BAB IV

IMPLEMENTASI DAN EVALUASI

Pada bab ini penulis akan menjelaskan mengenai implementasi dan evaluasi serta analisa dari aplikasi pembatasan dan pengatruan bandwidth dengan CBQ (*Class Based Queueing*) pada sistem operasi Linux berbasis web.

4.1 Kebutuhan Sistem

Sebelum melakukan implementasi dan menjalankan program pembatasan dan pengaturan bandwidth dengan menggunakan CBQ (*Class Based Queueing*) pada sistem operasi linux berbasis web, terlebih dahulu harus komponen-komponen utama komputer yang mendukung setiap proses harus sudah terpasang. Berikut ini adalah perangkat keras (*hardware*) yang dibutuhkan untuk mengimplementasi aplikasi pembatasan dan pengaturan bandwidth dengan CBQ (*Class Based Queueing*) pada sistem operasi Linux berbasis web antara lain :

1. PC Pentium III 500 Mhz atau lebih.
2. Memori RAM 128 Mb atau lebih
3. Hard disk 10 Gb atau lebih
4. VGA 4 Mb.
5. Monitor SVGA 14".

Sedangkan perangkat lunak (*software*) yang diperlukan agar dapat berjalan sesuai dengan yang diharapkan antara lain :

1. Sistem Operasi Linux RedHat versi 9.0 dengan versi kernel 2.4.20
2. CBQ Script (*cbq.init-v0.7.3*)
3. Apache Web Server

4. Perl-CGI
5. PHP
6. RRDTools (*Round Robin Database Tools*)
7. JPGraph
8. Squid
9. Iptables

Sebuah mesin QoS-CBQ dapat langsung didukung oleh proxy server dengan menggunakan *squid* dan juga *firewall* menggunakan *iptables* yang juga berfungsi sebagai *router/gateway*.

4.1.1 Installasi Program

Untuk menjalankan aplikasi ini diperlukan sebuah komputer yang berfungsi sebagai router/gateway dengan 2 NIC (*Network Interface Card*) yang nantinya akan diinstal CBQ script. Sedangkat untuk aplikasi Perl-CGI, Squid, IPTables, dan MySQL bisa instal pada saat melakukan proses instalasi sistem operasi linux RedHat 9.0. Berikut langkah langkah dalam instalasi program yaitu:

1. Install Sistem Operasi Redhat Linux 9.0
2. Paket aplikasi Perl yang harus dipilih pada saat proses installasi sistem operasi Linux Redhat 9.0 antara lain:
   - perl-5.8.0-88.i386.rpm
   - perl-CGI-2.81-88.i386.rpm
   - perl-DBI-1.32-5.i386.rpm
   - perl-CPAN-1.61-88.i386.rpm
   - perl-suidperl-5.8.0-88.i386.rpm
   - perl-TimeDate-1.1301-5.noarch.rpm

- perl-Archive-Tar-0.22-29.noarch.rpm
- perl-HTML-Parser-3.26-17.i386.rpm
- perl-HTML-Tagset-3.03-28.noarch.rpm
- perl-libwww-perl-5.65-6.noarch.rpm

3. Paket aplikasi lain yang harus diinstall adalah :
   - squid-2.5.STABLE1-2.i386.rpm
   - iptables-ipv6-1.2.7a-2.i386.rpm
   - mysql-3.23.54a-11.i386.rpm
   - mysql-server-3.23.54a-11.i386.rpm
4. Setelah proses installasi linux, dilanjutkan dengan proses installasi paket Tarball antara lain :
   - httpd-2.0.54.tar
   - freetype-2.1.9.tar.gz
   - zlib-1.2.2.tar.gz
   - libpng-1.2.8-config.tar.gz
   - gd-2.0.33.tar.gz
   - php-4.3.8.tar
   - snmp-extension-0.2.0.tar.gz
   - net-snmp-5.1.4.pre1.tar.gz
   - cgilib-0.5.tar.gz
   - libart_lgpl-2.3.17.tar.gz
   - rrdtool-1.2.6.tar.gz

5. Copy script CBQ (*cbq.init-v0.7.3*) kedalam directory /etc/init.d/ dan membuat direktori tempat menyimpan kelas-kelas CBQ pada direktory /etc/sysconfig/cbq.
6. Buat direktori /qoscbq dan copy semua source kedalam direktori tersebut dan ubah kepemilikan direktori tersebut kepada user yang menangani web server apache dalam hal ini user:apache dan group apache dengan menggunakan perintah yang terdapat pada sistem operasi linux.

