

ANALISA MANAGEMENT BANDWIDTH MENGGUNAKAN QUEUE TREE SEBAGAI TRAFFIC CONTROL DENGAN METODE PFIFO

¹Ahmad Syahlan Danu Terto, Eka Purwa Laksana²

¹Teknik Elektro : Universitas Budi Luhur
Jakarta, Indonesia

1552500058@student.budiluhur.com

²Teknik Elektro : Universitas Budi Luhur
Jakarta, Indonesia

eka.purwalaksana@budiluhur.ac.id

ABSTRAK

Pada Penelitian ini dilakukan penelitian Management bandwidth yang merupakan suatu teknik pengaturan jumlah bandwidth dalam melakukan unggah dan unduh data. Management bandwidth dilakukan dengan menggunakan algoritma queue tree pada Router OS. Pada penelitian ini metode yang digunakan adalah metode pfifo, dimana metode tersebut diterapkan pada Router OS sehingga dapat dilakukan perbandingan terhadap parameter-parameter Quality of Service(QoS) sebelum dan sesudah diterapkan pada Router OS. Nilai dari parameter-parameter Quality of Service(QoS) didapatkan dengan melakukan unggah suatu data ke sebuah website kemudian dilakukan unduh data. Pada penelitian ini dilakukan traffic control yang memisahkan traffic unggah dan unduh, sekaligus menerapkan antrian paket yang melalui Router OS tersebut. Hasil yang didapat berupa perubahan traffic internet yang telah melebihi ukuran 22mb dipindahkan pada traffic berat.

Kata Kunci : Management Bandwidth, Queue tree, PFIFO, Traffic control, Quality of Service(QoS).

ABSTRACT

In this research conducted the research of bandwidth Management which is a technique of setting the amount of bandwidth in uploading and downloading data. Bandwidth Management is done using the queue tree algorithm on the OS Router. In this research method used is the method Pffifo, where the method is applied to the Router OS so that it can be done a comparison to the parameter-Parameter Quality of Service (QoS) before and after applied to the Router OS. The values of the Quality of Service (QoS) parameters are obtained by uploading a data to a website and then downloading the data. In this research, traffic control separates upload and download traffic, while applying packet queues through the OS Router. Results obtained in the form of Internet traffic changes that have exceeded the size of 22MB are moved on heavy traffic.

Keywords: Bandwidth Management, Queue tree, PFIFO, Traffic control, Quality of Service (QoS).

I. PENDAHULUAN

Semakin berkembangnya teknologi informasi sekarang ini mengakibatkan kebutuhan akan informasi semakin meningkat, maka diperlukan koneksi internet yang cepat dan stabil. Terbatasnya bandwidth internet, sehingga harus secara bijak dalam menggunakan bandwidth yang tersedia dengan sebaik mungkin. Jika tidak diatur, kemungkinan besar bandwidth akan digunakan oleh beberapa orang saja.

Management bandwidth merupakan pengaturan bandwidth atau prioritas antrian terhadap pengguna dalam jaringan. Karena bandwidth merupakan salah satu unsur yang sangat

vital pada sebuah jaringan maka sangat diperlukan adanya pengaturan yang baik agar pengguna dapat mengakses dengan cepat. Quality of Service (QoS) memegang peranan yang sangat penting dalam memfasilitasi proses Management bandwidth dalam suatu jaringan. Salah satu usaha yang dapat dilakukan untuk menjamin kelancaran internet dan maksimalnya penggunaan bandwidth dalam jaringan yang dapat diterapkan pada router/gateway internet adalah dengan menerapkan perbedaan pembagian bandwidth dengan kebutuhan yang diprioritaskan. Dengan memanfaatkan berbagai teknik khususnya dalam Management bandwidth pada router/gateway internet.

Pada jurnal (Pamungkas, 2016) manajemen *bandwidth* dilakukan dengan menggunakan *routerboard* dan melakukan konfigurasi TCP, DNS, dan DHCP[1].

Pada analisa jurnal lainnya menjelaskan (Zuli, 2015) penggunaan manajemen *bandwidth* dilakukan dengan menggunakan metode *simple queue* dengan membagi 100 Mbps / 102.400 Kbps setiap pengguna yang berdasarkan penggunaan internet[2].

