

17.871/H/03



**PERANCANGAN DAN PEMBUATAN  
PERANGKAT LUNAK PENGATURAN BANDWIDTH  
DI TC-net JURUSAN TEKNIK INFORMATIKA ITS**



R51f  
005,1  
Wah  
p-1  

---

2002

Oleh :

**ELOK SRI WAHYUNI**  
NRP. 5195100073

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA**

**2002**

PERPUSTAKAAN I T S	
Tgl. Terima	10-4-2003
Terima Dari	H1
No. Agenda Prp.	217182

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

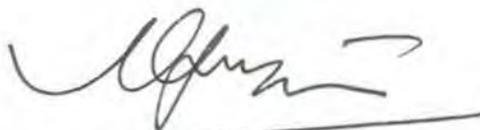
PENGATURAN BANDWIDTH DI TC-NET  
JURUSAN TEKNIK INFORMATIKA-ITS

## TUGAS AKHIR

Diajukan untuk memenuhi sebagian persyaratan  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya

Mengetahui/Menyetujui

Dosen Pembimbing I



IR. Moch. Husni, M.Kom  
NIP. 131 411 100

SURABAYA  
2002

## KATA PENGANTAR

Bismillahirrahmanirrahim

Syukur Alhamdulillah, berkat ridlo dan rahmat Allah SWT penulis dapat menyelesaikan pembuatan Tugas Akhir yang berjudul:

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

PENGATURAN BANDWIDTH

DI TC-net JURUSAN TEKNIK INFORMATIKA ITS

Tugas akhir ini dikerjakan untuk memenuhi syarat akademis dalam rangka ujian akhir bagi mahasiswa Strata 1 (S1) Jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Penulis menyadari banyak kekurangan dalam penyusunan Tugas Akhir ini. Untuk kritik dan saran yang membangun sangat diharapkan dapat mengurangi kekurangan – kekurangan yang ada.

Surabaya, Oktober 2002

Penulis



## DAFTAR ISI

Judul		i
Daftar Isi		ii
Daftar Gambar		iv
Daftar Tabel		v
Abstrak		vi
BAB I	Pendahuluan	1
1.1	Latar Belakang	1
1.2	Permasalahan	2
1.3	Batasan Masalah	3
1.4	Tujuan	3
1.5	Metode dan Sistematika Pembuatan Tugas Akhir	3
BAB II	TINJAUAN PUSTAKA	6
2.1	Quality of Servis	6
2.1.1	Komponen QoS	9
2.1.2	Hirarki Jaringan	10
2.1.3	QoS pada Elemen Jaringan	11
2.1.4	Shaping dan policing	14
2.1.5	Signaling	18
2.2	Bandwidth Manajemen	20
2.3	Telnet	24
2.4	Modul Net::Telnet	30
BAB III	PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK	33
3.1	Struktur Jaringan di Teknik Informatika	33
3.2	Perancangan Perangkat Lunak	34
3.2.1	Perancangan Sistem	34
3.2.2	Perancangan Database	38
3.3	Pebuatan Perangkat lunak	42
3.3.1	Pembuatan Antarmuka	42

3.3.2	Pembuatan program pencatatan konfigurasi	48
3.3.3	Pembuatan program penyettingan bandwidth	50
BAB IV	UJI COBA DAN EVALUASI	57
4.1	Data Ujicoba	57
4.2	Pelaksanaan Ujicoba	59
BAB V	PENUTUP	63
5.1	Kesimpulan	63
5.2	Kemungkinan Pengembangan	63
Daftar Pustaka		64

## DAFTAR GAMBAR

Gambar 2.1	Perbandingan aplikasi yang delay sensitive dan mission critical	22
Gambar 2.2	Proses Bandwidth Manajemen	23
Gambar 3.1	Jaringan TC-net	33
Gambar 3.2	DAD Level 0	35
Gambar 3.3	DAD Level 1	36
Gambar 3.4	DAD Level 2 sub proses pengaturan bandwidth	37
Gambar 3.5	Form Lihat Konfigurasi	43
Gambar 3.6	Form Lihat Profile Router	44
Gambar 3.7	Form Lihat data subnetwork	45
Gambar 3.8	Form tambah data konfigurasi	46
Gambar 3.8	Form menambah data profile router	47
Gambar 3.9	Form tambah data subnetwork	48
Gambar 4.1	Hasil ujicoba 1	61
Gambar 4.2	Hasil ujicoba 2	62

## DAFTAR TABEL

Tabel 2.1	Kode Kontrol ASCII	29
Tabel 3.1	Daftar Field Tabel Subnetwork	38
Tabel 3.2	Daftar Field Tabel Profile Router	39
Tabel 3.3	Daftar Field Tabel Konfigurasi	40

## ABSTRAK

Salah satu hal yang membuat pelayanan server menjadi kurang optimal adalah adanya kondisi beban server yang tidak stabil pada setiap saat. Beban server sangat fluktuatif bergantung pada banyaknya pengguna (user) dan lalu lintas data antar subnetwork atau antar jaringan. Supaya jumlah permintaan user pada traffic di jaringan setiap saat bisa lebih stabil maka bandwidth yang ada harus dialokasikan dengan benar. Secara umum semakin besar bandwidth yang tersedia maka semakin besar pula jumlah data yang bisa dikirimkan dari satu tempat ke tempat yang lain dalam satu satuan waktu (semakin besar data rate-nya). Sehingga untuk mengatur beban server dapat dilakukan dengan mengatur limit/batas data rate lalu lintas data yang ada. Jika jumlah permintaan data (data request) tinggi, data rate untuk semua subnetwork harus dibatasi, sehingga lalu lintas data bisa lebih dikendalikan. Untuk menentukan data rate untuk masing – masing lalu lintas sebelumnya masing – masing lalu lintas diklasifikasikan. Namun untuk mengecek beban server setiap saat dan mengklasifikasikan traffic kemudian mengatur limit data rate untuk masing – masing klasifikasi tersebut sangat sulit jika dilakukan sendiri oleh administrator jaringan, karena hal itu membutuhkan waktu tersendiri. Untuk itu diperlukan perangkat lunak yang bisa memonitor jumlah data request yang diterima pada gateway sebagai parameter tinggi rendahnya jumlah permintaan data, mengklasifikasikan traffic, kemudian mengatur limit data rate dari klasifikasi tersebut. Selain itu perangkat lunak juga mengatur limit data rate tersebut berdasarkan waktu – waktu yang diinginkan.



## BAB I

### PENDAHULUAN

#### I.1 LATAR BELAKANG

Jumlah sarana komputer yang disediakan di institusi pendidikan biasanya disesuaikan dengan banyaknya kebutuhan/pengguna, begitu juga yang ada di jurusan Teknik Informatika ITS. Namun jumlah user yang menggunakan komputer tidak bisa diperkirakan setiap saatnya. Pada jam – jam kerja/kuliah, banyak sekali jumlah user dan aplikasi yang menggunakan komputer dan melakukan pertukaran data antar subnetwork atau ke jaringan lain yang mengakibatkan beban server dalam melayani request dari user menjadi sangat tinggi. Disisi lain pada jam – jam tertentu jumlah user sedikit sehingga beban yang diterimanya ringan/sedikit. Jika banyak user yang mengakses suatu data maka respon yang diberikan server menjadi lambat karena proxy harus melayani permintaan pada waktu yang bersamaan. Sedangkan jika jumlah user yang melakukan permintaan pengaksesan data sedikit maka respon yang diberikan pada user menjadi cepat. Dari keadaan ini dapat dilihat bahwa beban yang diterima oleh server setiap saat tidak stabil.

Supaya jumlah permintaan user pada traffic di jaringan setiap saat bisa lebih stabil maka bandwidth yang ada harus dialokasikan dengan benar. Secara umum semakin besar bandwidth yang tersedia maka semakin besar pula jumlah

data yang bisa dikirimkan dari satu tempat ke tempat yang lain dalam satu satuan waktu (semakin besar data rate-nya). Sehingga untuk mengatur beban server dapat dilakukan dengan mengatur limit/batas data rate lalu lintas data yang ada.

Selain itu adanya beberapa level user di lingkungan TC. Ada beberapa user (misalnya dosen) yang mendapatkan prioritas dalam pengiriman/penerimaan data. Sehingga untuk keperluan itu harus ada pengklasifikasian lalu lintas data supaya data rate bisa diatur berdasarkan klasifikasi tersebut.

Pengaturan limit data rate dengan cara men-setting router secara manual membutuhkan waktu dan sumber daya tersendiri untuk mengerjakannya. Untuk itu diperlukan sebuah program yang mampu memonitor jumlah paket data yang masuk/lewat, menentukan dan mengeset besar limit data rate pada router untuk semua subnetwork berdasarkan jumlah paket data yang masuk/lewat tersebut atau mengeset besar limit data rate pada router sesuai dengan konfigurasi yang telah kita definisikan secara otomatis.

## 1.2 PERMASALAHAN

- Beban jaringan setiap saat bisa berubah. Maka dibutuhkan sebuah program yang dapat membaca data dari object/mesin yang dijadikan patokan seberapa besar beban jaringan dan program dapat secara otomatis mengantisipasi kondisi tersebut dengan menjalankan beberapa perintah supaya beban jaringan tidak terlalu berat.
- Pen-settingan router secara berkala dapat diotomatisasikan dengan program

yang dapat mengecek waktu sekarang dengan setting yang sudah diatur dan menjalankan setting secara otomatis.

### **1.3 BATASAN MASALAH**

Batasan dari sistem yang dibuat :

- Program diimplementasikan di jaringan TC-net
- Komponen router yang diatur hanya masalah besar bandwidth saja.
- Router yang digunakan jenis corebuilder 3500.

### **1.4 TUJUAN**

1. Membuat perangkat lunak yang mampu memonitor beban jaringan.
2. Mengotomatisasi pengaturan bandwidth untuk setiap subnetwork yang ada di jaringan TC berdasarkan beban jaringan yang ada.
3. Mengalokasikan besar bandwidth pada jam – jam tertentu pada subnetwork – subnetwork tertentu.

### **1.5 METODE DAN SISTEMATIKA PEMBUATAN TUGAS AKHIR**

Metode penelitian yang dilakukan menggunakan langkah-langkah berikut :

1. Studi literatur, yaitu mengumpulkan berbagai macam literatur yang berkaitan dengan rumusan masalah diatas untuk selanjutnya dipelajari dan dimengerti. Yang tercakup dalam langkah ini antara lain mempelajari konsep dasar pembuatan bandwidth manajemen serta mempelajari

- bagaimana membuat koneksi ke mesin router dengan telnet lewat program.
2. Perancangan dan pembuatan perangkat lunak, melakukan perancangan dan pembuatan perangkat lunak berdasarkan masalah dan tujuan yang ada. Hal – hal yang termasuk dalam hal ini adalah pembuatan antar muka pemakai, perancangan database, dan program untuk menjalankan setting router. Database yang digunakan disini dimaksudkan untuk menyimpan konfigurasi setting yang diinginkan.
  3. Pengujian perangkat lunak, pada tahap ini dilakukan pengujian terhadap perangkat lunak yang telah dibuat berdasarkan rancangan sebelumnya. Pengujian dilakukan berdasarkan 2 keadaan, yaitu berdasarkan kondisi beban jaringan yang terlalu tinggi dan berdasarkan waktu yang telah ditentukan.
  4. Evaluasi dan modifikasi perangkat lunak, melakukan evaluasi dan analisa terhadap hasil pengujian perangkat lunak dan penyempurnaan bila dianggap perlu.
  5. Dokumentasi, pada tahap ini dilakukan penulisan naskah diantaranya menjelaskan dasar teori, perancangan dan pembuatan perangkat lunak, pengujian dan evaluasi yang dilakukan untuk selanjutnya digunakan sebagai bahan penyusunan buku sebagai dokumentasi dari pelaksanaan tugas akhir.

Adapun sistematika dari pembuatan tugas akhir ini adalah sebagai berikut :

1. **Bab I Pendahuluan**, berisi tentang latar belakang, permasalahan, batasan permasalahan, tujuan serta metode dan sistematika pembuatan tugas akhir.
2. **Bab II Tinjauan Pustaka**, berisi tentang uraian dasar teori yang digunakan untuk menyusun dan menyelesaikan masalah dalam tugas akhir. Pada bab ini dijelaskan mengenai Quality of Servis, Bandwidth Manajemen, aplikasi TELNET dan modul Net::Telnet yang akan digunakan dalam program.
3. **Bab III Perancangan dan Pembuatan perangkat lunak**, berisi tentang rancangan perangkat lunak yang dibuat sebagai sarana untuk menyelesaikan masalah dilanjutkan dengan bahasan mengenai pengimplementasian perancangan perangkat lunak yang telah dibuat.
4. **Bab IV Uji coba dan Evaluasi**, berisi tentang uji coba perangkat lunak terhadap beberapa kasus yang berkaitan.
5. **Bab V Kesimpulan dan Saran**, berisi tentang kesimpulan – kesimpulan yang dapat diambil dari hasil penyelesaian masalah beserta saran – saran yang sebaiknya dilakukan untuk memperbaiki atau mengembangkan hasil tersebut.

