

# Optimasi Jaringan RT/RW-Net Menggunakan Software Defined Network Dan Load Balancing

Muhamad Fikri<sup>1\*</sup>, Fauzan Masykur<sup>1</sup>, Adi Fajaryanto Cobantoro<sup>1</sup>

<sup>1</sup>Prodi Teknik Informatika Fakultas Teknik  
Universitas Muhammadiyah Ponorogo  
Jl. Budi Utomo no.10 Ponorogo  
Indonesia

\*Corresponding Author Email: [muh44madfikri@gmail.com](mailto:muh44madfikri@gmail.com)

*Abstract: Meningkatnya kebutuhan akan akses internet yang stabil dan cepat, terutama di lingkungan RT/RW Net yang melayani pelanggan rumah tangga dengan berbagai kebutuhan seperti streaming dan pengunduhan, memunculkan tantangan dalam pengelolaan lalu lintas jaringan, terutama ketika menggunakan lebih dari satu penyedia layanan internet (ISP). Penelitian ini mengusulkan penerapan arsitektur Software Defined Networking (SDN) sebagai solusi untuk mengatasi masalah distribusi trafik yang tidak merata melalui mekanisme load balancing. SDN memberikan kontrol terpusat dan fleksibel dalam pengelolaan jaringan, memungkinkan pengaturan trafik secara dinamis dan efisien. Implementasi dilakukan dengan menggunakan controller OpenDaylight dan perangkat MikroTik sebagai router yang dikonfigurasi mendukung protokol OpenFlow untuk load balancing dua ISP. Hasil pengujian menunjukkan adanya peningkatan pada parameter Quality of Service (QoS) seperti throughput, delay, jitter, dan packet loss. Dengan penerapan SDN dan load balancing setelah dilakukan pengujian dengan QoS didapat hasil rata-rata dari masing-masing parameter yaitu, untuk throughput dengan rata-rata 8,36 mbps, untuk parameter packet loss didapat hasil rata-rata 0.09%, untuk delaynya dengan rata-rata 39 ms dan parameter jitter dengan rata-rata 12,67 ms. Dari semua hasil pengujian parameter diatas membuktikan adanya kenaikan kualitas jaringan dibandingkan dengan sebelum menerapkan SDN dan load balancing.*

*Key-Words: Jaringan Komputer, RT/RW Net, Software Defined Network, Load Balancing, Internet Service Provider, Quality of Service.*

Received: July 16, 2025. Revised: July 27, 2025. Accepted: July 29, 2025. Published: August 10, 2025

## 1 Pendahuluan

Seiring meningkatnya kebutuhan warga masyarakat akan akses internet yang cepat dan stabil, penyedia layanan internet dituntut untuk terus meningkatkan kualitas layanannya. Salah satu permasalahan yang dihadapi oleh penyedia layanan internet yakni terkait dengan cara mengelola lalu lintas internet yang semakin meningkat dan memastikan bahwa semua pelanggan mendapatkan kualitas layanan yang sama. Peningkatan permintaan akan akses internet yang berkualitas telah mendorong penyedia layanan internet untuk mencari solusi yang lebih efisien dalam mengelola jaringan[1].

Khususnya di daerah-daerah dengan topografi yang menantang seperti pegunungan, ketersediaan layanan internet yang stabil dan berkualitas tinggi masih menjadi tantangan. Penelitian ini berfokus pada peningkatan kualitas layanan internet di jaringan RT/RW net yang menggunakan lebih dari satu ISP[2].

Jaringan RT/RW yang menjadi objek penelitian ini umumnya melayani pelanggan perumahan yang kebutuhan utamanya adalah streaming video, menonton video, dan mengunduh data atau file dari internet. Namun, seringkali pengguna mengalami kendala seperti kecepatan akses yang lambat, latensi tinggi, dan koneksi yang tidak stabil. Situasi ini terjadi akibat sejumlah faktor, salah satunya adalah keterbatasan kapasitas jaringan, fluktuasi trafik yang tinggi, serta gangguan fisik pada infrastruktur jaringan seperti putus kabel akibat kondisi geografis yang ekstrem karena berada di wilayah pegunungan[3].

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan implementasi *Software Defined Networking* (SDN) untuk *load balancing*. SDN dipilih karena menawarkan fleksibilitas, visibilitas, dan kemampuan otomatisasi yang tinggi dalam mengelola jaringan. Dengan ini, diharapkan distribusi lalu lintas internet dapat dilakukan secara

optimal sehingga dapat meningkatkan kecepatan akses, mengurangi latensi, dan meningkatkan stabilitas koneksi[4].

Penelitian ini bertujuan untuk mengimplementasikan *load balancing* berbasis *Software Defined Networking* (SDN) pada jaringan penyedia akses internet dengan lebih dari satu *Internet Service Provider* (ISP). Melalui pendekatan SDN, diharapkan dapat meningkatkan kualitas layanan, meminimalkan latensi, dan mengoptimalkan penggunaan bandwidth. Penelitian ini melibatkan perancangan arsitektur SDN, implementasi *controller* SDN, dan konfigurasi aplikasi *load balancing*[5].

Hasil evaluasi menunjukkan bahwa sistem *load balancing* berbasis SDN berhasil meningkatkan kinerja jaringan secara signifikan. *Software defined networking* (SDN) merupakan sebuah pendekatan arsitektural yang inovatif dengan memisahkan fungsi kontrol dari fungsi data untuk menyederhanakan dan mempercepat pengelolaan jaringan besar. Ini berarti logika kontrol telah dipindahkan dari level perangkat keras jaringan ke level manajemen kontrol terpusat[6].

Tingkat kualitas dari sebuah jaringan komputer sangat bergantung pada penerapan *Quality of Service* (QoS). QoS digunakan dalam sistem jaringan guna membantu pengelola memberikan layanan internet yang optimal dan merata kepada seluruh pengguna atau pelanggan[7].