4.2 Implementasi Program

Implementasi program adalah implementasi dari jalannya sistem yang telah dibuat. Dengan adanya implementasi ini kita dapat memahami jalannya sistem tersebut. Berikut ini adalah penjelasan tentang implementasi dan bagian-bagian serta fungsi-fungsi dari masing masing menu dan form-form yang terdapat pada aplikasi pembatasan dan pengaturan bandwidth dengan *Class Based Queueing* (CBQ) pada sistem operasi linux berbasis web.

4.2.1 Menu Aplikasi Program

Pada saat menjalankan aplikasi ini pengguna akan diminta untuk memasukan nama user (*user name*) dan password (*password*) yang sudah terdaftar terlebih dahulu pada data aplikasi.. Hal ini berguna untuk membatasi penggunaan aplikasi agar hanya bisa digunakan oleh orang-orang tertentu dalam hal ini *Network Manager* disamping itu aplikasi ini juga didukung oleh OpenSSL untuk memberikan koneksi yang aman (*secure connection*) dengan menggunakan protokol 443:HTTPS (*Hyper Text Transfer Protocol Secure*). Untuk lebih jelasnya berikut ini adalah tampilan dari form login :

Gambar 4.1 Form Menu LOGIN

Proses login dengan menginputkan *username* dan *password* dengan benar maka tampilan form menu Home merupakan tampilan pertama. Form ini sebagai awalan (*intro*) yang berisikan tentang penjelasan singkat tentang CBQ dan informasi tentang perangkat lunak (*software*) serta informasi komputer server antara lain *hostname*, *IP Address*, dan sebagainya. Tampilan dari form Home seperti pada gambar 4.2 berikut :

Gambar 4.2 Form Home

Terdapat beberapa menu pada aplikasi ini, berikut penjelasan dari masing-masing menu yang terdapat pada aplikasi pembatasan dan pengaturan *bandwidth* dengan CBQ pada sistem operasi Linux berbasis web :

a. Menu CBQ

Menu ini digunakan untuk melakukan konfigurasi *script cbq.init* dimulai dengan membuat kelas CBQ menginputkan nilai parameter *cbq.init* pada masing-masing kelas, mengubah/mengganti parameter dari kelas maupun menghapus kelas tersebut. Pada menu ini juga proses meng-*compile* menjalankan, dan menghentikan proses *init* dari CBQ dan untuk mengetahui sejauh mana hasil konfigurasi yang dilakukan terdapat informasi *log* dan juga tabel traffik serta grafik untuk memelihat sejauh mana efek yang terjadi dengan adanya pembatasan dan pengaturan *bandwidth* pada masing-masing kelas layanan CBQ disamping itu pengguna juga dapat melihat data *history* dari besarnya traffik dari masing-masing kelas layanan CBQ..

b. Menu Support

Menu ini digunakan untuk konfigurasikan aplikasi pendukung dari dari CBQ yakni konfigurasi parameter *squid delaypools* dan konfigurasi NAT (*Network Translation Address*) dan *firewall* dengan iptables. Pada menu juga pengguna dapat melakukan administrasi aplikasi dan sistem yakni mengganti *password*, melakukan *setup* DNS-Client (*Domain Name Server*), dan juga melakukan administrasi *IP Address* dan *Routing* pada sistem.

Menu Diagnos & Log

Menu diagnosa dan Log digunakan untuk melakukan diagnosa interface dan *log system*, *log squid* dan *log history* dari iptables serta status koneksi iptables.

