

BAB 4

HASIL TESTBED DAN EVALUASI KINERJA PROGRAM FTS

Pada bab 3 telah dijelaskan model skenario *testbed* yang digunakan untuk menganalisa kinerja program FTS. Model *testbed* tersebut meliputi *routing* statik yang diterapkan pada *router* dan *server* sekaligus tujuan setiap skenario. Berdasarkan model *testbed* tersebut, maka pada bab 4 ini, akan diperlihatkan hasil *testbed* serta perbandingan antara hasil yang diperoleh menggunakan program yang dibuat dengan yang tidak menggunakan. Dalam setiap skenario, *bandwidth server* akan di seting maksimum 768kbps dimana *bandwidth* tersebut akan dishare untuk melayani *request* akses pengguna. Sedangkan nilai *bandwidth* minimum 256kbps, (nilai minimum ini bukanlah batas *bandwidth* minimum untuk pengguna, akan tetapi nilai batas bawah variabel *fuzzy* untuk *bandwidth* minimum) [BAB 3, persamaan 3.8 dan 3.9]. Adapun parameter yang diamati adalah *output bandwidth* yang diperoleh oleh pengguna berdasarkan *input rtt*, jumlah *hop* dan *loss ratio* masing-masing pengguna.

Tujuan lebih detail kami merancang ada dua skenario adalah :

- Skenario topologi acak (skenario 1), merepresentasikan kondisi sebenarnya *internet* dalam *Local Area Network* (LAN).
- Skenario topologi terkonsentrasi (skenario 2), untuk membuktikan apakah sifat TCP/IP yang *best effort* benar-benar terjadi, dan apakah pengaruhnya terhadap *output bandwidth* yang akan diperoleh dari setiap pengguna (4 pengguna terkonsentrasi). Apakah memperoleh *weight* dan besar *bandwidth* yang sama antar pengguna yang terkonsentrasi?. Sebagai pembandingan, kami meletakkan 1 pengguna paling dekat dengan *Server* (dilihat dari sisi jumlah *hop*).

Sebagai perbandingan *rule fuzzy*, kami menggunakan 2 *rule* yang berbeda. Yang pertama adalah yang diulas pada bab 3 (yang sesuai dengan tujuan perancangan program), selanjutnya disebut *Rule A*. Dan yang kedua adalah yang mengalami pembalikan inferensi *weight/* bobot, selanjutnya disebut *Rule B*. Berikut kedua *rule fuzzy* tersebut :

Rule A, Logika Fuzzy :

Tabel 4.1 *Rule A, Logika Fuzzy*

Condition (IF)			Impact (Then)
rtt	hop	Loss	bw
KECIL	SEDIKIT	KECIL	NAIK
KECIL	SEDIKIT	BESAR	NAIK
KECIL	BANYAK	KECIL	NAIK
KECIL	BANYAK	BESAR	TURUN
BESAR	SEDIKIT	KECIL	NAIK
BESAR	BANYAK	KECIL	TURUN
BESAR	SEDIKIT	BESAR	TURUN
BESAR	BANYAK	BESAR	TURUN

Rule B, Logika Fuzzy :

Tabel 4.2 *Rule B, Logika Fuzzy*

Condition (IF)			Impact (Then)
rtt	hop	Loss	bw
KECIL	SEDIKIT	KECIL	TURUN
KECIL	SEDIKIT	BESAR	TURUN
KECIL	BANYAK	KECIL	TURUN
KECIL	BANYAK	BESAR	NAIK
BESAR	SEDIKIT	KECIL	TURUN
BESAR	BANYAK	KECIL	NAIK
BESAR	SEDIKIT	BESAR	NAIK
BESAR	BANYAK	BESAR	NAIK

Berikut ini tabel-tabel yang diperoleh dari hasil *testbed* yang dilakukan berdasarkan kedua *rule* diatas. Setiap skenario, dilakukan 2 kali percobaan yaitu yang menggunakan FTS dan yang tidak menggunakan FTS. Adapun yang tidak menggunakan FTS, tidak kami lampirkan dalam tabel, karena *output* yang tidak dapat diprediksi. Akan tetapi kami melampirkan grafik keluarannya melalui SNMP Traffic Grapher (STG) pada akhir bab ini.

Tabel-tabel hasil percobaan berdasarkan Rule A :

Tabel 4.3 adalah hasil *Rule A*, skenario 1 (topologi acak).

Tabel 4.3 Hasil *Rule A*, skenario 1 (topologi acak)

IP Address	rtt(ms)	hop	loss ratio(%)
9.9.9.2	2.35	6	0
3.3.3.5	1.02	3	0
4.4.4.6	0.60	2	0
3.3.3.6	0.92	3	0
9.9.9.1	1.42	6	0
IP Address	weight(%)	bandwidth(kbps)	
9.9.9.2	11	85	
3.3.3.5	22	170	
4.4.4.6	25	190	
3.3.3.6	22	172	
9.9.9.1	20	152	

Tabel 4.4 adalah hasil *Rule A*, skenario 2 (topologi terkonsentrasi).

Tabel 4.4 Hasil *Rule A*, skenario 2 (topologi terkonsentrasi)

IP Address	rtt(ms)	hop	loss ratio(%)
9.9.9.1	1.64	6	0
4.4.4.6	0.67	2	0
9.9.9.4	1.51	6	0
9.9.9.5	1.47	6	0
9.9.9.2	1.99	6	0
IP Address	weight(%)	bandwidth(kbps)	
9.9.9.1	18	139	
4.4.4.6	28	217	
9.9.9.4	20	153	
9.9.9.5	20	157	
9.9.9.2	13	103	

Rule B kami lampirkan dengan tujuan untuk membandingkan apakah *Rule A* sudah sesuai dengan tujuan perancangan program.

