara efektif dan efisien[13], serta untuk menganalisis dan membandingkan performa kedua algoritma berdasarkan berbagai parameter seperti *CPU Load*, *Latency*, *Throughput*, dan penggunaan bandwidth[14].

Secara keseluruhan, penelitian ini diharapkan dapat memberikan kontribusi penting dalam pemahaman teknik *Load balancing*, serta memberikan rekomendasi bagi implementasi *Load balancing* yang lebih efektif dalam jaringan multi-WAN[15]. Temuan utama dari penelitian ini menunjukkan bahwa algoritma *Weighted Round Robin* memberikan hasil yang lebih optimal dibandingkan dengan *Round Robin* dalam kondisi jaringan yang

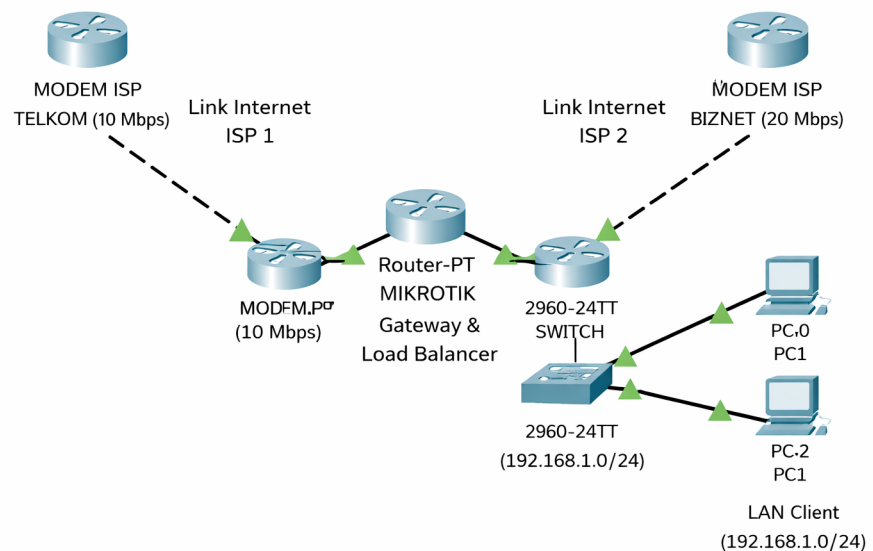
bervariasi, dengan distribusi beban yang lebih proporsional dan efisiensi yang lebih tinggi dalam penggunaan *bandwidth*[5].

## 2. Bahan dan Metode

Penelitian ini menggunakan pendekatan eksperimental kuantitatif untuk mengkaji perbandingan performa algoritma *Load balancing Round Robin* (RR) dan *Weighted Round Robin* (WRR) pada perangkat MikroTik. Penelitian dilakukan dalam lingkungan simulasi laboratorium dengan sistem topologi multi-WAN yang terdiri dari satu router MikroTik, dua jalur ISP, dan satu perangkat client. Kedua jalur ISP memiliki kapasitas berbeda, yakni 10 Mbps dan 20 Mbps, untuk mensimulasikan skenario nyata dalam pengaturan beban jaringan yang tidak seimbang.

### 2.1. Perancangan Topologi Jaringan

Topologi jaringan pada penelitian ini dirancang agar dapat digunakan untuk pengujian algoritma *Round Robin* dan *Weighted Round Robin* tanpa mengubah infrastruktur fisik jaringan. Gambar 1 menunjukkan topologi jaringan multi-WAN yang digunakan, di mana jaringan lokal (LAN) terhubung ke router MikroTik sebagai gateway dan load balancer utama. Router MikroTik selanjutnya terhubung ke dua jalur Internet yang berbeda, yaitu ISP 1 dengan kapasitas bandwidth 10 Mbps dan ISP 2 dengan kapasitas bandwidth 20 Mbps.



**Gambar 1.** Topologi Jaringan

Perbedaan penerapan algoritma *Round Robin* dan *Weighted Round Robin* tidak terletak pada perubahan topologi jaringan, melainkan pada konfigurasi logis yang diterapkan pada router MikroTik. Pada algoritma *Round Robin*, trafik dari jaringan lokal didistribusikan secara merata ke kedua jalur ISP tanpa mempertimbangkan perbedaan kapasitas bandwidth. Sementara itu, pada algoritma *Weighted Round Robin*, distribusi trafik dilakukan berdasarkan bobot 1:2 yang disesuaikan dengan kapasitas masing-masing jalur Internet, sehingga penggunaan bandwidth menjadi lebih proporsional dan efisien.

### 2.2. Konfigurasi Algoritma Round Robin pada Mikrotik

Konfigurasi algoritma *Round Robin* dilakukan untuk membagi trafik internet secara merata dengan rasio 50:50 ke dua jalur ISP tanpa mempertimbangkan perbedaan kapasitas bandwidth masing-masing. Pada implementasi ini, router MikroTik memanfaatkan fitur firewall mangle untuk melakukan penandaan koneksi (*connection-mark*) dan

menentukan jalur keluar trafik menggunakan routing-mark. Setiap koneksi baru yang berasal dari jaringan klien akan diarahkan secara bergiliran ke ISP 1 dan ISP 2.