## BAB II

### TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan teori yang melandasi pembuatan perangkat lunak pengaturan bandwidth yaitu mengenai Quality of Servis (QoS) pada jaringan, Bandwidth Management dan aplikasi Telnet yang akan digunakan dalam pembuatan perangkat lunak serta modul Net::Telnet yang digunakan dalam program.

#### 2.1 QUALITY OF SERVIS (QoS)

*Quality of Servis (QoS) adalah kemampuan suatu jaringan untuk memberikan servis yang lebih baik terhadap lalu lintas jaringan yang dipilih lewat berbagai jenis teknologi seperti frame relay, ATM, ethernet dsb.*

Fitur – fitur QoS menyediakan servis jaringan yang lebih baik dengan cara :

- Mendukung bandwidth yang ada
- Improving (meningkatkan) karakteristik kehilangan paket
- Menghindari dan mengatur terjadinya transient congestion.
- Membentuk lalu lintas jaringan (traffic shaping)
- Mengatur prioritas lalu lintas yang melewati jaringan.

*End-to-end QoS adalah kemampuan jaringan untuk memberikan servis pengiriman yang disyaratkan oleh lalu lintas jaringan tertentu dari satu end jaringan ke end jaringan yang lain.*

Sebuah servis model (atau disebut juga level servis) menjelaskan satu set kemampuan end – to - end QoS. QoS servis model berbeda antara satu dengan yang lain dalam hal bagaimana mereka mengizinkan aplikasi – aplikasi untuk mengirim data dan cara jaringan mencoba menyerahkan data tersebut. Untuk itu penggunaan servis model ini juga berbeda. Sebagai contoh servis model yang berbeda diterapkan untuk aplikasi real-time seperti konferensi audio dan video dari pada servis model yang diterapkan untuk aplikasi file transfer dan e-mail, karena jenis kebutuhan/servis yang diinginkan dari dua kelompok aplikasi tersebut berbeda. Aplikasi konferensi audio dan video menuntut /memprioritaskan servis waktu penyampaian data, sedangkan aplikasi file transfer dan email memprioritaskan kelengkapana data yang dikirim.

Beberapa servis model yang umum antara lain *best effort*, *integrated* dan *differentiated servis*. Best effort adalah sebuah servis model tunggal dimana sebuah aplikasi mengirim data kemanapun itu harus dikirimkan, dalam jumlah berapapun dan tanpa permintaan ijin atau pemberitahuan terlebih dahulu ke jaringan. Untuk best effort servis, jaringan mengirim data jika bisa dikirim tanpa jaminan apapun tentang reliabilitas, delay maupun hasil yang diterima.

Integrated servis adalah sebuah multiple servis model yang bisa mengakomodasi beberapa permintaan QoS. Dalam model ini aplikasi meminta sebuah jenis servis yang spesifik pada jaringan sebelum mengirim data. Permintaan dibuat dengan eksplisit signaling, aplikasi menginformasikan jaringan traffic profilnya dan meminta sebuah jenis servis tertentu yang dapat mencukupi

bandwidthnya dan delaynya. Aplikasi hanya akan mengirim data setelah mendapat konfirmasi dari jaringan. Aplikasi juga hanya mengirim data pada profile traffic yang sudah dijelaskan. Jaringan menjalankan ijin kontrol berdasarkan informasi dari aplikasi dan resource jaringan yang bisa digunakan. Jaringan juga memenuhi permintaan QoS dari aplikasi selama traffic mengingatkan spesifikasi profil. Jaringan menjalankan komitmen ini dengan menjaga state per-flow dan kemudian mendukung klasifikasi paket, policing dan queueing berdasarkan state tersebut.

Differentiated servis adalah sebuah multiple servis model yang dapat memenuhi permintaan QoS yang berbeda – beda. Namun demikian, tidak seperti model integrated servis, sebuah aplikasi menggunakan differentiated servis tidak secara eksplisit mengirim signal router sebelum mengirim data. Pada servis model ini jaringan mengirim sebuah jenis servis tertentu berdasarkan QoS yang dispesifikasikan oleh setiap paket. Differentiated servis model digunakan untuk beberapa aplikasi yang mengandung misi yang kritis dan untuk menyediakan end - to end QoS.

Ada beberapa faktor yang diperhatikan ketika memutuskan servis apa yang diterapkan di jaringan :

- aplikasi atau masalah apa yang akan diselesaikan
- jenis dari kemampuan yang ingin kita alokasikan pada sumber daya yang kita miliki.
- Analisa pengeluaran dan keuntungan.

### 2.1.1 KOMPONEN QoS

Suatu end-to-end QoS terbentuk dari rangkaian edge-to-edge QoS yang disediakan oleh tiap – tiap domain yang dilalui oleh sebuah lalu lintas data. Lebih jelasnya end-to-end QoS tergantung pada karakteristik QoS dari masing – masing hop pada tiap router. Sebagai contoh QoS yang diujicobakan pada aplikasi telpon intra-LAN hanya bergantung pada LAN itu sendiri. Sedangkan QoS yang diujicobakan pada aplikasi telpon untuk wide area bergantung pada LAN di kedua ujungnya, ISP pada masing – masing end dan IP backbone yang dilewati.

IP jaringan digunakan untuk menangani lalu lintas data dari pelanggan dengan berbagai jenis keperluan, sebagai contoh IP telpon, e-commerce dll. Setiap konsumen mengajukan permintaan layanan yang sifatnya unik.

Fitur QoS bisa dikonfigurasi di seluruh bagian dari sebuah jaringan untuk menyediakan pengiriman end – to-end QoS. Komponen – komponen yang penting untuk mengirim QoS lewat berbagai jaringan adalah :

- QoS pada sebuah elemen jaringan tunggal, dimana disitu termasuk queueing, scheduling, dan fitur traffic shaping.

Elemen terkecil jaringan yang bisa dikontrol adalah node (router atau switch) yang menghubungkan dua link atau lebih. Node – node tersebut diatur berdasarkan suatu arsitektur yang mengijinkan queueing dan scheduling diterapkan pada masing – masing hop dan dapat menggunakan karakteristik QoS dari inter-node link dengan semestinya.

- Policing QoS dan manajemen fungsi untuk mengkontrol dan

mengadministrasi lalu lintas yang melewati sebuah jaringan.

Ada banyak jalur paralel pada jaringan. Dengan membagi lalu lintas data ke jalur – jalur tersebut dapat mengurangi beban rata – rata dan kelebihan beban sepanjang jalur – jalur tersebut. Hal ini dapat meningkatkan kualitas pelayanan jaringan, karena setiap router yang kelebihan beban cenderung untuk men-drop paket yang melewati router tersebut.

- Teknik pen-sinyalan QoS untuk mengkoordinasi end –to-end QoS antara elemen – elemen jaringan.

### 2.1.2 HIRARKI JARINGAN

Setiap jaringan komputer dibangun dari tingkatan komponen – komponen. Setiap jalur dari satu titik ke titik lain biasanya dibentuk dari rangkaian jalur terpendek (hop) yang mempunyai level yang sama. Sebuah path pada beberapa level N menjadi satu hop pada path di level N+1. Sebagai contoh pada IP layer, jaringan dibentuk dari router – router yang bertindak sebagai point yang men-switch paket – paket IP dan link – link yang membawa paket – paket IP antara router – router. Setiap link adalah sebuah hop IP tunggal, sedangkan link itu sendiri terbentuk dari sejumlah hop – hop dan node – node miliknya sendiri. Link bisa jadi adalah sebuah single ethernet, IP tunnel atau virtual connection ATM.

QoS di level IP (IP-level QoS) antara dua titik bergantung pada router dan karakter QoS dari setiap teknologi link. Lebih jelasnya pengiriman paket inter-router membangun kemampuan QoS pada tiap – tiap link. Jika tekhnologi link

tidak menyediakan fasilitas kontrol QoS, router – router tersebut tidak bisa memberi kontribusi terlalu banyak, karena router – router tersebut bergantung pada setiap link untuk menyediakan hubungan antar router yang bagus.

### 2.1.3 QoS PADA ELEMEN JARINGAN

Tujuan dari lingkungan jaringan yang mendukung QoS adalah untuk memungkinkan pemberian layanan pengiriman ke kelas atau tipe traffic tertentu yang lebih bisa diprediksi, tanpa mengindahkan traffic apa yang melewati jaringan pada saat itu.

Jalur end – to- end yang dilalui oleh sebuah paket single user adalah sebuah rangkaian link – link dan router – router. Sedangkan elemen terkecil jaringan yang bisa dikontrol adalah node (router atau switch) yang menghubungkan dua link atau lebih. Meskipun tradisional router terutama difokuskan pada fungsi kemana harus mengirim paket (membuat keputusan untuk memforward suatu paket berdasarkan alamat tujuan pada setiap paket dan secara lokal menangani tabel informasi), router dengan IP level QoS-enabled harus dapat mengontrol “kapan” harus mengirim paket.

#### 2.1.3.1 TRANSIENT CONGESTION, DELAY DAN PACKET LOSS

Setiap router adalah titik temu dan titik pisah terkecil yang dapat dikontrol untuk puluhan, ratusan dan ribuan aliran paket yang tidak berhubungan. Pada data jaringan kebanyakan, traffic datang dalam kondisi ledakan yang fluktuatif. Pada

kondisi biasa, adanya ledakan paket yang terus menerus dari multiple link yang ditujukan ke output link yang sama (dimana link tersebut mempunyai kapasitas yang terbatas) akan meninggalkan paket dengan jumlah melebihi kapasitas yang bisa dikirim pada sebuah router.

Untuk menangani keadaan tersebut semua router menggunakan queue untuk menyimpan paket – paket tersebut sampai paket tersebut bisa dikirim kemudian. Dalam situasi seperti ini, paket – paket yang berusaha melewati router mengalami penundaan (*delay*). Router semacam ini dikatakan sedang mengalami tabrakan sementara (*transient congestion*).

Jadi keterlambatan pengiriman suatu paket dari satu titik ke titik lain adalah sebuah kombinasi penundaan – penundaan transmisi sepanjang link dan proses penundaan yang dialami di setiap router. Pada dasarnya penundaan yang diakibatkan oleh *transient congestion* pada setiap router tidak bisa benar – benar diprediksikan. Penundaan ini berfluktuasi dengan pola – pola tabrakan yang berubah. Komponen fluktuatif dari keterlambatan end-to-end sering juga disebut sebagai *jitter*.

Router mempunyai kapasitas queue yang terbatas, untuk bertahan dari *transient congestion* menyebabkan queue menaikkan kapasitasnya. Ketika paket datang untuk mengisi queue yang penuh, paket harus di drop samapai queue bisa diisi yang menyebabkan terjadinya *packet loss*.

Beberapa jenis traffic ( seperti koneksi TCP yang membawa email) lebih bertoleransi terhadap keterlambatan dibandingkan terhadap paket yang didrop

(packet loss), dengan kata lain queue yang panjang lebih ideal. Sedangkan tipe lain seperti UDP yang membawa streaming video atau audio lebih baik dihilangkan oleh network jika terlalu panjang, dengan kata lain queue yang pendek lebih ideal.

### 2.1.3.2 PENGKLASIFIKASIAN, ANTRIAN DAN PENJADWALAN

Transient congestion dan packet loss di setiap IP jaringan menurunkan QoS dari link dan fungsi dari queue dan manajemen queue itu sendiri di setiap router. Jika jaringan menaikkan tingkat servisnya, satu queue di setiap titik tabrakan internal tidak akan efisien, untuk itu dibutuhkan satu queue untuk masing – masing kelas traffic. Setiap queue tersebut harus mempunyai kebijaksanaan sendiri untuk membuang paket (sebagai contoh, digunakan standart yang berbeda untuk membuang paket secara acak atau dengan ketentuan tertentu). Tentu saja adanya multiple queue ditiap output interface tidak akan berguna tanpa adanya mekanisme untuk memberi tanda pada paket untuk masuk ke queue yang tepat. Metode pengklasifikasian dibutuhkan pada router yang digunakan. Ada variasi metode yang biasa digunakan untuk mengklasifikasikan paket.

