

BAB IV

Hasil dan Pembahasan

Bab 4 ini mengungkapkan hasil penelitian mengenai optimalisasi QoS, manajemen *bandwidth*, dan penggunaan *Simple Queue* dan *Policy-based Routing* pada *server* dengan fitur *Load Balance* dan *Failover*.

4.1 Implementasi Topologi Skenario pengujian

Topologi skenario pengujian yang digunakan seperti yang ada pada BAB III di mana terdapat satu *client* yang terhubung pada satu *switch* dan satu *router*, konfigurasi yang dibuat sesuai dengan *topologi* yang ditentukan. Langkah-langkahnya berikut:

- Menghubungkan kedua *client* ke switch dengan kabel LAN.
- Konfigurasi alamat IP pada *client* dan alamat IP proses secara dinamis, IP akan otomatis didapatkan dari *router*.
- Menginstall Aplikasi Winbox, Wireshark pada *client*.

4.2 Implementasi Konfigurasi

4.2.1 Konfigurasi Dasar

Pada konfigurasi tahap ini hal yang dilakukan adalah konfigurasi *hardware*. Tahap awal yang dilakukan adalah pemasangan Kabel lan dari *router* mikrotik *server* ke hub lalu di hubungkan ke *device client*.

Setelah konfigurasi di atas dilakukan, langkah selanjutnya adalah pemberian alamat IP pada *interface* lokal yang tersambung antara *router* dan *client*, pada percobaan kali ini yang digunakan sebagai *interface* lokal adalah *Ether3*. Pada *interface* lokal, pemberian alamat IP dilakukan dengan *scrip* seperti di bawah ini:

```
add address=192.168.15.8/24 interface=ether3-Hotpot+PPPOE network=\
192.168.15.0
```

Konfigurasi urutan pertama merupakan perintah untuk menambahkan alamat IP baru dan pada baris kedua perintah yang digunakan untuk menambahkan alamat IP pada *interface* 3 di mana konfigurasi IP 192,168.15.8 dengan subneting /24. Hasil *konfigurasi* dapat dilihat pada gambar 7:

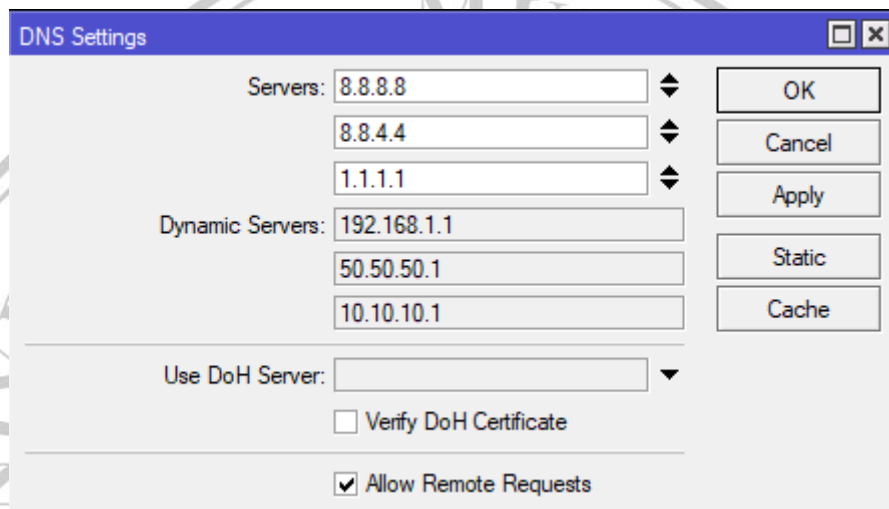
D	192.168.1.2/24	192.168.1.0	ether1-ISP1
	192.168.1.8/24	192.168.1.0	ether1-ISP1
	192.168.10.15/24	192.168.10.0	Ether2-LAN
	192.168.15.8/24	192.168.15.0	ether3-Hotpot+PPPOE

Gambar 7 konfigurasi IP Address

Langkah terakhir dalam konfigurasi dasar adalah dengan menambahkan DNS *server*. Di mana DNS *server* digunakan untuk mendeskripsikan *host name* komputer ke alamat IP. Terdapat 2 konfigurasi DNS dari *router* ISP, *static* dan *dynamic*, di mana untuk konfigurasi *static* menggunakan 8.8.8.8, 8.8.4.4, 1.1.1.1 dan perlu mengaktifkan *Remote Requests*. Konfigurasinya adalah sebagai berikut:

```
/ip dns  
set allow-remote-requests=yes servers=8.8.8.8,8.8.4.4,1.1.1.1
```

Konfigurasi DNS yang dihasilkan dapat dilihat pada gambar 8.



Gambar 8 Konfigurasi DNS

Pada langkah ini konfigurasi dasar telah selesai dilakukan. Untuk langkah berikutnya yaitu konfigurasi *Load balancing* dan *failover*, sesuai dengan flowchart penelitian.

1.2.2 Konfigurasi mangle

Mangle merupakan proses di mana paket-paket yang berasal dari sebuah *interface* akan diproses. Fungsi dari *Mangle* adalah untuk menandai paket agar dirutekan sesuai dengan aturan routing yang ada. Pada tahap ini, tiga metode *Load balancing* akan PCC Konfigurasinya adalah sebagai berikut:

```

/ip firewall mangle
add action=accept chain=prerouting comment=" Bypass+ISP" dst-address-list=\
    Client+ISP
    new-routing-mark=Rule-Route-Game passthrough=yes src-address-list=\
    Client+ISP
add action=mark-connection chain=input comment=" loadbalace PCC" \
    in-interface=ether1-ISP1 new-connection-mark=koneksiISP1 passthrough=yes
add action=mark-connection chain=input in-interface="ether4 ISP 2 WMS" \
    new-connection-mark=koneksiISP2 passthrough=yes
add action=mark-routing chain=output connection-mark=koneksiISP1 \
    new-routing-mark=Rule-Route-ISP1 passthrough=no
add action=mark-routing chain=output connection-mark=koneksiISP2 \
    new-routing-mark=Rule-Route-ISP2 passthrough=no
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
    dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
    passthrough=yes per-connection-classifier=both-addresses-and-ports:6/0 \
    src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
    dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
    passthrough=yes per-connection-classifier=both-addresses-and-ports:6/1 \
    src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
    dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP2 \
    passthrough=yes per-connection-classifier=both-addresses-and-ports:6/2 \
    src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
    dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
    passthrough=yes per-connection-classifier=both-addresses-and-ports:6/3 \
    src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
    dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
    passthrough=yes per-connection-classifier=both-addresses-and-ports:6/4 \

