

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bagian ini akan memaparkan langkah-langkah uji coba dan implementasi, dimulai dari menentukan parameter skenario uji coba, membuat konfigurasi manajemen *bandwidth* Queue Tree dan menganalisis QoS jaringan menggunakan *software* IPerf.

4.1. Langkah-Langkah Uji Coba

Bagian langkah langkah uji coba ini melakukan beberapa tahap untuk mengambil data manajemen *bandwidth* Queue Tree dengan metode PCQ, SFQ dan RED yang paling baik di implementasikan. Untuk mengambil data tersebut dipersiapkan dengan cara yaitu menggunakan *software* GNS3 dan *software* IPerf. Langkah lebih lanjut sebagai berikut :

4.2.1. Menentukan Parameter Uji coba

Berikut adalah parameter-parameter yang harus ditentukan untuk membuat skenario simulasi sesuai tabel 4.1 yaitu :

1. Jumlah Komputer dalam simulasi
2. Waktu Simulasi
3. Interval waktu simulasi
4. Protokol TCP dan UDP
5. Maksimum transfer data

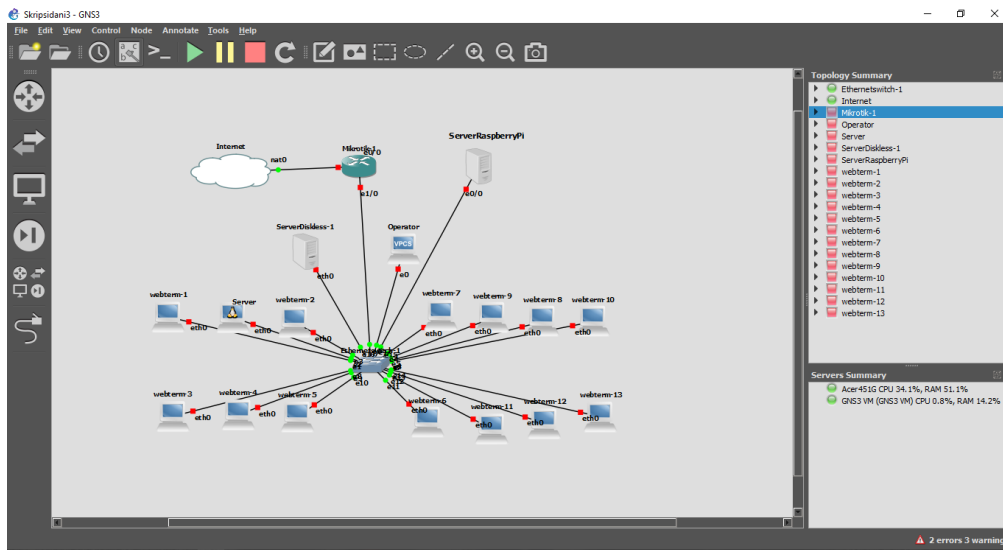
Tabel 4.1. Parameter Simulasi Skenario Uji Coba

Parameter	Nilai Parameter
MAC Layer	IEEE 802.11
Waktu Simulasi	10 s
Transmission Interval	1s
<i>Bandwidth</i> Protokol UDP	1 Mb/second
<i>Bandwidth</i> Protokol TCP	Unlimited
Jumlah Komputer	14 Komputer

4.2.2. Pengujian Menggunakan *Software* GNS3

Berdasarkan desain topologi yang diusulkan, digambarkan jaringan Warnet SweetNet yang menerapkan manajemen *bandwidth* Queue Tree dengan

beberapa metode PCQ, SFQ dan RED menggunakan router Mikrotik dan monitoring *software* Cacti (Gambar 4.1).



Gambar 4.1. Software GNS3

Skenario dalam simulasi dilakukan untuk menguji tingkat efektifitas serta mengukur kualitas jaringan setelah dilakukan manajemen *bandwidth*. Analisis difokuskan pada penggunaan beberapa metode seperti PCQ, SFQ dan RED untuk mendapatkan hasil kualitas QoS yang maksimal.

4.2.3. Analisis Menggunakan Software IPerf

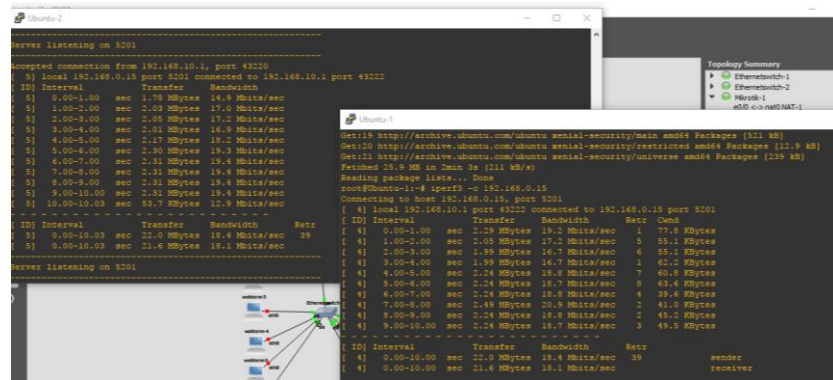
Software IPerf pada bagian tahap uji coba merupakan alat untuk mengukur kinerja traffic jaringan seperti mengukur *throughput*, *delay*, *jitter*, dan *packet loss*. Untuk menguji metode PCQ, SFQ dan RED *software* IPerf mempunyai 2 tahapan yaitu :

4.2.3.1. IPerf dengan Protokol TCP

Pada gambar 4.2 IPerf menggunakan protokol TCP untuk dapat menguji parameter *throughput*, *delay*, dan *packet loss*. Berikut ini perintah yang dapat dilakukan langsung pada *command prompt* (CMD) atau Terminal Linux.

```
# iperf3 -c ip_address
```

Keterangan IP Address dapat dirubah tergantung IP Server yang telah mengaktifkan IPerf.