Teknologi ini memungkinkan untuk mengelola dampak kemacetan lalu lintas jaringan pada aliran paket data. QoS menjadi metode untuk menilai seberapa baik performa jaringan serta berupaya mendefinisikan karakteristik dan sifat dari layanan jaringan[8].

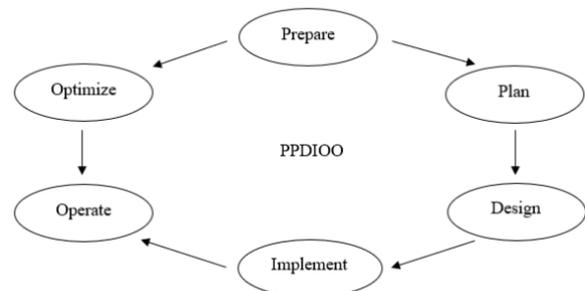
Performa jaringan komputer dapat dipengaruhi oleh berbagai faktor seperti *bandwidth*, *latency*, dan *jitter*, yang dapat berdampak besar pada sejumlah aplikasi dalam jaringan. QoS digunakan untuk mengukur baik atau buruknya performa jaringan. Dengan teknologi ini, pengelola jaringan dapat mengatasi efek kemacetan dalam lalu lintas paket data. Beberapa parameter QoS meliputi *delay*, *jitter*, *packet loss*, dan *throughput*. *Quality of Service* (QoS) menjamin kualitas layanan yang optimal dengan mengalokasikan bandwidth guna menangani masing-masing parameter tersebut[9].

Dalam upaya menjaga dan meningkatkan QoS, dibutuhkan metode pengelolaan jaringan yang dapat menggolongkan dan mengutamakan informasi sesuai karakteristik masing-masing jaringan, maka dari itu penelitian implementasi SDN untuk *load balancing* pada jaringan RT/RW net ini perlu dilakukan[10].

## 2 Metode Penelitian

### 2.1 Tahapan Penelitian

Metode PPDIIO dipilih sebagai pendekatan karena perancangan jaringan menitikberatkan pada kebutuhan pengguna. Metode ini menyediakan panduan penting yang mencakup seluruh tahapan dalam proses perancangan jaringan, mulai dari persiapan, perencanaan, desain, implementasi, operasional, hingga optimalisasi. Selain itu, PPDIIO mengarahkan struktur jaringan agar mampu menyesuaikan diri dengan berbagai aplikasi yang dibutuhkan oleh jaringan tersebut.



Gambar 1. Metode PPDIIO

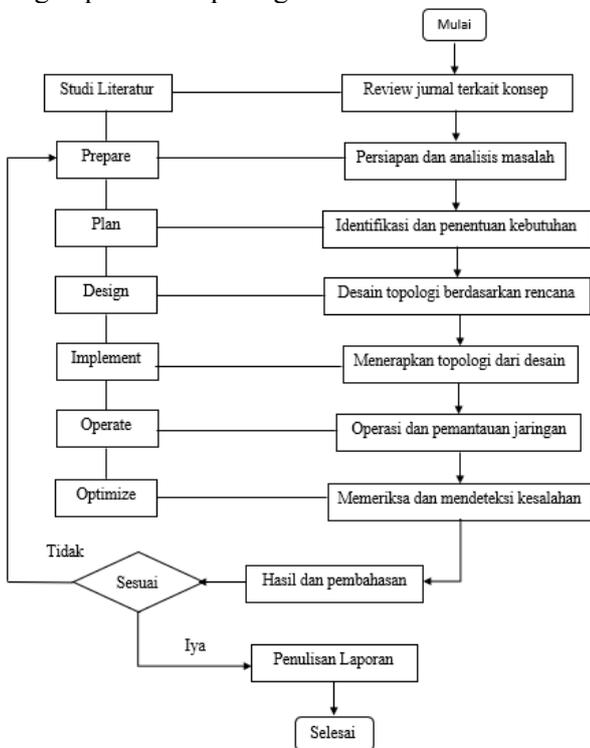
Gambar 1 memperlihatkan alur dari metode PPDIIO. Perancangan jaringan komputer dengan metode PPDIIO dilakukan melalui enam tahap, yaitu Prepare (persiapan), Plan (perencanaan), Design (perancangan), Implement (implementasi), Operate (operasi), dan Optimize (optimasi). Metode PPDIIO ini dipilih dikarenakan sangat terstruktur dan sistematis untuk membuat kerangka kerja dalam penelitian ini. Penjelasan singkat dari tahapan metode PPDIIO :

- Tahap pertama, Prepare (persiapan) yaitu diawali dengan mengidentifikasi kebutuhan dan informasi yang diperlukan untuk membangun sistem jaringan.
- Pada tahap Plan (perencanaan), setelah kebutuhan sistem teridentifikasi, selanjutnya melakukan analisis tentang kebutuhan perangkat keras, perangkat lunak, dan metode yang akan digunakan.
- Pada tahap Design (perancangan), yaitu mulai membuat desain jaringan dan sistem berdasarkan analisis yang telah dilakukan pada tahap perencanaan. Seperti membuat diagram jaringan, tata letak jaringan dan dokumentasi teknis.
- Tahap implementasi mencakup pengembangan sistem yang sesuai dengan desain. Semua perangkat keras maupun lunak mulai dikonfigurasi sesuai rancangan sebelumnya.
- Di tahap operasi, sistem yang sudah diimplementasikan mulai beroperasi untuk

memastikan semua komponen berjalan sesuai yang diharapkan. Dilakukan pula pemantauan untuk mendeteksi masalah yang mungkin akan terjadi.

- f. Tahap optimasi melibatkan evaluasi kinerja sistem dengan melakukan identifikasi dan uji coba untuk memperbaiki kesalahan yang muncul.