```

```

passthrough=yes per-connection-classifier=both-addresses-and-ports:6/4 \
src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
passthrough=yes per-connection-classifier=both-addresses-and-ports:6/5 \
src-address-list=Client+ISP
add action=mark-connection chain=prerouting dst-address-list=!Client+ISP \
dst-address-type=!local hotspot=auth new-connection-mark=koneksiISP1 \
passthrough=yes per-connection-classifier=both-addresses-and-ports:6/6 \
src-address-list=Client+ISP
add action=mark-routing chain=prerouting connection-mark=koneksiISP1 \
dst-address-list=!Client+ISP new-routing-mark=Rule-Route-ISP1 \
passthrough=yes src-address-list=Client+ISP
add action=mark-routing chain=prerouting connection-mark=koneksiISP2 \
dst-address-list=!Client+ISP new-routing-mark=Rule-Route-ISP2 \
passthrough=yes src-address-list=Client+ISP

```

Konfigurasi *firewall Mangle* pada Metode *Load Balance* bekerja dengan memecah traffic data yang melewati *router* kedalam beberapa *stream* dan beberapa *stream* tersebut akan dilewatkan melalui jalur yang berbeda, *Mangle Load Balance* PCC terdapat beberapa konfigurasi yang pertama Menambahkan Rule baru dengan *chain=prerouting* dan untuk parameter *Dst.address* adalah network-network yang terhubung langsung ke *router* di sini menggunakan " Bypass+ISP" *dst-address-list=\ Client+ISP* dengan *action=accept*, yang langsung terhubung dengan *dst-address list* yang berisi address *client* yang dapat dilihat pada gambar 9.

Name	Address	Timeout	Creation Time
Client+ISP	192.168.1.0/24		Nov/24/2021 21:59:21
Client+ISP	192.168.33.0/24		Nov/24/2021 22:00:33
Client+ISP	192.168.15.0/24		Nov/24/2021 22:00:50
Client+ISP	192.168.8.0/24		Nov/24/2021 22:01:05
Client+ISP	10.10.10.0/24		Dec/09/2021 21:14:01
Client+ISP	10.10.30.0/24		Dec/09/2021 21:27:06

Gambar 1 Dst address list

Fungsi rule *accept* di sini digunakan agar koneksi yang mengarah ke *network DAC* (*Dynamic Active Connect*) di *router* agar tidak dikirimkan menggunakan metode PCC.

Tahap yang ke dua menambahkan rule baru yang tujuannya untuk menandai koneksi-koneksi yang masuk dari *interface* WAN pada konfigurasi adalah *pppoe-ISP_1* dan *pppoe-ISP_2* di mana konfigurasi *chain=prerouting* dan *In.Interface=pppoe-ISP_1* kemudian pada parameter *Action=mark-connection* dengan nama *ISP_1*, selanjutnya buat lagi dengan *action=prerouting* dan *In.Interface=pppoe-ISP_2* *Mark-Connection = ISP_2*.

Tahap ke tiga adalah konfigurasi parameter PCC (*Per Connection Classifier*) dalam parameter ini terdapat 3 parameter baru kolom pertama adalah *classifier*, pada kolom *classifier* data yang akan diambil adalah data yang berasal dari IP Header bisa berupa *scr. address*, *dst. address*, atau bisa juga dari *port* atau bisa juga kombinasi dari ketiga data tersebut pada konfigurasi yang digunakan kali ini menggunakan *classifier=both addresses* yang merupakan *parameter classifier dari src. addresses* dan *dst. addresses* selanjutnya kolom yang kedua adalah denominator karena kita akan membagi menjadi 2 stream yang berbeda, maka pada kolom denominator ini kita isi dengan nilai 2 cara kerjanya adalah data yang berasal dari kolom *classifier* ini akan diolah menjadi data 32bit selanjutnya, bentuk data 32bit tersebut akan dimodulasi dengan nilai yang ada di kolom denominator ini hasil modulasi dari *classifier* dan denominator ini nantinya akan dibandingkan dengan nilai yang kita tentukan di kolom yang ketiga atau kolom remainder pada Rule ini, nilai remaindernya = 0, *action=mark-connection*, *parameter New Connection Mark=ISP_1* hal ini dilakukan agar koneksi yang memiliki nilai remainder=0 akan dikirimkan melalui ISP1, sama halnya dengan denominator yang pertama, denominator yang kedua menggunakan konfigurasi yang sama namun yang membedakan hanya pada nilai remainder = 1 dan *parameter New Connection Mark=ISP_2* agar koneksi yang memiliki nilai remainder=1 akan dikirimkan melalui ISP2.

Tahap ke empat adalah buat Rule *Mark Routing* dengan menambah rule baru dengan *chain=prerouting*, *In.Interface=ether3* atau *interface* yang mengarah ke jaringan lokal dan *parameter Connection Mark=ISP_1* dengan *Action=mark routing* dan *New Connection Mark =ke-ISP_1*, *Rule Mark*

Routing kedua untuk mengarah ke ISP2 dengan Chain=prerouting In.Interface=ether3, parameter Connection Mark=ISP_2, Action=mark routing dan New Routing Mark = ke-ISP_2.

Tahap ke lima menambahkan Mark Routing di Chain Output dengan membuat Rule baru Chain=output dengan Connection Mark=ISP_1, Action=mark routing dan parameter New Routing Mark = ke-ISP_1, penambahan Mark Routing pada chain Output ISP2 Chain=output dengan Connection Mark=ISP_2, Action=mark routing dan New Connection Mark kita pilih "ke-ISP_2. Konfigurasi yang dihasilkan dapat dilihat pada gambar 10.

#	Action	Chain	Protocol	Dst. Port	In. Inter...	Src. Ad...	Dst. Ad...	Bytes	Packets	Out
::: Bypass+ISP										
0	accept	prerouting					Client+I...	8352.7 GiB	7325 794...	
::: Routing ICMP										
1	mark conne...	prerouting	1 (icmp)					1366.5 MiB	13 179 166	
2	mark conne...	prerouting	17 (udp)	53,5353,853				164.8 MiB	2 045 146	
3	mark packet	forward						3500.3 MiB	24 196 109	
4	mark conne...	prerouting	6 (tcp)	53,5353,853				321.1 MiB	2 518 018	
::: loadbalance PCC										
5	mark conne...	input			ether1-l...			1931.5 MiB	15 227 919	
6	mark conne...	input			ether4 l...			0 B	0	
7	mark routing	output						5.4 GiB	9 685 271	
8	mark routing	output						435.1 KiB	3 963	
9	mark conne...	prerouting				Client+I...	!Client+...	22.3 GiB	145 035 ...	
10	mark conne...	prerouting				Client+I...	!Client+...	18.8 GiB	137 624 ...	
11	mark conne...	prerouting				Client+I...	!Client+...	19.9 GiB	141 080 ...	
12	mark conne...	prerouting				Client+I...	!Client+...	20.4 GiB	140 562 ...	
13	mark conne...	prerouting				Client+I...	!Client+...	19.3 GiB	137 557 ...	
14	mark conne...	prerouting				Client+I...	!Client+...	20.3 GiB	140 766 ...	
15	mark conne...	prerouting				Client+I...	!Client+...	0 B	0	
16	mark routing	prerouting				Client+I...	!Client+...	100.9 GiB	699 626 ...	
17	mark routing	prerouting				Client+I...	!Client+...	19.9 GiB	140 692 ...	
::: Koneksi Game										