Langkah-langkah konfigurasi yang dilakukan adalah sebagai berikut:

#### 1. IP Addressing:

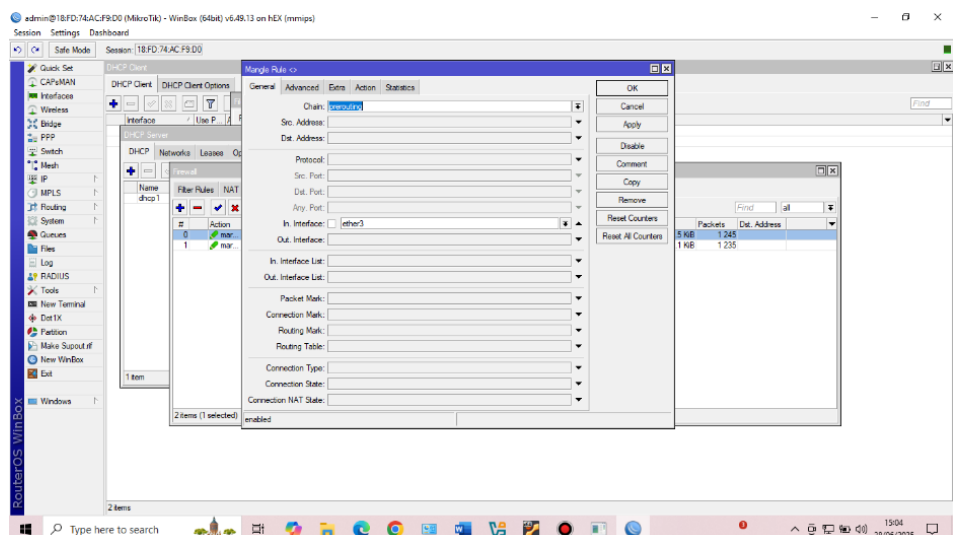
Alamat IP ditetapkan pada setiap antarmuka jaringan, baik pada antarmuka LAN maupun antarmuka yang terhubung ke masing-masing ISP. Konfigurasi ini memungkinkan router MikroTik berfungsi sebagai gateway bagi jaringan lokal dan penghubung ke dua jalur internet.

#### 2. DNS dan NAT:

DNS server publik ditambahkan pada router MikroTik untuk mendukung proses resolusi nama domain. Selain itu, konfigurasi Network Address Translation (NAT) dengan metode masquerade diterapkan pada masing-masing antarmuka ISP agar seluruh perangkat dalam jaringan lokal dapat mengakses internet melalui kedua jalur koneksi.

#### 3. Penandaan Koneksi dan Routing:

Penandaan koneksi dilakukan menggunakan fitur firewall mangle pada MikroTik. Rule mangle ditempatkan pada chain prerouting untuk memastikan setiap koneksi yang berasal dari jaringan LAN diproses sebelum proses routing dilakukan. Metode nth digunakan dengan parameter every = 2, di mana koneksi pertama diarahkan ke ISP 1 dan koneksi kedua diarahkan ke ISP 2 secara bergantian. Koneksi yang telah ditandai selanjutnya diberikan routing-mark sesuai dengan jalur ISP yang dituju. Implementasi konfigurasi firewall mangle untuk algoritma *Round Robin* ditunjukkan pada Gambar 2.



Gambar 2. Penandaan Koneksi

#### 4. Routing Table:

Default route ditambahkan berdasarkan routing-mark yang telah dibuat, masing-masing mengarah ke gateway ISP 1 dan ISP 2. Selain itu, satu default route tanpa routing-mark juga ditambahkan agar router MikroTik tetap dapat mengakses internet untuk keperluan manajemen sistem.

Konfigurasi algoritma *Round Robin* efektif digunakan pada jaringan dengan dua koneksi internet yang memiliki kapasitas relatif seimbang. Namun, pada kondisi jaringan dengan perbedaan bandwidth yang signifikan, metode ini berpotensi menimbulkan

bottleneck pada jalur dengan kapasitas lebih kecil karena algoritma *Round Robin* tidak mempertimbangkan kecepatan atau kapasitas masing-masing jalur koneksi.

### 2.3. Konfigurasi Algoritma *Weighted Round Robin* pada Mikrotik

Konfigurasi algoritma *Weighted Round Robin* (WRR) dilakukan untuk mendistribusikan trafik internet berdasarkan bobot kapasitas masing-masing jalur ISP. Pada penelitian ini digunakan dua ISP dengan rasio kapasitas 1:2, yaitu ISP1 sebesar 10 Mbps dan ISP2 sebesar 20 Mbps. Berdasarkan rasio tersebut, distribusi trafik diarahkan sehingga sekitar 1/3 koneksi dialokasikan ke ISP1 dan 2/3 koneksi dialokasikan ke ISP2.

Tahapan konfigurasi *Weighted Round Robin* yang diterapkan adalah sebagai berikut:

#### 1. IP Addressing:

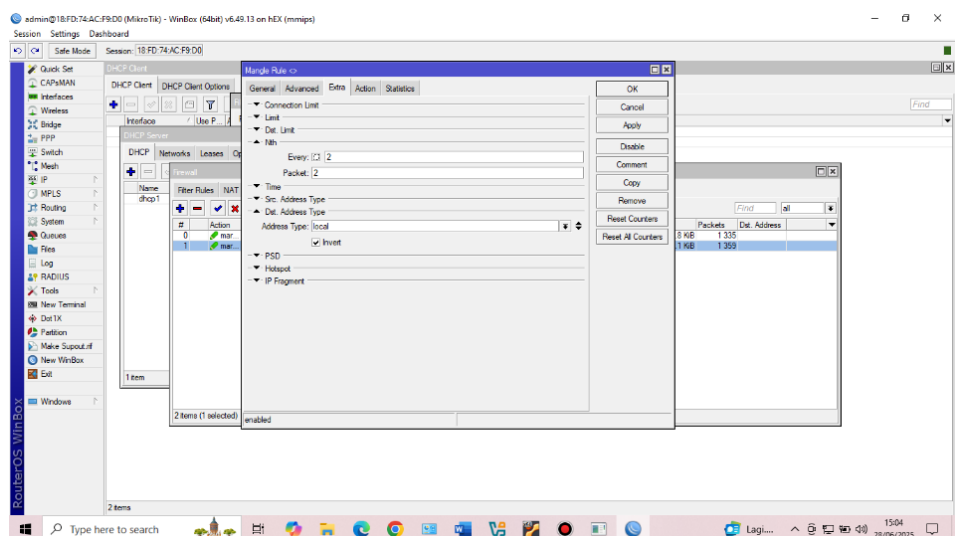
Alamat IP ditetapkan pada setiap antarmuka jaringan, baik antarmuka LAN maupun antarmuka yang terhubung ke masing-masing ISP. Konfigurasi ini sama dengan yang digunakan pada algoritma *Round Robin*, sehingga tidak diperlukan perubahan pada topologi jaringan fisik.