Dengan mengikuti setiap tahap secara berurutan, diharapkan penelitian dapat berjalan dengan lancar dan mencapai hasil yang diinginkan. Evaluasi dan optimasi yang berkelanjutan memastikan penelitian tetap relevan dan bermanfaat. Untuk lebih jelasnya bagaimana penelitian ini berjalan penulis membuat flowchart alur penelitian yang dapat dilihat pada gambar 2.



Gambar 2. Flowchart Alur Penelitian

### 3 Hasil dan Pembahasan

#### 3.1 Implementasi Sistem

Pada bagian ini dijelaskan langkah-langkah implementasi SDN dan konfigurasi *load balancing*. Proses instalasi SDN, konfigurasi *flow table* pada *router*, dan pengaturan *load balancing*. *Screenshot* konfigurasi dan penjelasan perintah konfigurasi disertakan untuk mendukung validitas implementasi.

##### 3.1.1 Instalasi SDN

*Instalasi Software Defined Networking* adalah proses menyiapkan perangkat lunak pengontrol jaringan yang bertugas memanajemen

dan mengontrol perilaku seluruh jaringan SDN melalui *interface* pemrograman (API) dan protokol manajemen seperti *OpenFlow*. *Controller* SDN merupakan otak dalam arsitektur SDN, yang memisahkan *control plane* dari *data plane* dengan perangkat jaringan seperti *switch* dan *router*. Penulis menggunakan *controller OpenDaylight*, Berikut langkah-langkah instalasi *controller SDN OpenDaylight*.

- a. Update Repositori

Pada tahap ini, kita memastikan bahwa sistem Ubuntu Desktop sudah menggunakan versi terbaru dari semua paket yang diperlukan. Dengan menjalankan *sudo apt update*, sistem akan memperbarui daftar repositori dan menginstal pembaruan terbaru. Hal ini penting untuk mencegah potensi konflik atau error ketika menginstal software baru. Ketik perintah pada terminal linux seperti gambar 3.

```
rendranet@SDN:~$ sudo apt update
```

Gambar 3. Apt Update

- b. Unduh Java JDK

Selanjutnya, *OpenDaylight* memerlukan *Java Development Kit* (JDK), jadi kita pastikan Java 21 sudah terpasang menggunakan perintah *java -version*. Jika belum ada, kita instal *OpenJDK 21* yang kompatibel dengan *OpenDaylight*. Ketik perintah pada terminal linux seperti gambar 4.

```
rendranet@SDN:~$ sudo apt install openjdk-11-jdk
```

Gambar 4. Install Java Jdk

- c. Unduh File Karaf

Setelah sistem siap, langkah berikutnya adalah mengunduh file distribusi *OpenDaylight*. Dengan masuk ke direktori home (*cd ~*), kita menjaga agar file distribusi tersimpan dengan rapi. Kemudian, perintah *wget* digunakan untuk mengunduh *file tar.gz* distribusi *OpenDaylight* dari server resmi. Penulis menggunakan *OpenDaylight* versi *Karaf-0.21.2*. Ketik perintah pada terminal linux diikuti alamat link download file karaf seperti gambar 5.

```
rendranet@SDN:~$ wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org.opendaylight/integration/karaf/0.21.2/karaf-0.21.2.tar.gz
```

Gambar 5. Download File Karaf

- d. Ekstrak File Karaf

File distribusi ini berisi semua file yang diperlukan untuk menjalankan SDN. Setelah diunduh, file diekstrak menggunakan perintah *tar -zxvf*, yang akan menghasilkan direktori dengan semua file eksekusi dan konfigurasi.

Ketik perintah pada terminal linux seperti gambar 6.

```
rendranet@SDN:~$ tar -zxvf karaf-0.21.2.tar.gz
```

Gambar 6. Ekstrak File Karaf

e. Masuk Direktori Karaf

Tahap ini adalah inti dari instalasi, yaitu menjalankan *software OpenDaylight*. Kita masuk ke direktori hasil ekstraksi `cd karaf-0.21.2` atau sesuai versi yang digunakan. Kita juga bisa memindahkan file karaf ke direktori baru seperti `/opt/opendaylight` dengan perintah `sudo mv karaf-<versi> /opt/opendaylight` memindahkan direktori hasil ekstraksi distribusi *OpenDaylight* ke lokasi standar `/opt`, agar lebih rapi dan sesuai standar *Linux*. Konfigurasinya dapat dilihat pada gambar 7.

```
rendranet@SDN:~$ cd karaf-0.21.2
rendranet@SDN:~/karaf-0.21.2$
```

Gambar 7. Masuk Direktori Karaf

f. Run Opendaylight Controller

Kemudian masuk direktori `/opt/opendaylight` lalu jalankan `./bin/karaf`. Perintah ini akan memulai *container OSGi Karaf* milik *OpenDaylight*, yang berfungsi sebagai kerangka kerja modular untuk menjalankan semua layanan dan plugin yang diperlukan. Ketika container Karaf aktif, kita akan melihat prompt `opendaylight-user@root>`, menandakan bahwa controller sudah aktif. Tampilannya dapat dilihat pada gambar 8.

```
rendranet@SDN:~/opt/opendaylight$ ./bin/karaf
Apache Karaf starting up. Press Enter to open the shell now...
100% [-----]
Karaf started in 33s. Bundle stats: 357 active, 358 total

Hit 'tab' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>
```

Gambar 8. Mengaktifkan Controller

Langkah instalasi sudah selesai controller sudah siap digunakan, Dengan perintah `feature:list`, kita bisa melihat semua fitur yang tersedia dalam distribusi *OpenDaylight*. Biasanya, fitur yang esensial seperti `odl-restconf`, `odl-l2switch-switch`, dan `odl-openflowplugin-flow-services-ui` harus diinstal agar controller dapat berfungsi penuh dalam mengelola switch *Openflow* dan menjalankan REST API. Perintah `feature:install` digunakan untuk menambahkan fitur-fitur ini sesuai kebutuhan.

g. Dashboard Login

Untuk mencoba tampilan pada DLUX UI nya kita dapat mengaksesnya dengan mengetikkan

`ipcontroller:8181/index.html` pada browser. Hasil nya akan menampilkan halaman login, tampilannya dapat dilihat pada gambar 9.