Gambar 2 Konfigurasi Firewall mangle PCC

1.2.3 Konfigurasi Routing

```

/ip route
add check-gateway=ping comment="Koneksi ISP 1" distance=1 gateway=8.8.8.8 \
    routing-mark=Rule-Route-ISP1 target-scope=30
add check-gateway=ping comment="Koneksi ISP 2" distance=1 gateway=8.8.4.4 \
    routing-mark=Rule-Route-ISP2 target-scope=30
add check-gateway=ping comment="Backup ISP WMS" distance=2 gateway=10.10.30.1
add comment="Fail Over Recursive Gateway ISP 2" distance=1 dst-address=\
    8.8.4.4/32 gateway=10.10.30.1
add comment="Fail Over Recursive Gateway ISP 1" distance=1 dst-address=\
    8.8.8.8/32 gateway=192.168.1.1

```

Konfigurasi *route* untuk metode *Loadbalace* PCC yang pertama membuat *route* baru dengan *check-gateway=ping*, tujuannya untuk cek kondisi jalur sedang *running* atau tidak kemudian *distance=1* menunjukkan jalur ke satu, *gateway=pppoe-ISP_1* parameter *Routing Mark=ke-ISP_1* untuk mengarahkan *gateway pppoe-ISP_1* ke *routing mark ke-ISP_1* pada *Mangle*, yang kedua menambahkan *router* baru *distance=2* menunjukkan jalur ke secondary atau cadangan, *gateway=pppoe-ISP_1* parameter *Routing Mark= ke-ISP_1* untuk mengarahkan *gateway pppoe-ISP_1* ke *routing mark ke-ISP_1* pada *Mangle*.

Membuat *route* baru lagi dengan *check-gateway=ping*, tujuannya untuk cek kondisi jalur sedang *running* atau tidak kemudian *distance=1* menunjukkan jalur ke satu, *gateway=pppoe-ISP_1* parameter *RoutingMark= ke-ISP_2* untuk mengarahkan *gateway pppoe-ISP_1* ke *routing mark ke-ISP_1* pada *Mangle*, yang kedua menambahkan *router* baru *distance=2* menunjukkan jalur ke secondary atau cadangan, *gateway=pppoe-ISP_1* parameter *Routing Mark= ke-ISP_1* untuk mengarahkan *gateway pppoe-ISP_1* ke *routing mark ke-ISP_1* pada *Mangle* dan seterusnya sampai *distance* ke 6 . Berikut konfigurasi yang dihasilkan dapat dilihat pada gambar 11.

	Dst. Address	Gateway	Check...	Type	Distance	Routing Mark	Scope	Target Scope	Pr
AS	0.0.0.0/0	8.8.4.4 recursive via 192.168.1.1 ether1-ISP1	ping	unicast	2	Rule-Route-ISP2	30	30	
AS	0.0.0.0/0	8.8.8.8 recursive via 192.168.1.1 ether1-ISP1	ping	unicast	1	Rule-Route-ISP1	30	30	
S	0.0.0.0/0	10.10.30.1 unreachable	ping	unicast	2	Rule-Route-ISP1	30	10	

Gambar 3 Konfigurasi Route PCC

1.2.4 Konfigurasi PBR

Konfigurasi kebijakan berbasis rute (*policy-based routing*) adalah metode untuk mengarahkan lalu lintas jaringan berdasarkan kebijakan tertentu yang ditentukan oleh administrator jaringan. Dalam pendekatan ini, lalu lintas diperiksa menggunakan aturan kebijakan yang telah ditentukan sebelumnya, dan rute yang sesuai dipilih berdasarkan kebijakan tersebut.

```
/ip firewall mangle
add action=accept chain=prerouting comment=" Bypass+ISP" dst-address-list=\
  Client+ISP
add action=mark-connection chain=prerouting comment="Routing ICMP" \
  new-connection-mark=DNS-ICMP passthrough=yes protocol=icmp \
  src-address-list=""
add action=mark-connection chain=prerouting dst-port=53,5353,853 \
  new-connection-mark=DNS-ICMP passthrough=yes protocol=udp
add action=mark-packet chain=forward connection-mark=DNS-ICMP \
  new-packet-mark=Paket-Ping-DNS passthrough=no
add action=mark-connection chain=prerouting dst-port=53,5353,853 \
  new-connection-mark=DNS-ICMP passthrough=yes protocol=tcp
add action=mark-connection chain=prerouting comment="Koneksi Game" \
  dst-address-list=IP-Game new-connection-mark=Koneksi-Game passthrough=yes \
  src-address-list=Client+ISP
add action=mark-routing chain=prerouting connection-mark=Koneksi-Game \
  new-routing-mark=Rule-Route-Game passthrough=yes src-address-list=\
  Client+ISP
add action=mark-packet chain=forward connection-mark=Koneksi-Game \
  new-packet-mark=Paket-Game passthrough=no
add action=mark-connection chain=prerouting comment=Rule_YouTube \
  dst-address-list=YouTube new-connection-mark=Koneksi_YouTube passthrough=\
  yes src-address-list=Client+ISP
add action=mark-routing chain=prerouting connection-mark=Koneksi_YouTube \
  new-routing-mark=Routing_YouTube passthrough=yes src-address-list=\
  Client+ISP
```



```
add action=mark-connection chain=prerouting comment="Koneksi Sosial Media" \
    dst-address-list=SosialMedia new-connection-mark=Koneksi-SosMed \
    passthrough=yes src-address-list=Client+ISP
add action=mark-routing chain=prerouting connection-mark=Koneksi-Telegram \
    new-routing-mark=Routing_SosMed passthrough=yes src-address-list=\
    Client+ISP
DLL
```

1. Bypass+ISP:

- a. Kebijakan ini digunakan untuk membiarkan lalu lintas dari daftar alamat tujuan yang ada dalam *dst-address-list* "Client+ISP" melalui tanpa perubahan rute.
- b. Aksi yang diambil adalah "*accept*", yang berarti lalu lintas akan diteruskan tanpa mengalami perubahan rute.

2. Routing ICMP:

- a. Kebijakan ini digunakan untuk menandai koneksi ICMP (ping) dan lalu lintas UDP ke port 53, 5353, dan 853 (DNS).
- b. Aksi yang diambil adalah "*mark-connection*" dengan mengatur *new-connection-mark* menjadi "DNS-ICMP", yang akan digunakan untuk mengelompokkan koneksi yang sesuai dengan kebijakan ini.
- c. Perintah "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.