#### 2. DNS dan NAT:

DNS server publik ditambahkan pada router MikroTik untuk mendukung resolusi nama domain. Selain itu, konfigurasi Network Address Translation (NAT) dengan metode masquerade diterapkan pada kedua jalur ISP agar seluruh trafik dari jaringan lokal dapat mengakses internet.

#### 3. Penandaan Koneksi dengan Rasio Bobot:

Penandaan koneksi dilakukan menggunakan fitur firewall mangle pada MikroTik dengan metode nth. Untuk merepresentasikan rasio bobot 1:2, digunakan pola distribusi tiga koneksi, di mana satu koneksi diarahkan ke ISP1 dan dua koneksi berikutnya diarahkan ke ISP2. Beberapa rule mangle dibuat pada chain prerouting untuk menandai koneksi secara proporsional sesuai dengan bobot masing-masing ISP. Implementasi konfigurasi firewall mangle untuk algoritma *Weighted Round Robin* ditunjukkan pada Gambar 3.



Gambar 3. Penandaan koneksi

#### 4. Penandaan Routing dan Jalur:

Penandaan koneksi dilakukan menggunakan fitur firewall mangle pada MikroTik dengan metode nth. Untuk merepresentasikan rasio bobot 1:2, digunakan pola distribusi tiga koneksi, di mana satu koneksi diarahkan ke ISP1 dan dua koneksi berikutnya diarahkan ke ISP2.

Beberapa rule mangle dibuat pada chain prerouting untuk menandai koneksi secara proporsional sesuai dengan bobot masing-masing ISP. Implementasi konfigurasi firewall mangle untuk algoritma *Weighted Round Robin* ditunjukkan pada Gambar Y.

#### 5. Routing Table:

Default route ditambahkan berdasarkan routing-mark yang telah dibuat untuk masing-masing ISP. Selain itu, satu default route tanpa routing-mark juga ditambahkan agar router MikroTik tetap memiliki konektivitas internet untuk keperluan manajemen sistem.

Konfigurasi *Weighted Round Robin* memungkinkan router MikroTik mendistribusikan trafik lebih banyak ke jalur ISP dengan kapasitas bandwidth yang lebih besar, yaitu ISP2. Dengan pendekatan ini, risiko terjadinya bottleneck pada jalur berkapasitas lebih kecil dapat dikurangi, sehingga pemanfaatan bandwidth menjadi lebih efisien dibandingkan dengan algoritma *Round Robin*.

#### 2.4. Spesifikasi Hardware dan Software

Rincian dari spesifikasi Hardware dan Software yang di butuhkan dalam penelitian ini sebagaimana pada tabel 1 dan 2. Perangkat keras yang di gunakan seperti pada tabel 1 dan 2:

**Tabel 1.** Hardware

No	Hardware	Spesifikasi
1	Router Mikrotik	RB750Gr3
2	Laptop Client	CPU 4-core, RAM 8GB, SSD 100GB
3	Modem ISP 1	Tenda – 10 Mbps
4	Modem ISP 2	Tenda – 20 Mbps
5	Kabel UTP	Standart CAT5e

Sementara itu, untuk perangkat lunak yang di gunakan seperti tabel berikut:

**Tabel 2.** Software

No	Software	Fungsi
1	Mikrotik RouterOs	System Operasi Mikrotik
2	Winbox	Konfigurasi Grafis Mikrotik
3	Terminal atau Ping	Uji <i>Latency</i> Dan Packet Loss

#### 2.5. Parameter Pengujian

Parameter pengujian dilakukan selama 10 menit per scenario, dengan 100 request yang di kirim ke tujuan internet (8.8.8.8) untuk mengamati *Latency* dan packet loss. Selain itu, data performa jaringan seperti bandwidth dan CPU usage juga dikumpulkan secara manual dari interface mikrotik. Berikut adalah parameter pengujian yang digunakan dalam penelitian pada tabel 3:

**Tabel 3.** Parameter Pengujian

No	Parameter	Satuan	Deskripsi
1	Bandwidth	Mbps	Kecepatan Transfer Data
2	<i>Latency</i>	Ms	Waktu Tunda Pengiriman Data
3	Packet Loss	%	Presentase Packet Hilang Selama Transmisi
4	Jumlah Packet	Packet/s	Total Packet yang diproses per-detik
5	CPU Load	%	Beban kerja CPU Load di Mikrotik

Data dikumpulkan secara manual menggunakan fitur monitoring real-time di Winbox, dan hasilnya dianalisis secara deskriptif-komparatif. Teknik analisis ini dilakukan dengan membandingkan nilai setiap parameter dari kedua algoritma. Hasil analisis bertujuan untuk mengetahui efisiensi pembagian beban trafik dan kestabilan performa

jaringan. Selain itu, kelebihan dan kekurangan dari masing-masing algoritma juga dianalisis berdasarkan kompleksitas konfigurasi dan efisiensi sumber daya.