Skema queue yang digunakan juga harus dievaluasi, hal ini penting untuk menentukan paket mana yang akan didrop. Kebijakan ini dapat bekerja dengan cara men-drop paket yang datang paling akhir (berdasarkan waktu minimal queue bisa diisi dan/atau atribut tertentu yang dibawa oleh paket itu sendiri).

#### 2.1.4 SHAPING DAN POLICING

Fokus utama dari pengklasifikasian, queue dan penjadwalan (Classify, Queue, Schedule – CQS) adalah bagaimana melindungi traffic pada setiap queue dari ledakan traffic pada queue lain. Jika kapasitas cadangan bisa digunakan, penjadwal yang baik bisa mengalokasikan kapasitas tersebut ke sembarang queue yang mempunyai paket – paket yang menunggu dikirim.

Memberi batas atas untuk maximum bandwidth ( atau nilai minimum pada interval antar paket) pada sebuah kelas traffic dikenal sebagai *traffic shaping*. Shaping scheduler dikonfigurasi untuk memberikan baik minimum service interval ( waktu jeda untuk mengambil paket dari queue yang sama) dan maximum service interval. Paket yang datang dengan interval antar paket yang lebih pendek daripada yang diijinkan oleh scheduler diantri terlebih dahulu sampai bisa ditransmisikan.

Shaping bukan fungsi yang sederhana untuk router. Meskipun tidak terlalu bagus solusi alternatif diperkenalkan untuk penghapusan paket ketika ada ledakan paket. Ketika terlalu banyak paket datang dalam interval yang terlalu pendek, paket di drop. Proses ini dikenal sebagai *policing*.

Policing bisa diimplementasikan tanpa queue – queue atau scheduler, meskipun hal ini umumnya membutuhkan beberapa bentuk klasifikasi untuk membedakan antara aturan kebijakan yang diterapkan pada kelas traffic yang berbeda. Dalam bentuk yang sederhana, setiap kelas traffic mempunyai sebuah counter yang saling berhubungan. Counter akan menaikkan nilainya setiap T detik

dan mengurangi nilainya setiap ada sebuah paket dikirim balik. Jika sebuah paket datang untuk ditransmisikan ketika counter nilainya nol, maka paket tersebut akan didrop. Ketika tidak ada paket yang akan dikirim, counter akan menaikkan nilainya sampai batas limit  $L$ . Namun jika ada lebih dari  $L$  paket yang datang dalam interval waktu kurang dari  $T$  detik, counter akan mencapai nilai nol dan kelebihan paket akan di drop. Nilai  $L$  mempengaruhi toleransi terhadap ledakan paket dan nilai  $T$  diset lebih rendah sampai traffic aman. Cara ini meskipun aneh namun masih efektif untuk diterapkan dalam menghadapi ledakan traffic pada kebijakan router.

Kegunaan dari policing didasarkan pada asumsi bahwa ledakan traffic pada umumnya berasal dari aplikasi yang menggunakan adaptif end-to-end transport protokol seperti TCP. Kehilangan paket (packet loss) diasumsikan untuk mengindikasikan adanya transient congestion dan reaksi TCP adalah memperlambat rate paket yang masuk ke jaringan. Policing mengizinkan operator jaringan untuk memalsukan keberadaan transient congestion ini untuk kelas-kelas traffic tertentu sebelum tabrakan benar-benar mulai terjadi di sepanjang jalur paket. Bahkan ketika kelas traffic tidak menggunakan adaptif end-to-end transport protokol, kebijakan tetap memproteksi jaringan yang dapat dengan terus men-drop paket yang melebihi batas parameter yang diijinkan.

Meskipun shaping bisa menjadi solusi yang bagus untuk melancarkan adanya ledakan traffic, namun policing yang sederhana tidak terlalu berguna. Sejumlah variasi sudah diperkenalkan untuk memperhalus efek dari policing di

edge router. Sebuah policing mungkin hanya digunakan untuk menandai paket – paket (suatu pilihan lain dari pada langsung men-drop paket tersebut) jika paket yang datang melebihi kapasitas yang ditetapkan. Router disepanjang jalur mengenali paket yang ditandai untuk diberi prioritas lebih rendah dibandingkan paket yang tidak ditandai. Jika transient congestion mulai terjadi pada queue, manajemen queue bisa mulai mendrop paket yang ditandai sebelum paket – paket tersebut men-drop paket – paket yang tidak bertanda.

Sebagai metode tambahan, node policing bisa mengimplementasikan sebuah batas ledakan yang mengejutkan ( jika sebuah ledakan paket datang lebih rendah dari batas maka paket – paket tersebut akan ditandai dan ditransmisikan, jika ledakan berlanjut dan datang melebihi batas yang ditetapkan, paket di-drop. Alternatif lainnya, node policing mungkin diimplementasikan pada level yang bertingkat – tingkat berdasarkan rata – rata rate paket yang datang yang diijinkan. Rate yang lebih rendah dimana paket dikirim balik dan tidak ditandai, range tingkat lanjut dimana paket – paket ditandai dan dikirimkan dan batas diatasnya dimana paket – paket langsung didrop. Pengaruhnya pada jaringan pusat adalah ledakan menjadi lebih halus dibandingkan jika diterapkan policing sederhana karena akan ada lebih banyak paket pada ledakan yang akan ditandai daripada paket – paket yang didrop.

Keuntungan dari skema ini adalah tidak adanya tabrakan jaringan lain pada jaringan pusat, kelas traffic tertentu ini dapat menggunakan lebih dari bandwidth yang ada. Banyak algoritma yang digunakan untuk memberikan level –

level penilaian yang beragam dan perhitungan batasan.

Point utama yang menjadi perhatian adalah penyusunan ulang potensial dari paket – paket yang ditandai relatif terhadap paket – paket yang tidak ditandai pada kelas traffic. Situasi ini dapat terjadi jika router menggunakan dua queue yang terpisah untuk membedakan antara paket yang ditandai dan paket yang tidak ditandai pada kelas traffic yang sama. Karena paket yang ditandai mempunyai prioritas lebih rendah, sebuah pengimplementasian mungkin memilih mempengaruhi prioritas relatif ini dengan cara memberi bandwidth lebih banyak pada queue milik paket tidak bertanda dibandingkan bandwidth untuk queue milik paket bertanda. Sebagai konsekuensinya sebuah paket bertanda yang datang terlebih dahulu daripada paket tidak bertanda mungkin akan dikirim setelah paket tidak bertanda dikirim. Asumsikan jika paket bertanda menggunakan semua jalur menuju ke tujuan, maka aplikasi penerima akan menganggap bahwa traffic membawa paket yang rusak.

Meskipun spesifikasi IP tidak menghalangi paket untuk disusun ulang oleh jaringan, konsep ini harus dihindari karena protokol end – to –end umumnya tidak menangani kasus ini dengan efisien. Pada jaringan dimana penandaan paket ini dimaksudkan untuk menaikkan kemungkinan paket drop, solusinya adalah dengan membiarkan router pusat tidak mengindahkan policing penanda ketika mengklasifikasikan paket kedalam queue, memastikan semua paket pada kelas traffic yang sama ditempatkan pada queue yang sama tanpa mengindahkan prioritas penge-drop-an paket. Kemudian memodifikasi batasan penjatuhan paket

untuk queue tersebut berdasarkan ditandai atau tidak ditandainya sebuah paket. Algoritma penjatuhan/pengedropan paket pada router utama aktif secara lebih agresif terhadap paket yang ditandai.

### 2.1.5 SIGNALING

Asumsikan kita bisa memberikan queue dan penjadwalan yang berbeda pada sebuah hop dan link layer – link layer yang dapat dikontrol yang mendasarinya. Kondisi ini mensyaratkan adanya koordinasi perilaku disepanjang path(jalur). Istilah umum untuk proses ini adalah *signaling* (pensinyalan) yaitu suatu aksi untuk memberi informasi kepada setiap hop disepanjang jalur – jalur tentang bagaimana mengenali traffic dimana sebuah proses spesial disyaratkan dan apa jenis/tipe proses spesial yang disyaratkan tersebut.

Signaling dapat diterima dalam sejumlah cara dengan tingkat batasan waktu, fleksibilitas dan campur tangan manusia yang bervariasi. Pada suatu kasus edge – to –edge signaling yang ekstrim, sebuah jaringan diberi informasi setiap kali sebuah kelas traffic yang baru membutuhkan/meminta dukungan yang spesial. Jaringan itu sendiri merespon permintaan tersebut dengan membangun informasi tambahan secara internal ( atau memodifikasi informasi yang telah ada ) pada tiap hop untuk mencapai persyaratan edge – to –edge yang diminta.

Teknologi jaringan yang baru seringkali tidak mempunyai protokol dynamic signaling yang terdefiniskan. Dibawah beberapa kondisi tertentu, jaringan kadang – kadang di tetapkan untuk jenis servis yang baru – seringkali

melibatkan campur tangan manusia untuk mengkonfigurasi atau mengkonfigurasi ulang pengontrol link dan node sepanjang jalur yang dikenai. Provisioning adalah suatu bentuk dari signaling, meskipun waktu responnya kadang lebih pelan dari pada dinamik signaling.

Karena jumlah link dan node pada sebuah jaringan sangat besar, banyak vendor membangun pengontrol yang terpusat atau server yang mengatasi konfigurasi atau provisioning. Pengontrol ini secara otomatis mendistribusikan rule yang sesuai ke link atau node pada jaringan. Pengontrol juga otomatis bereaksi pada perubahan kondisi jaringan yang sesuai dengan kebijakan umum yang diatur oleh operator. Meskipun skema terpusat juga menyangkut mekanisme dinamik, skema ini berbeda dengan edge – to – edge signaling dimana skema tidak bisa dikontrol oleh user. Internet sudah lama menggunakan bentuk dinamik signaling yaitu protokol routing.

Ketika sebuah signaling membutuhkan keadaan – keadaan beberapa tujuan QoS tertentu, ada sejumlah variabel untuk diperhatikan. Dalam teori baik path dan resource sepanjang path bisa dimodifikasi.

Masalah lain dari signaling adalah jumlah state informasi tambahan yang harus dibawa oleh router. State informasi adalah segala sesuatu yang router butuhkan untuk mengkararakteristikan traffic spesial (sebagai contoh informasi IP header untuk mengklasifikasikan paket – paket ) dan memproses spesial traffic ( seperti asosiated queue, parameter paket drop, dan prioritas penjadwal)

## 2.2 BANDWIDTH MANAJEMEN

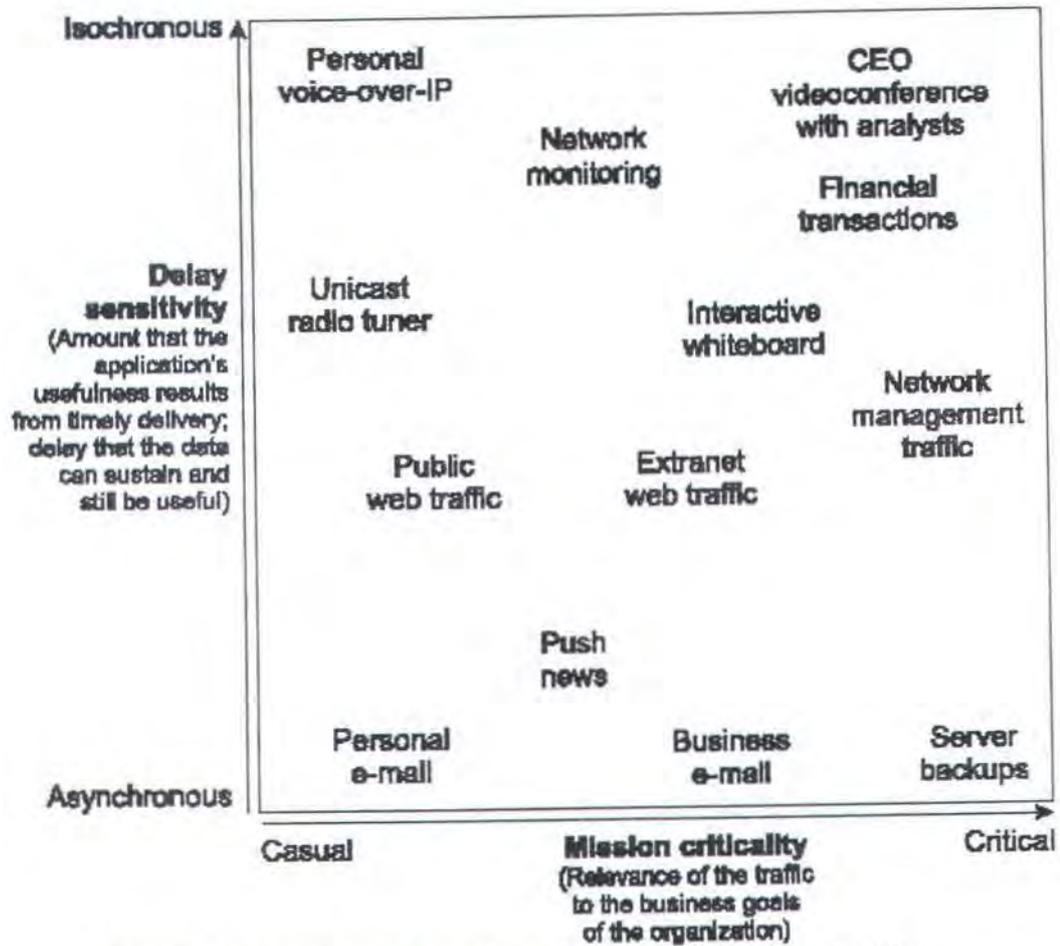
Bandwidth Management secara umum adalah konsep yang sederhana. Ide dasarnya adalah membagi dan mengklasifikasikan user traffic berdasarkan berbagai macam kriteria dan memberi prioritas pada masing – masing klasifikasi paket atau session. Tujuan utamanya adalah memberi kapasitas ( data rate) yang sesuai untuk semua level user dan aplikasi di jaringan. Contoh penerapannya misalnya bandwidth management memungkinkan administrator untuk memberi prioritas lebih untuk tipe traffic HTTP dibandingkan tipe traffic SMTP. Selain itu bandwidth management dapat melacak/melihat besar bandwidth yang digunakan oleh masing – masing aplikasi dan mengeset batas/limit data rate yang bisa digunakan pada masing – masing traffic.