Didalam percobaan yang kami lakukan, ada kondisi tertentu yang mana jika semua pengguna mempunyai jumlah *hop* dan *loss* yang sama, maka *input* yang akan dimasukkan ke aturan logika *fuzzy* hanya *rtt* saja. Maka dari itu didalam *source code* program yang kami buat terdapat penyeleksian kondisi yang hanya melibatkan *rtt* sebagai *input* logika *fuzzy*.

Tabel-tabel hasil percobaan berdasarkan Rule B :

Tabel 4.5 adalah hasil *Rule B*, skenario 1 (topologi acak).

Tabel 4.5 Hasil *Rule B*, skenario 1 (topologi acak)

IP Address	rtt(ms)	hop	loss ratio(%)
4.4.4.6	0.61	2	0
9.9.9.2	1.47	6	0
9.9.9.1	1.78	6	0
3.3.3.5	0.92	3	0
3.3.3.6	0.87	3	0
IP Address	weight(%)	bandwidth(kbps)	
4.4.4.6	16	126	
9.9.9.2	23	176	
9.9.9.1	26	200	
3.3.3.5	17	133	
3.3.3.6	17	133	

Tabel 4.6 adalah hasil *Rule B*, skenario 2 (topologi terkonsentrasi).

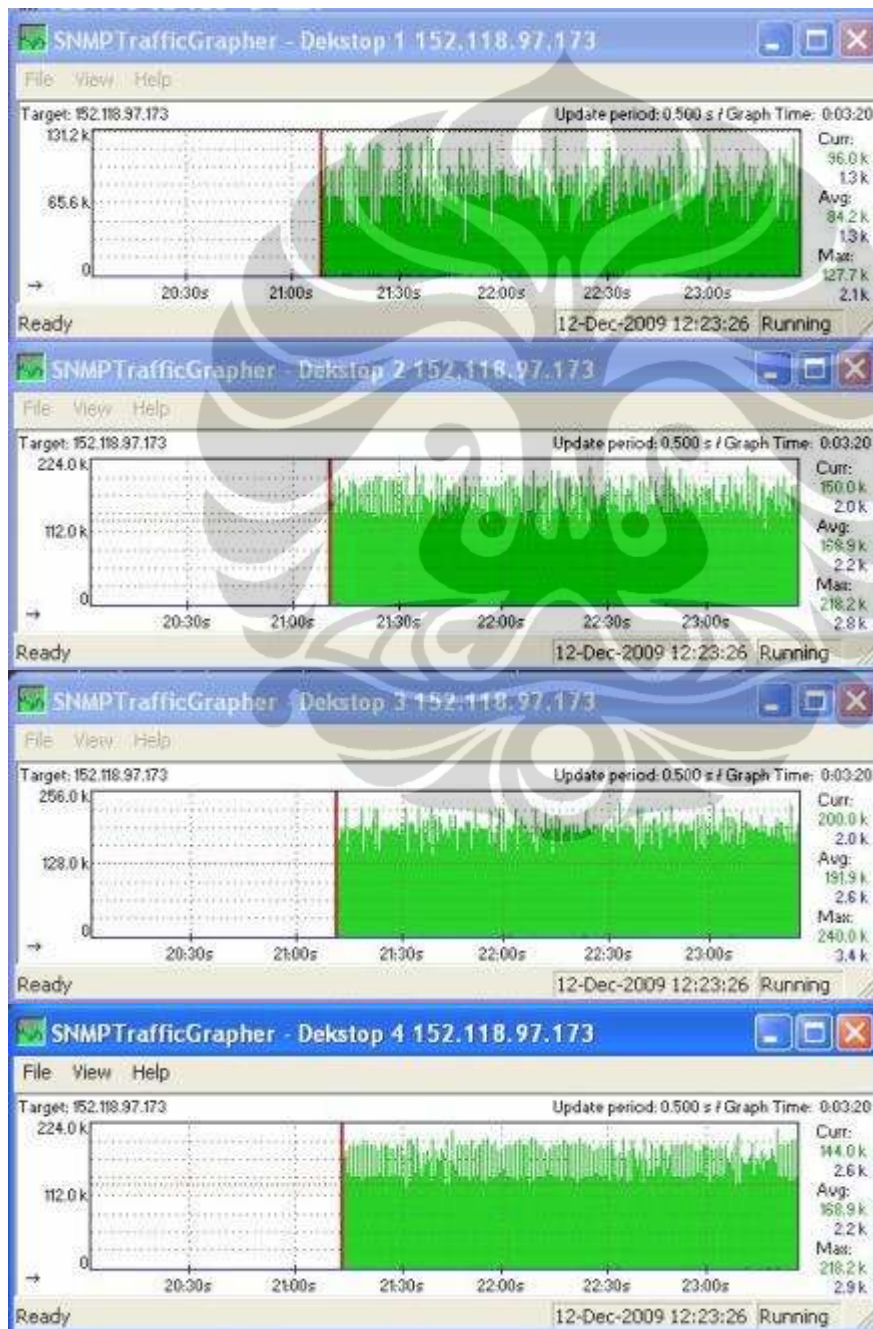
Tabel 4.6 Hasil *Rule B*, skenario 2 (topologi terkonsentrasi)

IP Address	rtt(ms)	hop	loss ratio(%)
9.9.9.5	1.33	6	0
4.4.4.6	0.65	2	0
9.9.9.4	1.65	6	0
9.9.9.1	1.65	6	0
9.9.9.2	1.44	6	0
IP Address	weight(%)	bandwidth(kbps)	
9.9.9.5	20	151	
4.4.4.6	15	113	
9.9.9.4	23	173	
9.9.9.1	23	173	
9.9.9.2	21	158	

Adapun penjelasan dan representasi hasil *output bandwidth* (dalam kbps) dari setiap skenario percobaan diatas akan ditampilkan dibawah ini dengan menggunakan *software* SNMP Traffic Grapher (STG).