Gambar 4.2. IPerf Transmission Control Protocol (TCP)

Parameter QoS yang dapat dihitung menggunakan *software* IPerf dengan protokol TCP yaitu throughput, delay, dan packet loss. Cara untuk menghitung 3 parameter tersebut menggunakan persamaan 2.1, persamaan 2.2, dan persamaan 2.4. Pada pengujian awal adalah menghitung throughput menggunakan persamaan 2.1 dan menggunakan data IPerf (gambar 4.2) seperti berikut :

$$\text{Throughput} = \frac{21600000 \text{ bytes/s}}{10 \text{ second}}$$

$$\text{Throughput} = 17280 \text{ kbps}$$

$$\text{Throughput} = \frac{17280 \text{ kbps} * 100 \%}{18100 \text{ kbps}}$$

$$\text{Throughput} = 95,46\%$$

Pada pengujian selanjutnya adalah menghitung delay menggunakan persamaan 2.2, namun pada *software* IPerf sudah terlihat delay tanpa harus menghitung kembali. Terlihat pada interval sender dan receiver 0.00 – 10.03, yang berarti terjadi delay sebesar 3 second atau di konversi menjadi 30 millisecond.

Packet loss dapat dihitung menggunakan persamaan 2.4 dan menggunakan data dari *software* IPerf (gambar 4.2). Untuk menghitung packet loss sebagai berikut :

$$\text{Packet Los} = \frac{22 - 21,6}{21,6} \times 100$$

$$\text{Packet Loss} = 1,85 \%$$

4.2.3.2. IPerf dengan Protokol UDP

```
Client
Sending select for 192.168.10.252...
Lease of 192.168.10.252 obtained, lease time 359200
root@Client:~#
root@Client:~# iperf3 -c 192.168.0.5 -u
Connecting to host 192.168.0.5, port 5201
[ 4] local 192.168.10.252 port 48472 connected to 192.168.0.5 port 5201
[ ID] Interval      Transfer      Bandwidth      Total Datagrams
[ 4] 0.00-1.00 sec  120 KBytes    983 Kbits/sec  15
[ 4] 1.00-2.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 2.00-3.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 3.00-4.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 4.00-5.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 5.00-6.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 6.00-7.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 7.00-8.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 8.00-9.00 sec  128 KBytes    1.05 Mbits/sec  16
[ 4] 9.00-10.00 sec 128 KBytes    1.05 Mbits/sec  16
-----
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datagrams
[ 4] 0.00-10.00 sec 1.24 MBytes    1.04 Mbits/sec  1.300 ms  0/159 (0%)
[ 4] Sent 159 datagrams

iperf Done.
root@Client:~#
```

Gambar 4.3. IPerf User Datagram Protocol (UDP)

Untuk menguji parameter *jitter* diperlukan protokol UDP untuk mengukurnya menggunakan IPerf. Pada gambar 4.3 merupakan IPerf menggunakan protokol UDP. Perintah yang dapat dilakukan langsung pada *command prompt* (CMD) atau terminal Linux hanya menambahkan parameter “-u”.

iperf3 -c ip_address -u

Keterangan IP Address dapat dirubah tergantung IP Server yang telah mengaktifkan IPerf. Parameter QoS yang dapat dihitung menggunakan *software* IPerf dengan protokol UDP yaitu *jitter*. Cara untuk menghitung *jitter* menggunakan parameter 2.3, namun pada *software* IPerf parameter *jitter* sudah di tunjukan seperti gambar 4.2 *jitter* bernilai 1,3 millisecond.

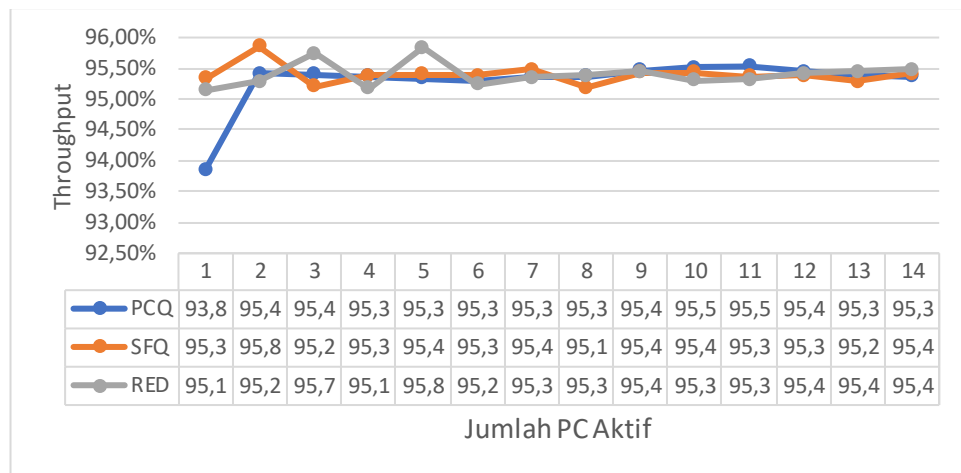
4.2. Hasil Uji Coba

Sub bab ini memaparkan data yang diperoleh dari hasil simulasi GNS3 menggunakan metode PCQ, SFQ, dan RED pada manajemen *bandwidth* Queue Tree. Setiap tabel menyajikan data hasil simulasi berdasarkan kinerja metode *queue* dengan perbedaan jumlah komputer klien.