Proses yang dijalankan dalam Bandwidth management antara lain mengklasifikasikan traffic, mengatur alokasi bandwidth dan mapping kelas – kelas traffic.

Ada sejumlah cara untuk mengklasifikasikan traffic, berdasarkan nama server, network subnet, IP dan port tujuan, IP dan Port asal, jenis file yang di-request, aplikasi atau bahkan faktor lain seperti waktu dll. Pengklasifikasian paket memungkinkan network manager untuk melacak request dan mengalokasikan resource berdasarkan prioritas. Apakah sebuah FTP request dari bagian accounting lebih penting dibandingkan HTTP request dari bagian human resource. Fungsi ini juga bisa memungkinkan suatu network untuk memberi proritas berdasarkan penggunaan aplikasi. Beberapa aplikasi sangat penting bagi bisnis

proses suatu organisasi atau perusahaan. Sehingga aplikasi yang berbeda mempunyai tingkat "mission criticality" yang berbeda pula bagi organisasi atau perusahaan. Aplikasi mana yang tergolong dalam kelompok yang mempunyai "mission criticality" yang tinggi dan mana yang tidak sangat tergantung pada tujuan bisnis dari organisasi tersebut. Jadi suatu aplikasi bisa dipandang sebagai aplikasi dengan "mission criticality" yang tinggi oleh suatu organisasi, tapi bagi organisasi lain tidak.

Selain itu hal lain yang perlu diperhatikan dalam pemberian besar data rate bagi suatu aplikasi adalah apakah aplikasi tersebut sensitive terhadap *delay* atau tidak. Misalnya aplikasi seperti file transfer dan e-mail umumnya bisa bertahan dalam berbagai variasi waktu *delay*. Meskipun informasi yang dibawa oleh aplikasi tersebut penting bagi bisnis, tetapi data itu akan tetap valid dan berguna meskipun mengalami *delay* yang cukup lama. Sedangkan aplikasi seperti voice chat tidak akan terlalu berfungsi jika mengalami *delay* yang cukup lama, sehingga aplikasi seperti ini seharusnya mendapat prioritas/alokasi data rate yang cukup tinggi.



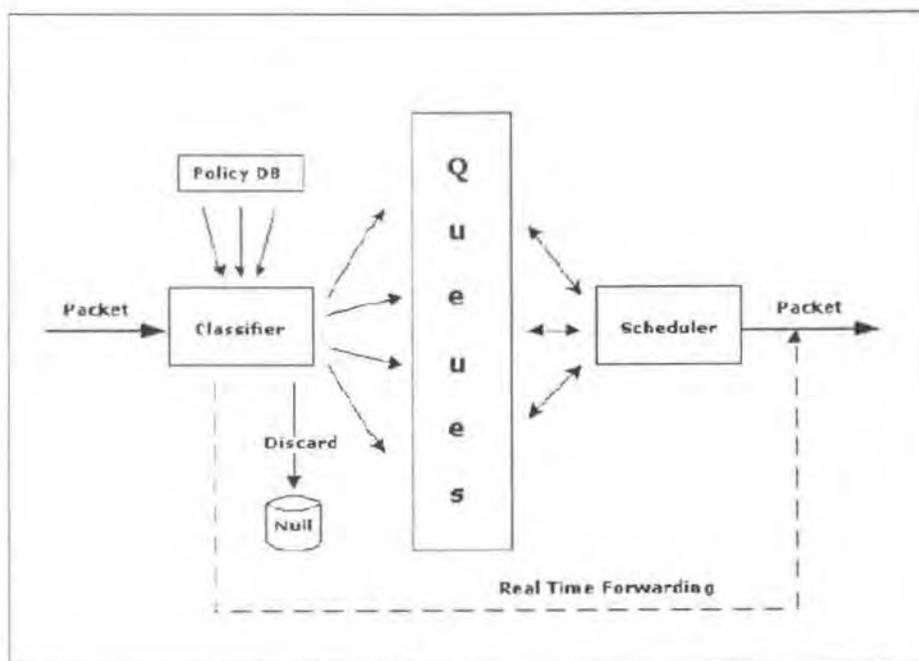
Gambar 2.1 Perbandingan aplikasi yang delay sensitive dan mission critical

Setelah traffic diklasifikasikan, konfigurasi yang spesifik dapat didefinisikan untuk mengontrol bagaimana bandwidth didistribusikan. Dua metode yang sering dipakai adalah *Partition* dan *policy*.

Partition membuat channel yang berbeda dan khusus untuk traffic yang mengatur total pemakaian jaringan berdasarkan tipe traffic – sebagai contoh, jenis traffic FTP diberi 10% dari nilai resource, email 10% dan incoming request

HTTP diberi 50%. Sisa bandwidth yang tidak digunakan dialokasikan untuk pada pool yang bisa digunakan oleh aplikasi lain untuk memperlancar jaringan secara keseluruhan.

Ada beberapa cara dalam menjalankan bandwidth management *Policy* yang dapat dikonfigurasi oleh administrator jaringan, cara yang paling sederhana adalah menghancurkan paket yang datang ketika buffer yang dialokasikan sudah penuh. Cara yang lebih kompleks antara lain kebijakan berdasarkan besar TCP rate (TCP rate based policies) dan kebijakan pengaturan queue berdasarkan prioritas (priority based queueing).



Gambar 2.2 Proses Bandwidth Manajemen

Policy berdasarkan TCP rate menggunakan mekanisme inherent flow control dari protocol TCP. Dengan mengeset parameter pada header paket TCP, bandwidth

management dapat memberi sinyal ke end node untuk memperlambat arus pengiriman paket. Mekanisme ini hanya berlaku untuk sesi TCP. Rate shaping juga tidak berjalan dengan baik untuk protocol yang menggunakan short lived session seperti HTTP, session biasanya berakhir sebelum bandwidth manager memutuskan bagaimana mengatur rate dari session.

Policy yang kedua didasarkan pada prioritas. Prioritas dijalankan berdasarkan queueing, artinya semua klasifikasi paket ditempatkan pada queue yang masing – masing queue memiliki prioritas sendiri. Sejumlah queue bisa digunakan dan paket dikirim dari queue yang mempunyai prioritas tertinggi terlebih dahulu.

Kebijakan – kebijakan ini memberikan stabilitas pada jaringan dan efisien untuk semua request dan response.

### 2.3 TELNET

Sebelum World Wide Web menggunakan mode grafis untuk mengakses internet, komputer yang terhubung ke internet hanya mengerti command – command yang ditulis (seperti DOS). Karena aplikasi dan operating system pada waktu itu masih menggunakan bentuk CLI (command Line Interface). CLI (command Line Interface) adalah user interface dari sebuah *operating system* atau sebuah *aplikasi* dimana respon user ditujukan ke sebuah visual prompt dengan mengetikkan sebuah command/perintah pada line yang sudah dispesifikasikan, menerima respon kembali dari sistem dan kemudian memasukkan lagi command -

command lain dan seterusnya. Aplikasi MS-DOS Prompt pada operating sistem Windows adalah contoh dari bentuk command line interface. Kebanyakan operating sistem dan aplikasi sekarang menggunakan graphical user interface (GUI) seperti sistem operasi Windows, Macintosh dll. Sedangkan sistem berbasis UNIX umumnya menyediakan baik bentuk command line interface maupun bentuk graphical user interface.

Telnet adalah sebuah cara untuk menghubungi ke komputer – komputer tersebut dan aplikasi ini menggunakan bentuk CLI, sehingga untuk menjalankan perintah – perintah dengan menuliskan command – command tersebut. Umumnya kita mengakses ke komputer ini dari sebuah “terminal” – sebuah komputer yang terhubung ke jaringan yang lebih luas.

Program telnet berkomunikasi dengan komputer lain yang terhubung ke internet dengan menggunakan apa yang disebut dengan protokol telnet. Remote terminal protokol ini adalah salah satu Protokol TCP/IP, yang mengijinkan user masuk/login ke sebuah komputer pada jaringan lewat internet. TELNET menyediakan koneksi TCP dan mengirim penekanan tombol pada keyboard user langsung ke remote computer sama seperti jika terjadi penekanan tombol pada keyboard yang terhubung langsung dengan mesin remote tersebut. TELNET juga membawa output dari remote mesin ke layar user. Servis ini disebut *transparent* karena servis ini memberikan tampilan seolah – olah keyboard dan layar user terhubung langsung dengan mesin remote.

Meskipun TELNET tidak sebagus beberapa remote terminal protocol lain,

namun protocol ini cukup banyak digunakan. Biasanya software TELNET client mengijinkan user menyebutkan mesin remote yang dituju dengan nama domainnya atau alamat IP-nya.

TELNET menyediakan tiga servis dasar. Yang pertama protokol TELNET mendefinisikan sebuah *network virtual terminal* yang menyediakan sebuah interface standard ke remote system. Program client tidak harus mengerti semua detil dari remote system yang mungkin, tapi cukup menggunakan interface standard. Kedua TELNET mempunyai mekanisme yang mengijinkan client dan server untuk menegosiasikan option – option dan menyediakan satu set option tandard (misalnya salah satu option kontrol adalah apakah data yang melewati koneksi menggunakan standard 7-bit ASCII karakter set atau 8-bit karakter). Terakhir TELNET memberi servis pada kedua ujung /end dari koneksi secara simetris. Pada kondisi tertentu, TELNET tidak memaksa input client berasal dari keyboard atau tidak mengharuskan output client ditampilkan di layar.

Ketika user melakukan TELNET, sebuah program aplikasi pada mesin user menjadi client. Client berjalan pada sebuah koneksi TCP ke server sebagai sarana komunikasi. Sekali komunikasi dideklarasikan, client menerima penekanan tombol dari keyboard user dan mengirimnya ke server, dan secara konkuren menerima karakter yang dikirim balik oleh server dan menampilkannya pada layar user. Server harus menerima sebuah koneksi TCP dari client dan meneruskan data dari koneksi TCP ke lokal operating system.

Pada kenyataannya, server lebih kompleks dari yang dijelaskan diatas.



karena server harus mengatasi multiple komunikasi konkuren. Biasanya sebuah proses master server menunggu koneksi – koneksi baru dan membuat sebuah slave baru untuk mengatasi setiap koneksi tersebut.

Istilah *pseudo terminal* digunakan untuk menjelaskan titik masuk operating system yang mengijinkan program berjalan seperti TELNET server untuk mentransfer karakter ke operating system sama seperti jika karakter tersebut berasal dari sebuah keyboard. Jika system mendukung sebuah pseudo terminal, TELNET server bisa diimplementasikan dengan program aplikasi. Setiap slave server menghubungkan sebuah aliran TCP dari satu client ke sebuah pseudo terminal tertentu.