3. Paket-Ping-DNS:

- a. Kebijakan ini digunakan untuk menandai paket yang terkait dengan koneksi yang sudah ditandai dengan "DNS-ICMP".
- b. Aksi yang diambil adalah "*mark-packet*" dengan mengatur *new-packet-mark* menjadi "Paket-Ping-DNS".
- c. Pengaturan "*passthrough=no*" memastikan bahwa paket yang ditandai dengan "Paket-Ping-DNS" tidak diteruskan.

4. Koneksi Game:

- a. Kebijakan ini digunakan untuk menandai koneksi yang menuju ke alamat yang ada dalam *dst-address-list* "IP-Game" dari daftar alamat sumber yang ada dalam *src-address-list* "Client+ISP".
- b. Aksi yang diambil adalah "*mark-connection*" dengan mengatur *new-connection-mark* menjadi "Koneksi-Game".

- c. Perintah "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.
5. Rule-Route-Game:
- a. Kebijakan ini digunakan untuk menandai rute yang digunakan oleh koneksi yang sudah ditandai dengan "Koneksi-Game".
 - b. Aksi yang diambil adalah "mark-routing" dengan mengatur *new-routing-mark* menjadi "Rule-Route-Game".
 - c. Pengaturan "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.
6. Paket-Game:
- a. Kebijakan ini digunakan untuk menandai paket yang terkait dengan koneksi yang sudah ditandai dengan "Koneksi-Game".
 - b. Aksi yang diambil adalah "*mark-packet*" dengan mengatur *new-packet-mark* menjadi "Paket-Game".
 - c. Pengaturan "*passthrough=no*" memastikan bahwa paket yang ditandai dengan "Paket-Game" tidak diteruskan.
7. Rule_Youtube:
- a. Kebijakan ini digunakan untuk menandai koneksi yang menuju ke alamat yang ada dalam *dst-address-list* "YouTube" dari daftar alamat sumber yang ada dalam *src-address-list* "Client+ISP".
 - b. Aksi yang diambil adalah "mark-connection" dengan mengatur *new-connection-mark* menjadi "Koneksi_Youtube".
 - c. Perintah "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.
8. Routing_Youtube:
- a. Kebijakan ini digunakan untuk menandai rute yang digunakan oleh koneksi yang sudah ditandai dengan "Koneksi_Youtube".
 - b. Aksi yang diambil adalah "mark-routing" dengan mengatur *new-routing-mark* menjadi "Routing_Youtube".
 - c. Pengaturan "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.
9. Koneksi Sosial Media:

- Kebijakan ini digunakan untuk menandai koneksi yang menuju ke alamat yang ada dalam *dst-address-list* "SosialMedia" dari daftar alamat sumber yang ada dalam *src-address-list* "Client+ISP".
- Aksi yang diambil adalah "mark-connection" dengan mengatur *new-connection-mark* menjadi "Koneksi-SosMed".
- Perintah "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.

10. Routing_SosMed:

- Kebijakan ini digunakan untuk menandai rute yang digunakan oleh koneksi yang sudah ditandai dengan "Koneksi-SosMed".
- Aksi yang diambil adalah "mark-routing" dengan mengatur *new-routing-mark* menjadi "Routing_SosMed".
- Pengaturan "*passthrough=yes*" memastikan bahwa lalu lintas akan diteruskan tanpa perubahan.

Hasil dari konfigurasi PBR mangle di tampilkan pada gambar 12.

#	Action	Chain	Protocol	Dst. Port	Src. Address ...	Dst. Address List	Bytes	Packets	lr
0	✓ accept	prerouting				Client+ISP	11515.9 ...	10101 37...	
1	mark connection	prerouting	1 (icmp)				1707.0 MiB	15 886 185	
2	mark connection	prerouting	17 (udp)	53,5353,853			229.7 MiB	2 849 789	
3	mark packet	forward					4496.9 MiB	30 172 924	
4	mark connection	prerouting	6 (tcp)	53,5353,853			481.9 MiB	3 774 641	
5	mark connection	prerouting			Client+ISP	IP-Game	13.9 GiB	137 085 ...	
6	mark routing	prerouting			Client+ISP		14.0 GiB	137 313 ...	
7	mark packet	forward					24.6 GiB	194 585 ...	
8	mark connection	prerouting			Client+ISP	YouTube	131.7 GiB	1473 371...	
9	mark routing	prerouting			Client+ISP		131.7 GiB	1473 148...	
10	mark connection	prerouting			Client+ISP	SosialMedia	88.6 GiB	538 712 ...	
11	mark routing	prerouting			Client+ISP		3583.3 MiB	44 246 917	
12	mark connection	prerouting			Client+ISP	MaketPlace-List	17.4 GiB	185 010 ...	
13 X	mark routing	prerouting			Client+ISP		0 B	0	
14	mark packet	forward					132.9 GiB	166 687 ...	
15	mark connection	prerouting			Client+ISP	WhatsApp	99.3 GiB	345 234 ...	
16 X	mark routing	prerouting					0 B	0	
17	mark packet	forward					680.0 GiB	849 152 ...	
18	mark connection	prerouting			Client+ISP	Telegram-List	3790.1 MiB	46 625 355	
19 X	mark routing	prerouting			Client+ISP		0 B	0	
20	mark packet	forward					101.4 GiB	125 773 ...	
21	mark connection	prerouting			Client+ISP	Tiktok-List	144.7 GiB	1435 401...	
22	mark packet	forward					2916.9 GiB	3543 950...	

Gambar 4 Hasil Konfigurasi PBR mangle

1.2.5 Konfigurasi Simple Queue

```
/queue simple
add name=TOTAL_SPEED queue=default/default target=\
    192.168.15.0/24,192.168.10.0/24,192.168.33.0/24,192.168.10.0/24
add max-limit=20M/20M name=TRAFIK-MARKETPLACE packet-marks=Paket-MarketPlace \
    parent=TOTAL_SPEED priority=3/3 queue=\
    pcq-upload-default/pcq-download-default target=\
    192.168.33.0/24,192.168.15.0/24,192.168.10.0/24
add name=TRAFIK-PING-DNS-ICMP packet-marks=Paket-Ping-DNS parent=TOTAL_SPEED \
    priority=4/4 queue=pcq-upload-default/pcq-download-default target=\
    192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
add bucket-size=0.8/0.8 name=TRAFIK-WA packet-marks=Paket-WhatsApps parent=\
    TOTAL_SPEED priority=1/1 queue=pcq-upload-default/pcq-download-default \
    target=192.168.33.0/24,192.168.15.0/24,192.168.1.0/24,192.168.10.0/24
add bucket-size=1/1 max-limit=20M/20M name=TRAFIK-GAME packet-marks=\
    Paket-Game parent=TOTAL_SPEED priority=2/2 queue=\
    pcq-upload-default/pcq-download-default target=\
    192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
add max-limit=10M/10M name=TRAFIK-TELEGRAM packet-marks=Paket-Telegram \
    parent=TOTAL_SPEED priority=5/5 queue=\
    pcq-upload-default/pcq-download-default target=\
    192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
add name=HOTSOPT+PPPOE parent=TOTAL_SPEED queue=default/default target=\
    192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
```

Lebih spesifik nya akan di jelas pada text box dibawah ini.

```
add name=TOTAL_SPEED queue=default/default target=\
    192.168.15.0/24,192.168.10.0/24,192.168.33.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TOTAL_SPEED" dan menggunakan jenis antrian *default/default*.
- Target antrian ini adalah rentang alamat IP 192.168.15.0/24, 192.168.10.0/24, 192.168.33.0/24, dan 192.168.10.0/24 untuk lalu lintas yang cocok dengan antrian ini.