4.3.1. Hasil Uji Throughput

Pada simulasi uji coba jaringan yang pertama adalah menguji jaringan menggunakan parameter *throughput* dengan jumlah komputer klien sebanyak 14 komputer serta dilakukan secara bertahap. Pengujian dilakukan menggunakan *software* IPerf dengan mengirim data sebesar 1 Mbps yang

merupakan nilai *default* besar transfer data dan menggunakan protocol *Transmission Control Protocol* (TCP). Didapatkan data seperti gambar 4.4

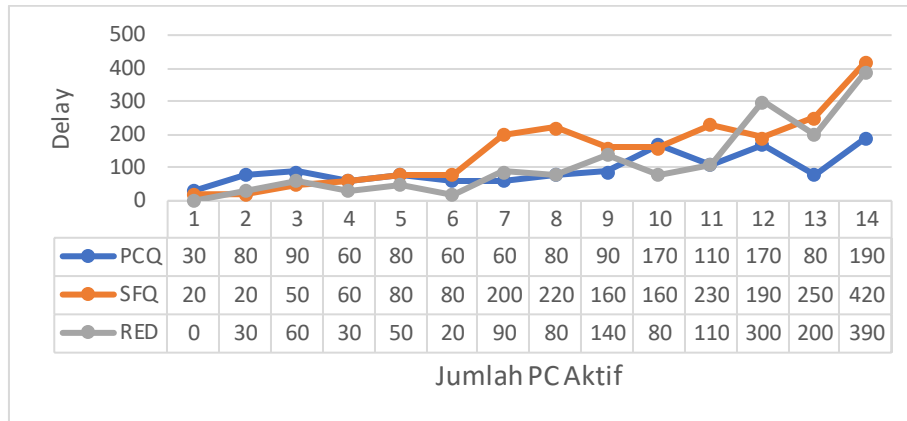


Gambar 4.4. Hasil Uji Throughput

Hasil uji coba menggunakan *software* Iperf dengan 14 komputer klien tampak pada gambar 4.4 Nilai Throughput yang paling stabil jika menggunakan metode *Peer Connection Queue* (PCQ) dengan nilai paling rendah hanya terjadi pada saat 1 komputer klien dengan nilai 93,8%, selain itu nilai Throughput dapat stabil sampai 14 komputer klien.

4.3.2. Hasil Uji Delay

Pada simulasi uji coba jaringan yang selanjutnya adalah menguji jaringan menggunakan parameter *delay* dengan jumlah komputer klien sebanyak 14 komputer serta dilakukan secara bertahap. Pengujian dilakukan menggunakan *software* IPerf dengan mengirim kecepatan data sebesar 1 Mbps yang merupakan nilai *default* besar transfer data dan menggunakan protokol *Transmission Control Protocol* (TCP). Didapatkan data seperti gambar 4.5

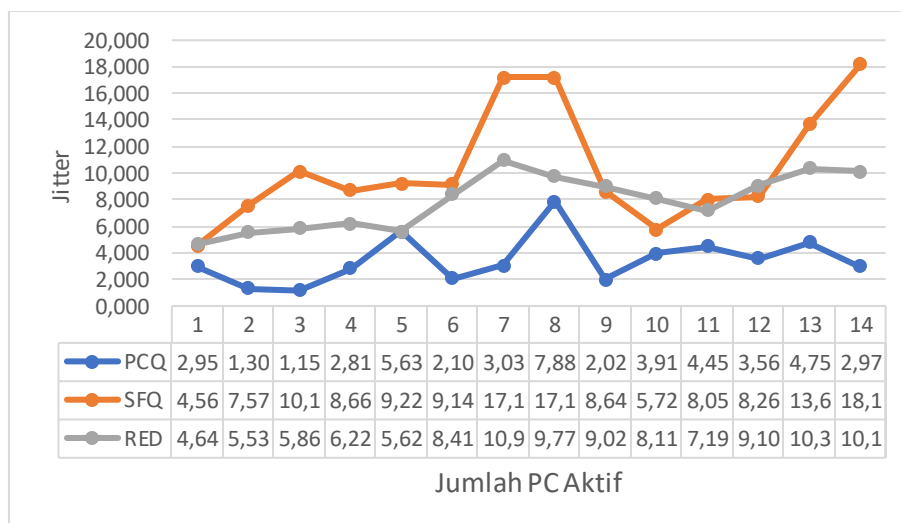


Gambar 4.5. Hasil Uji Delay

Hasil uji coba menggunakan *software* Iperf dengan 14 komputer klien tampak pada gambar 4.5 Nilai Delay yang paling kecil jika menggunakan metode *Peer Connection Queue* (PCQ) dengan nilai nilai 190 milisecond pada saat 14 komputer klien aktif bersama-sama .

4.3.3. Hasil Uji Jitter

Simulasi uji coba jaringan berikutnya adalah menguji jaringan menggunakan parameter *jitter* dengan jumlah komputer klien sebanyak 14 komputer serta dilakukan secara bertahap. Pengujian dilakukan menggunakan *software* IPerf dengan *bandwidth* sebesar 1 Mbps dan menggunakan protokol *User Datagram Protocol* (UDP). Didapatkan data seperti gambar 4.6