Mengatur TELNET server menjadi sebuah aplikasi level program mempunyai keuntungan dan kerugian. Keuntungan yang paling umum adalah TELNET lebih mudah dimodifikasi dan lebih mudah mengontrol server daripada jika kode ditempelkan pada operating system. Kerugiannya mungkin hanya masalah ketidak efisienan. Setiap penekanan tombol berjalan dari keyboard user melewati operating system menuju ke program client, dari program client kembali melewati operating system dan lewat internet ke mesin server. Setelah mencapai mesin tujuan, data harus berjalan melewati operating system server ke program aplikasi server dan dari program aplikasi server kembali ke operating system server pada sebuah titik masuk pseudo terminal. Akhirnya remote operating system mengirim karakter ke program aplikasi yang dijalankan user. Selama itu output (termasuk didalamnya remote karakter echo jika option sudah dipilih)

berjalan balik dari server ke client lewat jalur yang sama.

Supaya TELNET bisa dioperasikan antar berbagai jenis system, maka TELNET harus bisa mengakomodasi detail dari berbagai jenis komputer dan operating system. Sebagai contoh, beberapa system mengharuskan baris – baris teks diakhiri dengan karakter ASCII *carriage control* (CR). System yang lain meminta karakter ASCII *linefeed* (LF). Sementara system yang lain meminta urutan dua karakter CR-LF. Sebagai tambahan, pada system yang interaktif, menyediakan cara untuk seorang user menekan sebuah tombol yang bias menginterupsi sebuah program yang sedang berjalan. Penekanan tombol spesifik yang digunakan untuk menginterupsi sebuah program bervariasi antar satu system dengan system yang lain (misalnya ada system yang menggunakan Control-C, ada system yang menggunakan ESCAPE).

Untuk mengakomodasi keheterogenan, TELNET mendefinisikan bagaimana data dan runtutan command dikirim lewat internet. Pendefinisian ini dikenal sebagai *network virtual terminal* (NVT).

Software client menerjemahkan penekanan tombol dan runtutan command dari terminal user ke format NVT dan mengirimnya ke server. Software server menerjemahkan data yang masuk dan command – command dari format NVT ke format yang diminta oleh remote system. Untuk pengembalian data, remote server menerjemahkan dari format mesin remote ke NVT dan client lokal menerjemahkan dari NVT ke format mesin lokal.

Pendefinisian dari format NVT sangat kompleks, semua komunikasi mengandung 8-bit bytes. Sebagai permulaan NVT menggunakan 7-bit USASCII untuk merepresentasikan data dan menyembunyikan bytes dengan high order bit set untuk runtutan command – command. Set karakter USASCII termasuk didalamnya adalah 95 karakter yang mempunyai bentuk grafis ( seperti huruf – huruf, angka, dan tanda) sebagai 33 kode control. Semua karakter yang mempunyai bentuk diberi tanda sama dengan arti dari set karakter USASCII standard. Standard NVT mendefinisikan interpretasi dari karakter control sebagai berikut :

Tabel 2.1 Kode kontrol ASCII

ASCII control code	Nilai desimal	Arti tanda
NULL	0	Tidak ada operasi (tidak mempunyai efek terhadap output)
BEL	7	Sound audible/visible signal
BS	8	Move left satu posisi karakter
HT	9	Move right ke horizontal tab stop berikutnya
LF	10	Move down (secara vertical) ke line berikutnya
VT	11	Move down ke vertical tab stop berikutnya
FF	12	Bergerak ke bagian atas halaman berikutnya
CR	13	Bergerak ke batas kiri pada line sekarang
Control lain	-	Tidak ada operasi

Sebagai tambahan untuk interpretasi dari karakter kontrol pada table diatas, NVT mendefinisikan terminasi garis standard menjadi dua urutan karakter CR-LF. Ketika user menekan tombol yang berhubungan dengan akhir garis pada terminal lokal (misal ENTER atau RETURN), TELNET client harus mengkonfigurasi itu ke CR-LF untuk tranmisi. TELNET server menerjemahkan CR-LF ke karakter akhir garis yang sesuai untuk mesin remote.

#### 2.4 MODUL Net::Telnet

Modul Net::Telnet digunakan untuk berinteraksi dengan port TELNET atau port TCP yang lainnya. Dengan modul ini user diijinkan untuk membuat koneksi sebagai client ke port TCP dan melakukan operasi Input/Output jaringan, terutama untuk port dengan menggunakan protokol TELNET. Kita bisa melakukan metode I/O seperti print, get, dan getline.

Kemudahan lain ketika menggunakan modul ini dibandingkan langsung mengakses port TELNET adalah kita bisa mengatur waktu time-out ketika melakukan koneksi, pembacaan data dan penulisan data.

Untuk membuat object Net::Telnet adalah :

```
$obj = Net::Telnet->new([Binmode = $mode,] [Dump_Log =
$filename,] [Errmode = $errmode,] [Fhopen = $filehandle,]
[Host = $host,] [Input_log = $file,] [Input_record_separator =
$char,] [Output_log = $file,] [Output_record_separator =
$char,] [Port = $port,] [Prompt = $matchop,] [Telnetmode =
$mode,] [Timeout = $secs,]);
```

Perintah diatas akan membuat object `Net::Telnet` baru. `$host` menampung data host yang akan dihubungi dan object akan melakukan buka koneksi ke TCP pada `$port` pada `$host`.

Nilai default untuk masing – masing variabel diatas adalah :

- Nilai default **host** adalah `localhost`
- Nilai default **port** adalah `23`
- Nilai default **prompt** adalah `'[S%#>] S/`
- Nilai default **timeout** adalah `10`
- Nilai default **errmode** adalah `'die'`
- Nilai default **output\_record\_separator** adalah `“\n”`
- Nilai default **input\_record\_separator** adalah `“\n”`
- Nilai default **binmode** adalah `0`, yang berarti melakukan penerjemahan baris baru

Berikut ini contoh perintah untuk melakukan koneksi ke sebuah remote host bernama “sparky”. Untuk melakukan koneksi ke suatu remote host selain kita harus tahu login dan passwordnya juga bentuk shell prompt dari remote host tersebut. Misalkan dalam contoh ini bentuk shell promptnya adalah `bash$`

```
Use Net::Telnet();
$t = new Net::Telnet(Timeout => 10,
  Prompt => '/bash\$ $/');
$t -> open ("sparky");
$t -> login($username, $passwd);
@lines = $t -> cmd("ls");
print @lines;
```

Metode *login()* dan *cmd()* menggunakan setting prompt pada object untuk menentukan ketika sebuah login atau sebuah command selesai dilaksanakan. Jika prompt yang ditangkap tidak sesuai maka command tersebut dianggap timeout.

Untuk menutup koneksi dilakukan perintah sebagai berikut

```
$ok = $obj-close;
```

metode ini akan menutup socket, file, or pipe yang berhubungan dengan object.

Untuk menjalankan command pada host yang dihubungi :

```
$ok = $obj-> cmd(String = $string, [Output = $ref,]
[Prompt = $match,] [Timeout = $secs,]); @output = $obj-
cmd($string); @output = $obj-cmd(String = $string, [Output =
$ref,] [Prompt = $match,] [Timeout = $secs,]);
```

metode ini mengirim command \$string dan membaca karakter yang dikirim balik oleh remote host sampai terakhir diterima prompt yang sesuai. Diasumsikan bahwa program yang dihubungi menggunakan command prompt interpreter seperti shell.

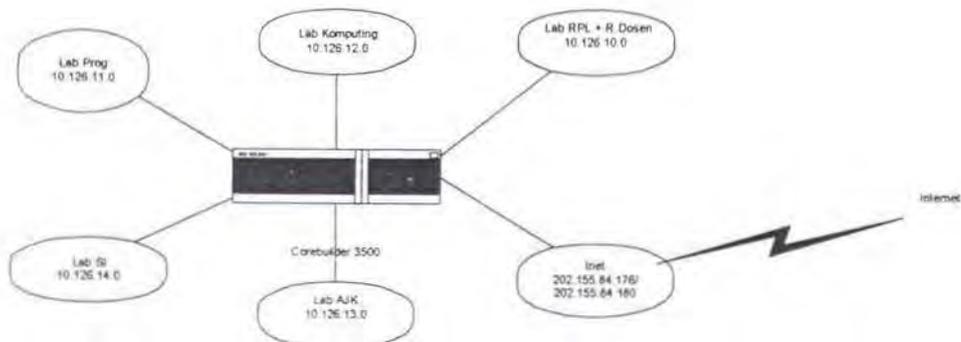
Didalam sebuah array hanya output yang dihasilkan dari command yang dikembalikan dan prompt yang ditampilkan, satu baris per elemen. Jika command tidak memberi hasil sebuah array mengandung satu elemen yaitu string null.

## BAB III

### PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Bab ini menjelaskan tentang tahapan proses perancangan dan pembuatan perangkat lunak. Tahapan ini terdiri dari sub bab struktur jaringan di lingkungan implementasi perangkat lunak yaitu TC-net, perancangan perangkat lunak dan pembuatan perangkat lunak. Sub bab perancangan perangkat lunak terdiri dari penjelasan tentang alur proses yang terjadi dan database yang akan dipergunakan dalam perangkat lunak. Sedangkan pada sub bab pembuatan perangkat lunak akan dijelaskan tentang antarmuka dengan pengguna dan program yang dibuat sesuai dengan alur proses yang dirancang.

#### 3.1 STRUKTUR JARINGAN DI JURUSAN TEKNIK INFORMATIKA



Gambar 3.1 Jaringan TC-net

Struktur jaringan TC-net mempunyai 6 subnetwork yaitu Laboratorium Pemrograman(Lab Prog), Laboratorium Komputing (Lab Komputing), Laboratorium Rancangan Perangkat Lunak (Lab RPL) + Ruang Dosen, Laboratorium Arsitektur Jaringan Komputer (Lab AJK), dan Laboratorium Sistem Informasi (Lab SI.).

Setiap subnetwork tersebut terhubung ke sebuah router CoreBuilder tipe 3500, dari router ini besar bandwidth untuk masing – masing subnetwork diatur dan dialokasikan. Router terhubung ke sebuah gateway (I-net) yang menghubungkan jaringan dengan internet.

### **3.2 PERANCANGAN PERANGKAT LUNAK**

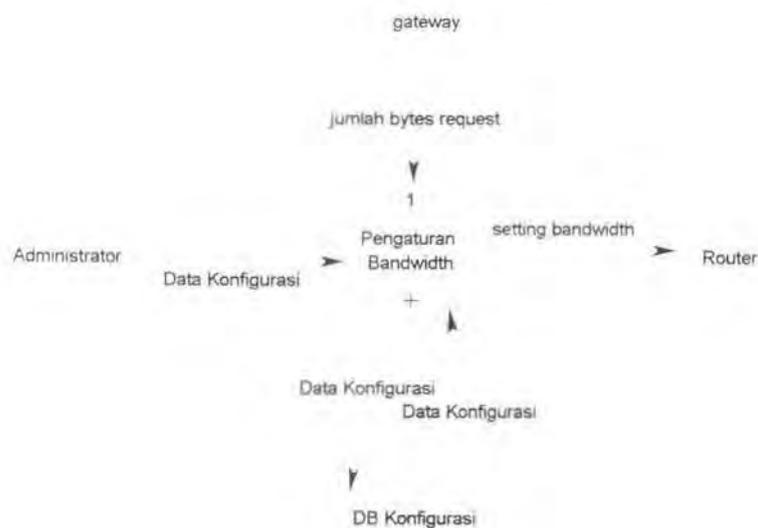
Perancangan perangkat lunak yang akan dijelaskan pada sub bab ini terdiri dari dua bagian yaitu perancangan sistem yang akan menjelaskan rancangan alur dari perangkat lunak dan kedua perancangan database yang akan menjelaskan database yang dibuat beserta keterangan tabel – tabel yang ada didalamnya.

#### **3.2.1 PERANCANGAN SISTEM**

Perancangan proses dari perangkat lunak yang akan dibuat ini digambarkan dalam bentuk data flow diagram atau Diagram aliran data (DAD). Disini akan dijelaskan bagaimana urutan proses yang terjadi dalam perangkat lunak yang dibuat.

Diagram tingkat 0 (gambar 3.2) menggambarkan proses aplikasi secara

keseluruhan. User/Administrator jaringan memasukkan data konfigurasi yang diinginkan kedalam database lewat aplikasi. Setting bandwidth yang diinginkan dijalankan di router yang digunakan/terhubung ke jaringan. Untuk mengetahui tingkat penggunaan/beban jaringan yang mengajukan request data ke internet perangkat lunak mengecek jumlah request data lewat gateway yang terhubung ke jaringan.