Penelitian ini tidak melibatkan subjek manusia atau hewan, sehingga tidak memerlukan persetujuan etik. Semua konfigurasi, data pengujian, dan protocol tersedia dalam bentuk dokumentasi teknis dan dapat direplikasi oleh pihak lain dengan perangkat dan kondisi yang setara.

### 3. Hasil

#### 3.1. Hasil Implementasi

Penelitian ini dilakukan untuk mengevaluasi kinerja dua algoritma *Load balancing*, yakni algoritma *Round Robin* dan algoritma *Weighted Round Robin* dalam lingkungan multi-WAN menggunakan perangkat mikrotik. Implementasi dilakukan pada dua jalur ISP dengan kapasitas yang berbed, yaitu 10 Mbps dan 20 Mbps. Konfigurasi *Round Robin* membagi trafik secara merata tanpa mempertimbangkan kapasitas pada tiap jalur ISP, sedangkan *Weighted Round Robin* mendistribusikan trafik berdasarkan bobot, sesuai dengan perbandingan kapasitas bandwidth antar jalur dengan rasio(1:2).

Pengujian dilakukan selama 10 menit untuk masing-masing algoritma dengan jumlah request yang sama, yaitu sebanyak 100 permintaan koneksi. Parameter performa yang diamati mencakup *Packet Loss*, *Latency* (waktu tunda), bandwidth maximum (TX rate), dan CPU load dari interface perangkat mikrotik. Data diambil secara langsung melalui interface winbox untuk mendapatkan gambaran real-time mengenai kinerja jaringan dalam dua scenario yang di uji.

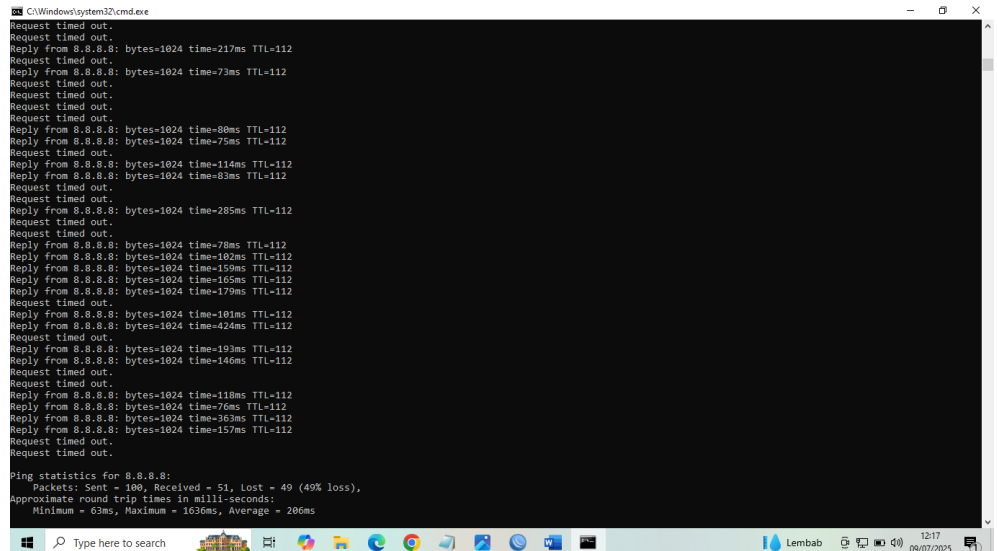
```

C:\Windows\system32\cmd.exe
Reply from 8.8.8.8: bytes=300 time=81ms TTL=112
Reply from 8.8.8.8: bytes=300 time=276ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=58ms TTL=112
Reply from 8.8.8.8: bytes=300 time=74ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=246ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=69ms TTL=112
Reply from 8.8.8.8: bytes=300 time=77ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=308ms TTL=112
Reply from 8.8.8.8: bytes=300 time=74ms TTL=112
Reply from 8.8.8.8: bytes=300 time=74ms TTL=112
Reply from 8.8.8.8: bytes=300 time=151ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=106ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=108ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=72ms TTL=112
Reply from 8.8.8.8: bytes=300 time=154ms TTL=112
Reply from 8.8.8.8: bytes=300 time=219ms TTL=112
Reply from 8.8.8.8: bytes=300 time=109ms TTL=112
Reply from 8.8.8.8: bytes=300 time=92ms TTL=112
Reply from 8.8.8.8: bytes=300 time=169ms TTL=112
Request timed out.
Reply from 8.8.8.8: bytes=300 time=124ms TTL=112
Reply from 8.8.8.8: bytes=300 time=224ms TTL=112
Reply from 8.8.8.8: bytes=300 time=109ms TTL=112
Reply from 8.8.8.8: bytes=300 time=92ms TTL=112
Reply from 8.8.8.8: bytes=300 time=51ms TTL=112
Reply from 8.8.8.8: bytes=300 time=64ms TTL=112
Reply from 8.8.8.8: bytes=300 time=122ms TTL=112
Reply from 8.8.8.8: bytes=300 time=224ms TTL=112
Reply from 8.8.8.8: bytes=300 time=212ms TTL=112
Reply from 8.8.8.8: bytes=300 time=144ms TTL=112
Reply from 8.8.8.8: bytes=300 time=207ms TTL=112

Ping statistics for 8.8.8.8:
    Packets: Sent = 100, Received = 74, Lost = 26 (26% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 51ms, Maximum = 378ms, Average = 133ms
  
