BAB III

PERANCANGAN DAN SIMULASI SOFTSWITCH

Berdasarkan pada penjelasan dari bab sebelumnya, maka dibuatlah suatu perancangan pemodelan *softswitch* sebelum simulasi dilakukan. Perancangan suatu pemodelan *softswitch* ini dilakukan agar mampu memenuhi kebutuhan PSTN dalm bermigrasi menuju jaringan data.

Simulasi ini berperan penting dalam memodelkan karakter *softswitch* sehingga memungkinkan dilakukannya penggalian arsitektur aplikasi *softswitch* yang berbeda dimasa yang akan datang. Selain itu, simulasi dilakukan untuk menurunkan ongkos dan waktu evaluasi dengan tingkat akurasi yang sedang jika dibandingkan dengan pengukuran di laboratorium. Simulasi *softswitch* ini dilakukan dengan *network simulator (NS-2)*, yang merupakan perangkat lunak simulasi yang banyak digunakan untuk penelitian dibidang jaringan.

1.1 Tahap Perancangan Softswitch

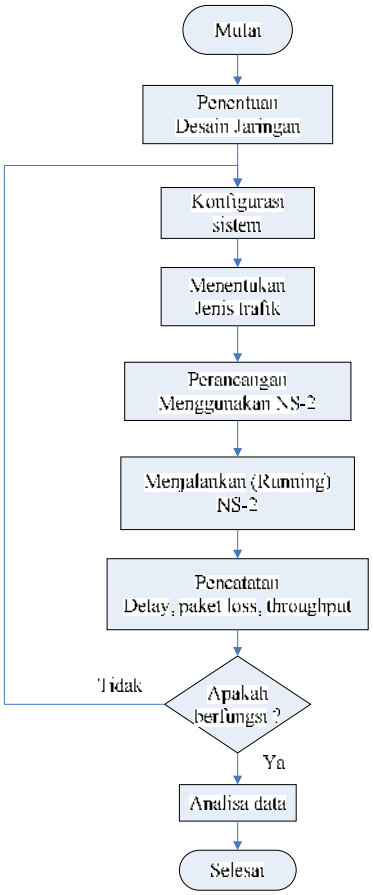
1.1.1 Diagram Alir Desain Sistem

Pada perancangan simulasi *softswitch* ini diperlukan suatu perencanaan yang baik dan tepat agar hasil yang didapat sesuai dengan yang diinginkan. Berikut ini adalah *flowchart* perancangan simulasi *softswitch* pada **Gambar 3.1.**

Perancangan simulasi *softswitch* ini adalah membuat software simulasi yang akan digunakan untuk mensimulasikan karakteristik *softswitch* dengan titik berat untuk mengetahui kinerja *softswitch*. Simulasi ini diharapkan dapat digunakan sebagai *prototype tools softswitch* dan sebagai analisis kerja *softswitch*

pada jaringan yang sedang beroperasi ataupun spesifikasi *softswitch* dalam perencanaan.

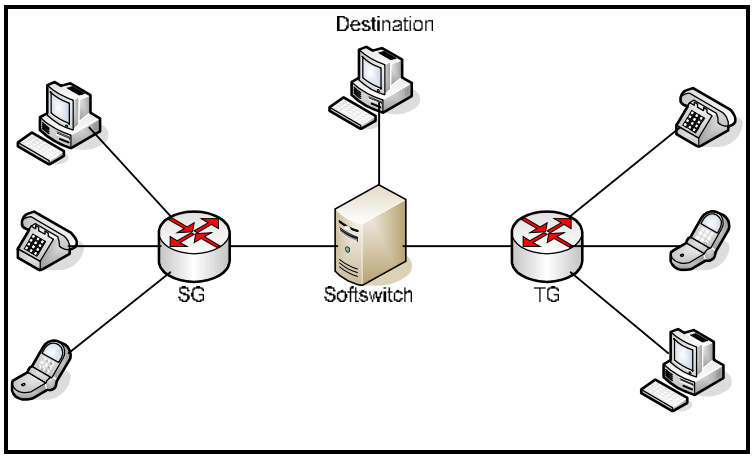
Softswitch merupakan suatu konsep jaringan masa depan yang dikembangkan dari pendekatan teknologi PSTN, VoIP dan jaringan data. Sistem komunikasi ini dirancang untuk dapat memberikan layanan suara dan data, disamping itu juga dirancang untuk melakukan penetrasi terhadap PSTN dalam bermigrasi ke jaringan paket.