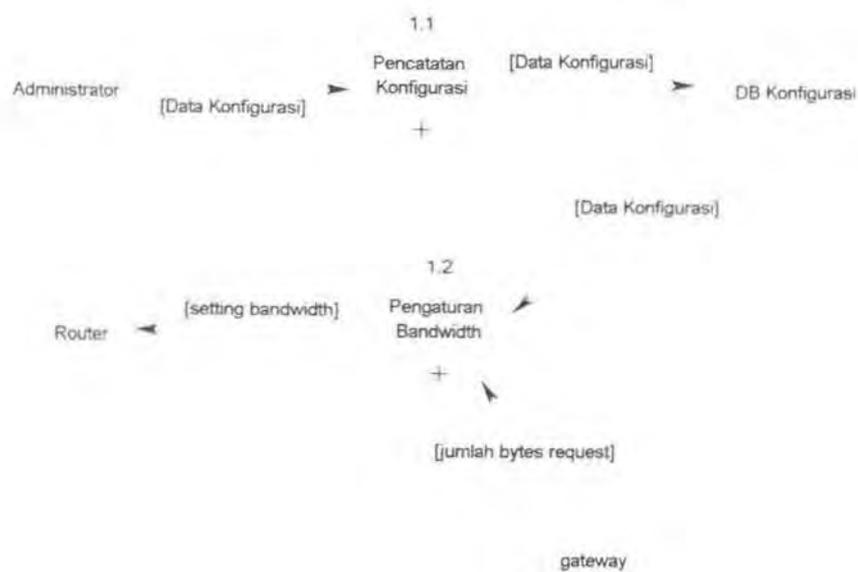


Gambar 3.2 DAD Level 0

Pada DAD level 1 (gambar 3.3) digambarkan bahwa proses secara keseluruhan terbagi menjadi dua, yang pertama proses pencatatan konfigurasi yang diinginkan, kemudian data konfigurasi tersebut disimpan dalam database. Data dimasukkan oleh administrator jaringan. Pada proses ini administrator juga

bisa mengaktifkan atau me-non aktifkan konfigurasi yang sudah didaftar/disimpan dalam database.

Kemudian proses kedua adalah proses menjalankan setting bandwidth ke router yang dilakukan secara otomatis oleh aplikasi berdasarkan data konfigurasi yang ada di database.



Gambar 3.3 DAD Level I



Gambar 3.4 DAD Level 2 sub proses pengaturan bandwidth

Pada proses ini program mengambil data konfigurasi dari database. Kemudian dilakukan klasifikasi traffic berdasarkan subnetwork jaringan atau bisa berdasarkan IP dan Port tujuan, IP dan Port Asal. Proses pengalokasian besar bandwidth ke router untuk masing – masing traffic tersebut didasarkan pada dua kondisi. Yang pertama perangkat lunak mengecek besar beban jaringan dengan menjalankan perintah pembacaan data request yang masuk di gateway. Pengecekan ini dilakukan secara berkala (setiap 5 menit sekali). Jika beban server melampaui batas yang ditentukan maka besar bandwidth di seluruh kelas traffic dialokasikan sebesar nilai default. Nilai default ditentukan oleh administrator jaringan. Sedangkan jika beban server masih dibawah batas yang ditentukan maka pengalokasian bandwidth untuk masing – masing kelas traffic dilakukan berdasarkan data konfigurasi yang ada di database dan berlaku pada jam – jam tertentu berdasarkan data konfigurasi tersebut.

### 3.2.2 PERANCANGAN DATABASE

Untuk menyimpan data – data yang diperlukan dalam menjalankan setting router sesuai dengan kebutuhan dibuat database dengan MySQL. Database ini mempunyai tabel – tabel antara lain tabel subnetwork, tabel Profile Router dan tabel Konfigurasi.

Tabel subnetwork berisi data – data dari subnetwork yang terhubung dalam jaringan. Dalam hal ini yaitu subnetwork yang terhubung di jaringan TC-net.

Berikut ini adalah field – field yang ada pada tabel subnetwork :

Tabel 3.1 Daftar Field Tabel Subnetwork

Field	Tipe Data
Kode_subnet	Text
Nama_subnet	Text
IP_subnet	Text
Netmask_subnet	Text

#### Keterangan fungsi field :

Kode\_subnet : untuk mencatat kode dari subnetwork

Nama\_subnet : untuk mencatat nama dari subnetwork yang ada di jaringan

contoh : Lab AJK, Lab RPL dll

IP\_subnet : untuk mencatat nomer IP untuk subnetwork yang bersangkutan

Netmask\_subnet : untuk mencatat netmask yang berlaku untuk subnetwork

bersangkutan.

Tabel Profile Router berisi data – data dari router yang digunakan dalam jaringan dan yang akan diset komponennya (dalam kasus ini komponen yang diset adalah besar bandwidth). Meskipun dalam tugas akhir ini dibatasi hanya digunakan satu jenis router (Corebuilder 3500) namun dalam perkembangannya ada kemungkinan pada lingkungan implementasi (TC-net) digunakan router yang lain sehingga dibutuhkan tabel profile router ini untuk mencatat router yang digunakan dalam lingkungan implementasi.

Berikut ini adalah field – field yang ada pada tabel profile router :

Tabel 3.2 Daftar Field Tabel Profile Router

Field	Tipe Data
ID_Router	Text
Nama_router	Text
File_Awal	Text
File_Akhir	Text
IP_router	Text
Prompt	Text
Login	Text
Password	Text

**Keterangan fungsi field :**

- ID\_Router : untuk mencatat ID dari router yang digunakan
- Nama\_Router : untuk mencatat nama router yang digunakan
- File\_Awal : untuk mencatat nama dan path filetext yang mencatat

command – command router untuk menjalankan perintah pengklasifikasian traffic dan kontrol untuk masing – masing kelas traffic tersebut.

- File\_Akhir : untuk mencatat nama dan path filetext yang mencatat command – command router untuk menjalankan perintah penghapusan kontrol kelas – kelas traffic
- IP\_Router : untuk mencatat nomer IP dari router yang bersangkutan.
- Prompt : untuk mencatat bentuk prompt dari command line interface (CLI) milik router yang bersangkutan
- Login : mencatat login yang berlaku di mesin router yang bersangkutan
- Password : mencatat password yang berlaku di router yang bersangkutan

Sedangkan tabel konfigurasi berisi data klasifikasi traffic yang ingin diterapkan dan kontrol untuk klasifikasi tersebut, misalnya subnetwork mana yang akan dialokasikan bandwidthnya, jam berapa setting tersebut dijalankan dsb.

Berikut ini adalah field – field yang ada pada tabel profile router :

Tabel 3.3 Daftar Field Tabel Konfigurasi

Field	Tipe Data
Kode_konf	Text
Kode_subnet_asal	Text
Port_asal	Text
Kode_subnet_tujuan	Text

Port_tujuan	Text
Besar	Text
Jam_awal	Time
Jam_akhir	Time
Is_aktif	YesNo

**Keterangan fungsi field :**

- Kode\_konf : mencatat kode konfigurasi yang masuk di database dan berfungsi sebagai primary key.
- Kode\_subnet\_asal : mencatat kode subnetwork dari asal data yang akan dibatasi bandwidthnya
- Port\_asal : mencatat port asal data yang akan dibatasi bandwidthnya
- Kode\_subnet\_tujuan : mencatat kode subnetwork dari tujuan data dikirim yang akan dibatasi bandwidthnya
- Port\_tujuan : mencatat port tujuan data dari subnetwork tujuan.
- Besar : mencatat besar bandwidth yang dialokasikan, dalam satuan Kbytes/Sec
- Jam\_awal : mencatat jam dimulainya konfigurasi ini dijalankan.
- Jam\_akhir : mencatat jam konfigurasi ini di akhiri/tidak dijalankan.
- Is\_aktif : untuk mencatat apakah konfigurasi ini masih diberlaku atau tidak (aktif/non aktif).

### 3.3 PEMBUATAN PERANGKAT LUNAK

Pada proses pembuatan perangkat lunak terdiri dari tiga bagian yaitu pembuatan antarmuka perangkat lunak, perangkat lunak pencatatan data konfigurasi dan yang terakhir perangkat lunak untuk penyettingan bandwidth/data rate itu sendiri.

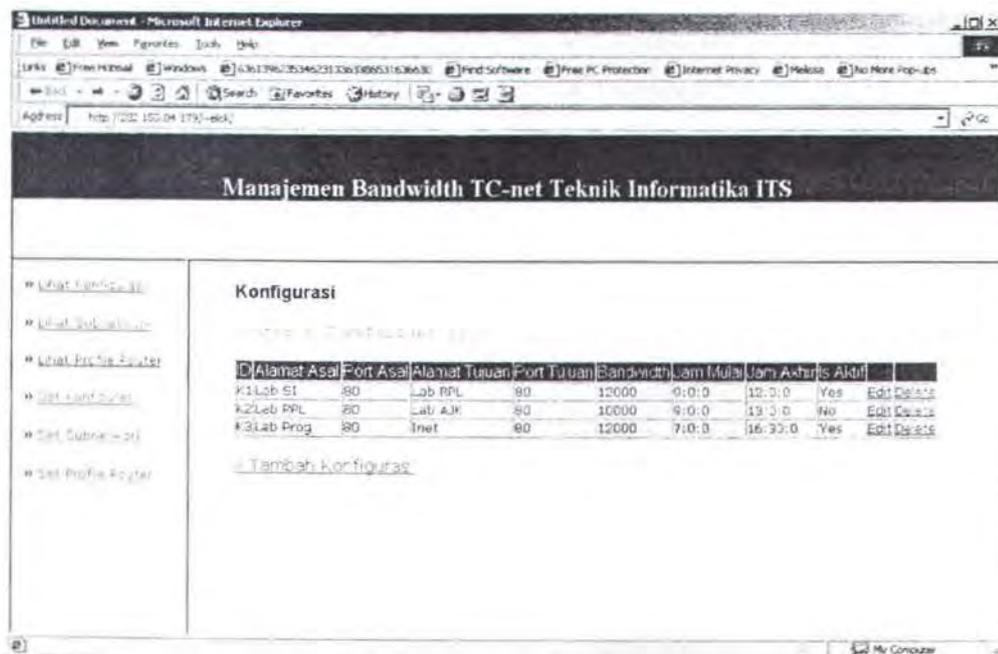
#### 3.3.1 PEMBUATAN ANTARLUKA

Berikut ini adalah rancangan antarmuka yang digunakan untuk mencatat, menghapus, mengedit dan menampilkan konfigurasi, router yang digunakan, dan subnetwork yang terhubung dengan jaringan.

##### 1. Form Lihat Konfigurasi

Form ini digunakan untuk menampilkan data konfigurasi yang pernah dicatat, baik yang aktif (sekarang digunakan) maupun yang tidak aktif . Data ditampilkan dalam bentuk tabel- tabel dan dikelompokkan berdasarkan Profile router yang digunakan dalam konfigurasi tersebut.

Dari form ini user bisa mengedit dan menghapus data konfigurasi yang ada.



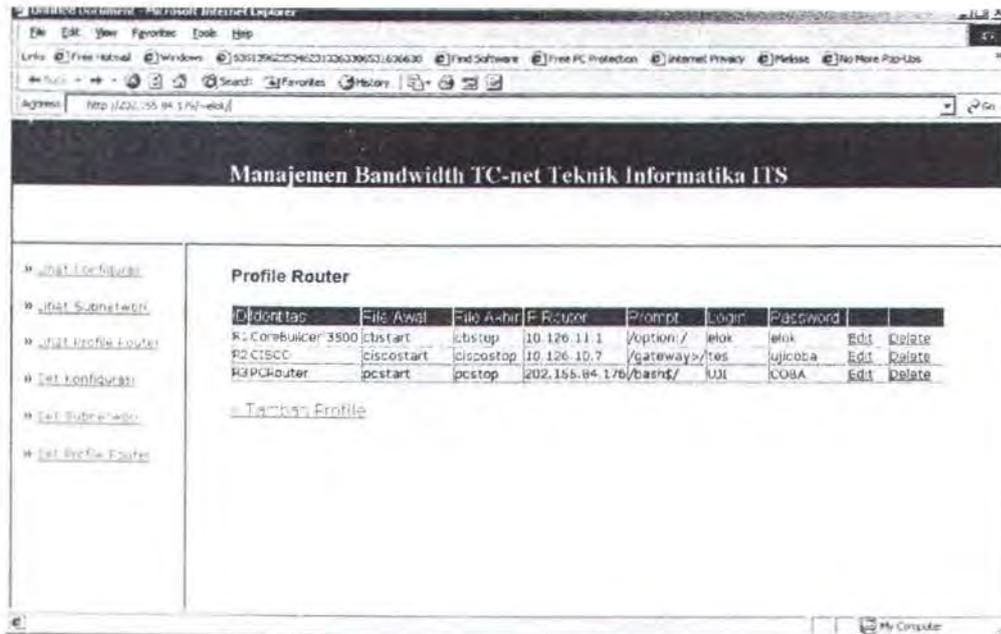
Gambar 3.5 Form Lihat Konfigurasi

## 2. Form Lihat Profile Router

Form ini digunakan untuk menampilkan data router yang bisa digunakan.