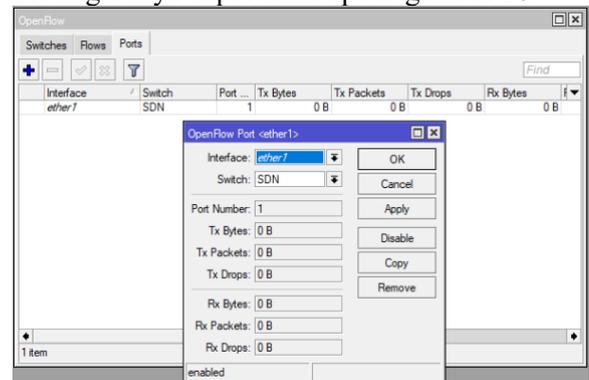
Gambar 9. Halaman Login

### 3.1.2 Konfigurasi Load Balancing

Pada tahap ini penulis akan memaparkan langkah-langkah mulai dari pengaturan *Openflow* pada mikrotik agar bisa terhubung ke kontroler SDN yang sudah diinstall sebelumnya. Setelah terhubung dengan kontroler SDN dilanjutkan konfigurasi load balancing 2 ISP yang bertujuan untuk mendistribusikan trafik dari dua atau lebih jalur WAN (ISP) ke *host* pelanggan secara merata, bukan untuk menggandakan *bandwidth* total, melainkan untuk membagi beban koneksi agar seimbang sehingga tidak ada salah satu dari ISP yang terlalu terbebani, dengan cara ini juga memungkinkan router untuk secara otomatis mendeteksi ketika salah satu koneksi ISP mati atau bermasalah, dan kemudian secara otomatis mengalihkan semua trafik internet ke ISP yang masih berfungsi. Ini memastikan bahwa jaringan tetap memiliki konektivitas internet tanpa interupsi yang signifikan, bahkan jika salah satu ISP mengalami gangguan.

a. Setting Openflow

Tahap pertama yang dilakukan adalah konfigurasi protokol *Openflow* pada router mikrotik agar nantinya dapat terhubung dengan kontroler SDN. Caranya login ke router mikrotik kemudian pilih menu *Openflow*, kemudian klik tanda +, pada *interfaces* isi dengan *port ethernet* yang digunakan, kemudian beri nama SDN, kemudian *apply* lalu *ok*. Konfigurasinya dapat dilihat pada gambar 10.



Gambar 10. Setting Openflow

b. Cek Koneksi

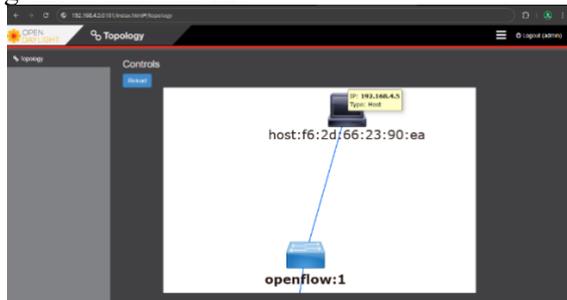
Kemudian cek koneksi dengan melakukan ping ke *IP host* yang telah dipasang *controller SDN*. Jika hasil menunjukkan tidak ada paket yang tidak terkirim menandakan koneksi sudah terhubung. Hasil ping dapat dilihat pada gambar 11

```
[admin@MikroTik] > ping 192.168.4.5
SEQ HOST                               SIZE TTL TIME STATUS
0 192.168.4.5                          56 128 0ms
1 192.168.4.5                          56 128 0ms
2 192.168.4.5                          56 128 0ms
3 192.168.4.5                          56 128 0ms
sent=4 received=4 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=0ms
```

Gambar 11. Ping IP Kontroler

c. Koneksi SDN

Setelah berhasil terhubung kemudian nyalakan *controller SDN* dan cek koneksinya pada DLUX UI di browser. Ketik IP dari controller pada halaman web browser. Jika berhasil dan terhubung akan tampil topologi *Openflow* yang menghubungkan *controller SDN* dengan *router mikrotik*. Hasil dapat dilihat pada gambar dibawah 12.



Gambar 12. Openflow

d. Reset Konfigurasi Mikrotik

Selanjutnya kita akan melakukan konfigurasi *load balancing* untuk mengelola dan menyeimbangkan distribusi trafik pada jaringan RT/RW net. Langkah pertama dalam konfigurasi load balancing ini adalah mereset konfigurasi Mikrotik agar bersih dari konfigurasi sebelumnya. Hal ini sangat disarankan terutama jika menggunakan router baru atau ingin memulai dari awal tanpa konfigurasi default yang bisa mengganggu proses. Gunakan perintah seperti gambar 13.

```
[admin@MikroTik] > /system
reset-configuration no-
default=yes
Dangerous! Reset anyway? [y/N]:
```

Gambar 13. Reset Mikrotik

e. Konfigurasi Interface

Setelah router reboot, agar lebih mudah dalam mengelola, kita beri nama antarmuka (interface) sesuai fungsinya: ether1 menjadi WAN1, ether2 menjadi WAN2, dan ether3 sebagai LAN. Konfigurasinya dapat dilihat pada gambar 14.

```
[admin@MikroTik] > interface
ethernet
[admin@MikroTik] /interface
ethernet> set ether2 name=WAN1
[admin@MikroTik] /interface
ethernet> set ether3 name=WAN2
[admin@MikroTik] /interface
ethernet> set ether4 name=LAN
```