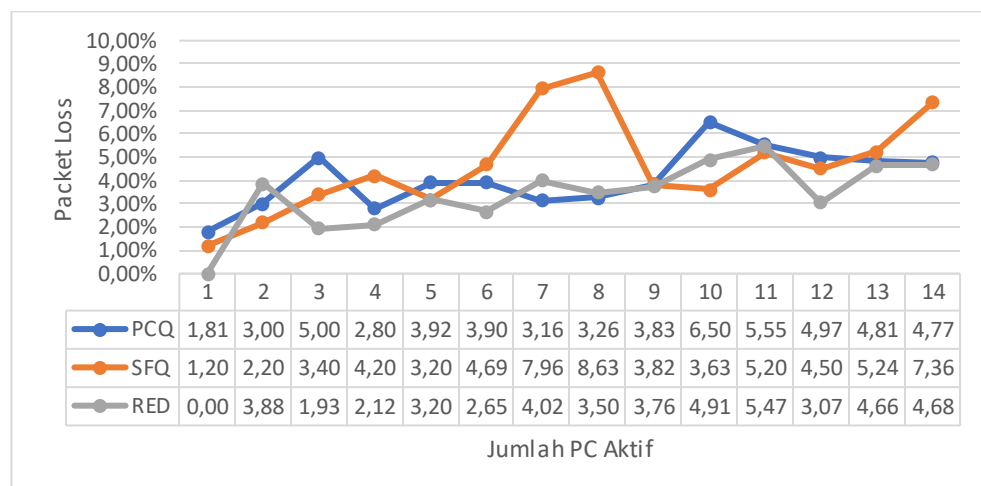


Gambar 4.6. Hasil Uji Jitter

Terlihat pada gambar 4.6 menggunakan *software* IPerf hasil nilai Jitter yang paling kecil menggunakan metode *Peer Connection Queue* (PCQ). Dari 1 komputer dengan nilai jitter 2,95 milisecond sampai dengan 14 komputer dengan nilai jitter 2,97 milisecond, jauh dibandingkan dengan menggunakan metode *Stochastic Fairness Queue* (SFQ) dan *Random Early Drop* (RED).

4.3.4. Hasil Uji Packet Loss

Pada simulasi uji coba jaringan yang berikutnya adalah menguji jaringan menggunakan parameter *packet loss* dengan jumlah komputer klien sebanyak 14 komputer serta dilakukan secara bertahap. Pengujian dilakukan menggunakan *software* IPerf dengan mengirim kecepatan transfer data sebesar 1 Mbps yang merupakan nilai *default* dan menggunakan protokol *Transmission Control Protocol* (TCP). Didapatkan data seperti gambar 4.7



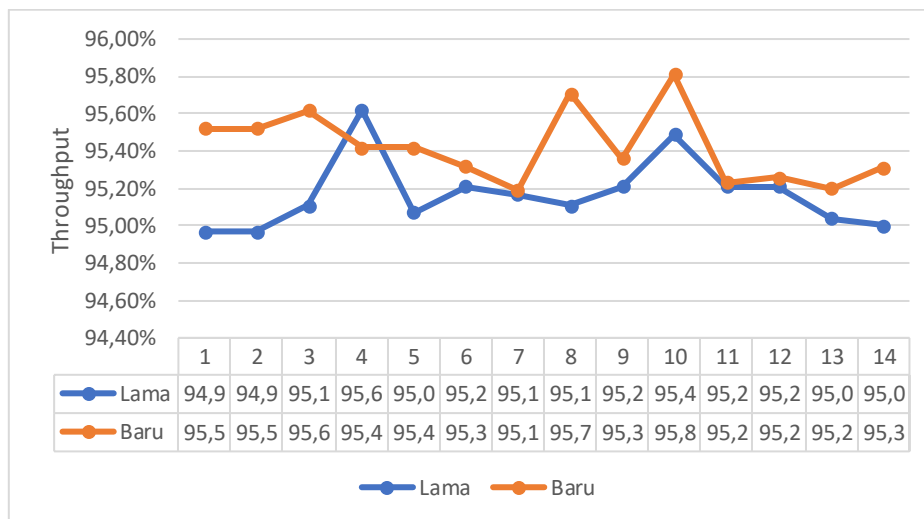
Gambar 4.7. Hasil Uji Packet Loss

Terlihat pada gambar 4.7 hasil uji menggunakan *software* IPerf dengan parameter *packet loss* bahwa metode *Random Early Drop* (RED) dan *Peer Connection Queue* (PCQ) hampir memiliki nilai yang sama. Namun metode PCQ memiliki nilai *packet loss* tertinggi 6,50%, yang terjadi pada saat 10 komputer klien aktif bersama-sama. Untuk metode RED nilai *packet loss* tertinggi yaitu 5,20%, yang terjadi pada saat 11 komputer aktif bersama-sama.

4.3. Pembahasan

Berdasarkan hasil simulasi pengujian pada topologi baru, dapat disimpulkan bahwa dari ketiga manajemen *bandwidth* Queue Tree (PCQ, SFQ, dan RED), manajemen *bandwidth* menggunakan PCQ memiliki nilai terbaik dari sisi QoS (throughput, delay, jitter, dan packet loss) sehingga metode PCQ dipilih untuk diimplementasikan pada Warnet SweetNet. Sistem jaringan yang baru dengan mengimplementasikan metode PCQ pada manajemen *bandwidth* Queue Tree kemudian dibandingkan dengan sistem jaringan yang lama. Hal ini dilakukan untuk melihat apakah kinerja sistem jaringan yang baru lebih baik dari sistem jaringan sebelumnya. Uji kualitas jaringan baru dan jaringan lama tetap menggunakan parameter QoS (throughput, delay, jitter, dan packet loss) sebagai ukuran kualitas dan *software* IPerf sebagai data yang nanti akan diolah menjadi informasi. Berikut ini perbandingan hasil uji jaringan lama dan jaringan baru :