Dari form ini user bisa mengedit dan menghapus data profile router yang ada.

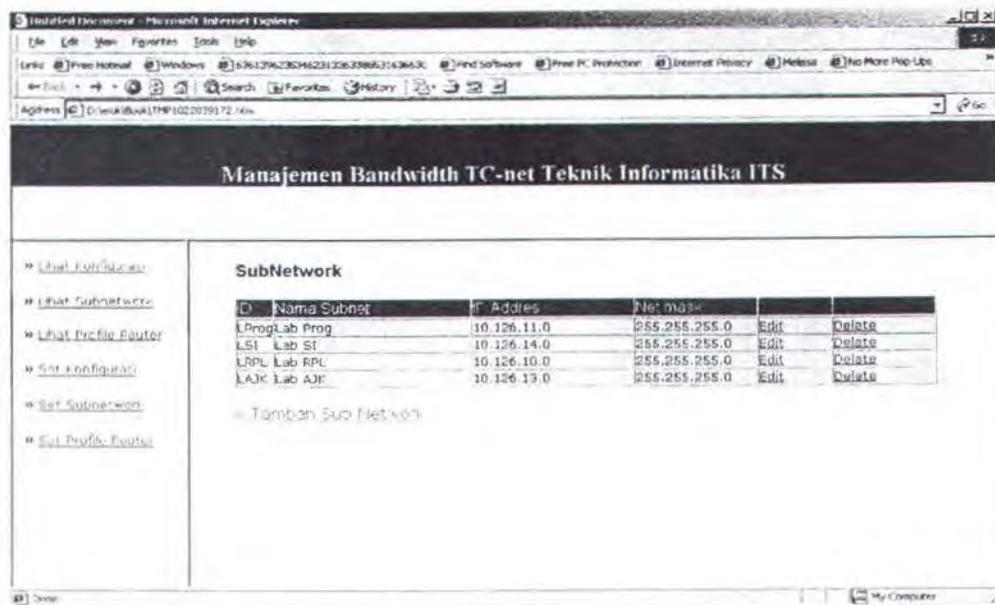




Gambar 3.6 Form Lihat Profile Router

### 3. Form Lihat Subnetwork

- Form ini digunakan untuk menampilkan data subnetwork yang terdaftar dalam jaringan TC-net. Dari form ini user bisa mengedit dan menghapus data subnetwork yang ada.



Gambar 3.7 Form Lihat data subnetwork

#### 4. Form tambah data konfigurasi

Form ini digunakan untuk menambahkan data konfigurasi ke dalam database.

Data disimpan dalam tabel Konfigurasi.

The screenshot shows a web browser window with the following details:

- Browser Title:** Manajemen Bandwidth TC-net Teknik Informatika ITS
- Address Bar:** http://100.155.84.173/~wks/
- Form Title:** Set Konfigurasi baru
- Form Fields:**
  - ID: [Text Input]
  - Alamat Asal: [Text Input] (Dropdown: Lat A...)
  - Port Asal: [Text Input]
  - Alamat Tujuan: [Text Input] (Dropdown: Lat P...)
  - Port Tujuan: [Text Input]
  - Bandwidth: [Text Input] /bps
  - Jam Mulai: [Dropdown: 12] [Dropdown: 00] [Dropdown: 00]
  - Jam Berakhir: [Dropdown: 12] [Dropdown: 00] [Dropdown: 00]
  - ID Profile: [Dropdown: C...-E...-icer 3500]
  - IsAktif:
- Buttons:** Submit, Reset

Gambar 3.8 Form tambah data konfigurasi

##### 5. Form tambah data Profile Router

Form ini difungsikan untuk mencatat data profile router yang digunakan ke dalam database.

The screenshot shows a web browser window with the following content:

- Page Title:** Manajemen Bandwidth TC-net Teknik Informatika ITS
- Navigation Menu:**
  - ↳ Lihat konfigurasi
  - ↳ Lihat Subnetwork
  - ↳ Lihat Profile Router
  - ↳ Set konfigurasi
  - ↳ Set Subnetwork
  - ↳ Set Profile Router
- Main Form: Set Router baru**
  - ID:
  - Nama Router:
  - File Awal:
  - File Akhir:
  - IP Router:
  - Prompt:
  - Login:
  - Password:
  - Buttons:

Gambar 3.8 Form menambah data profile router

#### 6. Form tambah data subnetwork

Form ini digunakan untuk menambahkan data subnetwork baru yang terhubung ke jaringan TC-net ke dalam database

The screenshot shows a web browser window with the title "Manajemen Bandwidth TC-net Teknik Informatika ITS". The address bar shows "http://202.155.84.179/~eou". The main content area contains a form titled "Set Sub Network baru". The form has the following fields and buttons:

- ID:
- Nama Subnet:
- IP subnet:
- Netmask:
- Buttons:

On the left side, there is a navigation menu with the following items:

- » Lihat Konfigurasi
- » Lihat Subnet word
- » Lihat Profile Router
- » Set Konfigurasi
- » Set Subnet baru
- » Set Profile Router

Gambar 3.9 Form tambah data subnetwork

### 3.3.2 PEMBUATAN PROGRAM PNCATATAN KONFIGURASI

Program pencatatan dan menampilkan data konfigurasi, subnetwork dan profile router yang digunakan ke dalam database dibuat dengan menggunakan bahasa pemrograman PHP.

Berikut ini adalah salah satu prosedur pencatatan data yaitu prosedur untuk mencatat data – data profile router ke dalam database :

```
<?
$username="radio";
$password="radio";
$host="202.155.84.179";
mysql_connect("$host", "$username", "$password");
mysql_select_db("ProjectTA");
$per="insert into Profile values('$RutID',
'$Identitas', '$Fname', '$RutIP',
'$Fnameak', '$Prom', '$log', '$pwd)";
$result = mysql_query($per);
```

```
mysql_close();
?>
```

Berikut ini adalah prosedur untuk menampilkan data tersebut ke antarmuka yang sudah dirancang diatas sehingga user bisa melihat semua informasi yang sudah disimpan .

```
<?
$username="radio";
$password="radio";
$host="202.155.84.179";
mysql_connect("$host", "$username", "$password");
mysql_select_db("ProjectTA");
$hasil=mysql_query("select * from Profile");
while($data=mysql_fetch_row($hasil)){
    echo("<tr><td>$data[0]</td><td>$data[1]</td>
        <td>$data[2]</td><td>$data[3]</td>
        <td>$data[4]</td><td>$data[5]</td>
        <td>$data[6]</td><td>$data[7]</td>
        <td> &nbsp;<a href=a>Edit</a>&nbsp;<a href=hapus.php?str=$data[0]&db=");
    }
mysql_close();
?>
```

untuk proses pengeditan data digunakan form pengisian data. Setiap kali terjadi pengeditan dan penghapusan data di database maka program PHP ini memberhentikan dan menjalankan ulang program daemon (program penyettingan bandwidth), sehingga data yang dibaca oleh program daemon dari database selalu sesuai dengan data yang terbaru.

### 3.3.3 PEMBUATAN PROGRAM PENSETTINGAN BANDWIDTH

Untuk program penyettingan router ini digunakan bahasa pemrograman

CGI-Perl. Ide dasar dari program ini adalah mengotomatisasi proses yang biasa dijalankan oleh user/administrator jaringan dalam mengalokasikan bandwidth untuk subnetwork.

Sebelum mengalokasikan bandwidth yang ada, program mengambil data konfigurasi yang sudah dimasukkan oleh administrator jaringan dan disimpan dalam database. Data ini disimpan dalam suatu variable array sehingga pengambilan data dari database hanya perlu dilakukan sekali saja yaitu ketika program pertama kali dijalankan.

Perintah untuk pengambilan data dari database adalah sebagai berikut :

```

use DBI;
my($dsn) = "DBI:mysql:ProjectTA:202.155.84.179";
my($user_name) = "radio";
my($password) = "radio";
$dbh = DBI -> connect($dsn, $user_name, $password);
$sth = $dbh -> prepare("select * from Konfigurasi where
IsAktif='Y'");
$sth -> execute;
$jmlklass = 0;
while (@dta = $sth -> fetchrow_array())
{
    $baris = join(" ", @dta);
    $hasil[$jmlklass] = $baris;
    $jmlklass++;
}
$sth -> finish();

$countl = 0;
while ($countl < $jmlklass)
{
    @baris = split(" ", $hasil[$countl]);
    $sth = $dbh -> prepare("select NetAddr, NetMask from
NetTable
        where NetID=$baris[1]");
    $sth -> execute;
    while (@dta = $sth -> fetchrow_array())
    {
        $baris = join(" ", $baris, @dta);
        $hasil[$countl] = $baris;
    }
}

```

```

    $sth -> finish();
    $count1++;
}

```

Setelah pembacaan data dari database selanjutnya dilakukan klasifikasi traffic berdasarkan data konfigurasi dan data subnetwork yang sudah diambil dari database tersebut.

Untuk melakukan klasifikasi ini perlu dilakukan login ke mesin router (Core Builder 3500). Untuk masuk/login kedalam mesin router dan menjalankan perintah/command – command router, bisa menggunakan aplikasi telnet. Aplikasi telnet menggunakan command line interface untuk menjalankan perintah – perintah yang ada. Dalam program digunakan modul Net::Telnet untuk memasukkan perintah lewat command line interface telnet dan menangkap hasilnya sama seperti jika kita menjalankan perintah telnet langsung ke router.

Perintah untuk login ke mesin router adalah sebagai berikut :

```

use Net::Telnet();
$buka = new Net::Telnet(Timeout=>10, Prompt =>'/:/');
$buka -> open("10.126.11.1");
$buka -> login("write", "tuliskan226");

```

Setelah berhasil dilakukan login, selanjutnya akan dijalankan command – command classifier dari mesin router. Pada router Corebuilder 3500 command – command untuk melakukan klasifikasi adalah sebagai berikut :

```

Qos
Classifier
Define
3
→ nomer klasifikasi

```

tesclassifier	→ nama klasifikasi
all	→ tipe cast (unicast/multicast)
all	→ tipe IP protocol (TCP/UDP)
10.126.11.0	→ Alamat IP asal
255.255.255.0	→ Mask Alamat IP asal
10.126.12.0	→ Alamat IP tujuan
255.255.255.0	→ Mask Alamat IP tujuan
0	→ nilai awal range port asal
23	→ nilai akhir range port asal
0	→ nilai awal range port tujuan
23	→ nilai akhir range port tujuan
no	
q	
q	

Perintah/command – command router untuk melakukan klasifikasi traffic tersebut sebelumnya sudah dicatat dalam sebuah file teks. Program membaca command – command tersebut dari file teks dan mengirimkan ke mesin router baris per baris dan menangkap hasilnya. Prosedur pembacaan dan pengiriman command – command tersebut adalah sebagai berikut :

```

if (open(isifile,"cbstart"))
{
    $n=0;
    $isi[$n]=<isifile>;
    while($isi[$n] ne "")
    {
        @lines = $buka -> cmd($isi[$n]);
        $n++;
        $isi[$n]=<isifile>;
    }
}
$ok = $buka->close;

```

Setelah berhasil dilakukan klasifikasi traffic, selanjutnya dilakukan kontrol terhadap klasifikasi traffic tersebut, yaitu pengalokasian besar bandwidth untuk

masing – masing kelas traffic.

Seperti dijelaskan dalam perancangan sistem, proses pemberian kontrol ini didasarkan pada dua kondisi, yaitu yang pertama berdasarkan beban jaringan yang ada dan yang kedua berdasarkan data konfigurasi di database. Untuk mengetahui besar beban jaringan diambil dari banyaknya data request yang masuk lewat gateway (202.155.84.177). Pembacaan ini dilakukan setiap lima menit sekali untuk memastikan perubahan beban jaringan. Sama halnya dengan proses klasifikasi traffic, untuk mengetahui besar data request yang masuk ke gateway program melakukan login terlebih dahulu ke gateway ( 202.155.84.177 ). Kemudian menjalankan perintah “*netstat - i*”.