Gambar 3.1 Flowchart Perencanaan Simulasi Softswitch

1.1.2 Desain Sistem Jaringan

Adapun pemodelan sistem secara umum pada tugas akhir ini dapat dimodelkan seperti gambar di bawah ini :



Gambar 3.2 Pemodelan Sistem Softswitch

Pendeskripsian sistem diatas adalah sebagai berikut :

Tiap *node* sumber terhubung dengan *signaling gateway* dimana trafiknya ditandai sesuai dengan parameter yang akan dispesifikasikan kemudian. Untuk pensinyalan antara jaringan *SS7* dengan jaringan *softswitch*, SIP tidak dapat langsung terhubung dengan *softswitch*, tetapi harus melalui suatu *gateway* yang disebut *Signalling gateway*. *Trunk gateway* dan *Signalling gateway* terhubung ke *softswitch*, dan kemudian ke *node* tujuan.

1.1.3 Konfiguarasi Sistem Softswitch

Dalam sistem yang akan dibuat pada proyek akhir ini, terdapat *node* sumber sebanyak-n dan tiap nodenya akan membangkitkan aliran *RTP*. Untuk keterangan lebih lanjutnya adalah :

⇒ Link antara node sumber dan *signaling gateway* memiliki *delay* sebesar 10 ms dan *bandwidth* sebesar 2 Mbps.

⇒ Link antara node sumber dan *trunk gateway* memiliki *delay* sebesar 10 ms dan *bandwidth* sebesar 2 Mbps.

⇒ Link antara *signaling gateway* dan *softswitch* memiliki *delay* sebesar 10 ms dan *bandwidth* sebesar 2 Mbps.

⇒ Link antara *trunk gateway* dan *softswitch* memiliki *delay* sebesar 10 ms dan *bandwidth* sebesar 2 Mbps.

⇒ Link antara *softswitch* dan node tujuan memiliki *delay* sebesar 10 ms dan *bandwidth* sebesar 2 Mbps.

Dalam mendesain sistem ini, digunakan beberapa asumsi yang akan digunakan dalam melakukan simulasi. Berikut ini adalah asumsi-asumsi yang dipakai dalam perancangan sistem :

⇒ TCP dan RTP digunakan pada sisi pengirim (sumber), dan TCP Sink pada sisi penerima.

⇒ Pola kedatangan panggilan (trafik telepon) diasumsikan berdistribusi *poisson* dan trafik data berdistribusi *pareto* pada sisi pengirim (sumber).

⇒ Aplikasi yang dipakai adalah FTP, yang merupakan aplikasi yang sering dipakai untuk pengiriman data dan internet kepada pengguna internet untuk menirimkan paket data.

⇒ Setting lama waktu pengamatan selama 1 jam.

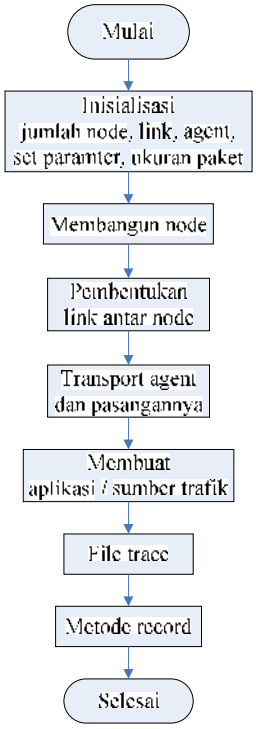
⇒ Parameter performansi berupa *throughput*, *delay* dan *packet loss*.

1.1.4 Menentukan Jenis Trafik

Layanan trafik yang akan diasumsikan dalam perencanaan ini adalah VoIP dan data yang akan memungkinkan terjadinya interaksi yang baik pada komunikasi jaringan. Dalam perancangan simulasi ini digunakan link dua arah (*duplex*) antara node satu dengan node yang lain.

1.2 Perancangan Dengan Menggunakan Network Simulator

Untuk memberikan ilustrasi dalam menginisialisasikan pembuatan sistem, di bawah ini menunjukkan diagram alir inisialisasi pembuatan sistem agar hasil yang didapat sesuai dengan yang diinginkan.