4.3.1. Uji Throughput Jaringan Lama dan Baru

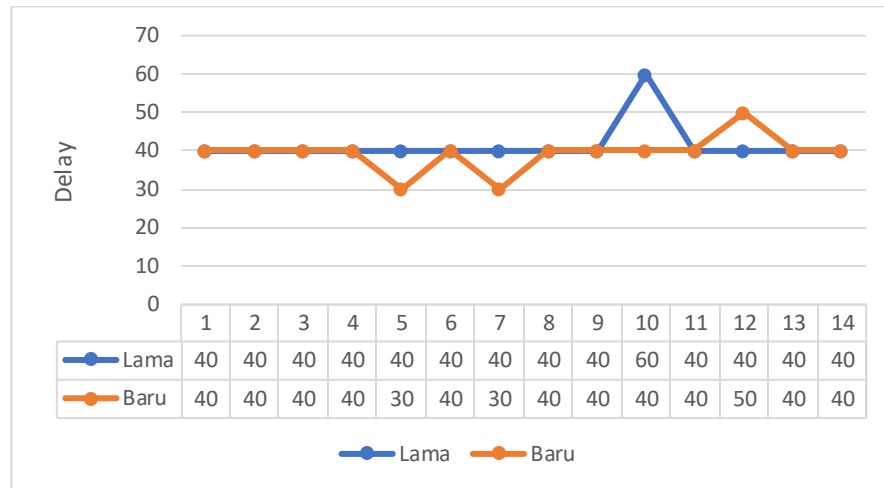


Gambar 4.8. Hasil Throughput

Berdasarkan grafik pada gambar 4.8 Hasil Throughput pada kondisi saat Warnet SweetNet sebelum dan sesudah mengimplementasikan manajemen *bandwidth* Queue Tree dengan metode *Peer Connection Queue* (PCQ). Terlihat nilai throughput awal pada saat 1 komputer aktif, jaringan lama memiliki nilai 94,9% sedangkan jaringan baru memiliki nilai 95,5%. Untuk

nilai throughput tertinggi pada jaringan baru yaitu 95,8% dengan 10 komputer aktif.

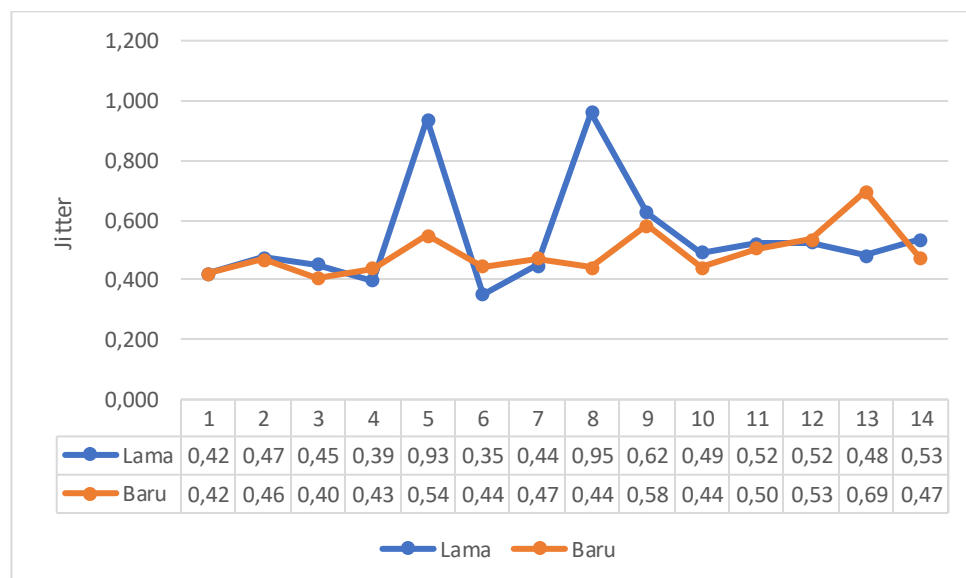
4.3.1. Uji Delay Jaringan Lama dan Baru



Gambar 4.9. Hasil Delay Warnet SweetNet

Grafik pada gambar 4.9 menunjukkan hasil delay jaringan lama dan jaringan baru hampir sama. Namun pada saat 5 komputer dan 7 komputer aktif bersama pada jaringan baru dapat memperkecil delay dengan nilai 30 milisecond. Untuk jaringan lama nilai delay tertinggi pada saat 10 komputer aktif dengan nilai delay sebesar 60 milisecond.

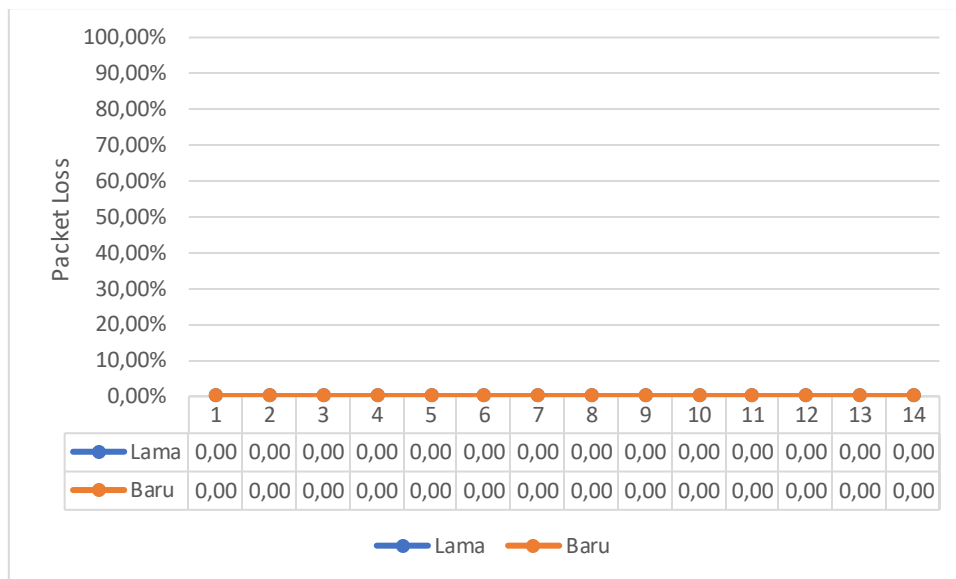
4.3.2. Uji Jitter Jaringan Lama dan Baru



Gambar 4.10. Hasil Jitter Warnet SweetNet

Grafik Gambar 4.10 menunjukkan nilai jitter pada jaringan baru berada pada nilai stabil tanpa adanya nilai tinggi. Nilai terendah jitter pada jaringan lama hanya terjadi pada saat 6 komputer aktif yaitu dengan nilai jitter 0,35 milisecond. Namun pada jaringan lama terjadi kenaikan nilai jitter pada saat 5 dan 7 komputer aktif dengan nilai 0,93 milisecond dan 0,95 milisecond.