```
add max-limit=20M/20M name=TRAFIK-MARKETPLACE packet-marks=Paket-MarketPlace \parent=TOTAL_SPEED priority=3/3 queue=\pcq-upload-default/pcq-download-default target=\192.168.33.0/24,192.168.15.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TRAFIK-MARKETPLACE" dan batasan maksimum 20 Mbps untuk *upload* dan *download*.
- Menggunakan tanda paket "Paket-MarketPlace" untuk mencocokkan lalu lintas.
- Antrian induk (parent *queue*) adalah "TOTAL_SPEED".
- Prioritas diatur menjadi 3/3.
- Menggunakan jenis antrian *pcq-upload-default/pcq-download-default*.
- Target rentang alamat IP untuk lalu lintas yang cocok dengan antrian ini adalah 192.168.33.0/24, 192.168.15.0/24, dan 192.168.10.0/24.

```
add name=TRAFIK-PING-DNS-ICMP packet-marks=Paket-Ping-DNS \parent=TOTAL_SPEED \priority=4/4 queue=pcq-upload-default/pcq-download-default target=\192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TRAFIK-PING-DNS-ICMP" untuk mengelola lalu lintas ping, DNS, dan ICMP.
- Menggunakan tanda paket "Paket-Ping-DNS" untuk mencocokkan lalu lintas.
- Antrian induk adalah "TOTAL_SPEED".
- Prioritas diatur menjadi 4/4.
- Menggunakan jenis antrian *pcq-upload-default/pcq-download-default*.
- Target rentang alamat IP untuk lalu lintas yang cocok dengan antrian ini adalah 192.168.15.0/24, 192.168.33.0/24, dan 192.168.10.0/24.

```
add bucket-size=0.8/0.8 name=TRAFIK-WA packet-marks=Paket-WhatsApps parent=\TOTAL_SPEED priority=1/1 queue=pcq-upload-default/pcq-download-default \target=192.168.33.0/24,192.168.15.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TRAFIK-WA" untuk mengelola lalu lintas WhatsApp.
- Menggunakan tanda paket "Paket-WhatsApps" untuk mencocokkan lalu lintas.
- Antrian induk adalah "TOTAL_SPEED".
- Prioritas diatur menjadi 1/1.
- Menggunakan jenis antrian *pcq-upload-default/pcq-download-default*.

- f. Target rentang alamat IP untuk lalu lintas yang cocok dengan antrian ini adalah 192.168.33.0/24, 192.168.15.0/24, 192.168.1.0/24, dan 192.168.10.0/24.

```
add bucket-size=1/1 max-limit=20M/20M name=TRAFIK-GAME packet-marks=\
Paket-Game parent=TOTAL_SPEED priority=2/2 queue=\
pcq-upload-default/pcq-download-default target=\
192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TRAFIK-GAME" untuk mengelola lalu lintas game.
- Menggunakan tanda paket "Paket-Game" untuk mencocokkan lalu lintas.
- Antrian induk adalah "TOTAL_SPEED".
- Prioritas diatur menjadi 2/2.
- Menggunakan jenis antrian *pcq-upload-default/pcq-download-default*.
- Target rentang alamat IP untuk lalu lintas yang cocok dengan antrian ini adalah 192.168.15.0/24, 192.168.33.0/24, dan 192.168.10.0/24.

```
add max-limit=10M/10M name=TRAFIK-TELEGRAM packet-marks=Paket-Telegram \
parent=TOTAL_SPEED priority=5/5 queue=\
pcq-upload-default/pcq-download-default target=\
192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "TRAFIK-TELEGRAM" untuk mengelola lalu lintas Telegram.
- Menggunakan tanda paket "Paket-Telegram" untuk mencocokkan lalu lintas.
- Antrian induk adalah "TOTAL_SPEED".
- Prioritas diatur menjadi 5/5.
- Menggunakan jenis antrian *pcq-upload-default/pcq-download-default*.
- Target rentang alamat IP untuk lalu lintas yang cocok dengan antrian ini adalah 192.168.15.0/24, 192.168.33.0/24, dan 192.168.10.0/24.

```
add name=HOTSOPT+PPPOE parent=TOTAL_SPEED queue=default/default target=\
192.168.15.0/24,192.168.33.0/24,192.168.10.0/24
```

- Perintah ini membuat simple *queue* dengan nama "HOTSOPT+PPPOE" dan menggunakan jenis antrian *default/default*.
- Target antrian ini adalah rentang alamat IP 192.168.15.0/24, 192.168.33.0/24, dan 192.168.10.0/24 untuk lalu lintas yang cocok dengan antrian ini.

Hasil dari konfigurasi Simple quequ di tampilkan pada gambar 13.