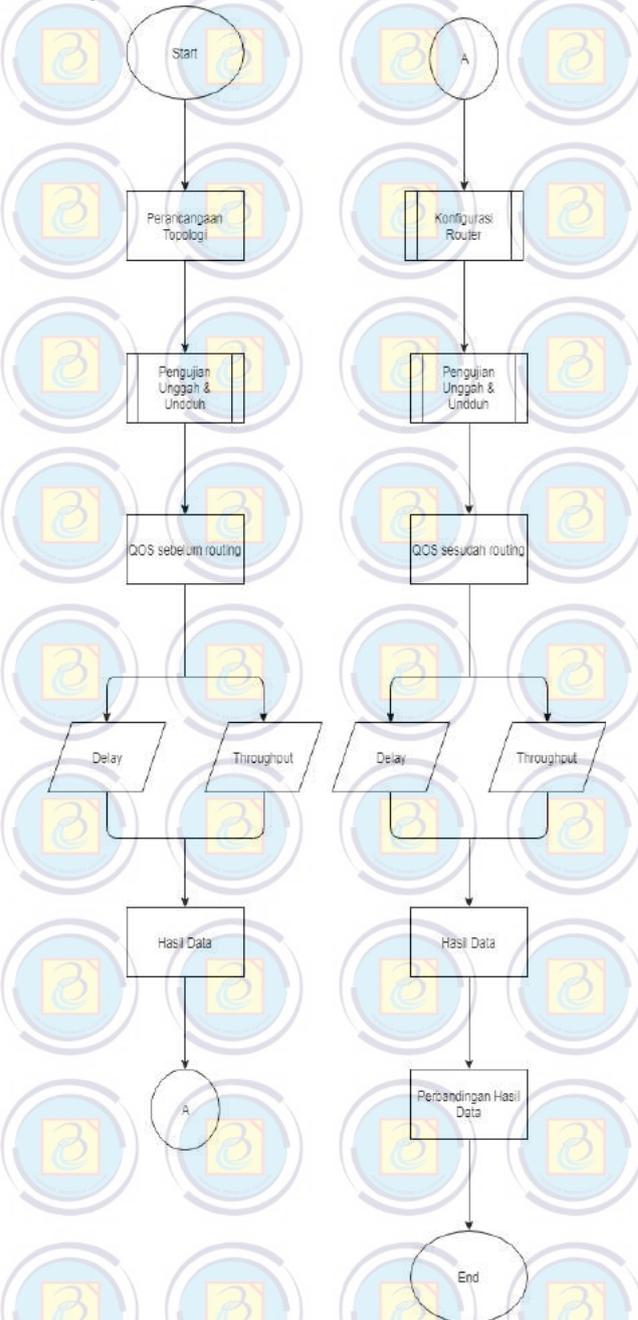
Pada jurnal (Ziomek, Kinross-Wright and Plato, 2015) melakukan perbandingan metode *simple queue* dan *queue tree* didapatkan hasil dengan metode *queue tree* mengoptimalkan manajemen *bandwidth* setiap pengguna dan dapat menyesuaikan kebutuhan[3]. Pada jurnal (Galeh Fatma Eko Ardiansa, Primananda and Hanafi, 2017) menjelaskan manajemen *bandwidth* jaringan *wireless* mesh dengan penelitian self-healing dan self-configure diperoleh waktu masing – masing adalah 27,5 detik dan 26 detik[4]. Pada jurnal lain menjelaskan (Zamuswara, 2017) Manajemen *Bandwidth* Menggunakan Metode Hierarchical Token Bucket (HTB) pedistribusi *bandwidth* untuk komputer user telah berhasil dioptimalisasi dan pemerataan *bandwidth* per divisi telah berhasil dilakukan[5].

Pada penelitian ini dilakukan *Management bandwidth* dengan menggunakan *queue tree* dengan metode *pfifo* (*packet first in first out*) yang akan dibagi menjadi empat trafik bagian yaitu trafik unggah ringan, unggah berat, unduh berat dan unduh ringan yang akan dianalisa dengan wireshark dengan 2 mengunggah dan mengunduh file besar (520 MB) dan file ringan (22 MB). Pengunggahan dan pengunduhan file dilakukan pada *google drive* dan dilakukan analisa *Quality of Service(QoS)*.

II. PERANCANGAN SISTEM DAN METODE PENELITIAN

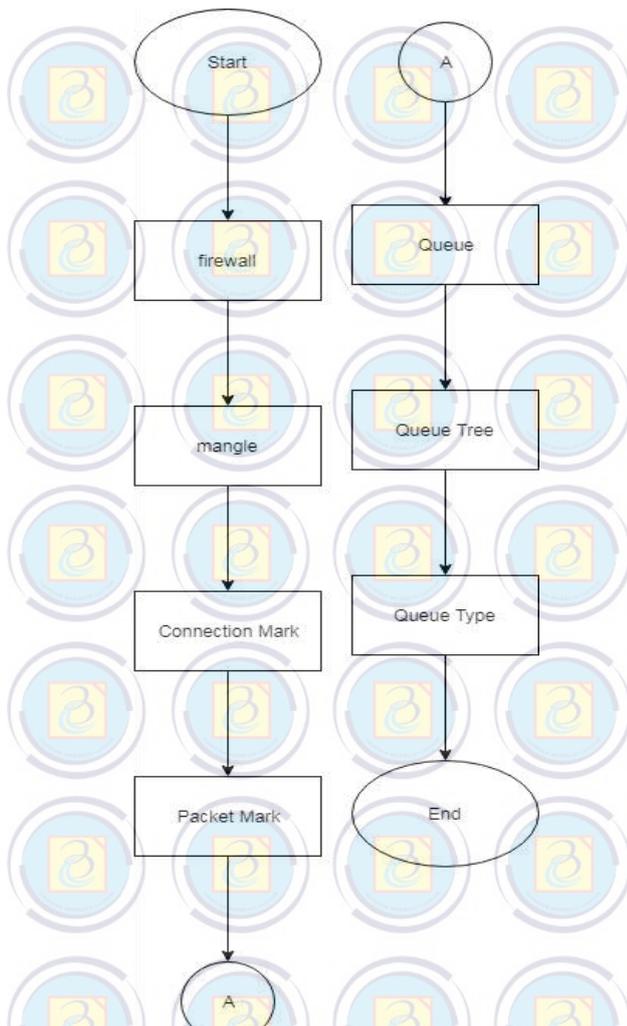
Perancangan sistem dengan metode *pfifo* (*packet first in first out*). Penelitian untuk mendapatkan *Quality of Services* berupa parameter *delay*, dan *throughput* pada unggah dan unduh file yaitu 520 MB dan 22 MB. *Quality of services* yang didapatkan sebelum dilakukan konfigurasi dan sesudah konfigurasi pada *Router OS*. QoS (*Quality of Services*) didapatkan dengan melakukan pengujian unggah dan unduh pada *google drive* dari dua file yaitu 520 MB dan 22 MB. Konfigurasi jaringan menggunakan VMware.

A. Diagram Alir Sistem



Gambar 1. Diagram Alir Sistem

Gambar 2. Sub diagram alir konfigurasi router



B. Skema Penelitian

Skenario yang dilakukan pada penelitian ini, dengan cara melakukan pengujian unggah dan unduh untuk mendapatkan nilai *QoS* (*Quality of Service*) berupa *delay* dan *throughput*.

1. Pengujian unggah

Pengujian unggah akan dilakukan dengan cara mengunggah file dari laptop menuju website *google drive* dengan dua ukuran yang yaitu ukuran 22 MB dan ukuran ke 520 MB. Pengujian ini akan dibandingkan hasil *Quality of Service* (*QoS*) berupa *delay* dan *throughput* yang di dapatkan dari sebelum konfigurasi dan sesudah konfigurasi pada *Router OS*.