Perintah untuk login ke gateway dan melakukan pengambilan data request serta memasukkan hasil pembacaan kedalam suatu variable adalah sebagai berikut :

```
use Net::Telnet();
$buka = new Net::Telnet(Timeout=>10, Prompt =>'/>\$
$/');
$buka -> open("202.155.84.177");
$buka -> login("roy", "123");
@lines = $buka -> cmd("netstat -i");
@hasil = split(" ", $lines[2]);
$ok = $buka->close;
```

Jika jumlah data request yang masuk diatas batas yang ditetapkan maka besar bandwidth untuk masing – masing klasifikasi traffic dialokasikan sesuai nilai default.

Untuk keperluan tersebut program melakukan login ke mesin router dan

selanjutnya akan dijalankan command – command control classifier dari mesin router.

Pada router Corebuilder 3500 contoh command – command untuk melakukan suatu kontrol terhadap suatu kelas traffic adalah sebagai berikut :

```

Qos
Control
Define
3          → nomer kontrol
tescontrol → nama kontrol
none       → tipe rate limit
           (none/aggregate/receivePort)
low        → jenis servis untuk conforming paket
           (high,best,low)
no
low        → jenis servis untuk excess paket
           (high,best,low,drop)
no
Kbytes/sec → Jenis ekspresi limit rate
           (Kbytes/sec,percentage)
2          → Nilai limit rate
16         → Nilai burst (dalam Kbytes)
all        → Port yang dikenai kontrol
no
no
no
3          → nomer klasifikasi yang dikenai
kontrol
q
q

```

Program membaca command – command tersebut dan mengirimkan ke mesin router. Pada kondisi ini hanya perlu dibuat satu pengontrol dan digunakan untuk mengontrol semua kelas traffic.

Jika kondisi pertama tidak terjadi (beban jaringan dalam kondisi normal/

dibawah batas) maka pengaturan bandwidth didasarkan pada konfigurasi yang sudah dicatat dalam database.

Program membaca data konfigurasi yang sudah disimpan dalam variabel. Pada kondisi kedua ini kontrol hanya berlaku pada jam – jam tertentu saja. Dari semua klasifikasi dicek klasifikasi mana yang harus dikontrol pada jam tersebut. Karena satu kelas traffic dalam waktu yang sama hanya bisa berlaku satu pengontrol saja maka sebelum dibuat kontrol baru untuk masing – masing kelas traffic tersebut, harus dicek terlebih dahulu apakah kontrol default sedang dijalankan atau tidak. Jika sedang dijalankan kontrol default dihapus dengan menjalankan perintah remove kontrol

```
Qos
Control
Remove
3           → nomer kontrol yang dihapus
q
q
```

baru setelah dipastikan tidak ada pengontrol yang berlaku untuk kelas traffic tersebut, dibuatkan pengontrol baru sesuai dengan kondisi yang berlaku untuk kelas traffic tersebut. Sehingga pada kondisi ini satu klasifikasi traffic mempunyai pengontrol sendiri – sendiri. Pembuatan kontrol untuk masing – masing klasifikasi tidak berbarengan namun tergantung jam mulai berlakunya kontrol dan jam berakhirnya kontrol digunakan.

Perintah/command – command router untuk melakukan kontrol klasifikasi traffic tersebut sebelumnya sudah dicatat dalam sebuah file teks. Program

membaca command – command tersebut dari file teks dan mengirimkan ke mesin router baris per baris dan menangkap hasilnya. Prosedur/perintah pembacaan dan pengiriman command – command tersebut sama dengan prosedur/perintah pembacaan dan pengiriman command – command untuk klasifikasi traffic.

Selain mengecek kontrol mana yang harus pengontrolnya diaktifkan, program juga melakukan pengecekan terhadap semua klasifikasi untuk mengetahui klasifikasi mana yang kontrolnya harus dimatikan/dinonaktifkan pada jam tersebut dengan menjalankan perintah remove kontrol yang bersangkutan.

Prosedur pengecekan data request sampai proses pengecekan kontrol aktif/nonaktif dijalankan secara terus – menerus, sehingga beban server terus terpantau dan terkendali dan administrator hanya perlu sewaktu – waktu untuk mengecek atau merubah konfigurasi.

## BAB IV

### UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan data yang akan diujicobakan dalam perangkat lunak yang dibuat kemudian dilakukan evaluasi terhadap hasil yang diperoleh dari uji coba tersebut

#### 4.1 DATA UJI COBA

Untuk menguji dan mengevaluasi perangkat lunak diambil beberapa kasus yang berbeda dengan data ujicoba adalah sebagai berikut :

1. Data yang dimasukkan dalam database dan masing – masing tabel adalah sebagai berikut :

- Tabel Konfigurasi

Pada table ini dimasukkan dua data, data yang pertama yaitu :

FIELD	NILAI
Kode_konf	2
Kode_subnet_asal	1
Port_asal	20
Kode_subnet_tujuan	2
Port_tujuan	20
Besar	3
Jam_awal	19:00
Jam_akhir	02:00
Is_aktif	Yes

Data yang kedua :

FIELD	NILAI
Kode_konf	3
Kode_subnet_asal	2
Port_asal	21
Kode_subnet_tujuan	1
Port_tujuan	21
Besar	3
Jam_awal	21:00
Jam_akhir	02:00
Is_aktif	Yes

- Tabel Profile Router

FIELD	NILAI
ID_Router	CB3500
Nama_router	COREBUILDER 3500
File_Awal	CBSTART
File_Akhir	CBSTOP
IP_router	10.126.11.1
Prompt	:/
Login	Write
Password	tulis226

- Tabel Subnetwork

Kode_subnet	Nama_subnet	IP_subnet	Netmask_subnet
1	Lab Prog	10.126.11.200	255.255.255.255

2	R-Dosen	202.155.84.179	255.255.255.255
---	---------	----------------	-----------------

2. Ketentuan yang digunakan dalam ujicoba :

- Jika jumlah data request yang masuk pada gateway diberi batas paling tinggi sebesar 300 kb/sec.
- Pembacaan data pada gateway dilaksanakan tiap lima menit sekali.
- Kontrol default untuk semua klasifikasi ditetapkan besar batas data rate sebesar 2 Kbytes/sec.

#### 4.2 PELAKSANAAN UJI COBA

Sesuai dengan data yang telah dimasukkan dalam database dilakukan ujicoba dengan menggunakan aplikasi Pengaturan Bandwidth. Ketika program dijalankan didapat hasil sebagai berikut :

Klasifikasi traffic yang dibuat ada 2 kelas traffic, karena data konfigurasi yang dimasukkan ada dua kelas traffic:

1. nomer klasifikasi : 2

nama klasifikasi : klass2

alamat IP asal : 10.126.11.200

netmask IP asal : 255.255.255.255

alamat IP tujuan : 202.155.84.179

netmask IP tujuan : 255.255.255.255

Port asal : 20

Port tujuan : 20

2. nomer klasifikasi :3

nama klasifikasi : klass3

alamat IP asal : 202.155.84.179

netmask IP asal : 255.255.255.255

alamat IP tujuan : 10.126.11.200

netmask IP tujuan : 255.255.255.255

Port asal : 21

Port tujuan : 21

Pada data ujicoba ditetapkan jika rata – rata data request digateway diatas 300 kb/sec, dianggap beban server tinggi. Setelah program dijalankan didapatkan nilai rata – rata data request sebesar 1020 kb/sec, sehingga program membuat kontrol default yang berlaku untuk semua klasifikasi yang ada. Hasilnya ketika dilihat pada router Corebuilder adalah sebagai berikut:

```

Type "q" to return to the previous menu or ? for help
-----
Select menu option (qos/control) : suk
-----
Control      Loss      Excess      Excess      802-1p
Name         Service  Eligible    Service     Eligible    Tag    Controlled
1 Default/Best Effort  high  no         best        yes       --       499
50 kontrolutama      low   no         best        yes       --       23
-----
Menu options (RC):
summary      - Display summary information
detail       - Display detail information
define       - Define a new control
modify       - Modify an existing control
remove       - Remove an existing control
-----
Type "q" to return to the previous menu or ? for help
-----
Select menu option (qos/control) :

```

Gambar 4.1 Hasil ujicoba 1

Kontrol default diberi nomer 50 dan nama kontrol "kontrolutama", kontrol berlaku/digunakan untuk mengontrol klasifikasi nomer 2 dan 3 yang telah dibuat diatas.

Pada jam 16:00 dari data konfigurasi diinginkan besar limit data rate dari subnet Lab Prog ke R. Dosen adalah sebesar 3Kb/sec. Tetapi karena pada jam tersebut data request pada gateway masih melebihi 300 kb/sec (ketika dicek nilainya sekitar 1400 kb/sec, maka program tidak membuat kontrol tersebut, sehingga kontrol yang berlaku untuk klasifikasi tersebut (nomer 2) masih tetap menggunakan kontrol default.

Pada jan 21:00 dari data konfigurasi diinginkan besar limit data rate dari subnet R. Dosen ke subnet Lab Prog adalah sebesar 3Kb/sec. Ternyata nilai data request pada jam tersebut turun (rendah) sehingga program membuat kontrol untuk klasifikasi tersebut. Sebelumnya kontrol default yang ada dihapus terlebih dahulu. Karena jika kontrol default masih ada, kontrol yang baru tidak bisa

mengontrol klasifikasi nomer 3. Akibat dari penghapusan kontrol default untuk klasifikasi nomer dua tidak dikontrol. Hasilnya ketika dilihat pada router Corebuilder adalah sebagai berikut :

```

Type "q" to return to the previous menu or ? for help.
Select menu option (qos/control) sum
Control
  Name          Service  Loss   Excess  Excess  Loss  802 1p Classifier
 1 Default/Best Effort high    no     best   yes   --   499   Controlled
 3 kontrol3     low     no     low    no    --   3
Menu options (RC)
summary      - Display summary information
detail      - Display detail information
define      - Define a new control
modify      - Modify an existing control
remove      - Remove an existing control
Type "q" to return to the previous menu or ? for help.
Select menu option (qos/control)

```

Gambar 4.2 Hasil ujicoba 2

Pada jam 22:00 ditetapkan bahwa klasifikasi nomer 2 dan 3 tidak di atur besar limit data ratenya dengan kata lain kontrol yang berlaku untuk klasifikasi tersebut harus dihapus. Karena sebelumnya kontrol nomer 2 tidak dibuat, maka program hanya menghapus kontrol nomer 3. Ketika dicek di gateway jumlah data request dikategorikan rendah (dibawah 300 kb/sec) maka program tidak menjalankan kontrol default. Sehingga setelah jam tersebut klasifikasi nomer dua dan tiga tidak dikontrol/ tidak dibatasi data ratenya.

## **BAB V**

### **PENUTUP**

Dalam bab V ini akan dijelaskan kesimpulan dan kemungkinan pengembangan lebih lanjut dari perangkat lunak yang dibuat.

#### **5.1 KESIMPULAN**

Beberapa kesimpulan yang dapat ditarik dari pembuatan tugas akhir ini dan pelaksanaan ujicoba adalah sebagai berikut :

1. Program mampu membaca besar data masuk pada gateway setiap lima menit sekali.
2. Program mampu membuat klasifikasi traffic berdasarkan ketentuan yang dimasukkan dalam database oleh administrator jaringan.
3. Program mampu mengontrol klasifikasi traffic yang sudah dibuat tersebut berdasarkan kondisi jumlah data yang ada pada gateway dan berdasarkan jam – jam yang sudah ditetapkan oleh administrator jaringan

#### **5.2 KEMUNGKINAN PENGEMBANGAN**

Kemungkinan pengembangan yang bisa dilakukan untuk aplikasi ini supaya lebih berguna adalah :

1. Router yang digunakan tidak terbatas pada router CoreBuilder 3500

2. komponen router yang diset tidak hanya masalah bandwidth saja.
3. Program diterapkan di jaringan lain selain TC-net, dengan mengikuti konfigurasi jaringan yang lain.

## DAFTAR PUSTAKA

1. William Stallings, Data and Computer Communication, Prentice Hall International, Inc, 1994
2. Behrouz A. Forouzan , TCP/IP Protocol Suite, Behrouz A. Forouzan, McGraw Hill, 2000
3. Jay Rogers, Net::Telnet –interact with TELNET port or other TCP ports, [http://www.perldoc.com/cpan/Net\\_Telnet.html](http://www.perldoc.com/cpan/Net_Telnet.html), 2000