4.1 Rule A Skenario 1 (Topologi Acak)

Pada Tabel 4.3, menunjukkan bahwa koneksi terbaik adalah *IP Address* 4.4.4.6 (*rtt* 0,60ms dan *hop* 2), sehingga memperoleh *weight* terbesar (25%) dan alokasi *bandwidth* terbesar (190kbps). Sedangkan koneksi terburuk adalah *IP Address* 9.9.9.2 (*rtt* 2,35ms dan *hop* 6), sehingga memperoleh *weight* terkecil (11%) dan alokasi *bandwidth* terkecil (85kbps).

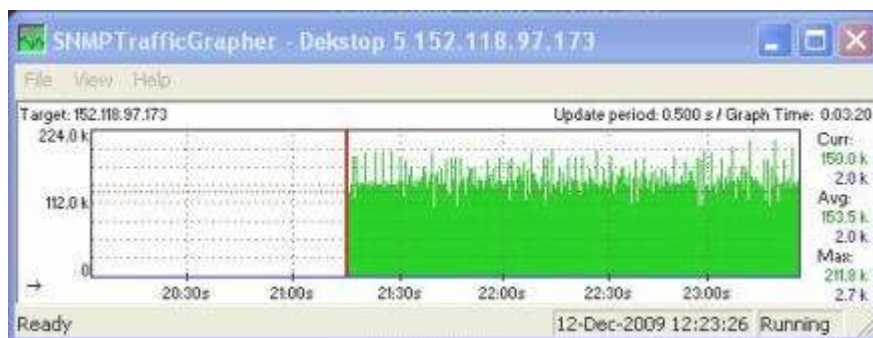


IP : 9.9.9.2
Downstream :
Curr = 96,0 kbps
Avg = 84,2 kbps
Max = 127,7 kbps

IP : 3.3.3.5
Downstream :
Curr = 150,0 kbps
Avg = 168,9 kbps
Max = 218,2 kbps

IP : 4.4.4.6
Downstream :
Curr = 200,0 kbps
Avg = 191,9 kbps
Max = 240,0 kbps

IP : 3.3.3.6
Downstream :
Curr = 144,0 kbps
Avg = 168,9 kbps
Max = 218,2 kbps



Gambar 4.1 *Output bandwidth, Rule A* skenario 1

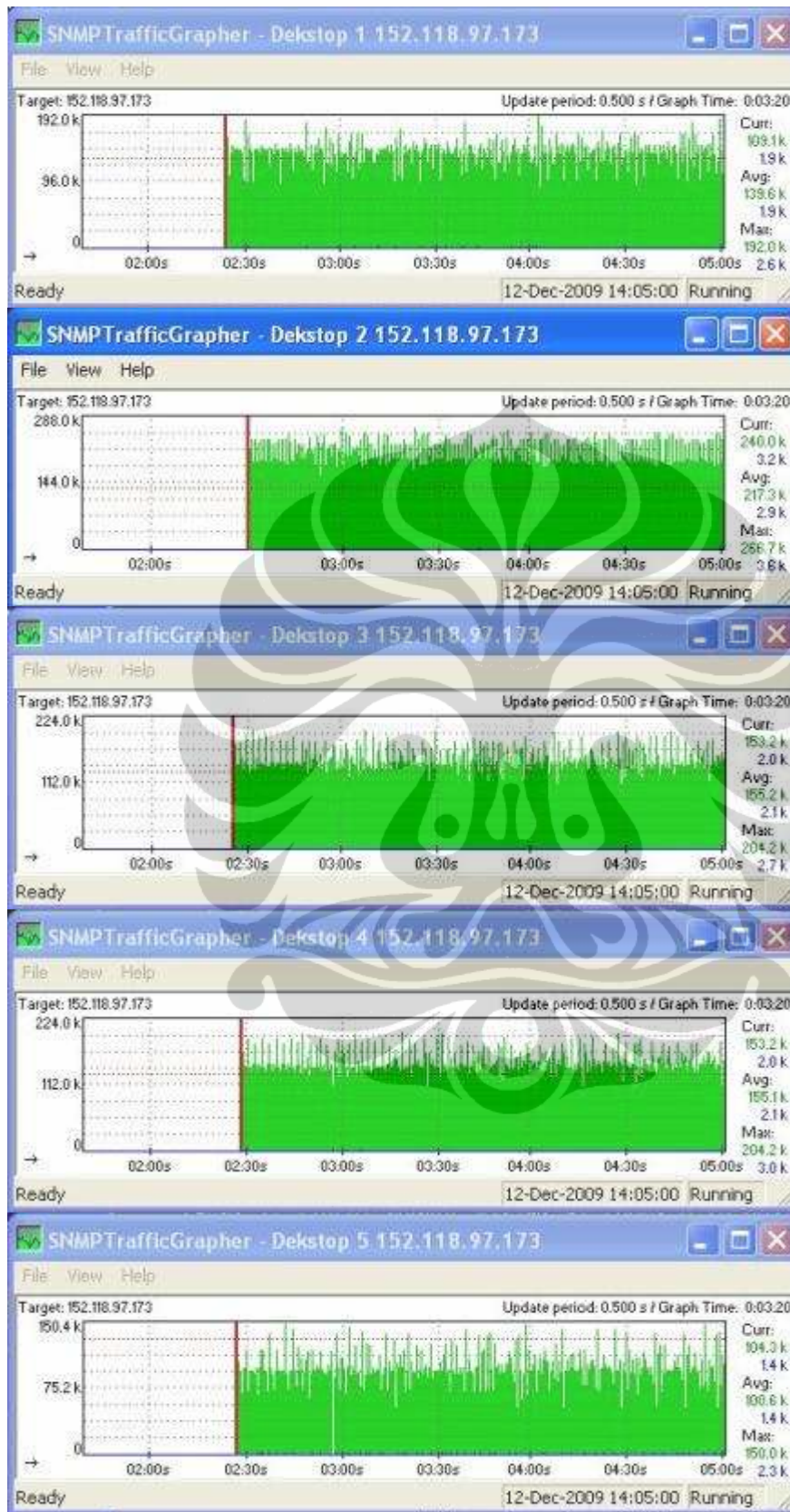
Gambar diatas menunjukkan hasil keluaran trafik *bandwidth* menggunakan *software* STG. Pada kotak sebelah kanan dari setiap gambar, dapat dilihat nilai “Avg”/ (rata-rata) *bandwidth* yang merepresentasikan *output* trafik *bandwidth* dari *Rule A* skenario 1. *IP Address* 4.4.4.6 memperoleh alokasi *bandwidth* terbesar dengan Avg 191,9 kbps, sedangkan *IP Address* 9.9.9.2 dengan Avg 84,2 kbps merupakan koneksi terburuk. Dari hasil *output bandwidth* semua pengguna dapat dilihat bahwa fluktuasi *bandwidth* lebih rata atau bisa diredam, karena menggunakan FTS yang melibatkan *traffic shaping*.