2. Pengujian unduh

Pengujian unduh akan dilakukan dengan unduh file dari *google drive* menuju laptop yang digunakan dengan dua ukuran yang yaitu ukuran 520 MB dan ukuran 22 MB. Unduh dilakukan dengan menggunakan *software internet download manager* yang bisa memaksimalkan *traffic* unduh yang ada.

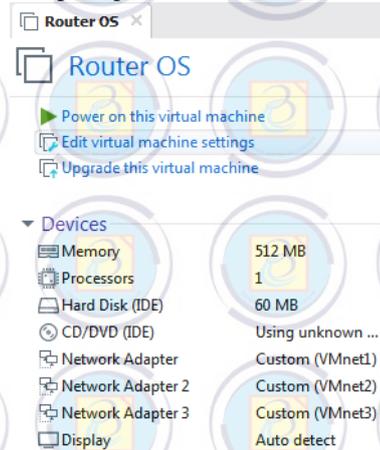
Pengujian ini akan dibandingkan hasil *Quality of Service* (*QoS*) berupa *delay* dan *throughput* yang di dapatkan dari sebelum konfigurasi dan sesudah konfigurasi pada *Router OS*.

C. Langkah Pengujian

Pengujian yang digunakan dalam melakukan simulasi ini, diantaranya yaitu konfigurasi *VMWare*, konfigurasi *ipv4* pada *Router OS*, konfigurasi *ipv4* pada *Personal Komputer*, konfigurasi *ipv4* pada *access point*.

1. Konfigurasi *VMWare*

Perangkat untuk simulasi yang digunakan pada penelitian ini adalah *VMWare*. Tahap awal simulasi yaitu menentukan penggunaan *hardware* yang akan digunakan seperti pada Gambar 3.



Gambar 3. Setting *VMware*

Setelah selesai dengan pemilihan penggunaan *hardware* pada *VMware*. Selanjutnya menentukan *dynamips* emulator. *Dynamips* emulator mirip dengan perangkat nyata *router*, karena dapat mengemulasikan *router* lengkap dengan OS-nya. Dilanjutkan dengan memilih *Router OS* yang digunakan.

2. Konfigurasi IP Firewall Mangle

Selanjutnya dilakukan konfigurasi *ip firewall mangle* pada *Router OS*, dengan syntaks atau perintah :

```

/ip firewall
add action=mark-connection chain=forward
connectionmark=!koneksiiberatnewconnect
ion=mark=semua-koneksi
add action=mark-connection chain=forward
connection-bytes=2200000
connection=mark=semuakoneksinewconnect
ion-mark=koneksi-berat protocol=tcp
add action=mark-connection chain=forward
connection-bytes=2200000 connection-
  
```

```

mark=semua-koneksi      new-connection-
mark=koneksi-berat protocol=udp
add action=mark-packet    chain=forward
connection-mark=koneksi-berat new-
mark-packet=traffic-berat
passthrough=no
add action=mark-packet    chain=forward
connection-mark=semua-koneksi new-
mark-packet=traffic-ring
passthrough=no

```

Syntax atau perintah di atas menandakan bahwa router telah diaktifkan sesuai dengan ip firewall magle yang diberikan. Setelah selesai memasukan semua perintah Router OS, maka dilanjutkan dengan mengkonfigurasi queue tree.

3. Konfigurasi Queue tree

Dalam penelitian digunakan queue tree untuk memisahkan traffic. Berikut konfigurasi dari queue tree:

```

/queue tree
add max-limit=15M name=upload parent=ether1
add limit-at=10M max-limit=10M name=traffic-
ringan-upload packet-mark traffic-
ringan parent=upload priority=1
add limit-at=5M max-limit=5M name=traffic-
berat-upload packet-mark traffic-berat
parent=upload
add maxlimit=10M name=download parent=ether2
add limit-at=6M max-limit=6M name=traffic-
ringan-download packet-mark traffic-
ringan parent=download priority=1
add limit-at=4M max-limit=4M name=traffic-
berat-download packet-mark traffic-
berat parent=download

```

4. Parameter Quality of Service (QoS)

Quality of Service merupakan kemampuan untuk memberikan prioritas yang berbeda untuk berbagai aplikasi, pengguna, atau aliran data, atau untuk menjamin tingkat kinerja pada aliran data. QoS bertujuan untuk menyediakan kualitas layanan yang berbeda-beda untuk beragam kebutuhan akan layanan di dalam jaringan IP. Parameter-parameter dalam QoS antara lain: Delay dan throughput.

5. Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu node ke node lain yang menjadi tujuannya Tabel 1.

Tabel 1. Kategori Delay (Sumber : TIPHON)

Kategori Latensi	Besar delay(ms)
Sangat Bagus	< 150 ms
Bagus	150 ms s/d 300 ms

Sedang	300 ms s/d 450 ms
Jelek	> 450 ms

6. Throughput

Throughput adalah kecepatan rata-rata data yang diterima oleh suatu node dalam selang waktu pengamatan tertentu. Berikut Tabel 2 merupakan kategori dari throughput.

Tabel 2. Kategori Throughput (Sumber : TIPHON)

Kategori Throughput	Throughput
Sangat Bagus	> 2,1 Mbps
Bagus	1200 kbps – 2,1 Mbps
Sedang	700 – 1200 kbps
Jelek	338 – 700 kbps
Sangat Jelek	0 – 338 kbps

III. PENGUJIAN DAN ANALISA

Menguraikan hasil dan kinerja sistem serta analisa menggunakan parameter Quality of Service berupa delay dan throughput. Pengukuran ini di uji coba sebanyak lima kali dengan ukuran unggah 520MB (file berat) dan 22 MB (file ringan). Uji coba dilakukan sebelum konfigurasi dan sesudah konfigurasi Router OS. Capture dilakukan dari mulai proses unggah dan unduh hingga selesai pengujian.