d. Menu Monitoring

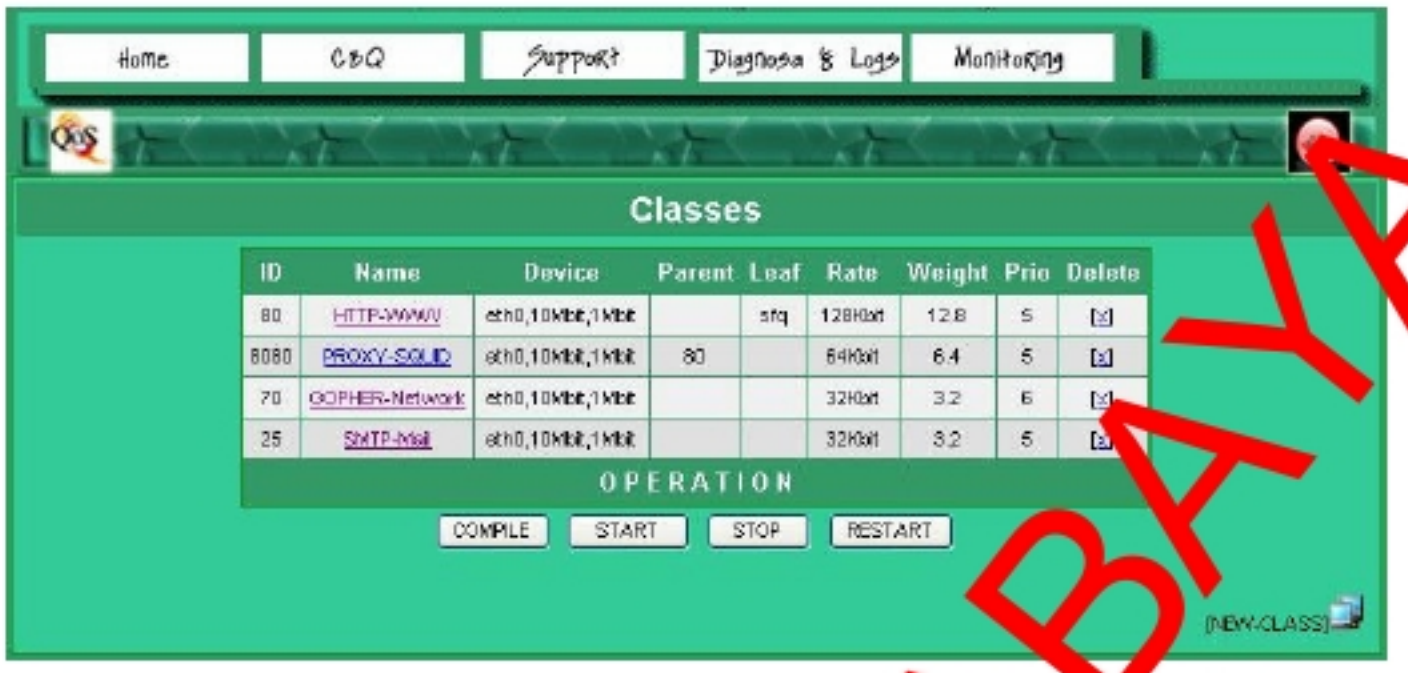
Menu ini berisi representasi grafik dari trafik rata-rata untuk koneksi keluar (*outgoing*) dan koneksi masuk (*incomming*) masing-masing *interface* (*Network Interface Card*) dengan beberapa interval waktu yang terdapat pada sistem dan juga informasi *load average* dari sistem.

A. Menu CBQ

Form-form yang terdapat pada menu CBQ antara lain sebagai berikut :