4.3.3. Uji Packet Loss Jaringan Lama dan Baru



Gambar 4.11. Hasil Packet Loss Warnet SweetNet

Grafik gambar 4.11 menunjukkan tidak adanya nilai packet loss yang terjadi semua rata-rata pada nilai 0% yang berarti tidak adanya data yang hilang selama komputer dari 1 sampai 14 aktif bersama melakukan kegiatan internet. Hal ini bisa terjadi dikarenakan *bandwidth* yang tersedia besar sangat cukup untuk dapat melayani.

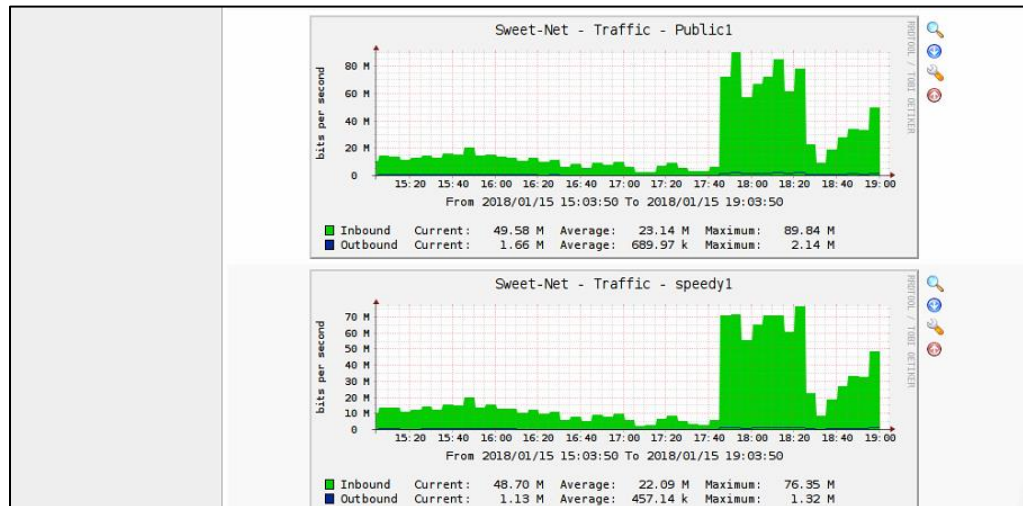
4.4. Monitoring

Pada tahap monitoring akan melakukan beberapa tahap sebagai berikut :

4.4.1. Memantau Traffic Jaringan dengan Software Cacti

Sebagai admin pada Warnet atau pemilik Warnet ingin mengetahui apakah *traffic* yang berjalan di jaringan sudah sesuai dengan semestinya. Cara nya yaitu dengan menggunakan monitoring jaringan menggunakan *software* Cacti yang *diinstall* di perangkat Raspberry Pi. Berikut langkahnya :

Buka Browser Firefox, Chrome, atau Opera masukan Ip address perangkat Raspberry Pi : <http://192.168.0.15/cacti>. Lalu masuk dengan username dan password yang sudah di tentukan. Masuk ke tab Graph dan akan terlihat monitoring *traffic* jaringan (gambar 4.12).



Gambar 4.12. Monitoring *Software Cacti*

4.4.2. Ping (*Packet Internet Groper*)

Pada Router Mikrotik dapat dilakukan ping secara langsung menggunakan Winbox melalui menu **Tools >> Ping**, kemudian masukkan alamat IP Address. Pada gambar 4.13 konektivitas jaringan yang di monitor yaitu connect atau tidaknya komputer asal ke komputer atau IP Address tujuan.

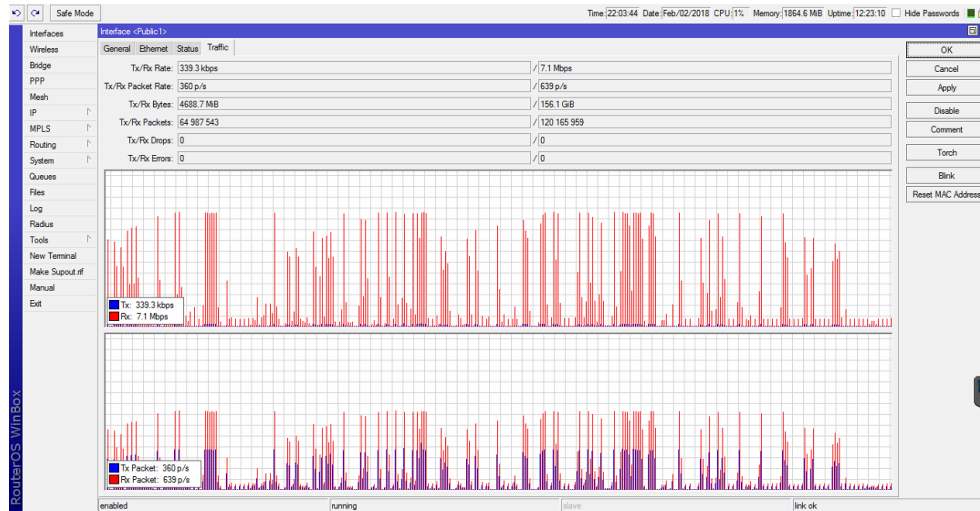
The screenshot shows the Winbox Ping tool interface. The 'General' tab is active, showing 'Ping To: 8.8.8.8', 'Interface: speedy1', and 'Packet Count: 35'. The 'Advanced' tab is also visible. Below the settings, a table displays the results of 35 ping attempts.