1. PENGUJIAN UNGGAH

Pada pengujian ini dilakukan unggah data melalui google drive yang digunakan untuk penyimpanan data. Untuk dapat menganalisis unggah data maka akan diambil capture data dari software network analyzer wireshark, guna mendapatkan hasil parameter Quality of Service. Pengujian dilakukan dengan lima kali pengujian dengan dua ukuran file unggah untuk mendapatkan hasil perbandingan sebelum dan sesudah konfigurasi dari kedua file tersebut. Untuk kemudian didapatkan rata-rata dari hasil pengujian. Capture diambil pada waktu 07.30 WIB – 12.00 WIB melalui network analyzer wireshark, setelah itu dilakukan pemberhentian capture. Kemudian didapat hasil rata-rata delay dan throughput. Dari pengukuran analisa data dari software network analyzer wireshark didapatkan data hasil waktu delay saat unggah. Pengujian.

- Waktu Delay saat Unggah

Pengujian waktu *delay* ini diambil data pengujian dari software wireshark dengan ukuran file yang sama yaitu 22MB dan 520MB dengan lima kali pengujian. Dari hasil pencarian nilai *delay* pada *wireshark* ditunjukkan statistiknya pada Gambar 4.

Statistics			
Measurement	Captured	Displayed	Marked
Packets	29780	29637 (99.5%)	—
Time span, s	56.933	49.277	—
Average pps	523.1	601.4	—
Average packet size, B	872	875	—
Bytes	25959607	25939637 (99.9%)	0
Average bytes/s	455 k	526 k	—
Average bits/s	3647 k	4211 k	—

Gambar 4. Pencarian Waktu *Delay* Pada Wireshark

Dari Gambar 4 dapat dihitung rata-rata *delay* menggunakan persamaan, sebagai acuan dalam mengetahui besarnya rata-rata *delay* dari total paket yang dikirim pada waktu pengujian, sebagai acuan untuk mengetahui rata-rata *delay* menggunakan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{\text{total delay}}{\text{total paket yang diterima}}$$

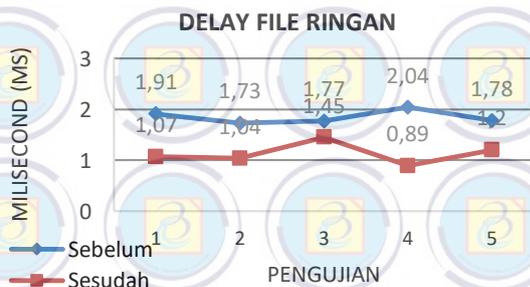
Perhitungan menggunakan rumus diambil sample pengujian pada pengujian ke-1 pada unggah file ringan sebelum konfigurasi, untuk dilakukan perhitungan dengan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{56,933}{29780} = 0,00191 \text{ s} = 1,91 \text{ ms}$$

Perhitungan *delay* diatas sebagai *sample* pembuktian yang dihitung pada pengujian pertama sebelum konfigurasi. Hasil nilai rata - rata *delay* yang didapatkan ditunjukkan pada Tabel 3.

Tabel 3. Hasil *delay* unggah file ringan(22MB)

Pengujian ke	Milisecond (ms)	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	1,91	1,07
2	1,73	1,04
3	1,77	1,45
4	2,04	0,89
5	1,78	1,2
Jumlah	9,23	5,65
Rata-rata	1,84	1,13



Gambar 5. Hasil pengujian unggah *delay* file ringan (22MB)

Hasil pengujian *delay* dari paket yang diambil dari lima kali pengujian pada ukuran file ringan (22 MB) menggunakan *network analyzer*

wireshark. Pada tabel 3 terlihat perbedaan nilai rata-rata *delay* dari lima kali pengujian dengan ukuran file yang sama saat diunggah pada *google drive*. Pengujian di lakukan pada waktu 7.30 – 12.00 WIB dan dilakukan perbandingan sebelum dan sesudah konfigurasi pada *Router OS*.

Statistics			
Measurement	Captured	Displayed	Marked
Packets	667556	666739 (99.9%)	—
Time span, s	1138.686	1138.686	—
Average pps	586.3	585.5	—
Average packet size, B	934	935	—
Bytes	623819211	623691273 (100.0%)	0
Average bytes/s	547 k	547 k	—
Average bits/s	4382 k	4381 k	—

Pengujian waktu *delay* dengan ukuran file 520MB dengan lima kali pengujian. Hasil pencarian nilai *delay* pada *wireshark* ditunjukkan pada gambar 6.

Gambar 6. Pencarian Waktu *Delay* Pada Wireshark

Dari Gambar 6 dapat dihitung rata-rata *delay* menggunakan persamaan, sebagai acuan dalam mengetahui besarnya rata-rata *delay* dari total paket yang dikirim pada waktu pengujian, sebagai acuan untuk mengetahui rata-rata *delay* menggunakan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{\text{total delay}}{\text{total paket yang diterima}}$$

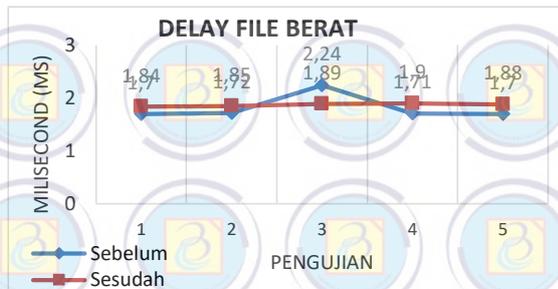
Perhitungan menggunakan rumus diambil sample pengujian pada pengujian ke-1 pada unggah file berat sebelum konfigurasi, untuk dilakukan perhitungan dengan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{1138,686}{667556} = 0,0017 \text{ s} = 1,7 \text{ ms}$$

Perhitungan *delay* diatas sebagai *sample* pembuktian yang dihitung pada pengujian pertama sebelum konfigurasi. Hasil nilai rata - rata *delay* yang didapatkan ditunjukkan pada Tabel 5.