4.2 Rule A Skenario 2 (Topologi Terkonsentrasi)

Pada Tabel 4.4, menunjukkan bahwa koneksi terbaik adalah *ip address* 4.4.4.6 (*rtt* 0,67ms dan *hop* 2), sehingga memperoleh *weight* terbesar (28%) dan alokasi *bandwidth* terbesar (217kbps).

Adapun 4 *ip address* pengguna dengan *network* 9.9.9.0 mempunyai harga *rtt* yang berbeda, meskipun jumlah *hop*nya sama dari arah *server*. Hal tersebut membuktikan bahwa TCP/IP mempunyai sifat *best effort*.

Keempat pengguna (*network* 9.9.9.0) tersebut, mempunyai *weight* dan *bandwidth* yang lebih kecil dari *ip address* 4.4.4.6. Hal tersebut dikarenakan *network* 9.9.9.0 mempunyai *rtt* yang lebih besar dan jumlah *hop* yang lebih jauh dari arah *Server*.



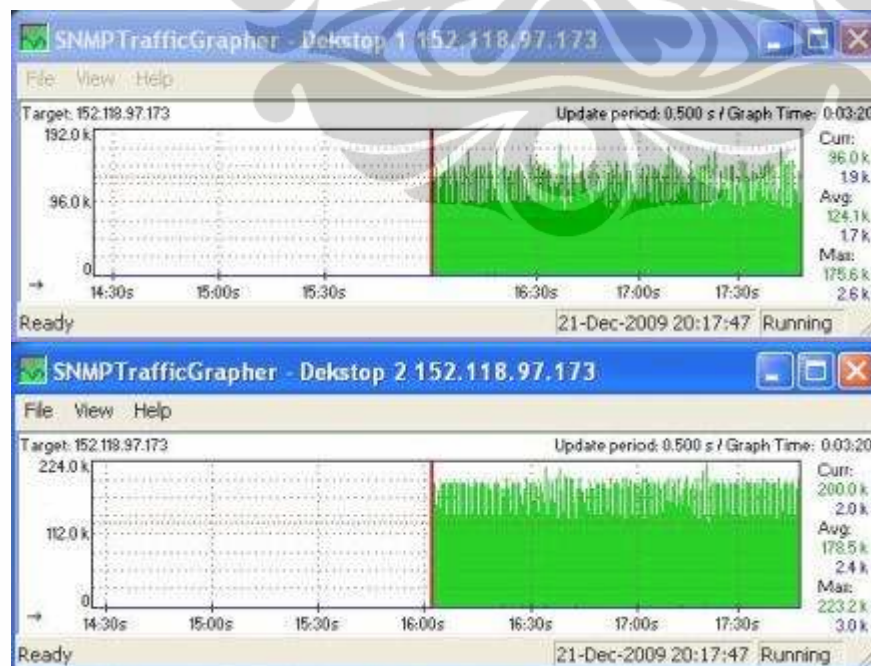
Gambar 4.2 Output bandwidth, Rule A skenario 2