#	Name	Target	Upload Max Limit	Download Max Limit	Packet Marks
0	TOTAL_SPEED	192.168.15.0/24, 192.168.10.0/2...	unlimited	unlimited	
3	TRAFIK-WA	192.168.33.0/24, 192.168.15.0/2...	unlimited	unlimited	Paket-WhatsApps
5	TRAFIK-TELEGRAM	192.168.15.0/24, 192.168.33.0/2...	15M	15M	Paket-Telegram
2	TRAFIK-PING-DNS-ICMP	192.168.15.0/24, 192.168.33.0/2...	15M	15M	Paket-Ping-DNS
1	TRAFIK-MARKETPLACE	192.168.33.0/24, 192.168.15.0/2...	15M	15M	Paket-MarketPlace
4	TRAFIK-GAME	192.168.15.0/24, 192.168.33.0/2...	20M	20M	Paket-Game
6	HOTSOPT+PPPOE	192.168.15.0/24, 192.168.33.0/2...	unlimited	unlimited	
57 D	<pppoe-wildanaira>	<pppoe-wildanaira>	1796k	3592k	
37 D	<pppoe-udinsatu>	<pppoe-udinsatu>	2155k	4311k	
34 D	<pppoe-udinpo>	<pppoe-udinpo>	2275k	4551k	
42 D	<pppoe-tanteday>	<pppoe-tanteday>	1796k	3592k	
28 D	<pppoe-sultan>	<pppoe-sultan>	2275k	4551k	
40 D	<pppoe-siah>	<pppoe-siah>	1896k	3792k	
12 D	<pppoe-shaka1>	<pppoe-shaka1>	1896k	3792k	
9 D	<pppoe-santyh>	<pppoe-santyh>	1796k	3592k	
13 D	<pppoe-saha>	<pppoe-saha>	1796k	3592k	
18 D	<pppoe-riki>	<pppoe-riki>	1706k	5120k	
53 D	<pppoe-ria1>	<pppoe-ria1>	1796k	3592k	
27 D	<pppoe-rendy>	<pppoe-rendy>	1796k	3592k	
44 D	<pppoe-pakrt9>	<pppoe-pakrt9>	1896k	3792k	
60 D	<pppoe-noryana>	<pppoe-noryana>	1706k	5120k	
41 D	<pppoe-nisa>	<pppoe-nisa>	2275k	4551k	
15 D	<pppoe-nasulpoe>	<pppoe-nasulpoe>	2275k	4551k	
29 D	<pppoe-nasrul>	<pppoe-nasrul>	2275k	4551k	
32 D	<pppoe-martul>	<pppoe-martul>	2275k	4551k	
19 D	<pppoe-mariana>	<pppoe-mariana>	2275k	4551k	
50 D	<pppoe-mardhiiah>	<pppoe-mardhiiah>	16M	16M	
17 D	<pppoe-mamarobi>	<pppoe-mamarobi>	2275k	4551k	
80 D	<pppoe-mamaemi>	<pppoe-mamaemi>	2275k	4551k	
16 D	<pppoe-lia>	<pppoe-lia>	1796k	3592k	
24 D	<pppoe-lastripoe>	<pppoe-lastripoe>	1796k	3592k	
55 D	<pppoe-jube>	<pppoe-jube>	1796k	3592k	
7 D	<pppoe-januaripoe>	<pppoe-januaripoe>	2275k	4551k	
20 D	<pppoe-icha>	<pppoe-icha>	1896k	3792k	
22 D	<pppoe-hana>	<pppoe-hana>	1706k	5120k	

Gambar 5 Hasil konfigurasi Simple Quequ

1.2.6 Konfigurasi Iper3

Pada Iper3-3.1.3 sendiri tidak banyak konfigurasi yang digunakan dikarenakan Iper3 merupakan *tool* untuk membuat simulasi testing pada jaringan yang berjalan pada Base sistem atau CMD pada sistem operasi windows pengguna hanya perlu memanggil Iper3 pada CMD untuk menjalankannya.

The image shows two side-by-side Windows Command Prompt windows. The left window, titled 'Command Prompt - iperf3.exe', displays the output of a server running Iperf3. It shows multiple client connections from 192.168.15.182 to 192.168.15.182 on port 5201, with throughput statistics for each connection and a summary. The right window, titled 'Command Prompt', shows the command 'C:\Users\muham\Downloads\iperf-3.1.3>iperf3.exe -c 192.168.15.182 -t 20 -p 5 -b 1024k' and its output, including connection details and throughput statistics.

gambar 15 konfigurasi iperf3 1

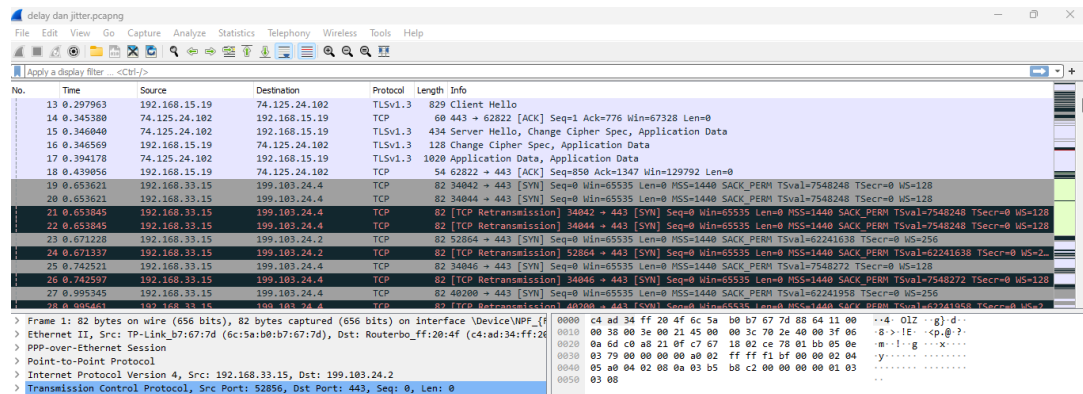
Perintah yang digunakan pada Iperf3 “`iperf-3.1.3>iperf3.exe -c 192.168.15.182 -t 20 -p 5 -b 1024k`”:

- `iperf3.exe`: Ini adalah alat yang digunakan untuk memanggil extension pada Iperf3.
- `-c 192.168.15.182` adalah fitur untuk memeriksa seberapa cepat data dapat dikirim ke perangkat tertentu dengan alamat 192.168.15.182.
- `-t 300` : Pengujian akan berjalan selama 5 Menit, seperti mengukur seberapa cepat sesuatu dilakukan dalam waktu 5 Menit.
- `-p 5`: merupakan fitur untuk membuat semualsi Pararel stream pada client dimana angka 5 berarti simulasi dilakukan dengan membuat 5 steam pada client
- `-b 2M`: fitur ini dapat memberikan beban bandwidth pada client, 2 megabyte per detik (2Mbps) selama pengujian.

1.2.7 Konfigurasi Wireshark

Pada Wireshark sendiri tidak banyak konfigurasi yang digunakan dikarenakan Wireshark hanya merupakan *tool Network Analyzer*. Pada konfigurasi awal untuk mencari *throughput* adalah membuka data hasil pengujian, pada menu pilih stastistic kemudian pilih capture file properties untuk menampilkan data hasil uji dan dilakukan perhitungan menggunakan rumus *throughput* yang sudah ada, untuk mencari *packet loss* menggunakan perintah *tcp.analysis.lost_segment* yang berguna filter data, pada menu pilih stastistic kemudian pilih capture file properties untuk menampilkan data hasil uji dan dilakukan perhitungan menggunakan rumus *packet loss* yang sudah ada, mencari *Delay* dan *jitter* hanya

menggunakan perintah tcp untuk filter data kemudian data disimpan pada format CSV untuk mempermudah perhitungan. untuk tampilan Wireshark dapat ditampilkan pada gambar 14.