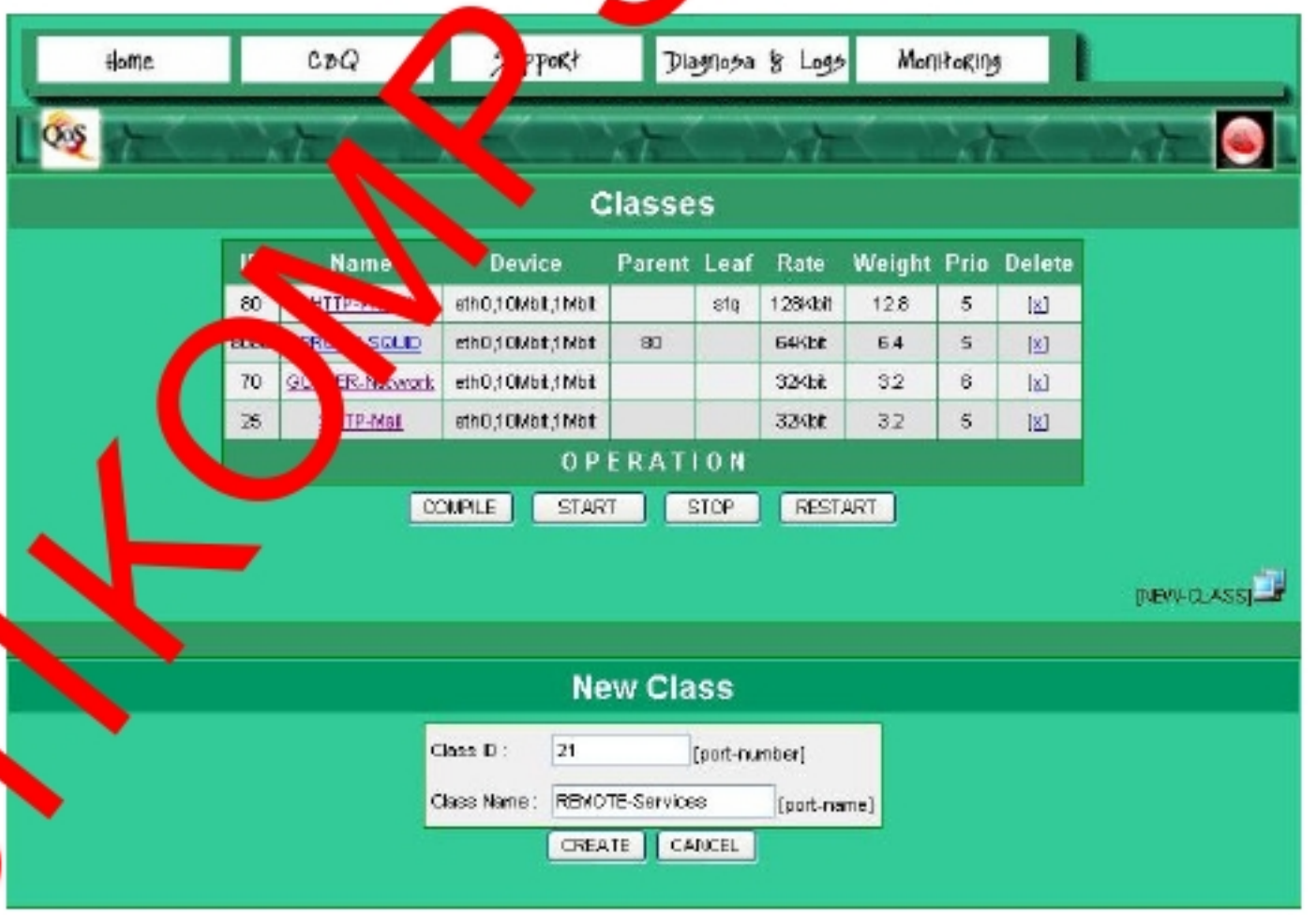
- **CBQ-Init**

Pada form Classes terdapat daftar dari kelas kelas CBQ yang telah dibuat sebelumnya dan informasi nilai parameter dari masing-masing kelas antara lain *ID, Name, Device, Parent, Leaf, Rate, Weight,* dan *Prio*. Kelas ID merupakan *Port Number* dan kelas *Name* merupakan *Port Name* dari layanan-layanan yang tersedia pada jaringan komputer. Sedangkan *field* yang terakhir yakni Delete digunakan untuk menghapus kelas yang bersangkutan jika tidak dipergunakan lagi. Pada bagian *Operation* terdapat empat buah tombol yakni *Compile*, *Start*, *Stop*, dan *Restart* digunakan untuk mengeksekusi, menjalankan, dan menghentikan proses *script cbq.init*. Tombol *Compile* dan *Restart* digunakan jika penggguna berhasil menambahkan sebuah kelas baru pada daftar kelas CBQ hal ini agar kelas tersebut dimasukan kedalam proses *script cbq.init*. Tampilan dari form Classes seperti pada gambar 4.3 berikut :