Tabel 5. Hasil *delay* unggah file berat (520 MB)

Pengujian ke	Milisecond (ms)	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	1,7	1,84
2	1,72	1,85
3	2,24	1,89
4	1,71	1,9
5	1,7	1,88
Jumlah	9,07	9,36
Rata-rata	1,81	1,87

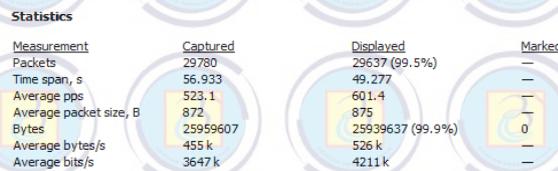


Gambar 7. Hasil pengujian unggah delay file berat (520 MB)

Hasil pengujian *delay* dari paket yang diambil dari lima kali pengujian pada ukuran file berat (520 MB) menggunakan *network analyzer wireshark*. Pada tabel 5 terlihat perbedaan nilai rata-rata *delay* dari lima kali pengujian dengan ukuran file yang sama saat diunggah pada *google drive*. Pengujian di lakukan pada waktu 7.30 – 12.00 WIB dan dilakukan perbandingan sebelum dan sesudah konfigurasi pada *Router OS*.

• Pengujian *Throughput* saat Unggah

Throughput merupakan parameter yang menunjukkan jumlah bit rata-rata data yang dapat ditransfer dari satu *node* ke *node* yang lain perdetiknya dalam suatu jaringan yang terbentuk. Pengujian *throughput* ini diperoleh dari hasil rata-rata 5 kali percobaan pada ukuran file 22MB dan 520MB yang dikirim pada waktu 07.30 WIB - 12.00 WIB dan di *capture* oleh *network analyzer wireshark*. Ditunjukkan pada Gambar 8.



Gambar 8. Pencarian *Throughput* Unggah File Ringan Pertama Pada *Wireshark*

Karena *throughput* menunjukkan kecepatan transfer data suatu jaringan, semakin besar nilai *throughput* akan semakin baik performa jaringan tersebut. Untuk mencari nilai *throughput* dari hasil statistik pada Gambar 8 digunakan persamaan sebagai acuan dalam mengetahui besarnya *throughput* pada masing-masing waktu yaitu :

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{lama pengamatan paket data}}$$

$$\text{Throughput} = \frac{25959607}{56,933} = 455485,2412476554 \text{ byte/s}$$

Didapatkan hasil 455485,2412476554 byte/s, selanjutnya dikonversikan ke Megabit/s dengan perhitungan sebagai berikut:

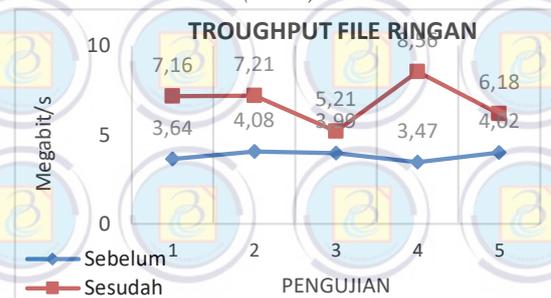
$$\frac{455485,2412476554 \times 8}{1000000} = 3,64 \text{ Mb/s}$$

Perhitungan *throughput* diatas sebagai *sample* pembuktian yang dihitung pada pengujian ke-1 sebelum konfigurasi pada unggah file ringan. Pada Tabel 6 adalah hasil *throughput* unggah file ringan (22MB).

Tabel 6. Hasil *throughput* unggah file ringan (22 MB)

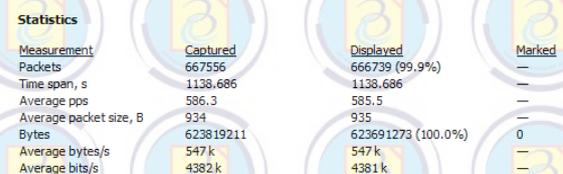
Pengujian ke	Average Mbps	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	3,64	7,16
2	4,08	7,21
3	3,99	5,21
4	3,47	8,56
5	4,02	6,18
Jumlah	19,2	34,32
Rata-rata	3,84	6,86

Gambar 9. Hasil pengujian *throughput* unggah file ringan (22 MB)



Pada Tabel 6 dapat dilihat bahwa nilai *throughput* sebelum konfigurasi memiliki rata-rata nilai atau kecepatan akses data paling besar sebesar 4,08 Mbit/s, sedangkan pada sesudah konfigurasi memiliki rata-rata atau kecepatan akses data sebesar 8,56 Mbit/s. Dari grafik pengujian unduh file ringan *throughput* ini didapatkan bahwa pengujian sesudah konfigurasi terdapat *throughput* yang cukup tinggi yaitu dengan rata-rata sebesar 8,56 Mbit/s, sedangkan *throughput* terendah terjadi sebelum konfigurasi dengan rata-rata yaitu sebesar 3,47 Mbit/s.

Pengujian *throughput* dengan ukuran file 520MB dengan lima kali pengujian. Hasil pencarian *throughput* pada *wireshark* ditunjukkan pada gambar 10.



Gambar 10. Pencarian *Throughput* Unggah File Berat Pertama Pada *Wireshark*

Untuk mencari nilai *throughput* dari hasil statistik pada Gambar 10 digunakan persamaan

sebagai acuan dalam mengetahui besarnya *throughput* pada masing-masing waktu yaitu :

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{lama pengamatan paket data}}$$

$$\text{Throughput} = \frac{623819211}{1138,686} = 547841,2933855339 \text{ byte/s}$$

Didapatkan hasil 547841,2933855339 byte/s, selanjutnya dikonversikan ke Megabit/s dengan perhitungan sebagai berikut:

$$\frac{547841,2933855339 \times 8}{1000000} = 4,38 \text{ Mb/s}$$

Perhitungan *throughput* diatas sebagai *sample* pembuktian yang dihitung pada pengujian ke-1 sebelum konfigurasi pada unggah file berat. Pada Tabel 7 adalah hasil *throughput* unggah file berat (520MB).