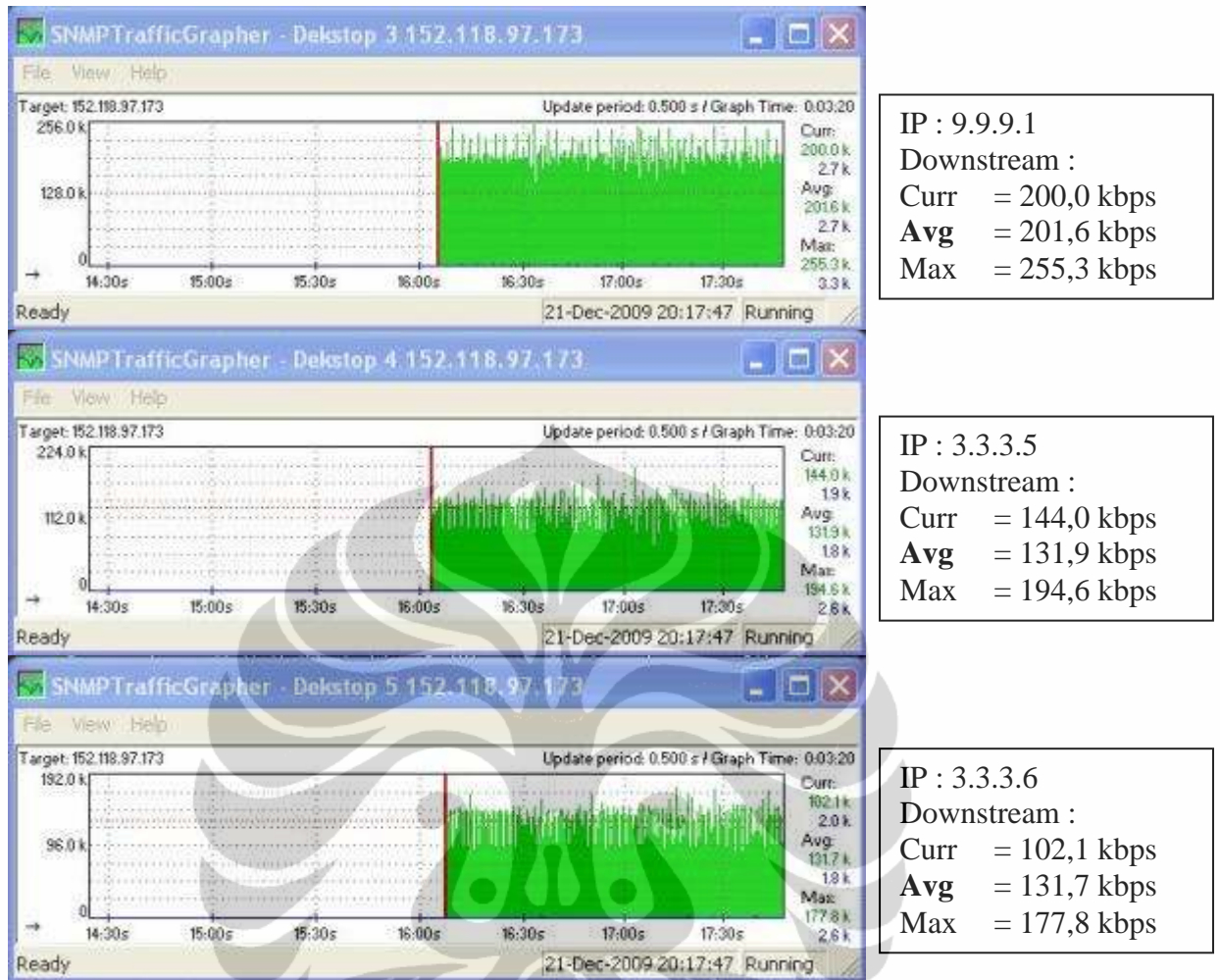
Gambar diatas menunjukkan hasil keluaran trafik *bandwidth* menggunakan *software* STG. Pada kotak sebelah kanan dari setiap gambar, dapat dilihat nilai “**Avg**”/ (rata-rata) *bandwidth* yang merepresentasikan *output* trafik *bandwidth* dari *Rule A* skenario 2. *IP Address* 4.4.4.6 memperoleh alokasi *bandwidth* terbesar dengan *Avg* 217,3 kbps, sedangkan *IP Address* 9.9.9.2 dengan *Avg* 100,6 kbps merupakan koneksi terburuk. Dari hasil *output bandwidth* semua pengguna dapat dilihat bahwa fluktuasi *bandwidth* lebih rata atau bisa diredam, karena menggunakan FTS yang melibatkan *traffic shaping*.

4.3 Rule B Skenario 1 (Topologi Acak)

Pada Tabel 4.5, menunjukkan bahwa koneksi terbaik adalah *IP Address* 4.4.4.6 (*rtt* 0,61ms dan *hop* 2), akan tetapi memperoleh *weight* terkecil (16%) dan alokasi *bandwidth* terkecil (126kbps). Sedangkan koneksi terburuk yaitu *IP Address* 9.9.9.1 (*rtt* 1,78ms dan *hop* 6), akan tetapi memperoleh *weight* terbesar (26%) dan alokasi *bandwidth* terbesar (200kbps).

Hal tersebut menunjukkan kegagalan *rule fuzzy* yang diterapkan pada FTS. Sehingga *Rule B* tidak sesuai dengan tujuan perancangan program.





Gambar 4.3 Output bandwidth, Rule B skenario 1

Gambar diatas menunjukkan hasil keluaran trafik *bandwidth* menggunakan *software* STG. Pada kotak sebelah kanan dari setiap gambar, dapat dilihat nilai “Avg”/ (rata-rata) *bandwidth* yang merepresentasikan *output* trafik *bandwidth* dari Rule B skenario 1.

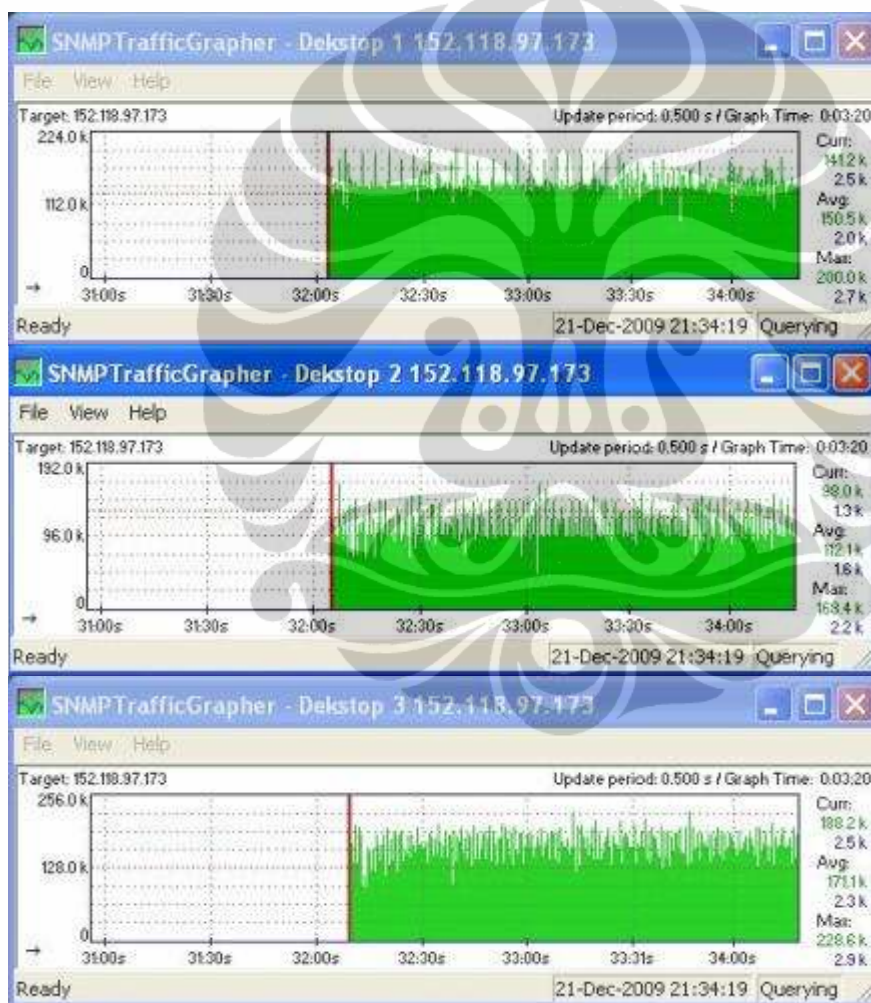
IP Address 4.4.4.6 merupakan koneksi terbaik, akan tetapi memperoleh alokasi *bandwidth* terkecil dengan Avg 124,1 kbps.