Gambar 3.3 Diagram Alir Inisialisasi Desain Sistem Pada NS

Network simulator-2 yang dipakai dalam simulasi *softswitch* ini adalah *NS-2* versi 2.29 yang berbasis *under windows*. Pada *network simulator* ini,

pemodelan desain jaringan dihasilkan dengan menggunakan modul-modul pembangunan, yaitu *node*, *link*, *agent*, *application* serta *quee management*. *Agent* merupakan objek aktif yang menggerakkan simulasi, ini berupa protokol transport seperti *TCP, UDP* dan *RTP*. Berbagai modul protokol *transport* merupakan objek simulasi *ns* yang dinamai *agent*, sedangkan sumber trafik yang dikirimkan ke lapisan transport merupakan objek simulasi *ns* yang disebut *application*. Setelah sumber trafik dan *agent* dibuat, kemudian digabungkan ke node dengan metode *attach-agent*.

Langkah-langkah pembuatan simulasi NS-2 :

- Pemodelan jaringan. *Client* sebagai node *sender* dan node *receiver* serta node untuk *server* dan node untuk *signaling gateway (SG), trunk gateway (TG), softswitch.*

  Untuk memulai suatu topologi jaringan, maka sebelumnya dibuatlah suatu inisialisasi dalam mengawali pembuatan simulasi.

  Script inisialisasi untuk memulai simulasi :

```
set ns [new Simulator]
set nf [open soft.nam w]
$ns namtrace-all $nf

proc finish {} {
global ns nf
$ns flush-trace
close $nf

puts "running nam.."
```

```
exec nam soft.nam &
exit 0
}
$ns at 60.0 "finish"
$ns run
```

Di dawah ini merupakan contoh *command* node untuk membangun suatu node :

```
set n0 [$ns node]        # membuat node dengan nama n0
set n1 [$ns node]        # membuat node dengan nama n1
set n2 [$ns node]        # membuat node dengan nama n2
set n3 [$ns node]        # membuat node dengan nama n3
set n4 [$ns node]        # membuat node dengan nama n4
set n5 [$ns node]        # membuat node dengan nama n5
set n6 [$ns node]        # membuat node dengan nama n6
set n7 [$ns node]        # membuat node dengan nama n7
set n8 [$ns node]        # membuat node dengan nama n8
set n9 [$ns node]        # membuat node dengan nama n9
```

Untuk membuat link antar *node* dengan manajemen antrian yang akan digunakan, dapat dituliskan sebagai berikut :

```
$ns duplex-link $n0 $n6 2Mb 10ms DropTail
$ns duplex-link $n1 $n6 2Mb 10ms DropTail
$ns duplex-link $n2 $n6 2Mb 10ms DropTail
```

$ns duplex-link $n3 $n8 2Mb 10ms DropTail

$ns duplex-link $n4 $n8 2Mb 10ms DropTail

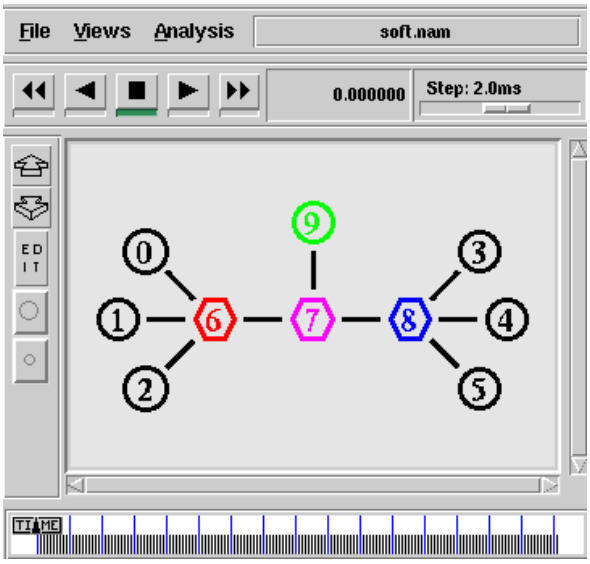
$ns duplex-link $n5 $n8 2Mb 10ms DropTail

$ns duplex-link $n6 $n7 2Mb 10ms DropTail

$ns duplex-link $n8 $n7 2Mb 10ms DropTail

$ns duplex-link $n7 $n9 2Mb 10ms DropTail

Keterangan dalam gambar :



Gambar 3.4 Pemodelan Softswitch pada NS-2

- Proses pengiriman data pada NS dilakukan dengan membuat *transport agent* dan aplikasinya. *Transport agent* dibuat berpasangan, yang satu berfungsi sebagai sumber (*source)* dan yang satunya lagi berfungsi sebagai tujuan (*destination)*. Pada *node sender* menggunakan UDP dan RTP yang diimplementasikan secara bergantian. Sedangkan pada node *receiver* menggunakan *agent null*, yang merupakan pasangan UDP sebagai tujuan trafik.