Seq #	Host	Time	Reply Size	TTL	Status
12	8.8.8.8	46ms	50	57	
13	8.8.8.8	34ms	50	57	
14	8.8.8.8	50ms	50	57	
15	8.8.8.8	55ms	50	57	
16	8.8.8.8	47ms	50	57	
17	8.8.8.8	41ms	50	57	
18	8.8.8.8	52ms	50	57	
19	8.8.8.8	46ms	50	57	
20	8.8.8.8	46ms	50	57	
21	8.8.8.8	50ms	50	57	
22	8.8.8.8	45ms	50	57	
23	8.8.8.8	83ms	50	57	
24	8.8.8.8	37ms	50	57	
25	8.8.8.8	49ms	50	57	
26	8.8.8.8	65ms	50	57	
27	8.8.8.8	16ms	50	57	
28	8.8.8.8	81ms	50	57	
29	8.8.8.8	50ms	50	57	
30	8.8.8.8	82ms	50	57	
31	8.8.8.8	85ms	50	57	
32	8.8.8.8	44ms	50	57	
33	8.8.8.8	45ms	50	57	
34	8.8.8.8	39ms	50	57	
35	8.8.8.8	48ms	50	57	

Gambar 4.13. Tes Ping

4.4.3. Evaluasi *Bandwidth*

Gambar 4.14 menggambarkan perbandingan sistem lama, yang tidak memakai mikrotik dibandingkan dengan sistem baru yang menggunakan router Mikrotik dan Winbox.



Gambar 4.14. Graphing *Bandwidth*

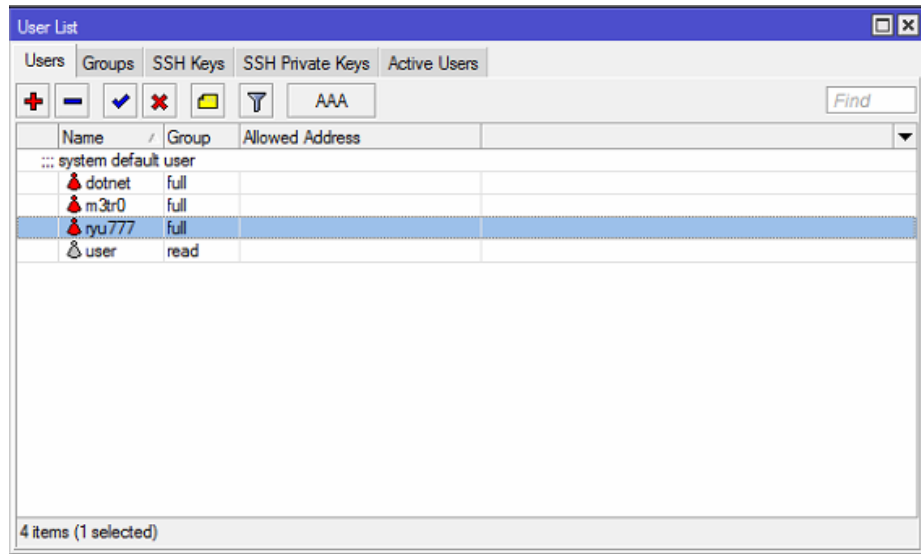
Jika sebelumnya disistem lama Warnet SweetNet tidak terdapat monitoring *bandwidth*, sekarang dapat melihat hasil pengukuran *bandwidth* Warnet SweetNet.

4.5. Manajemen

Pada tahap manajemen meliputi aktifitas perawatan dan pemeliharaan keseluruhan sistem yang sudah dibangun. Tahap manajemen dilakukan setelah semua sistem dapat berjalan dengan baik pada Warnet Sweet-Net. Berikut ini beberapa langkah manajemen untuk sistem yang telah dibangun agar dapat dengan optimal bekerja.

4.5.1. Manajemen User dan Password Mikrotik

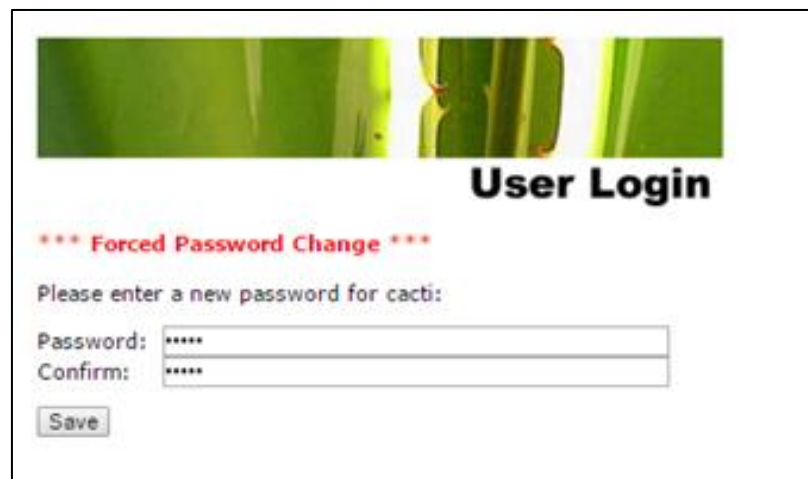
Pada tahap gambar 4.15 merupakan tahap memajemen user dan password router Mikrotik. Tujuan nya untuk mengatur siapa saja yang dapat mengakses router atau menggunakan router Mikrotik. User default admin dari router Mikrotik di hapus dan di buatkan kembali user baru dengan group full agar dapat melihat dan mengkonfigurasi router Mikrotik.