Gambar 14. Konfigurasi Interface

f. Konfigurasi IP Pada Ethernet

Kemudian kita atur alamat IP untuk masing-masing interface. Misalnya, jika ISP1 memberikan IP statik 192.168.10.2/24, maka IP tersebut kita pasang di WAN1. Sedangkan WAN2 akan menggunakan DHCP Client agar IP-nya didapat otomatis. Untuk LAN, kita gunakan subnet 192.168.88.0/24 dan set gateway-nya di 192.168.88.1. Konfigurasi dapat dilihat pada gambar 15.

```
[admin@MikroTik] /ip address> add
address=192.168.10.2/24
interface=WAN1
[admin@MikroTik] /ip address> /ip
dhcp-client
[admin@MikroTik] /ip dhcp-client>
add interface=WAN2 add-default-
route=no disabled=no
[admin@MikroTik] /ip dhcp-client>
/ip address
[admin@MikroTik] /ip address> add
address=192.168.88.1/24
interface=LAN
```

Gambar 15. Konfigurasi IP Address

g. Konfigurasi DNS

Selanjutnya, kita atur DNS Server menggunakan DNS publik milik Google (8.8.8.8 dan 8.8.4.4) dan mengizinkan permintaan DNS dari klien lokal. Buka menu terminal pada mikrotik lalu ketik konfigurasinya. Konfigurasinya dapat dilihat pada gambar 16.

```
[admin@MikroTik] /ip address> /ip
dns
[admin@MikroTik] /ip dns> set
servers=8.8.8.8,8.8.4.4 allow-
remote-requests=yes
```

Gambar 16. Konfigurasi DNS

h. Konfigurasi Firewall NAT

Untuk mengizinkan koneksi internet dari jaringan lokal ke luar melalui masing-masing ISP, kita buat aturan NAT (Network Address Translation) menggunakan masquerade. Buka menu terminal pada mikrotik lalu ketik konfigurasinya. Konfigurasinya dapat dilihat pada gambar 17.

```
[admin@MikroTik] /ip dns> /ip
firewall nat
```

```
[admin@MikroTik] /ip firewall
nat> add chain=srcnat out-
interface=WAN1 action=masquerade
comment="NAT for ISP1"
[admin@MikroTik] /ip firewall
nat> add chain=srcnat out-
interface=WAN2 action=masquerade
comment="NAT for ISP2"
```

Gambar 17. Konfigurasi Firewall NAT

i. Konfigurasi Mangle

Langkah inti dari load balancing adalah konfigurasi mangle dengan metode PCC. Di sini, koneksi dari LAN akan dibagi dua berdasarkan kombinasi IP dan port menggunakan per-connection-classifier. Koneksi akan ditandai untuk diarahkan ke masing-masing ISP. Buka menu terminal pada mikrotik lalu ketik konfigurasinya seperti gambar 18.

```
[admin@MikroTik] /ip firewall
mangle> # Tandai koneksi masuk
dari ISP
[admin@MikroTik] /ip firewall
mangle> add chain=input in-
interface=WAN1 action=mark-
connection new-connection-
mark=ISP1_conn passthrough=yes
[admin@MikroTik] /ip firewall
mangle> add chain=input in-
interface=WAN2 action=mark-
connection new-connection-
mark=ISP2_conn passthrough=yes
[admin@MikroTik] /ip firewall
mangle>
[admin@MikroTik] /ip firewall
mangle> # Tandai output
berdasarkan koneksi
[admin@MikroTik] /ip firewall
mangle> add chain=output
connection-mark=ISP1_conn
action=mark-routing new-routing-
mark=to_ISP1 passthrough=no
[admin@MikroTik] /ip firewall
mangle> add chain=output
connection-mark=ISP2_conn
action=mark-routing new-routing-
mark=to_ISP2 passthrough=no
[admin@MikroTik] /ip firewall
mangle>
[admin@MikroTik] /ip firewall
mangle> # PCC: Tandai koneksi
dari LAN agar terbagi ke ISP1 dan
ISP2
[admin@MikroTik] /ip firewall
mangle> add chain=prerouting in-
interface=LAN dst-address-
type=!local \
... per-connection-
classifier=both-addresses-and-
ports:2/0 \
```

```
\... action=mark-connection
new-connection-mark=ISP1_conn
passthrough=yes comment="PCC
WAN1"
[admin@MikroTik] /ip firewall
mangle>
[admin@MikroTik] /ip firewall
mangle> add chain=prerouting in-
interface=LAN dst-address-
type=!local \
... per-connection-
classifier=both-addresses-and-
ports:2/1 \
... action=mark-connection
new-connection-mark=ISP2_conn
passthrough=yes comment="PCC WAN2"
[admin@MikroTik] /ip firewall
mangle>
[admin@MikroTik] /ip firewall
mangle> # Tandai routing
berdasarkan connection-mark
[admin@MikroTik] /ip firewall
mangle> add chain=prerouting
connection-mark=ISP1_conn in-
interface=LAN action=mark-routing
new-routing-mark=to_ISP1
passthrough=no
[admin@MikroTik] /ip firewall
mangle> add chain=prerouting
connection-mark=ISP2_conn in-
interface=LAN action=mark-routing
new-routing-mark=to_ISP2
passthrough=no
```

Gambar 18. Konfigurasi IP Firewall Mangle

j. Konfigurasi Routing

Konfigurasi selanjutnya adalah routing, di mana koneksi yang sudah ditandai akan diarahkan ke gateway ISP masing-masing. Kita juga menambahkan default route tanpa tanda sebagai failover. Jika koneksi ke ISP1 mati, MikroTik otomatis akan beralih ke ISP2. Buka menu terminal pada mikrotik lalu ketik konfigurasinya seperti gambar 19.

```
[admin@MikroTik] /ip route> #
Routing berdasarkan tanda
[admin@MikroTik] /ip route> add
dst-address=0.0.0.0/0
gateway=192.168.10.1 routing-
mark=to_ISP1 check-gateway=ping
distance=1
[admin@MikroTik] /ip route> add
dst-address=0.0.0.0/0
gateway=192.168.20.1 routing-
mark=to_ISP2 check-gateway=ping
distance=1
[admin@MikroTik] /ip route>
[admin@MikroTik] /ip route> #
Default route failover
[admin@MikroTik] /ip route> add
dst-address=0.0.0.0/0
```

```
gateway=192.168.10.1 check-
gateway=ping distance=1
comment="Default route ISP1
Primary"
[admin@MikroTik] /ip route> add
dst-address=0.0.0.0/0
gateway=192.168.20.1 check-
gateway=ping distance=2
comment="Default route ISP2
Backup"
```