Berikut adalah contoh membangun hubungan protokol *transport* :

```
set rtp1 [new Agent/RTP]            # rtp1 = agen RTP
$ns attach-agent $n0 $rtp1          # n0 diberi agen rtp1
set udp [new Agent/UDP]             # udp = agen UDP
$ns attach-agent $n0 $udp           # n0 diberi agen udp
set tcp [new Agent/TCP]             # tcp = agen TCP
$ns attach-agent $n3 $tcp           # n3 diberi agen tcp
set null1 [new Agent/LossMonitor]   # null1 = agen RTP
$ns attach-agent $n9 $null1         # n9 diberi agen RTP

set null [new Agent/Null]           # null = agen UDP
$ns attach-agent $n9 $null          # n9 diberi agen udp
set sink [new Agent/TCPSink]        # sink = agen TCP
$ns attach-agent $n6 $sink          # n6 diberi agen tcp
$ns connect $rtp1 $null1            # rtp1 dihubungkan null1
$ns connect $udp $null              # udp dihubungkan null
$ns connect $tcp1 $sink1            # tcp dihubungkan sink
```

♣ Pembangkitan trafik yang digunakan masing-masing node adalah fungsi *Eksponential*. Hal ini dilakukan karena pada ns-2 pembangkitan trafik dilakukan terpisah. Di bawah ini merupakan contoh pembangkitan trafik menggunakan fungsi *Eksponential* pada protokol *transport RTP*.

```
set vois1 [new Application/Traffic/Exponential]
$vois1 attach-agent $rtp1           # vois1 diberi agen rtp
set vois2 [new Application/Traffic/Exponential]
$vois2 attach-agent $udp            # vois2 diberi agen udp
```

set ftp [new Application/FTP] # ftp = aplikasi FTP

$ftp attach-agent $tcp # ftp diberi agen tcp

Untuk mengatur waktu proses sending data, menggunakan perintah sebagai berikut :

$ns at 0.5 "$<aplikasi> start"

$ns at 60.0 "$<aplikasi> stop"

Contoh :

$ns at 0.5 "$vois2 start" # waktu mulai simulasi vois2

$ns at 3599.5 "$vois2 stop" # waktu selesai simulasi vois2

$ns at 0.5 "$data start" # waktu mulai simulasi data

$ns at 3599.5 "$data stop" # waktu selesai simulasi data

$ns at 0.5 "$ftp start" # waktu mulai simulasi ftp

$ns at 3550.0 "$ftp stop" # waktu selesai simulasi ftp

♣ Mengkonfigurasi *softswitch* di NS-2. Menentukan parameter-parameter yang akan digunakan pada *softswitch*. Di bawah ini merupakan contoh pembangkitan model konfigurasi *softswitch* :

membentuk *2 Proxy node*

membentuk *2 softswitch*

set udp_1 [new Agent/UDP/UDPSIP]

$ns attach-agent $PR1 $udp_1

set udp_2 [new Agent/UDP/UDPSIP]

$ns attach-agent $PR2 $udp_2

 # connects peers

$ns connect $udp_1 $udp_2

…..

...
aplikasi : *2 SIP Servers*

set sip_1 [new Application/Traffic/SipUdp]

$sip_1 set packetSize_ $sip_packet_size # ukuran pesan (dalam bytes)

$sip_1 set burst_time_ 0 # eksponensial jarak *burst*

$sip_1 set idle_time_ $interarrivo # eksponensial waktu *iddle*

$sip_1 set rate_ 2.5Mb # sumber *bit rate*

$sip_1 set print_ 0

$sip_1 set min_ss7_delay_ 0.05 # nilai minimum delay

$sip_1 set max_ss7_delay_ 0.1 # nilai maksimum delay

$sip_1 set call_ $max_call_ 100 # mengatur panggilan (dalam detik)

$sip_1 set end_call_management_ 1

$sip_1 set siptimer_ 1

$sip_1 attach-agent $udp_1

set sip_2 [new Application/Traffic/SipUdp]

$sip_2 set packetSize_ $sip_packet_size # ukuran pesan (dalam bytes)

$sip_2 set burst_time_ 0 # eksponensial jarak *burst*

$sip_2 set idle_time_ $interarrivo # eksponensial waktu *iddle*

$sip_2 set rate_ 2.5Mb # sumber *bit rate*

$sip_2 set print_ 0

$sip_2 set min_ss7_delay_ 0.05 # nilai minimum delay

$sip_2 set max_ss7_delay_ 0.1 # nilai maksimum delay

$sip_2 set call_ $call_time_ 100 # mengatur panggilan (dalam detik)

$sip_2 set end_call_management_ 1

$sip_2 set siptimer_ 1

$sip_2 attach-agent $udp_2

...