Gambar 4.15. Manajemen User dan Password Mikrotik

4.5.2. Manajemen Password Cacti

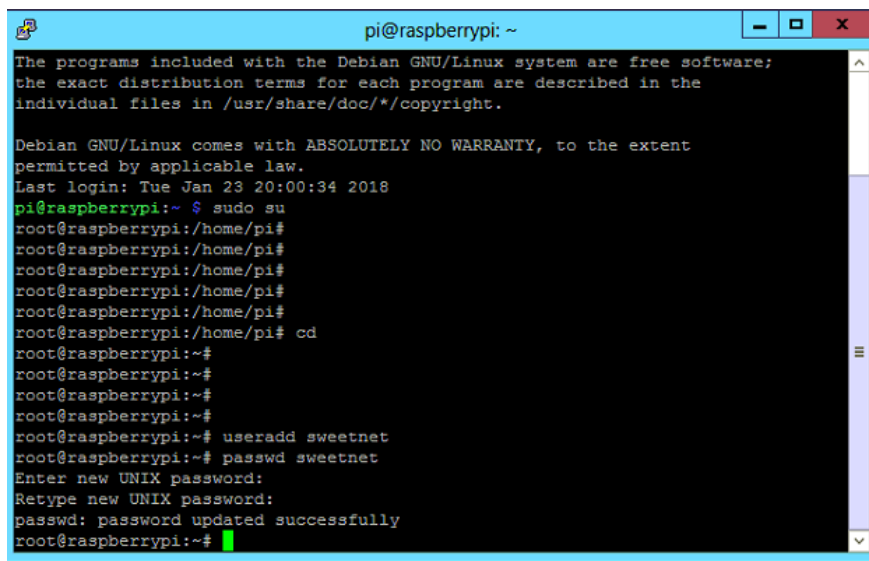
Manajemen password Cacti diperlukan untuk memanajemen user yang dapat masuk ke dalam sistem monitoring jaringan. Ada beberapa user yang di buat untuk dapat membaca dan merubah, adapula user yang hanya dapat membaca saja tanpa bisa merubah. Untuk *username* dan *password default* menggunakan **admin**. Selanjutnya *software* Cacti meminta agar merubah password *default* tersebut (gambar 4.16).



Gambar 4.16. Manajemen Password *Software* Cacti

4.5.3. Manajemen Password Raspberry Pi

Manajemen password Raspberry Pi termasuk hal terpenting. Perangkat Raspberry Pi terinstall *software* Cacti yang berguna untuk monitoring jaringan. Jika Password Raspberry Pi tidak di rubah akan terjadi pengguna yang tidak berhak untuk menggunakannya dan secara langsung dapat merubah konfigurasi *software* Cacti. Pada gambar 4.17 menjelaskan bagaimana mengganti *password* Raspberry Pi dengan user **Root**.



```

pi@raspberrypi: ~
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 23 20:00:34 2018
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi# cd
root@raspberrypi:~#
root@raspberrypi:~#
root@raspberrypi:~#
root@raspberrypi:~# useradd sweetnet
root@raspberrypi:~# passwd sweetnet
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@raspberrypi:~#

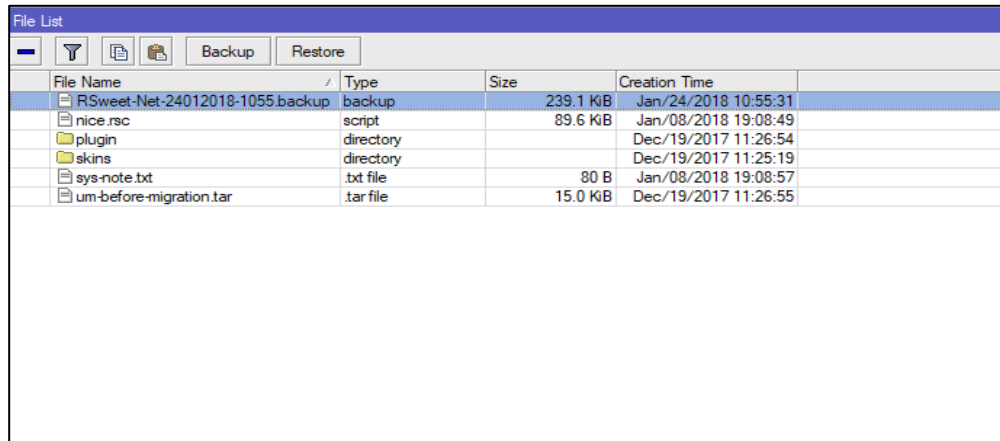
```

Gambar 4.17. Manajemen Password Raspberry Pi

4.5.4. Manajemen Backup Router Mikrotik

Konfigurasi backup di router Mikrotik hal yang sangat penting. Apabila suatu konfigurasi router Mikrotik atau router Mikrotik bermasalah dapat dengan mudah di restore kembali dari backup yang telah di lakukan.

Cara untuk membackup router Mikrotik adalah dengan menggunakan Winbox yaitu Pilih menu **File >>Backup >>Selanjutnya File backup klik copy diatas >>Buka Windows Explorer lalu paste** (gambar 4.18).



The screenshot shows a 'File List' window with a toolbar containing a filter icon, a folder icon, a document icon, and buttons for 'Backup' and 'Restore'. Below the toolbar is a table with the following columns: File Name, Type, Size, and Creation Time. The table contains the following entries:

File Name	Type	Size	Creation Time
RSweet-Net-24012018-1055.backup	backup	239.1 KB	Jan/24/2018 10:55:31
nice.rsc	script	89.6 KB	Jan/08/2018 19:08:49
plugin	directory		Dec/19/2017 11:26:54
skins	directory		Dec/19/2017 11:25:19
sys-note.txt	.txt file	80 B	Jan/08/2018 19:08:57
um-before-migration.tar	.tar file	15.0 KB	Dec/19/2017 11:26:55

Gambar 4.18. Manajemen Backup Router Mikrotik