Gambar 19. Konfigurasi Routing

k. Konfigurasi DHCP

Untuk memberikan IP secara otomatis ke perangkat klien di jaringan LAN, kita bisa mengaktifkan DHCP Server. Buka menu terminal pada mikrotik lalu ketik konfigurasinya seperti gambar 20.

```
[admin@MikroTik] /ip pool> add
name=dhcp_pool
ranges=192.168.88.100-
192.168.88.254
[admin@MikroTik] /ip pool>
[admin@MikroTik] /ip pool> /ip
dhcp-server
[admin@MikroTik] /ip dhcp-server>
add address-pool=dhcp_pool
disabled=no interface=LAN
name=dhcp_server
[admin@MikroTik] /ip dhcp-server>
[admin@MikroTik] /ip dhcp-server>
/ip dhcp-server network
[admin@MikroTik] /ip dhcp-server
network> add
address=192.168.88.0/24 dns-
server=192.168.88.1
gateway=192.168.88.1
```

Gambar 20. Konfigurasi DHCP Server

Setelah semua konfigurasi load balancing dilakukan, proses selanjutnya yaitu menguji kualitas jaringan menggunakan parameter QoS.

### 3.2 Hasil Pengujian Sistem

#### 3.2.1 Pengujian QoS

Setelah SDN dan *load balancing* diimplementasikan, dilakukan pengujian jaringan. Metode pengujian menggunakan parameter QoS. Pengujian ini dilakukan untuk memperoleh data seputar jaringan setelah penerapan SDN dan load balancing. Pengujian ini meliputi pengukuran parameter kualitas layanan jaringan seperti *Throughput*, *Packet loss*, *Delay* dan *Jitter*. Pengujian dilakukan menggunakan tool Wireshark untuk *Throughput*, *Packet loss*, *Delay* dan *Jitter*.

a. Data Hasil Pengujian *Throughput*.

Tabel 1. Perbandingan Pengujian *Throughput*

Pengujian	Sebelum Optimasi (Mbps)	Setelah Optimasi (Mbps)
1	4.67 Mbps	7.86 Mbps

2	5.43 Mbps	8.60 Mbps
3	4.31 Mbps	8.63 Mbps
<b>Rata-rata</b>	4.80 Mbps	8.36 Mbps

Evaluasi terhadap hasil pengujian *throughput* dapat dilihat pada Tabel 1. Berdasarkan tiga kali percobaan yang ditampilkan dalam tabel tersebut, terlihat bahwa perbandingan nilai *throughput* antara jaringan yang menggunakan manajemen SDN dan load balancing dengan jaringan tanpa keduanya tidak menunjukkan perbedaan yang signifikan saat proses pengunduhan file berlangsung. Rata-rata dari ketiga percobaan tersebut menunjukkan bahwa performa *throughput* pada jaringan yang menggunakan manajemen SDN dan load balancing memiliki hasil yang lebih baik dengan rata-rata 8.36 Mbps.

b. Data Hasil Pengujian *Packet Loss*.

Tabel 2. Perbandingan Pengujian *Packet Loss*.

Pengujian	Sebelum Optimasi (%)	Setelah Optimasi (%)
1	10.3%	0,20 %
2	11%	0,03 %
3	9.4%	0,04 %
<b>Rata-rata</b>	10.23%	0.09%

Hasil evaluasi pengujian *packet loss* ditampilkan pada Tabel 2. Berdasarkan tiga kali percobaan dalam tabel tersebut, terlihat adanya perbedaan yang cukup mencolok pada hasil *packet loss* antara jaringan yang menerapkan SDN dan load balancing dengan jaringan yang tidak menggunakannya saat proses pengunduhan file. Rata-rata dari ketiga pengujian menunjukkan bahwa jaringan dengan SDN dan load balancing memiliki *packet loss* sebesar 0,09%, sedangkan jaringan tanpa keduanya menunjukkan rata-rata sebesar 10,23%. Hal ini menunjukkan bahwa penerapan SDN dan load balancing memberikan kinerja jaringan yang lebih baik, ditandai dengan tingkat *packet loss* yang jauh lebih rendah.

c. Data Hasil Pengujian *Delay*.

Tabel 3. Perbandingan Pengujian *Delay*.

Pengujian	Sebelum Optimasi (ms)	Setelah Optimasi (ms)
1	47 ms	42 ms
2	59 ms	36 ms
3	66 ms	39 ms
<b>Rata-rata</b>	57.33 ms	39 ms

Hasil evaluasi terhadap pengujian delay ditampilkan pada Tabel 3. Dari tiga kali percobaan yang dilakukan, terlihat adanya perbedaan yang signifikan pada nilai delay antara jaringan yang menggunakan SDN dan load balancing dengan jaringan yang tidak menerapkannya saat melakukan pengunduhan file. Rata-rata delay pada jaringan dengan SDN dan load balancing tercatat sebesar 39 ms, sementara pada jaringan tanpa keduanya mencapai 57,33 ms. Temuan ini mengindikasikan bahwa penggunaan SDN dan load balancing mampu meningkatkan performa jaringan dengan menurunkan tingkat delay secara signifikan.

d. Data Hasil Pengujian *Jitter*.