Tabel 7 Hasil *throughput* unduh file berat (520 MB)

Pengujian ke	Average Mbps	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	4,38	4,04
2	4,34	4,02
3	3,36	3,94
4	4,4	3,91
5	4,43	3,96
Jumlah	20,91	19,87
Rata-rata	4,18	3,97



Gambar 11. Hasil pengujian *throughput* unggah file berat (520 MB)

Pada Tabel 7 dapat dilihat bahwa nilai *throughput* sebelum konfigurasi memiliki rata-rata nilai atau kecepatan akses data paling besar sebesar 4,4 Mbit/s, sedangkan pada sesudah konfigurasi memiliki rata-rata atau kecepatan akses data sebesar 4,04 Mbit/ dalam setiap ukuran *file* yang sama. Dari grafik pengujian unduh berat *throughput* ini didapatkan bahwa pengujian sebelum konfigurasi terdapat *throughput* yang cukup tinggi yaitu dengan rata-rata sebesar 4,4 Mbit/s, sedangkan *throughput* terendah terjadi sebelum konfigurasi dengan rata-rata yaitu sebesar 3,36 Mbit/s.

2. PENGUJIAN UNDUH

Pada pengujian ini dilakukan unduh data melalui *google drive* yang digunakan untuk penyimpanan data. Untuk dapat menganalisis unduh data maka akan diambil data dari *software network analyzer wireshark*, guna mendapatkan hasil parameter *Quality of Service*. Pengujian dilakukan dengan lima kali pengujian dengan dua ukuran file unduh 22 MB dan 520 MB untuk mendapatkan hasil perbandingan sebelum dan sesudah konfigurasi dari kedua file tersebut. kemudian didapatkan rata-rata dari hasil pengujian. *Capture* di ambil pada waktu 07.30 WIB – 12.00 WIB melalui *network analyzer wireshark*, lalu dilakukan pemberhentian *capture*. Kemudian didapat hasil rata-rata *delay dan throughput*. Dari pengukuran berdasarkan analisa data dari *software network analyzer wireshark* didapatkan statistik.

• Pengujian *Delay* saat Unduh

Delay merupakan waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu *node* ke *node* lain yang menjadi tujuannya pengirim dan penerima. Pengukuran unduh *delay* ini diambil paket yang diperoleh dari masing-masing paket yang dikirim dengan ukuran file yang sama yaitu 22MB dan 520MB melalui lima kali pengujian. Hasil pengukuran berdasarkan analisis data dari *wireshark* ditunjukkan pada Gambar 12.

Measurement	Captured	Displayed	Marked
Packets	25854	25725 (99.5%)	—
Time span, s	34.255	28.568	—
Average pps	754.8	900.5	—
Average packet size, B	984	986	—
Bytes	25447735	25375067 (99.7%)	0
Average bytes/s	742 k	888 k	—
Average bits/s	5943 k	7105 k	—

Gambar 12 Pencarian *Delay* saat Unduh File Ringan Pertama Pada Wireshark

Pada gambar 12 dapat dihitung rata-rata *delay* digunakan persamaan sebagai acuan dalam mengetahui berapa besarnya rata-rata *delay* dari total paket yang dikirim pada waktu pengujian, sebagai acuan untuk mengetahui rata-rata *delay* menggunakan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{\text{total delay}}{\text{total paket yang di terima}}$$

Perhitungan menggunakan rumus diambil 1 sample pada pengujian ke-1 sebelum konfigurasi, untuk dilakukan perhitungan dengan rumus pada waktu pengujian unduh berat sebelum konfigurasi yaitu

$$\text{delay rata - rata} = \frac{34,255}{25854} = 0,00132 \text{ s} = 1,32 \text{ ms}$$

Lalu didapatkan rata-rata *delay* pada masing-masing pengujian yang telah diamati. Dari pengukuran berdasarkan analisis data dari *wireshark*

rata-rata *delay* didapatkan hasil perhitungan pada Tabel 8.

Tabel 8 Hasil *delay* unduh file ringan (22 MB)

Pengujian ke	Milisecond (ms)	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	1,32	0,74
2	1,08	0,58
3	0,93	0,82
4	1,06	0,89
5	1,21	0,98
Jumlah	4,67	4,01
Rata-rata	1,16	0,8



Gambar 13 Hasil pengujian unduh *delay* file ringan

Hasil pengujian *delay* dari paket yang diambil dari lima kali pengujian pada ukuran file ringan (22 MB) menggunakan *network analyzer wireshark*. Pada Tabel terlihat perbedaan nilai rata-rata *delay* dari lima kali pengujian dengan ukuran file yang sama saat diunduh pada *google drive*. Pengujian di lakukan pada waktu 7.30 – 12.00 WIB dan dilakukan perbandingan sebelum dan sesudah konfigurasi pada *Router OS*.

Penelitian *delay* unduh dengan ukuran file 520MB dengan lima kali penelitian. Hasil pencarian *delay* pada *wireshark* ditunjukkan pada Gambar 14.