```

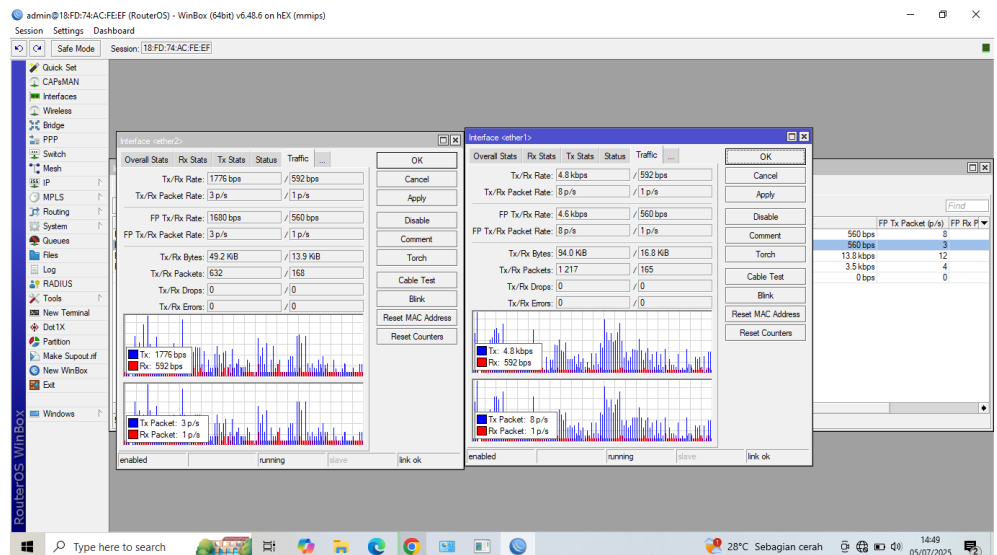
Gambar 4. Pengujian algoritma *weighted Round Robin*

Gambar 4 menunjukkan hasil pengujian koneksi menggunakan perintah ping ke server Google DNS (8.8.8.8) menunjukkan bahwa dari 100 paket data yang dikirim, hanya 74 paket yang berhasil diterima, sedangkan 26 paket hilang sehingga terjadi packet loss sebesar 26%. Waktu respon yang diperoleh bervariasi dengan nilai minimum 51 ms, maksimum 378 ms, dan rata-rata 133 ms.



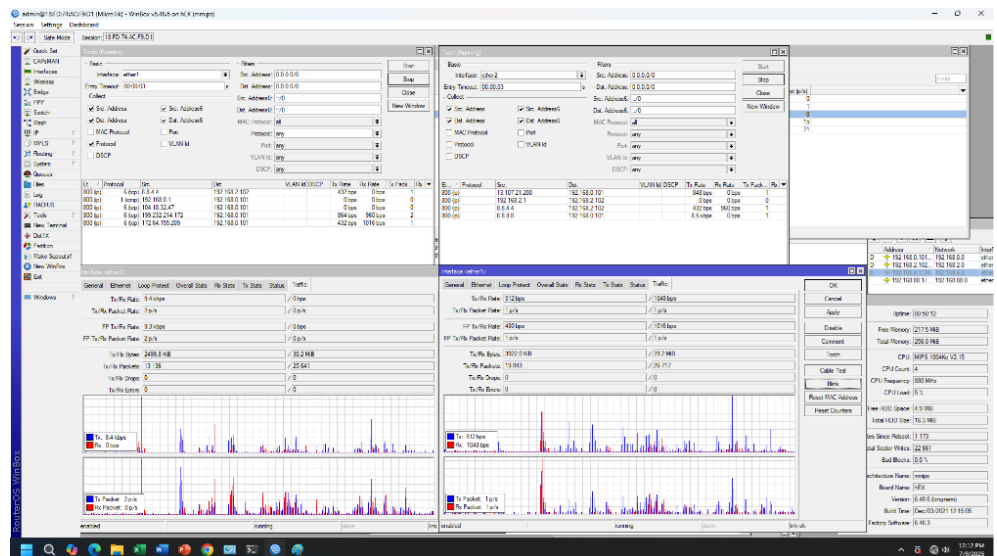
Gambar 5 Pengujian algoritma Round Robin

Gambar 5 menunjukkan hasil pengujian koneksi pada gambar menunjukkan ping ke server Google DNS (8.8.8.8) dengan ukuran paket 1024 byte. Dari total 100 paket data yang dikirim, hanya 51 paket yang berhasil diterima dan 49 paket hilang, sehingga terjadi packet loss sebesar 49%. Waktu respon (*Latency*) juga sangat fluktuatif, dengan nilai minimum 63 ms, maksimum 1636 ms, dan rata-rata 206 ms.



Gambar 6 Pengujian algoritma weighted Round Robin pada Winbox

Gambar 6 menunjukkan hasil dari pengujian algoritma *weighted Round Robin* pada interface winbox, dengan kondisi jaringan pada ether 1 dan ether 2 menunjukkan trafik jaringan bandwidth yang rendah dan itu menunjukkan bahwa kedua interface berjalan dengan normal, namun dengan aktivitas trafik yang minimum.



Gambar 7 Pengujian algoritma Round Robin pada winbox

Gambar 7 menunjukkan hasil dari pengujian algoritma Round Robin pada interface winbox dan kondisi ether 1 dan ether 2 berjalan dengan normal, namun dengan aktivitas bandwidth yang rendah.