Sedangkan *IP Address* 9.9.9.1 merupakan koneksi terburuk, akan tetapi memperoleh alokasi *bandwidth* terbesar dengan Avg 201,6 kbps.

Gambar diatas merupakan representasi kegagalan dari *rule fuzzy* yang diterapkan pada FTS, sehingga Rule B tidak sesuai dengan tujuan perancangan program.

4.4 Rule B Skenario 2 (Topologi Terkonsentrasi)

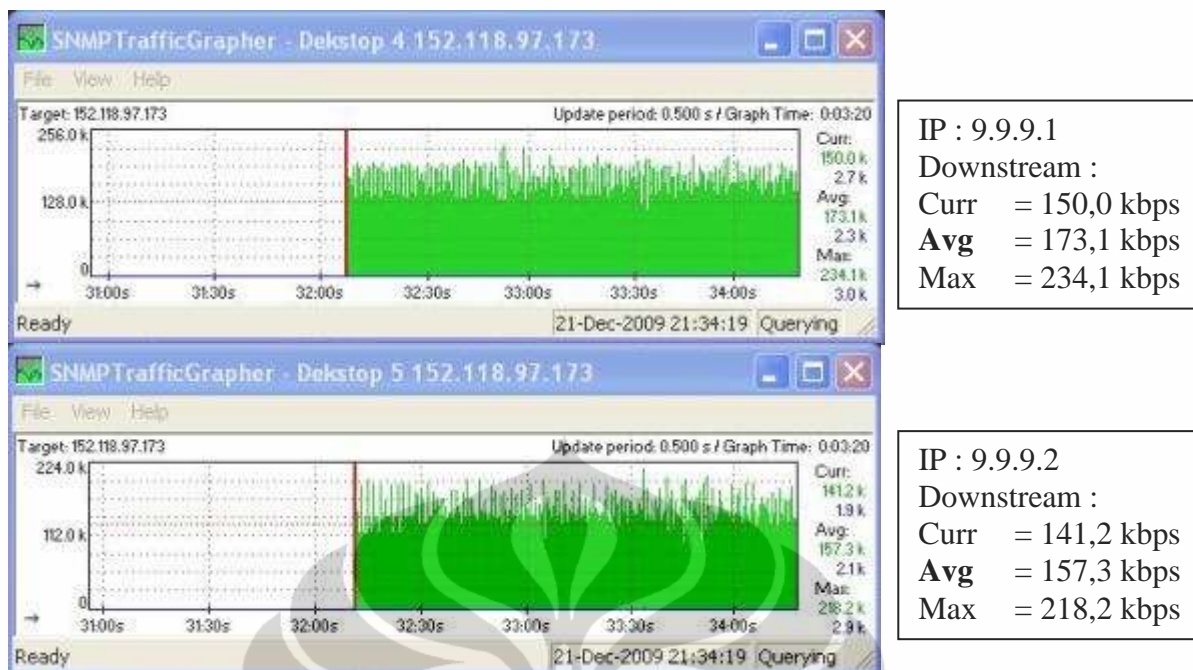
Pada Tabel 4.6, menunjukkan bahwa koneksi terbaik adalah *IP Address* 4.4.4.6 (*rtt* 0,65ms dan *hop* 2), akan tetapi memperoleh *weight* terkecil (15%) dan alokasi *bandwidth* terkecil (113kbps). Sedangkan koneksi terburuk yaitu *IP Address* 9.9.9.4 dan 9.9.9.1 dengan (*rtt* 1,65ms dan *hop* 6), akan tetapi memperoleh *weight* terbesar (23%) dan alokasi *bandwidth* terbesar (173kbps). Hal tersebut menunjukkan kegagalan *rule fuzzy* yang diterapkan pada FTS. Sehingga *Rule B* tidak sesuai dengan tujuan perancangan program.



IP : 9.9.9.5
Downstream :
Curr = 141,2 kbps
Avg = 150,5 kbps
Max = 200,0 kbps

IP : 4.4.4.6
Downstream :
Curr = 98,0 kbps
Avg = 112,1 kbps
Max = 168,4 kbps

IP : 9.9.9.4
Downstream :
Curr = 188,2 kbps
Avg = 171,1 kbps
Max = 228,6 kbps



Gambar 4.4 Output bandwidth, Rule B skenario 2

Gambar diatas menunjukkan hasil keluaran trafik *bandwidth* menggunakan *software* STG. Pada kotak sebelah kanan dari setiap gambar, dapat dilihat nilai “Avg”/ (rata-rata) *bandwidth* yang merepresentasikan *output* trafik *bandwidth* dari Rule B skenario 2.