Measurement	Captured	Displayed	Marked
Packets	25854	25725 (99.5%)	—
Time span, s	34.255	28.568	—
Average pps	754.8	900.5	—
Average packet size, B	984	986	—
Bytes	25447735	25375067 (99.7%)	0
Average bytes/s	742 k	888 k	—
Average bits/s	5943 k	7105 k	—

Gambar 14 Pencarian *Delay* saat Unduh Pertama File Berat Pada *Wireshark*

Pada Gambar 14 dapat dihitung rata-rata *delay* digunakan persamaan sebagai acuan dalam mengetahui berapa besarnya rata-rata *delay* dari total paket yang dikirim pada waktu pengujian, sebagai acuan untuk mengetahui rata-rata *delay* menggunakan persamaan sebagai berikut :

$$\text{delay rata - rata} = \frac{\text{total delay}}{\text{total paket yang di terima}}$$

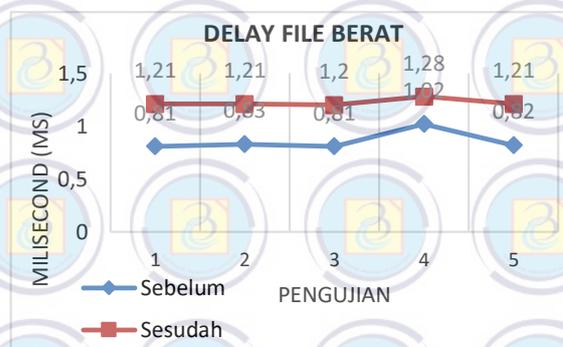
Perhitungan menggunakan rumus diambil 1 *sample* pada pengujian ke-1 sebelum konfigurasi, untuk dilakukan perhitungan dengan rumus pada waktu pengujian unduh berat sebelum konfigurasi yaitu :

$$\text{delay rata - rata} = \frac{481,101}{592770} = 0,00081 \text{ s} = 0,81 \text{ ms}$$

Lalu didapatkan rata-rata *delay* pada masing-masing pengujian yang telah diamati. Dari pengukuran berdasarkan analisis data dari *wireshark* rata-rata *delay* didapatkan hasil perhitungan pada Tabel 9.

Tabel 9 Hasil *delay* unduh file berat (520 MB)

Pengujian ke	Milisecond (ms)	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	0,81	1,21
2	0,83	1,21
3	0,81	1,2
4	1,02	1,28
5	0,82	1,21
Jumlah	4,29	6,11
Rata-rata	0,85	1,22



Gambar 15 Hasil pengujian *delay* unduh file berat (520 MB)

Hasil pengujian *delay* dari paket yang diambil dari lima kali pengujian pada ukuran file berat (520 MB) menggunakan *network analyzer wireshark*. Pada Tabel 9 terlihat perbedaan nilai rata-rata *delay* dari lima kali pengujian dengan ukuran file yang sama saat diunduh pada *google drive*. Pengujian di lakukan pada waktu 7.30 – 12.00 WIB dan dilakukan perbandingan sebelum dan sesudah konfigurasi pada *Router OS*.

- Pengujian unduh *throughput*

Throughput merupakan parameter yang menunjukkan jumlah bit rata-rata data yang dapat ditransfer dari satu *node* ke *node* yang lain perdetiknya dalam suatu jaringan yang terbentuk. Pengujian *throughput* ini diperoleh dari hasil dari rata-rata 5 kali percobaan pada ukuran file 22MB yang dikirim pada waktu 07.30 WIB - 12.00 WIB di

capture oleh network analyzer wireshark ditunjukkan pada Gambar 16.

Measurement	Captured	Displayed	Marked
Packets	592770	591688 (99.8%)	—
Time span, s	481.101	481.101	—
Average pps	1232.1	1229.9	—
Average packet size, B	992	994	—
Bytes	588202244	588005735 (100.0%)	0
Average bytes/s	1222 k	1222 k	—
Average bits/s	9780 k	9777 k	—

Gambar 16 Pencarian throughput unduh ringan pertama

Karena throughput menunjukkan kecepatan transfer data suatu jaringan, semakin besar nilai throughput akan semakin baik performa jaringan tersebut. Untuk mencari nilai throughput dari gambar 17 digunakan rumus persamaan sebagai acuan dalam mengetahui besarnya throughput pada masing-masing waktu yaitu :

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{lama pengamatan paket data}}$$

$$\text{Throughput} = \frac{25447735}{34,255} = 742891,1107867465 \text{ byte/s}$$

Didapatkan hasil 888449,838075007 Byte/s, selanjutnya dikonversikan ke Megabit/s dengan perhitungan sebagai berikut:

$$\frac{742891,1107867465 \times 8}{1000000} = 5,94 \text{ Mb/s}$$

Perhitungan throughput diatas sebagai sample pembuktian yang dihitung pada pengujian ke-1 sebelum konfigurasi pada throuhput unduh file ringan. Dari pengukuran berdasarkan analisis data wireshark nilai throughput unduh file ringan ditunjukkan pada Table 10.

Tabel 10 Hasil throughput unduh file ringan (22 MB)

Pengujian ke	Average Mbps	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	5,94	9,39
2	7,24	11
3	8,44	8,32
4	7,39	8,05
5	6,52	7,79
Jumlah	35,53	44,55
Rata-rata	7,1	8,91



Gambar 18 Hasil pengujian throughput unduh file ringan

Pada Tabel 10 dapat dilihat bahwa nilai throughput sebelum konfigurasi memiliki rata-rata nilai

atau kecepatan akses data paling besar sebesar 7,1 Mbit/s, sedangkan pada setelah konfigurasi memiliki rata-rata atau kecepatan akses data sebesar 7,44 Mbit/ dalam setiap ukuran file yang sama. Dari grafik pengujian unduh ringan throughput ini didapatkan bahwa pengujian sesudah konfigurasi terdapat throughput yang cukup tinggi yaitu dengan rata-rata sebesar 7,44 Mbit/s, sedangkan throughput terendah terjadi sebelum konfigurasi dengan rata-rata yaitu sebesar 4,62 Mbit/s.

Pengujian throughput ini diperoleh dari hasil dari rata-rata 5 kali percobaan pada ukuran file 520MB yang dikirim pada waktu 07.30 WIB - 12.00 WIB di capture oleh network analyzer wireshark ditunjukkan pada Gambar 19.