Tabel 4 menyajikan data hasil dari performa antara algoritma Round Robin dengan algoritma weighted Round Robin sesuai dengan parameter utama yang diuji:

Tabel 4 Hasil Parameter Pengujian

Parameter	Round Robin ISP 1	Round Robin ISP 2	Weighted Round Robin ISP 1	Weighted Round Robin ISP 1
Packet Loss	68 %	76 %	11 %	15 %
Latency	77 ms	85 ms	62 ms	66 ms
Bandwidth	21.3 Kbps	18.3 Kbps	24.5 Kbps	23.5 Kbps
CPU Load	3 %	4 %	1 %	2 %
Max Request	6.5 %	5 %	5 %	5 %

Berdasarkan data dari tabel 4, terlihat bahwa algoritma weighted Round Robin secara konsisten memberikan hasil performa yang lebih baik dibandingkan dengan algoritma Round Robin di semua parameter yang diuji. Algoritma weighted Round Robin menunjukkan packet loss yang lebih rendah hingga 4-6 kali dibanding Round Robin, Latency yang lebih stabil, dan bandwidth yang digunakan secara optimal. Selain itu, penggunaan CPU weighted Round Robin juga lebih efisien meskipun konfigurasi yang digunakan lebih kompleks. Hal ini menjelaskan bahwa algoritma weighted Round Robin lebih sesuai diterapkan pada jaringan dengan jalur ISP yang memiliki kapasitas tidak seimbang, karena mampu mendistribusikan beban berdasarkan proporsi kapasitas yang ada.

### 3.2. Interpretasi Hasil

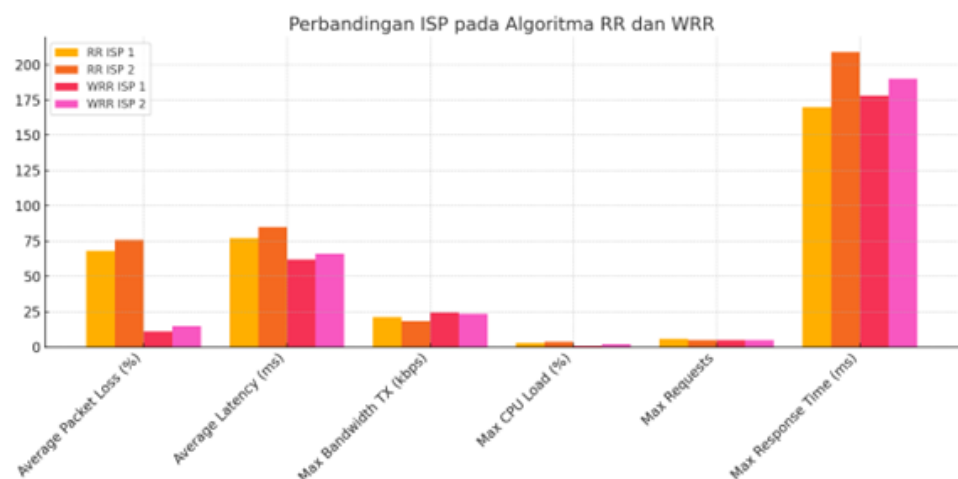
Hasil pengujian menunjukkan bahwa algoritma weighted Round Robin memberikan kinerja yang lebih baik dibanding dengan algoritma Round Robin dalam scenario jaringan yang menggunakan jalur dengan kapasitas tidak seimbang. Packet Loss pada algoritma Round Robin tercatat sangat tinggi, mencapai 68% di ISP 1 dan 76% pada ISP 2. Hal ini menunjukkan bahwa pembagian trafik secara merata tanpa memperhitungkan kapasitas jalur dan menyebabkan bottleneck, terutama pada jalur dengan bandwidth rendah. Sebaliknya, weighted Round Robin yang menerapkan distribusi berbasis bobot mampu menurunkan packet loss secara signifikan menjadi 11% dan 15% untuk ISP 1 dan ISP 2 secara berturut-turut.

Dalam hal Latency, weighted Round Robin menunjukkan performa yang lebih stabil, dengan rata-rata waktu tunda 62ms dan 66ms, lebih baik dibandingkan Round Robin yang

mencatatkan 77ms dan 85ms. Pengurangan *Latency* ini menunjukkan bahwa distribusi trafik yang lebih proporsional memungkinkan aliran data berjalan lebih lancar dan tidak menumpuk di jalur yang sempit.

Selain itu, bandwidth maksimum yang tercapai pada *weighted Round Robin* lebih tinggi dibandingkan *Round Robin*, yakni 24.5 Kbps dan 23.5 Kbps untuk ISP 1 dan ISP 2. Ini menandakan bahwa *weighted Round Robin* mampu memaksimalkan pemanfaatan bandwidth pada jalur dengan kapasitas lebih besar. Menariknya, meskipun konfigurasi *weighted Round Robin* lebih kompleks karena membutuhkan lebih banyak rule routing, penggunaan CPU pada mikrotik justru lebih ringan, hanya mencapai 1% dan 2%, dibandingkan dengan *Round Robin* yang memerlukan 3% dan 4%.

Gambar 3 menggambarkan secara visual grafik dari semua parameter yang diuji pada kedua algoritma.



Gambar 8 Grafik Perbandingan

Visualisasi grafik dapat memperkuat data tabel, Dimana packet loss pada *Round Robin* terlihat jauh lebih tinggi dan fluktuatif, sedangkan *weighted Round Robin* menghasilkan grafik yang lebih stabil dan rendah. Demikian pula pada *Latency*, *Round Robin* memperlihatkan puncak-puncak delay yang ekstrem, sedangkan *weighted Round Robin* mempertahankan kestabilan waktu respons koneksi.