♣ Mengkonfigurasi *File Trace* di NS-2. *File Trace* merupakan pencatatan seluruh *event* (kejadian) yang dialami oleh suatu simulasi paket pada simulasi yang dibangun. Pembuatan *file trace* dilakukan dengan memanggil *object trace* pada *library*. Sama seperti file nam, pembuatan *output* trace file dinyatakan pada inisialisasi simulasi. Perintah inisialisasi untuk pencatatan seluruh kejadian yang akan disimulasikan pada *softswitch* ini adalah sebagai berikut :

set ns [new Simulator]

set nf [open soft.nam w]

$ns namtrace-all $nf

set f0 [open delay_vois1.tr w]

set f1 [open delay_data.tr w]

set f2 [open thrghpt_vois1.tr w]

set f3 [open thrghpt_data.tr w]

set f4 [open pktloss_vois1.tr w]

set f5 [open pktloss_data.tr w]

set f6 [open pkttrmdt_vois1.tr w]

set f7 [open pkttrmdt_data.tr w]

set f [open output.tr w]

```
$ns trace-all $f

proc finish {} {
global ns nf f  f1 f2 f3 f4 f5 f6 f7
$ns flush-trace
close $nf
close $f1
close $f2
close $f3
close $f4
close $f5
close $f6
close $f7
close $f
puts "running nam.."
exec nam soft.nam &
exit 0
}
$ns at 60.0 "finish"
$ns run
```

Keterangan :

- f0 = *delay* pada paket voip yang menggunakan *coding* G.729
- f1 = *delay* pada paket data yaitu paket TCP dengan aplikasi FTP
- f2 = *throughput* pada paket voip yang menggunakan *coding* G.729

- f3 = *throughput* pada paket data paket TCP dengan aplikasi FTP
- f4 = *packet loss* pada paket voip yang menggunakan *coding* G.729
- f5 = *packet loss* pada paket data paket TCP dengan aplikasi FTP
- f6 = jumlah paket yang dihasilkan oleh voip yang menggunakan metode *coding* G.729
- f7 = jumalh paket yang dihasilkan oleh data TCP dengan menggunakan aplikasi FTP.

♣ Metode *record* merupakan suatu metode yang dapat melakukan *record* sendiri pada file *output*. Metode ini menggunakan agen *Loss Monitor*, dimana agen *Loss Monitor* adalah agen penerima yang juga melakukan perhitungan statistik data. Agen ini hanya bisa dipakai sebagai pasangan agen UDP atau RTP. Perhitungan yang dilakukan oleh transport agen ini yaitu :

- Jumlah paket loss
- Jumlah paket yang diterima
- Jumlah *byte* yang diterima
- Waktu pada saat paket terakhir diterima

Perhitungan statistik dilakukan pada suatu prosedur yang biasanya diberi nama prosedur *record*. Pembangkitan trafik dan prosedur *record* bertujuan untuk menghitung *delay*, *throughput*, *packet loss*, paket terakhir diterima pada node.

♣ Untuk melakukan simulasi desain tersebut, dibuat *script* simulasi yang ditulis pada program *text editor* yang ada di windows, kemudian disimpan dalam folder dengan ekstensi *.tcl*. Hal ini dilakukan karena pada ns-2

menggunakan bahasa pemrograman *C++* dengan *interpreter Tcl* (Tool Command Language) / *Otcl* (Object Tcl). Untuk mencatat hasil simulasi yang diinginkan, dapat dihasilkan dari *trace file*.tr*.

- ♣ Hasil output simulasi menggunakan *Network Animator (Nam)* dalam bentuk *Gui*.nam* untuk melihat topologi dan mekanisme aliran TCP.
- ♣ Untuk menjalankan simulasi yang telah dibuat, dilakukan di *root console*.