Tabel 4. Perbandingan Pengujian *Jitter*

Pengujian	Sebelum Optimasi (ms)	Setelah Optimasi (ms)
1	57 ms	15 ms
2	30 ms	11 ms
3	45 ms	12 ms
<b>Rata-rata</b>	44 ms	12.67 ms

Hasil evaluasi pengujian jitter ditampilkan pada Tabel 4 Berdasarkan tiga kali percobaan yang dilakukan, terlihat perbedaan yang mencolok pada nilai jitter antara jaringan yang menerapkan SDN dan load balancing dengan jaringan yang tidak menggunakannya saat proses pengunduhan file. Rata-rata jitter pada jaringan yang menggunakan SDN dan load balancing tercatat sebesar 12,67 ms, sedangkan pada jaringan tanpa keduanya mencapai 44 ms. Hal ini menunjukkan bahwa penerapan SDN dan load balancing dapat meningkatkan kualitas performa jaringan dengan menurunkan tingkat jitter secara signifikan.

### 3.2.2 Analisis Jaringan

Bagian ini membahas hasil dari pengujian implementasi yang telah dilakukan, serta memberikan penjelasan terkait analisis perbandingan yang diperoleh. Pengujian dalam penelitian ini difokuskan pada membandingkan kinerja jaringan yang menerapkan manajemen SDN dan load balancing dengan jaringan sebelumnya yang tidak menerapkan SDN dan load balancing. Untuk memperjelas analisa dari jaringan ini disertai tabel hasil uji yang rangkumannya dapat dilihat pada tabel 5.

Tabel 5 Analisis Jaringan

No	Tgl	Pengukuran Ke	Parameter				Bandwith	
			Through-put	Pakcet Loss	Delay	Jitter	Down-load	Upload
1	20 Juni 2025	1	0,719,48 9 Mbps/7,0 26%	1,65 %	42 ms	15 ms	7.86 Mbps	8.20 Mbps
2	20 Juni 2025	2	0,719,49 0 Mbps/8,0 66%	0,90 %	36 ms	21 ms	6.68 Mbps	7.09 Mbps
3	24 Juni 2025	3	1,010,83 1 Mbps 9,871%	0,26 %	39 ms	15 ms	5.90 Mbps	7.51 Mbps
4	24 Juni 2025	4	1,010,83 1 Mbps/10, 218%	0,40 %	40 ms	11 ms	6.30 Mbps	7.50 Mbps
5	30 Juni 2025	5	0,739,51 2 Mbps /7,245%	0,20 %	38 ms	12 ms	6.80 Mbps	7.99 Mbps
6	30 Juni 2025	6	0,739,51 2 Mbps/7,3 30%	0,03 %	37 ms	11 ms	6.86 Mbps	8.00 Mbps
7	4 Juli 2025	7	1,141,89 3 Mbps/11, 151%	0,04 %	41 ms	31 ms	7.60 Mbps	7.60 Mbps
8	4 Juli 2025	8	1,141,89 3 Mbps/11, 275%	0,38 %	55 ms	27 ms	7.63 Mbps	8.31 Mbps
9	9 Juli 2025	9	0,865,86 7 Mbps/8,4 79%	0,02 %	38 ms	23 ms	6.80 Mbps	7.78 Mbps
10	9 Juli 2025	10	0,865,86 7 Mbps/8,0 68%	0,4 %	42 ms	25 ms	7.00 Mbps	8.30 Mbps

Berdasarkan hasil pengukuran QoS yang dilakukan pada rentang waktu 20 Juni hingga 24 Juni 2025, terlihat bahwa parameter throughput mengalami fluktuasi signifikan, dimulai dari sekitar 0,64 Mbps hingga mencapai 1,14 Mbps. Meskipun demikian, secara umum terlihat tren kenaikan throughput dari awal hingga akhir periode pengukuran. Parameter packet loss juga menunjukkan variasi yang cukup besar, dengan nilai terendah hanya 0,02% dan tertinggi mencapai 0.40% pada tanggal 20 Juni 2025, yang mengindikasikan adanya gangguan jaringan pada saat itu.

Sementara itu, delay umumnya stabil di kisaran 36–42 ms, kecuali pada tanggal 4 Juli 2025 yang mencapai 55 ms, kemungkinan disebabkan oleh lonjakan trafik atau gangguan sementara. Jitter, yang merupakan ukuran variasi delay, juga bervariasi dan

pernah melonjak signifikan hingga 57 ms, menunjukkan ketidakstabilan jaringan pada waktu tertentu. Untuk kecepatan download dan upload, nilai download cenderung stabil dan meningkat hingga 7,63 Mbps, sementara upload lebih fluktuatif, dengan nilai terendah hanya 1,09 Mbps dan tertinggi hingga 6,78 Mbps.

Jika dibandingkan dengan jaringan yang belum menerapkan SDN dan load balancing. Dari hasil pengujian parameter *throughput* mempunyai kinerja yang lebih baik karena memiliki rata-rata nilai *throughput* yang lebih besar. Dari hasil pengujian *packet loss* mempunyai performa yang lebih baik karena memiliki rata-rata persentase *packet loss* yang lebih kecil. Kemudian dari hasil pengujian *delay* mempunyai performa yang lebih baik karena memiliki rata-rata *delay* yang lebih kecil. Yang terakhir hasil dari pengujian *jitter* juga menunjukkan penurunan angka yang membuktikan jaringan lebih baik dari sebelum diterapkannya SDN dan load balancing.