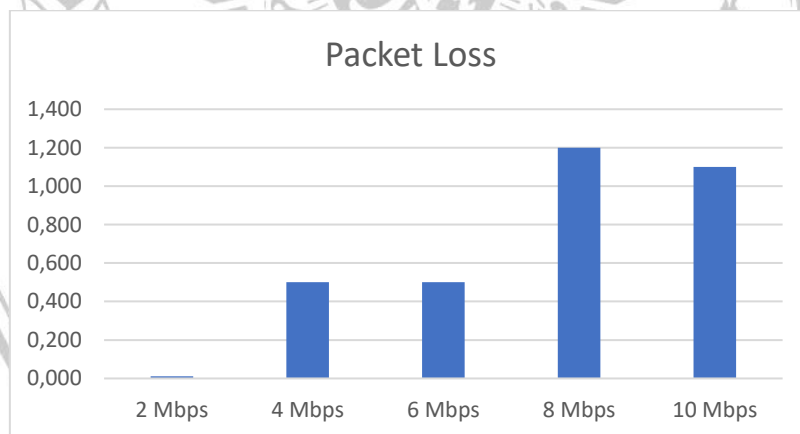
Gambar 6 konfigurasi Wireshark

4.3 Hasil pengujian dan analisis data

4.3.1 Hasil Pengujian Packet Loss

Table 1 Hasil testing Paket loss

BEBAN TRAFIK <i>bandwidth</i>	Paket Loss
2 Mbps	0,012 %
4 Mbps	0,5 %
6 Mbps	0,5 %
8 Mbps	1,2 %
10 Mbps	1,1 %



Gambar 7 Hasil Pengujian packet loss

Gambar 15 menunjukkan hasil pengukuran parameter QoS *Packet loss* berdasarkan skenario pengujian yang telah ditentukan sebelumnya pada Bab 3. Dari *Table 12* dan grafik pada gambar 15 dapat terlihat kenaikan nilai pembebanan yang diberikan pada jaringan menyebabkan terjadinya kenaikan pada *packet loss*. Untuk pembebanan 2 *Mbps* nilai *packet loss* yang didapat yaitu sebesar 0.012%, sedangkan untuk nilai pembebanan 4 *Mbps* terjadi kenaikan cukup signifikan pada *packet loss*, yaitu sebesar 0.488%. Demikian juga pada pembebanan jaringan

sebesar 8 Mbps terjadi kenaikan *packet loss* sebesar 0.7%. Secara keseluruhan *packet loss* rata-rata yang dihasilkan dengan variasi pembebanan yang diberikan masih kecil.

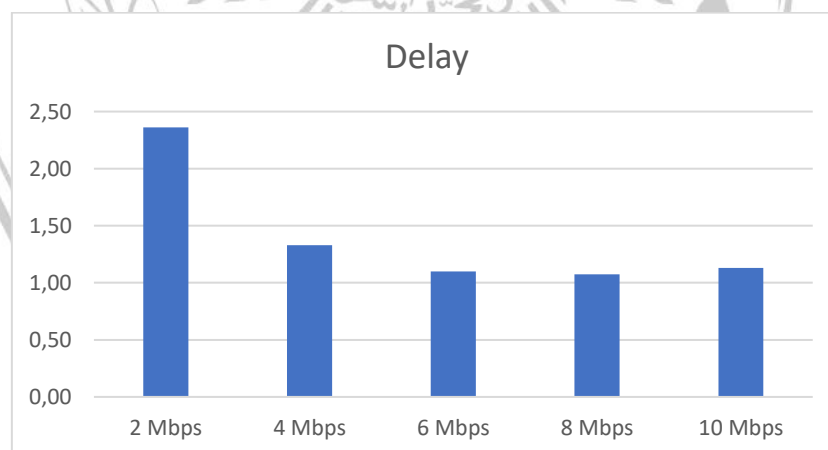
Menurut standarisasi THIPON, *packet loss* yang dihasilkan dalam pengujian, masih masuk dalam kategori yang wajar, dan masih dibawah rata-rata. Semakin kecil nilai *packey loss*, maka semakin baik jaringan tersebut berdasarkan standar TIPHON dimana pengujian *packet loss* memiliki golongan indeks yang sangat baik jika kurang dari 3%. Terjadinya kenaikan dan fluktuasi pada nilai *packet loss* terjadi karena faktor-faktor *external* di luar kedali pengujian seperti *divice* yang kurang memadai atau interferensi-interferensi jaringan lain ketika pengujian dilakukan.

Nilai rata-rata *packet loss* yang cukup kecil menunjukkan bahwa metode *policy-base routing* (PBR) yang digunakan dapat bekerja dengan baik. Metode PBR dapat memetakan layanan, mempersingkat jalur akses paket ke tujuan, mengurangi *waktu tunggu*, dan mengurangi terjadinya *miss* paket pada saat pengiriman paket ke tujuan.

4.3.2 Hasil Pengujian Delay

Table 2 Tabel Hasil testing Delay

BEBAN TRAFIK <i>bandwidth</i>	Delay
2 Mbps	2,36 ms
4 Mbps	1,33 ms
6 Mbps	1,10 ms
8 Mbps	1,07 ms
10 Mbps	1,13 ms



Gambar 8 Hasil Pengujian Delay

Gambar 16 menunjukkan hasil pengukuran parameter QoS *delay* berdasarkan skenario pengujian yang telah ditentukan sebelumnya pada Bab 3. Dari Table 13 dan grafik pada gambar 16 dapat terlihat pola grafik yang dengan nilai *delay* yang berbanding terbalik, seiring dengan penambahan nilai pembebanan. Dengan kata lain semakin tinggi nilai pembebanan jaringan

yang dilakukan semakin kecil nilai *delay*. Pada pembebanan 2 *mbps* memiliki hasil paling tinggi di angka 2,36 ms, namun menurun sebesar 1,03 ms pada pembebanan 4 *mbps* dengan angka *delay* di 1,33 ms. *Trend* penurunan ini berlanjut hingga pembebanan 6 dan 8 *mbps* yang mendapatkan nilai *delay* 1,10 ms untuk 6 *mbps* dan 1,07 ms untuk 8 *mbps*. Namun kondisi yang berbeda terjadi pada pembebanan 10 *mbps* yang mengalami kenaikan 0,06 ms dengan nilai pengujian 1,13 ms

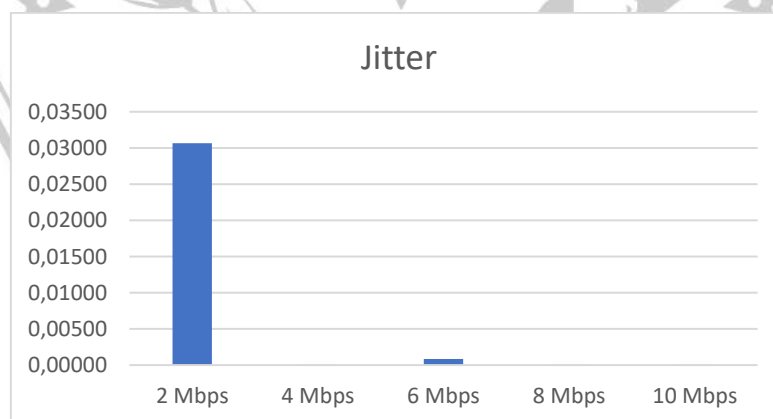
Dari hasil pengujian tersebut, didapatkan bahwa nilai *delay* yang dihasilkan jika dibandingkan dengan standarisasi THIPON masih masuk dalam degradasi **sangat bagus** menurut TIPPON.