Measurement	Captured	Displayed	Marked
Packets	592770	591688 (99.8%)	—
Time span, s	481.101	481.101	—
Average pps	1232.1	1229.9	—
Average packet size, B	992	994	—
Bytes	588202244	588005735 (100.0%)	0
Average bytes/s	1222 k	1222 k	—
Average bits/s	9780 k	9777 k	—

Gambar 19 Pencarian throughput unduh berat pertama

Untuk mencari nilai throughput dari gambar 19 digunakan rumus persamaan sebagai acuan dalam mengetahui besarnya throughput pada masing-masing waktu yaitu :

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{lama pengamatan paket data}}$$

$$\text{Throughput} = \frac{588202244}{481,101} = 1222616,964005479 \text{ byte/s}$$

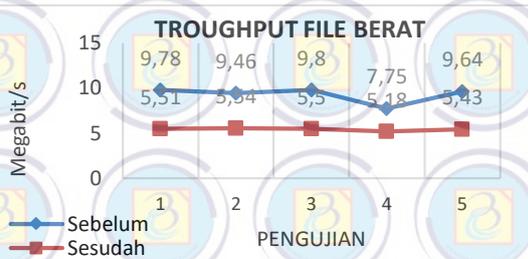
Didapatkan hasil 1222,616964005479 Byte/s, selanjutnya dikonversikan ke Megabit/s dengan perhitungan sebagai berikut:

$$\frac{1222616,964005479 \times 8}{1000000} = 9,78 \text{ Mb/s}$$

Perhitungan throughput diatas sebagai sample pembuktian yang dihitung pada pengujian ke-1 sebelum konfigurasi pada throuhput unduh file berat. Dari pengukuran berdasarkan analisis data wireshark nilai throughput unduh file berat ditunjukkan pada Tabel 11.

Tabel 11 Hasil throughput unduh file berat (520 mb)

Pengujian ke	Average Mbps	
	Sebelum Konfigurasi	Sesudah Konfigurasi
1	9,78	5,51
2	9,46	5,54
3	9,8	5,5
4	7,75	5,18
5	9,64	5,43
Jumlah	46,43	27,16
Rata-rata	9,28	5,43



Gambar 20 Hasil pengujian throughput unduh file berat

Pada Tabel 11 dapat dilihat bahwa nilai *throughput* sebelum konfigurasi memiliki rata-rata nilai atau kecepatan akses data paling besar sebesar 9,8 Mbit/s, sedangkan pada sesudah konfigurasi memiliki rata-rata atau kecepatan akses data sebesar 5,5 Mbit/ dalam setiap ukuran *file* yang sama. Dari grafik pengujian unduh ringan *throughput* ini didapatkan bahwa pengujian sebelum konfigurasi terdapat *throughput* yang cukup tinggi yaitu dengan rata-rata sebesar 9,8 Mbit/s, sedangkan *throughput* terendah terjadi sesudah konfigurasi dengan rata-rata yaitu sebesar 5,18 Mbit/s. Karena *throughput* menunjukkan kecepatan transfer data suatu jaringan, semakin besar nilai *throughput* akan semakin baik performa jaringan tersebut.

IV. KESIMPULAN DAN SARAN

1. Kesimpulan

Dari hasil pengujian dan analisis yang sudah dilakukan, dapat diambil kesimpulan sebagai berikut:

- *Management bandwidth* dengan *queue tree* dilakukan dengan membuat empat jalur trafik yang berbeda unggah ringan, unggah berat, unduh ringan, dan unduh berat.
- Perubahan terjadi dengan ditandai trafik internet yang telah melebihi ukuran lebih 22 MB dipindahkan pada trafik berat.
- Hasil skenario yang didapat berupa perubahan *Traffic Internet* yang lebih 22 mb dipindahkan *Traffic* berat.
- Hasil pengujian sesudah dilakukan konfigurasi *Management bandwidth* untuk ukuran berat (520 MB) didapatkan nilai *throughput* yang lebih kecil dari sebelum konfigurasi dilakukan.
- Hasil pengujian sesudah dilakukan konfigurasi *Management bandwidth* untuk ukuran file ringan (22 MB) didapatkan nilai *delay* yang lebih kecil dari sebelum konfigurasi dilakukan.
- Hasil pengujian sesudah dilakukan konfigurasi *Management bandwidth* untuk ukuran file berat (520 MB) didapatkan nilai *delay* yang lebih besar dari sebelum konfigurasi dilakukan.

2. Saran

Berdasarkan hasil pengujian *management bandwidth* maka didapatkan saran untuk Pengujian selanjutnya diharapkan dilakukan pengujian *management bandwidth* di jaringan dengan bandwidth yang kecil dengan user yang banyak agar dapat hasil yang maksimal.

V. REFRENSI

- [1] Pamungkas, C. A. (2016) 'Manajemen *Bandwidth* Menggunakan Mikrotik *Routerboard* Di Politeknik Indonusa Surakarta', Jurnal INFORMA Politeknik Indonusa Surakarta, 1(3), pp. 17–22. doi: 2442-7942.
- [2] Zuli, F. (2015) 'Penerapan Metode *Simple queue* untuk Manajemen Bandwith dengan *Router* Mikrotik', Satya Informatika, 1, pp. 23–33. doi: 10.1016/j.micron.2017.02.002.
- [3] Ziomek, C. D., Kinross-Wright, J. M. and Plato, J. G. (2015) 'Low-level RF control for the AFEL', Nuclear Inst. and Methods in Physics Research, A, 358(1–3), pp. 11–27. doi: 10.1016/0168-9002(94)01588-0.
- [4] Galeh Fatma Eko Ardiansa, Primananda, R. and Hanafi, M. H. (2017) 'Manajemen *Bandwidth* dan Manajemen Pengguna pada Jaringan *Wireless Mesh Network* dengan Mikrotik', Galeh Fatma Eko Ardiansa, Rakhmadhany Primananda, Mochammad Hannats Hanafi, 1(11), pp. 1226–1235.
- [5] Zamuswara, M. W., Jakarta, P. N. and Pendahuluan, I. (2017) 'Manajemen *Bandwidth* Menggunakan Metode Hierarchical Token Bucket (HTB) Pada Jaringan LAN PT . Waskita Beton Precast Plant Karawang', pp. 1–7.