#### 4. Pembahasan

Hasil yang diperoleh dari penelitian ini menunjukkan bahwa algoritma *weighted Round Robin* memiliki performa yang lebih baik secara signifikan dibandingkan algoritma *Round Robin* dalam konteks jaringan multi-WAN dengan kapasitas jalur yang tidak seimbang. Hasil ini memperkuat hipotesis awal bahwa *weighted Round Robin* yang menggunakan mekanisme pembobotan lebih efisien dalam mengatur trafik, terutama dalam menghindari bottleneck pada jalur berkapasitas kecil. Penurunan packet loss dan *Latency* secara drastic pada konfigurasi *weighted Round Robin* menunjukkan bahwa pemanfaatan jalur yang lebih proporsional terhadap kapasitas dapat memberikan stabilitas yang lebih tinggi dan efisiensi dalam penggunaan bandwidth.

Agar hasil penerapan dari masing-masing algoritma berjalan sesuai dengan fungsinya, terdapat beberapa parameter penting yang memerlukan perhatian dan pengaturan yang tepat. Parameter utama yang harus dikendalikan meliputi kapasitas bandwidth setiap jalur yang menjadi dasar dalam penentuan bobot pada algoritma *weighted Round Robin*, sebab kesalahan dalam penyesuaian bobot dapat menyebabkan distribusi trafik yang tidak proporsional. Selain itu, faktor seperti ukuran paket, *Latency*, jitter, serta tingkat packet loss pada tiap jalur juga berpengaruh besar terhadap kestabilan koneksi dan efisiensi transmisi data. Pengelolaan antrian paket (*queue management*) serta

metode *load balancing* yang digunakan pun perlu disesuaikan agar tidak terjadi bottleneck maupun pemutusan sesi koneksi. Dengan pengaturan parameter-parameter tersebut secara cermat dan pemantauan performa jaringan secara berkelanjutan, penerapan algoritma *Round Robin* maupun *weighted Round Robin* dapat memberikan hasil yang optimal sesuai dengan karakteristik jaringan multi-WAN yang digunakan.

Dari perspektif penelitian sebelumnya, temuan ini mendukung hasil penelitian oleh (Priya dan Rajendra 2025) yang menunjukkan bahwa *weighted Round Robin* mampu mengurangi waktu tunda dan meningkatkan throughput pada system cloud. Selain itu, hasil ini juga konsisten dengan penelitian oleh (hanifah 2021) yang mengaplikasikan *weighted Round Robin* untuk beban webserver dan menemukan adanya peningkatan performa yang signifikan. Dengan demikian, penelitian ini tidak hanya mengkonfirmasi validitas pendekatan *weighted Round Robin*, tetapi juga memberikan pembuktian empiris tambahan dalam konteks implementasi jaringan mikrotik.

Implikasi praktis dari penelitian ini menunjukkan bahwa administrator jaringan dapat mengadopsi konfigurasi *weighted Round Robin* terutama pada infrastruktur yang mengandalkan dua ISP dengan kecepatan yang berbeda. Meskipun memerlukan konfigurasi yang lebih kompleks, dan hasil yang diperoleh sebanding dengan stabilitas dan efisiensi trafik jaringan. Selain itu, metode pengamatan yang ada pada Gambar 4 sampai Gambar 7 menunjukkan hasil dari pengamatan antarmuka winbox terbukti cukup akurat untuk mendapatkan data performa jaringan secara real-time.

Untuk pengembangan penelitian di masa mendatang, disarankan agar dilakukan komparasi dengan algoritma *Load balancing* lainnya seperti *Per Connection Classifier (PCC)*, *NTH*, atau bahkan pendekatan berbasis AI dalam routing otomatis. Selain itu, perlu juga dilakukan pengujian pada jaringan skala besar atau lingkungan real-world untuk menguji skalabilitas konfigurasi *weighted Round Robin*.

## 5. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Round Robin* dan *Weighted Round Robin* menunjukkan karakteristik kinerja yang berbeda pada jaringan multi-WAN dengan kapasitas jalur Internet yang tidak seimbang, yaitu ISP1 sebesar 10 Mbps dan ISP2 sebesar 20 Mbps.

Algoritma *Round Robin*, yang mendistribusikan trafik secara merata tanpa mempertimbangkan kapasitas masing-masing jalur, menunjukkan performa yang kurang optimal pada skenario ini. Hasil pengujian memperlihatkan nilai packet loss yang tinggi, *Latency* yang fluktuatif, serta pemanfaatan bandwidth yang tidak efisien, terutama pada jalur dengan kapasitas lebih kecil (10 Mbps). Kondisi ini terjadi karena beban trafik dibagi secara sama rata, sehingga jalur dengan bandwidth rendah lebih cepat mengalami bottleneck.

Sebaliknya, algoritma *Weighted Round Robin* mampu memberikan kinerja yang lebih baik dengan mendistribusikan trafik berdasarkan rasio kapasitas bandwidth antar ISP, yaitu 1:2. Pada konfigurasi ini, jalur ISP dengan kapasitas 20 Mbps menerima porsi trafik yang lebih besar dibandingkan jalur 10 Mbps. Hasil pengujian menunjukkan bahwa *Weighted Round Robin* menghasilkan penurunan packet loss hingga 4–6 kali dibandingkan *Round Robin*, *Latency* yang lebih stabil, serta pemanfaatan bandwidth yang lebih optimal pada kedua jalur Internet. Selain itu, penggunaan sumber daya CPU pada router MikroTik juga relatif lebih efisien meskipun konfigurasi yang diterapkan lebih kompleks.