Trend penurunan nilai *delay* karena penerapan metode *policy-base routing* (PBR) yang dapat memetakan layanan sehingga menyebabkan *delay* yang relative lebih kecil jika dibandingkan dengan kondisi sebelum diterapkannya metode PBR. Metode PBR menyimpan akses data informasi alamat ataupun layanan yang sudah di akses sebelumnya ke dalam *cache* atau *memory* penyimpanan sementara yang ada di dalam *server* sehingga ketika mengakses layanan yang sama waktu tunggu dapat lebih kecil dari sebelumnya.

4.3.3 Hasil Pengujian *Jitter*

Table 3 Tabel Hasil testing *Jitter*

BEBAN TRAFIK <i>bandwidth</i>	<i>Jitter</i>
2 Mbps	0,03064 ms
4 Mbps	0,00003 ms
6 Mbps	0,00086 ms
8 Mbps	0,00006 ms
10 Mbps	0,00003 ms



Gambar 9 Hasil Pengujian *Jitter*

Gambar 17 menunjukkan hasil pengukuran parameter QoS pada *Jitter* berdasarkan skenario pengujian yang telah ditentukan sebelumnya pada Bab 3. Dari Table 14 dan grafik pada gambar

17 dapat dilihat bahwa terjadi penurunan nilai *jitter* seiring dengan kenaikan nilai pembebanan pada jaringan pada pembebanan 2 *mbps* dihasilkan nilai *jitter* 0,03064 ms, namun nilai *jitter* ini menurun secara signifikan pada pembebanan 2 *Mbps*, 6 *Mbps*, 8, *Mbps*, dan 10 *Mbps*, yang menghasilkan nilai *jitter* dibawah 0.005 ms.

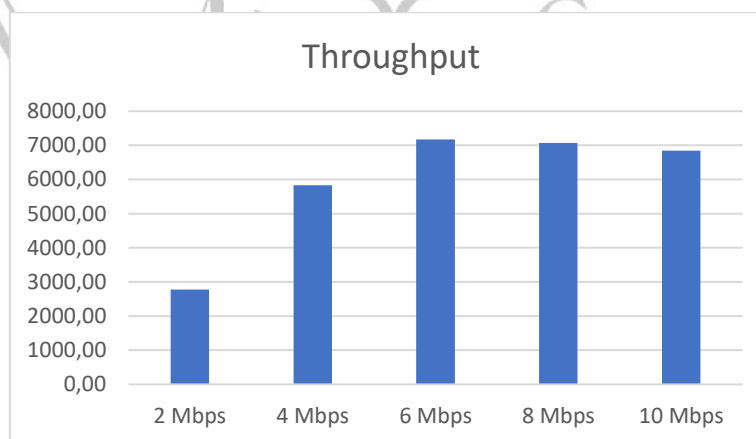
Nilai *jitter* yang sangat kecil ini disebabkan karena konfigurasi yang baik dan penerapan metode yang tepat sesuai dengan kebutuhan spesifik jaringan, faktor faktor yang membuat kecil nya nilai QoS *jitter* yaitu sistem pengalihan lalu lintas dan sistem prioritas lalu lintas. Dimana metode PBR berperan penting dalam kecil nya nilai *jitter*, karena PBR dapat dikonfigurasi untuk melakukan pengalihan jaringan sehingga administrator jaringan dapat memilih jenis layanan atau protokol tertentu, dengan PBR administrator jaringan juga dapat memberikan prioritas pada jenis lalu lintas atau alamat *IP* tertentu. Penerapan metode PBR memberikan fleksibilitas kepada jaringan untuk menyesuaikan kebutuhan spesifik jaringan, hal ini lah yang meningkatkan efektifitas dan efisiensi dari jaringan yang mempengaruhi kecil nya nilai *jitter* pada jaringan.

Berdasarkan standar THIPON, hasil pengujian nilai *jitter* ini masih termasuk dalam kategori indeks yang sangat baik. Nilai *jitter* dikatakan sangat baik jika kurang dari 0 ms s/d 75 ms menurut standarisasi THIPON.

4.3.4 Hasil Pengujian Throughput

Table 4 Hasil Testing Throughput

BEBAN TRAFIK <i>bandwidth</i>	Throughput
2 Mbps	2775,30 <i>bps</i>
4 Mbps	5832,57 <i>bps</i>
6 Mbps	7174,35 <i>bps</i>
8 Mbps	7066,99 <i>bps</i>
10 Mbps	6847,13 <i>bps</i>



Gambar 10 Hasil Pengujian Througput

Gambar 18 menunjukkan hasil pengukuran parameter QoS kategori *throughput* berdasarkan skenario pengujian yang telah ditentukan sebelumnya pada Bab 3. Dari *Table 15* dan grafik pada gambar 18 dapat dilihat bahwa terjadi kenaikan nilai *throughput* seiring dengan adanya kenaikan pembebanan pada jaringan. Pada pembebanan 2 *mbps* menghasilkan nilai *throughput* sebesar 2775,30 *bps*, namun meningkat sebesar 3.057,27 *bps* pada pembebanan 4 *mbps* dengan angka di 5832,57 *bps*, kondisi ini terus berlangsung sampai pada pembebanan 6 *mbps* yang mendapat nilai QoS 7174,35 *bps*. Kondisi berbeda terjadi pada pembebanan 8 dan 10 *mbps* dimana terjadi sedikit penurunan pada *throughput* yang dihasilkan, yaitu sebesar 7066,99 dan 6847,13.

Berdasarkan pada standarisasi THIPON *throughput* memiliki golongan indeks yang sangat baik jika lebih dari 100 *bps*. Pada hasil pengujian QoS nilai paling rendah terjadi pada pembebanan 2 *mbps* yaitu di 2775,30 *bps*, hasil ini menempati indeks sangat baik dalam standarisasi TIPHON.

Nilai *throughput* yang bagus tersebut dipengaruhi oleh sistem antrian yang memiliki hierarki atau prioritas yang terstruktur dengan baik, yang berpengaruh pada distribusi *bandwidth* efektif dan efisien, dimana penerapan metode *simple queue* yang berfungsi dalam pendistribusian *bandwidth* kepada *user*. Metode PBR juga berpengaruh terhadap nilai *throughput* yang bagus, karena PBR dapat memberikan fleksibilitas dalam pengelolaan *source* dalam jaringan, sehingga administrator jaringan dapat memaksimalkan *bandwidth* yang tersedia dalam jaringan.