#### 4 Kesimpulan

Penelitian ini secara komprehensif membuktikan bahwa implementasi Software Defined Networking (SDN) yang dikombinasikan dengan metode load balancing memberikan dampak positif yang signifikan dalam peningkatan Quality of Service (QoS) pada jaringan RT/RW Net yang memanfaatkan lebih dari satu penyedia layanan internet (ISP). Dengan mengadopsi arsitektur SDN, pengelolaan lalu lintas jaringan menjadi jauh lebih fleksibel dan terpusat, sehingga distribusi trafik antar jalur ISP dapat dilakukan secara lebih seimbang dan efisien. Hasil pengujian yang dilakukan menunjukkan adanya perbaikan nyata pada berbagai parameter QoS, seperti peningkatan *throughput*, pengurangan *delay*, penurunan *jitter*, serta berkurangnya *packet loss*, yang semuanya merupakan indikator penting dalam kualitas koneksi jaringan.

Selain itu, distribusi trafik yang merata membuat koneksi internet menjadi lebih stabil dan andal, dimana gangguan pada salah satu jalur ISP tidak lagi menyebabkan terganggunya layanan secara keseluruhan bagi pengguna. Dengan memanfaatkan controller OpenDaylight sebagai otak pengendali jaringan dan perangkat MikroTik sebagai pengelola perangkat keras jaringan, penelitian ini berhasil mengimplementasikan sistem load balancing berbasis SDN secara nyata pada lingkungan jaringan RT/RW Net. Hal ini membuktikan bahwa teknologi ini tidak hanya efektif di area perkotaan tetapi juga mampu beradaptasi dengan tantangan geografis yang kompleks, seperti wilayah pegunungan yang biasanya memiliki keterbatasan infrastruktur. Sistem

ini juga dilengkapi dengan mekanisme failover otomatis yang secara cepat dapat beralih ke jalur ISP alternatif saat terjadi kegagalan, sehingga meningkatkan ketersediaan dan keandalan layanan internet secara keseluruhan.

Dengan demikian, penelitian ini tidak hanya memberikan kontribusi pada pengembangan teknologi jaringan, tetapi juga menawarkan solusi praktis yang dapat diterapkan untuk meningkatkan kualitas layanan internet di komunitas-komunitas RT/RW Net di berbagai kondisi geografis

#### References :

- [1] A. Fajaryanto, I. A. Zulkarnain, F. Masykur, dan Y. Litanianda, "IMPLEMENTASI INTERNET AMAN DI FASILITAS UMUM DESA NGRUPIT KABUPATEN PONOROGO MENGGUNAKAN WEB PROXY," *Jurnal Pengabdian Kepada Masyarakat Bersinergi Inovatif*, vol. 2, no. 1, Art. no. 1, Jul 2024.
- [2] T. Sukendar, M. I. Saputro, A. Ishaq, A. Sumbaryadi, dan R. Santoso, "Implementasi Manajemen Voucher Internet Menggunakan Mikhmon pada Jaringan RT/RW Net Berbasis Mikrotik," *Jurnal Teknologi Informatika dan Komputer*, vol. 10, no. 2, Art. no. 2, Okt 2024, doi: 10.37012/jtik.v10i2.2331.
- [3] A. A. Putra dan Ubaidi -, "IMPLEMENTASI SISTEM KENDALI JARAK JAUH DAN MANAJEMEN USER BERBASIS OPENWRT PADA JARINGAN RT/RW NET DI DESA LADEN," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, Art. no. 3S1, Okt 2024, doi: 10.23960/jitet.v12i3S1.5200.
- [4] P. A. Khairunnisa, N. Annisa, dan J. Parhusip, "Penerapan Teknologi SDN (Software-Defined Networking) untuk Meningkatkan Keamanan Jaringan Perusahaan," *Teknik: Jurnal Ilmu Teknik dan Informatika*, vol. 4, no. 2, Art. no. 2, Nov 2024, doi: 10.51903/teknik.v4i2.569.
- [5] M. C. Anam, H. D. Septama, R. A. P, dan G. F. Nama, "PENGARUH PENGGUNAAN VLAN PADA SOFTWARE DEFINED NETWORK BERBASIS RYU CONTROLLER," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Art. no. 1, Jan 2024, doi: 10.23960/jitet.v12i1.3766.
- [6] K. Sabila, S. Rahayu, dan T. Sumarni, "Peningkatan Efisiensi Penggunaan Sumber

Daya Jaringan Melalui Teknik Load Balancing,” *CEMERLANG: Jurnal Manajemen dan Ekonomi Bisnis*, vol. 4, no. 3, hlm. 31–41, Jun 2024, doi: 10.55606/cemerlang.v4i3.2989.

- [7] A. Irfan, N. Nursakti, R. Riskayani, Z. Rachmat, dan F. Nurhidayah, “Penerapan Metode Quality of Service (QoS) untuk Menganalisis Kualitas Jaringan Wireless di STMIK Amika Soppeng,” *Jurnal Minfo Polgan*, vol. 14, no. 1, Art. no. 1, Mei 2025, doi: 10.33395/jmp.v14i1.14787.
- [8] Melina, B. Arifwidodo, dan J. G. A. Ginting, “Analisis Kualitas Video Conference Microsoft Teams Menggunakan Metode Quality Of Service (QoS) Dan Peer Connection Queue (PCQ),” *eProceedings of Engineering*, vol. 12, no. 3, Art. no. 3, Jul 2025.
- [9] F. Anissabilla dan R. Kusumarani, “ANALISIS DAN EVALUASI KINERJA JARINGAN INTERNET BERDASARKAN QUALITY OF SERVICE (QoS),” *Prosiding Seminar Nasional Sains dan Teknologi “SainTek,”* vol. 2, no. 1, Art. no. 1, Feb 2025.
- [10] A. F. Zulkarnain, N. Hijriana, A. Awaliyah, dan E. S. Wijaya, “ANALISIS QUALITY OF SERVICE DAN QUALITY OF EXPERIENCE JARINGAN INTERNET BERBASIS WIRELESS PADA LAYANAN TELKOMSEL ORBIT,” *Technologia: Jurnal Ilmiah*, vol. 16, no. 1, Art. no. 1, Jan 2025, doi: 10.31602/tji.v16i1.17022.