1.3 Skenario Perancangan

Perancangan sistem pada simulasi ini dibuat seperti pada topologi di atas (**Gambar 3.2**) dengan sumber node sebanyak n, satu buah *signaling gateway*, *trunk gateway* dan satu *softswitch*. Sesuai dengan tujuan dari pembuatan sistem yaitu membuat desain sistem yang akan digunakan untuk mensimulasikan karakteristik *softswitch* dengan titik berat untuk mengetahui kinerja dari *softswitch*. Hasil yang diharapkan dari perancangan ini adalah penganalisaan performansi sistem terhadap perubahan parameter yang akan divariasikan untuk mendapatkan analisis *throughput*, *delay* dan *packet loss*.

⇒ *Throughput*

Merupakan banyaknya paket yang diterima dengan baik oleh *receiver* dalam suatu selang waktu tertentu. Satuan dalam *bps (bit per second)*.

⇒ *Delay*

Merupakan waktu yang diperlukan suatu paket data untuk dilayani oleh *softswitch*. Satuan dalam *ms (millisecond)*. Besarnya delay maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah 150 ms,

sedangkan *delay* maksimum dengan kualitas suara yang masih diterima pengguna adalah 250 ms.

⇒ *Packet loss*

Merupakan perbandingan antara jumlah paket yang hilang (dibuang) dengan jumlah paket yang diterima oleh *router* pada selang waktu tertentu. *Loss rate* dihitung dalam persentase (%). Paket loss untuk aplikasi *voice* dan multimedia dapat toleransi sampai dengan 20% untuk *single Access Point*. Batas toleransi paket loss adalah sebagai berikut :

Tabel 3.1 Toleransi Packet Loss [1]

Kategori Degredasi	Packet Loss (%)
Sangat baik	0%
Baik	3%
Sedang	15%
jelek	25%

Pada *Network Simulator* (NS), pembangkitan trafik dilakukan terpisah sehingga harus menambahkan pembangkit trafik yang kita inginkan, pada simulasi ini digunakan *File Transfer Protocol (FTP)*. FTP ini selalu memiliki data untuk dikirim, kita dapat mengamati perubahan *throughput* akibat FTP yang menempati *bandwidth* saat *link iddle*. Di sisi penerima diimplementasikan dengan *TCP Sink*, yang berguna untuk mengirim *ACK* ketika menerima data dan digunakan sebagai umpan balik mengenai kondisi jaringan ke pengirim.

1.3.1 Skenario 1 : Pengujian Terhadap Perubahan Jumlah Node Sumber Trafik Dengan Manajemen Antrian DropTail.

Pemodelan yang digunakan pada skenario ini adalah dengan mengubah jumlah sumber terhadap manajemen antrian *DropTail*. Pada skenario ini N akan diubah-ubah sebesar 2 sumber, 4 sumber dan 6 sumber. Dari skenario ini maka dapat dilihat performansi masing-masing sumber trafik dengan membandingkan parameter-parameter *QoS* terhadap perubahan jumlah aliran sumber. Disini juga akan diketahui kemampuan *softswitch* dalam mengatur antrian terhadap perubahan jumlah aliran (N sumber).

1.3.2 Skenario 2 : Pengujian Terhadap Perubahan Jumlah Node Sumber Trafik Dengan Manajemen Antrian RED.

Sama halnya dengan skenario 1, pemodelan yang digunakan pada skenario ini adalah dengan mengubah jumlah sumber, namun yang membedakannya yaitu dengan menggunakan manajemen *RED*. Pada skenario ini N akan diubah-ubah sebesar 2 sumber, 4 sumber dan 6 sumber. Dari skenario ini maka dapat dilihat performansi masing-masing sumber trafik dengan membandingkan parameter-parameter *QoS* terhadap perubahan jumlah aliran sumber. Disini juga akan diketahui kemampuan *softswitch* dalam mengatur antrian terhadap perubahan jumlah aliran (N sumber).

1.3.3 Skenario 3 : Pengujian Perubahan Bandwidth Terhadap Node Sumber Trafik Dengan Manajemen Antrian DropTail dan RED

Pada skenario ini akan diuji pengaruh perubahan bandwidth terhadap kinerja dari *softswitch* yaitu antara *signaling gateway* dengan *softswitch* yang

terhubung ke node tujuan. Pada skenario ini jumlah sumber yang akan digunakan yaitu 6 sumber, kemudian perubahan bandwidth yang akan digunakan yaitu sebesar 2 Mbps, 6 Mbps dan 10 Mbps.