IP Address 4.4.4.6 merupakan koneksi terbaik, akan tetapi memperoleh alokasi *bandwidth* terkecil dengan Avg 112,1 kbps.

Sedangkan IP Address 9.9.9.1 merupakan koneksi terburuk, akan tetapi memperoleh alokasi *bandwidth* terbesar dengan Avg 173,1 kbps.

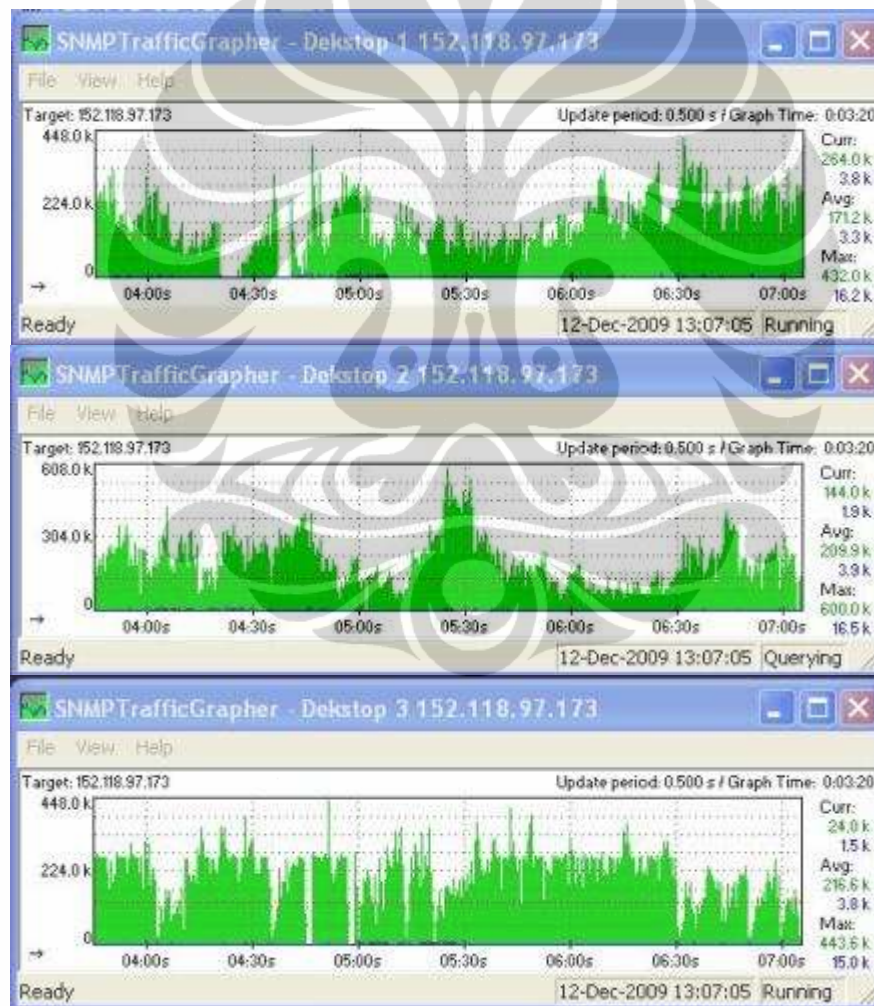
Gambar diatas merupakan representasi kegagalan dari *rule fuzzy* yang diterapkan pada FTS, sehingga Rule B tidak sesuai dengan tujuan perancangan program.

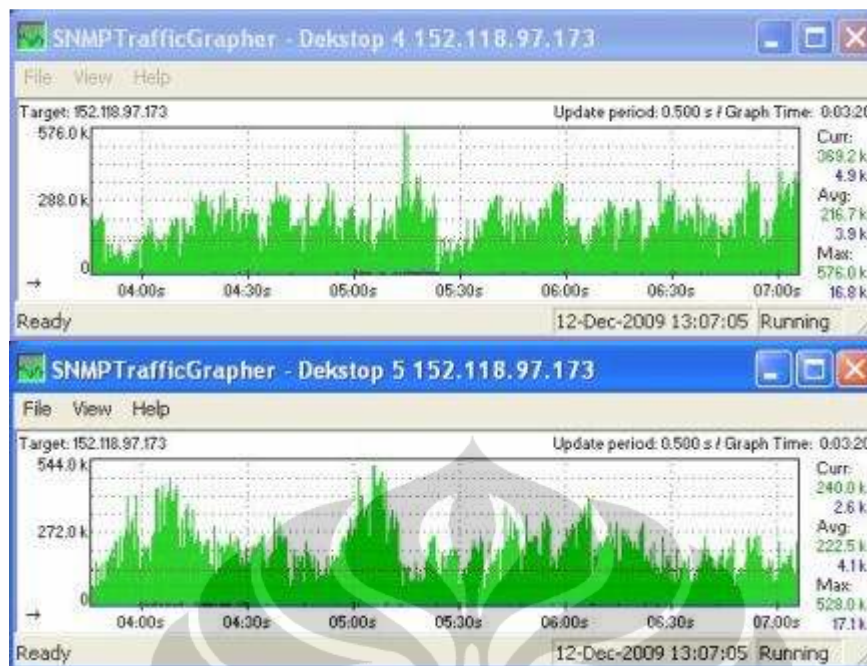
4.5 Percobaan Tidak Menggunakan FTS

Dibawah ini adalah gambar salah satu contoh percobaan yang tidak menggunakan FTS, yang mana tidak melibatkan adanya *traffic shaping*. Tanpa *traffic shaping*, maka fluktuasi *bandwidth* akan sulit diprediksi. Dengan fluktuasi *bandwidth* yang sulit diprediksi maka akan sulit memastikan mana pengguna yang

mempunyai koneksi terbaik dan mana pengguna yang mempunyai koneksi terburuk.