Dengan demikian, dapat disimpulkan bahwa algoritma *Weighted Round Robin* lebih sesuai untuk diterapkan pada jaringan multi-WAN dengan perbedaan kapasitas bandwidth yang signifikan, seperti pada koneksi Internet 10 Mbps dan 20 Mbps. Sementara itu, algoritma *Round Robin* masih dapat digunakan pada jaringan dengan kapasitas jalur

yang relatif seimbang, namun kurang direkomendasikan untuk kondisi jaringan dengan perbedaan bandwidth yang besar.

## Referensi

- [1] T. S. N. Azhar, "Perimbangan Beban Web Server Menggunakan Metode *Weighted Round Robin* algoritma *Round Robin* pada PT," *Jurnal Sains dan Teknologi*, vol. 5, no. 1, hlm. 183–192, 2023, <https://doi.org/10.55338/saintek.v5i1.1403>.
- [2] S. S. Priya dan T. Rajendran, "Enhanced *Weighted Round Robin*: A New Paradigm in Cloud *Load balancing*," *Indian J Sci Technol*, vol. 18, no. 15, hlm. 1220–1228, Mei 2025, <https://doi.org/10.17485/IJST/v18i15.3976>.
- [3] K. A. Sundara, H. Aspriyono, dan R. Supardi, "Perancangan Manajemen Bandwidth Menggunakan Mikrotik Router Wireless Pada Sekolah Menengah Kejuruan Negeri 4 Kota Bengkulu," *Jurnal Media Infotama*, vol. 18, no. 2, hlm. 279–290, 2022, <https://doi.org/10.37676/jmi.v18i2.2747>.
- [4] S. Safriadi dan R. Rahmadani, "Analisis Kinerja *Load balancing Round Robin* Pada Website Skalabel," *Jurnal of Information System Management (JOISM)*, vol. 5, no. 2, hlm. 227–232, 2024, <https://doi.org/10.24076/joism.2024v5i2.1441>.
- [5] A. Hanafiah, "Implementasi *Load balancing* Dengan Algoritma Penjadwalan *Weighted Round Robin* Dalam Mengatasi Beban Webserver," *IT Journal Research and Development*, vol. 5, no. 2, hlm. 226–233, Jan 2021, [https://doi.org/10.25299/itjrd.2021.vol5\(2\).5795](https://doi.org/10.25299/itjrd.2021.vol5(2).5795).
- [6] E. B. Pabelan, I. K. Sudaryana, "Implementasi *Load balancing* Menggunakan Metode NTH Pada PT. Zyrexindo Mandiri Buana Untuk Optimalisasi Jaringan," *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 8, no. 3, hlm. 2562–2567, 2024, <https://doi.org/10.35870/jtik.v8i3.1257>.
- [7] M. S. Pradana dan A. Prapanca, "Analisis Performa *Load balancing* Algoritma *Weighted Round Robin* di Infrastruktur BPBD Provinsi Jawa Timur" *Journal of Informatics and Computer Science (JINACS)*, vol. 1, no. 02, hlm. 109–114, 2020, <https://doi.org/10.26740/jinacs.v1n02.p109-114>.
- [8] 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)," *Journal of Informatics and Computer Science (JINACS)*, vol. 4, no. 01, hlm. 67–75, Jul 2022, <https://doi.org/10.26740/jinacs.v4n01.p67-75>.
- [9] H. K. Ramadhan dan S. Wardhana, "Seminar Nasional Pengaplikasian Telematika (SINAPTIKA) Optimalisasi Jaringan Komputer Menggunakan Algoritma *Load balancing* di Virtual Server Citrix ADC" Dalam Seminar Nasional Pengaplikasian Telematika (SINAPTIKA), vol. 1, no. 1, hlm. 137–146, 2021 <https://publikasi.mercubuana.ac.id/index.php/sinaptika/article/view/11428>.
- [10] A. Zakir, S. A. Dalimunthe, dan D. Irwan, "Penerapan Algoritma *Round Robin* pada Penjadwalan Preventive Maintenance di PT. Pasifik Satelit Nusantara," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 3, no. 2, hlm. 54, Jan 2021, <https://doi.org/10.37600/tekinkom.v3i2.142>.
- [11] M. Darip, S. Auliana, dan B. Rakhim Setya Permana, "Implementation of *Round Robin* Algorithm in Public Transportation Scheduling System at Pakupatan Terminal in Serang City-Indonesia," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 14, no. 2, Mei 2025, <https://doi.org/10.32736/sisfokom.v14i2.2362>.
- [12] I. M. Ibrahim., "Web Server Performance Improvement Using Dynamic *Load balancing* Techniques: A Review," *Asian Journal of Research in Computer Science*, hlm. 47–62, Jun 2021, <https://doi.org/10.9734/ajrcos/2021/v10i130234>.
- [13] R. Oktariyadi, I. Ruslianto, dan S. Bahri, "Analisa Kinerja *Load balancing* Menggunakan Metode *Round Robin* Dan *Weighted Round Robin*," *Coding: Jurnal Komputer dan Aplikasi*, vol. 9, no. 1, hlm. 131–141, 2021, <https://doi.org/10.26418/coding.v9i01.45871>.
- [14] T. D. Putra dan R. Purnomo, "Analisis Algoritma *Round Robin* pada Penjadwalan CPU," *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 15, no. 2, hlm. 85–90, Sep 2021, <https://doi.org/10.32815/jitika.v15i2.481>.
- [15] B. Arifwidodo, V. Metayasha, dan S. Ikhwan, "Analisis Kinerja *Load balancing* pada Server Web Menggunakan Algoritma *Weighted Round Robin* pada Proxmox VE," *Jurnal Telekomunikasi dan Komputer*, vol. 11, no. 3, hlm. 210, Des 2021, <https://doi.org/10.22441/incomtech.v11i3.11775>.