Adakalanya koneksi terbaik memperoleh garansi *bandwidth* terbesar dan adakalanya memperoleh garansi *bandwidth* terkecil. Hal tersebut dikarenakan sifat TCP/IP yang *best effort* telah mempengaruhi transmisi *packet* data didalam jaringan. Hal tersebut sesuai dengan RFC 791 tentang TCP/IP *Protocol suite*, *IP Packet* yang mana *type of services* (tos) mempunyai harga = 0. Adapun tos sendiri mempunyai parameter didalamnya yaitu *delay*, *throughput*, *reliability* dan *cost requirement*.





Gambar 4.5 Output bandwidth, tidak melibatkan *traffic shaping*

Dari gambar diatas dapat dilihat bahwa tanpa melibatkan *traffic shaping* maka fluktuasi *bandwidth* tidak dapat diprediksi. Adakalanya lonjakan *bandwidth* tinggi sekali dan adakalanya rendah sekali. Hal tersebut dikarenakan tidak adanya garansi *bandwidth*/ reservasi yang dialokasikan terhadap pengguna.

Dari sisi pengguna, kenikmatan dalam memperoleh layanan akan terganggu karena *buffering* terhadap aplikasi multimedia disisi pengguna yang waktunya tidak teratur.

4.6 Kinerja Hardware

Dari segi kinerja kedua *file* ketika berjalan, menunjukkan bahwa *resource* CPU yang dibutuhkan kedua *file* tersebut selama proses sangat kecil, sehingga tidak mempengaruhi kinerja Sistem Operasi dalam menjalankan proses yang lain, berikut dengan perintah *top*, *man* TOP(1) [*display and update information about the top cpu processes*] untuk melihat proses secara *realtime*.

```

152.118.97.173 - PuTTY
last pid: 26931; load averages: 0.59, 0.58, 0.38 up 0+00:16:09 12:03:05
62 processes: 2 running, 60 sleeping
CPU: 22.5% user, 0.0% nice, 23.6% system, 1.5% interrupt, 52.4% idle
Mem: 69M Active, 33M Inact, 58M Wired, 136K Cache, 45M Buf, 958M Free
Swap: 2000M Total, 2000M Free

  PID USERNAME   THR PRI NICE   SIZE    RES STATE   TIME  WCPU COMMAND
 1118 root         1  -58    0 13088K 10652K bpf      0:26  0.98% perl5.8.9
   967 mysql       11   4     0 45392K 20396K sbwait   0:00  0.10% mysqld
   856 root         1   8     0  5940K  3964K nanslp   0:16  0.00% snmpd
  1005 root         1  44     0 48052K 25760K select   0:01  0.00% httpd
  1119 root         1   8     0 12836K 10596K nanslp   0:01  0.00% perl5.8.9

```

Gambar 4.6 Perintah *top*Tabel 4.7 Tabel perintah *top*, man TOP(1)

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	TIME	WCPU	COMMAND
1118	root	1	-58	0	13088K	10652K	Bpf	0:26	0.98%	perl5.8.9
967	mysql	11	4	0	45392K	20396K	Sbwait	0:00	0.10%	mysqld
856	root	1	8	0	5940K	3964K	Nanslp	0:16	0.00%	snmpd
1005	root	1	44	0	48052K	25760K	Select	0:01	0.00%	httpd
1119	root	1	8	0	12836K	10596K	Nanslp	0:01	0.00%	perl5.8.9

PID = proses ID.

USERNAME = pemilik proses.

PRI = prioritas proses.

NICE = berhubungan dengan perubahan *priority scheduling*.

SIZE = total besar ukuran proses (teks, data dan *stack*), dalam kilobytes.

RES = *memory* yang ditempati proses ketika berjalan, dalam kilobytes.
(RES = *RESIDENT*)

STATE = nama *state* yang sedang berjalan
(*START, RUN, SLEEP, STOP, ZOMB, WAIT, LOCK* atau hal- hal yang terjadi selama proses menunggu).

TIME = lama waktu sistem dan *user cpu* selama proses berjalan (detik).

WCPU = *Weighted CPU percentage* (persentase bobot *cpu* digunakan), sama dengan *ps* (proses status).

COMMAND = nama perintah proses yang sedang *running*.

Proses sniff.pl (tabel 4.7 baris nomor 2 kolom 10) memerlukan *resource* (WCPU) berkisar 0.98% dan hit.pl (tabel 4.7 baris nomor 6 kolom 10) memerlukan *resource* (WCPU) berkisar 0.00%.

Proses yang dijalankan juga tidak menyebabkan *interface* (*device*) mengalami *error*, atau membebani *buffer interface*, atau menyebabkan data macet (*collission*) di *interface*, disini *interface* yang digunakan bfe0. Berikut perintah untuk melihat *error* di *interface*:

```
fauzi# netstat -in
```

Tabel 4.8 Tabel perintah *netstat* (*network status*) dengan *flags -in*

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
bfe0	1500	<Link#1>	00:02:e3:55:94:24	35204	0	61149	0	0
bfe0	1500	152.118.97.0/	152.118.97.173	35087	-	10159	-	-
sk0	1500	<Link#2>	00:1c:f0:0b:c4:29	0	0	1	0	0

Pada tabel 4.8 baris 2 kolom 6(Ierrs) dan kolom 8(Oerrs) menunjukkan harga = 0, yang artinya tidak ada *input* yang masuk ke *interface* mengalami *error* dan *output* yang keluar *interface* mengalami *error*. Sedangkan pada baris 2 kolom 9(Coll) menunjukkan harga = 0, yang berarti tidak ada *collision* di *interface*.