Gambar 4.3 Form Classes

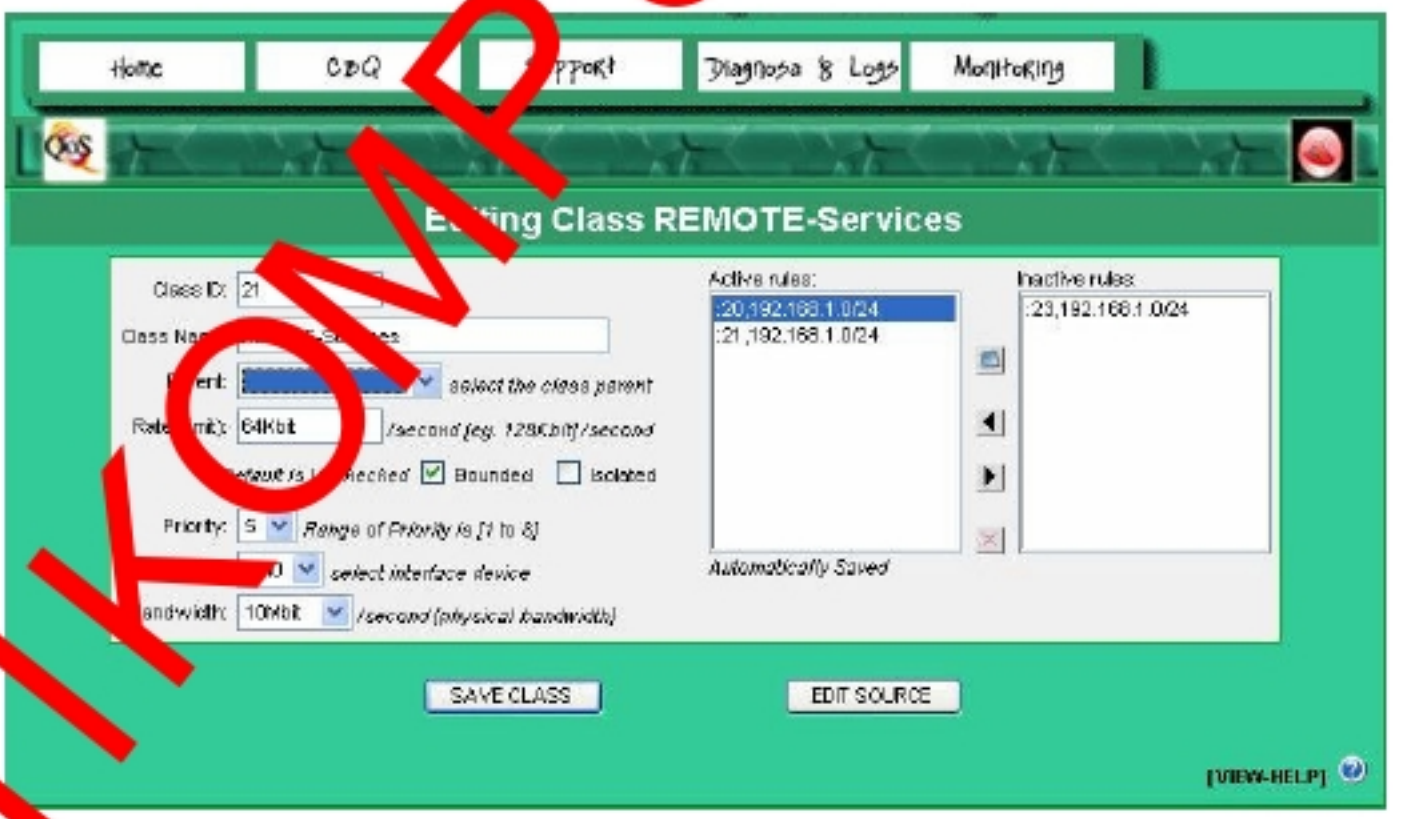
Untuk membuah kelas CBQ baru terdapat tombol *icon* NEW-CLASS, dimana jika diklik akan menampilkan form New Class yang digunakan untuk menginputkan kelas *ID* dan kelas Name dari kelas baru CBQ yang akan dibuat. Tampilan dari form New Class seperti pada gambar 4.4 berikut :



Gambar 4.4 Form New Class

Pada form New Class terdapat dua buah tombol yakni *Create* dan *Cancel*, tombol *Create* digunakan untuk membuat kelas baru CBQ sedangkan tombol *Cancel* untuk membatalkan pembuatan kelas baru. Form Edit Class digunakan untuk menginputkan parameter dari kelas CBQ yang baru dibuat, selain itu form Edit Class juga digunakan untuk merubah parameter dari kelas-kelas CBQ yang telah dibuat sebelumnya.

Parameter *parent* menentukan kelas yang baru dibuat berada dibawah kelas lain yang sudah ada sedangkan parameter *rate* menentukan besarnya kecepatan *bandwidth* dari kelas layanan ukurannya berupa Kbit/s (*Kilo bits per second*) / Kbps (*Kilo byte per second*), parameter *rate* menentukan nilai dari parameter *weigth* (weight = rate / 10). Parameter device menentukan interface dan parameter *rule* untuk menetukan kelas berlaku untuk *single host*, atau subnet (*sub network*). Tampilan dari form Edit Class seperti pada gambar 4.5 berikut :



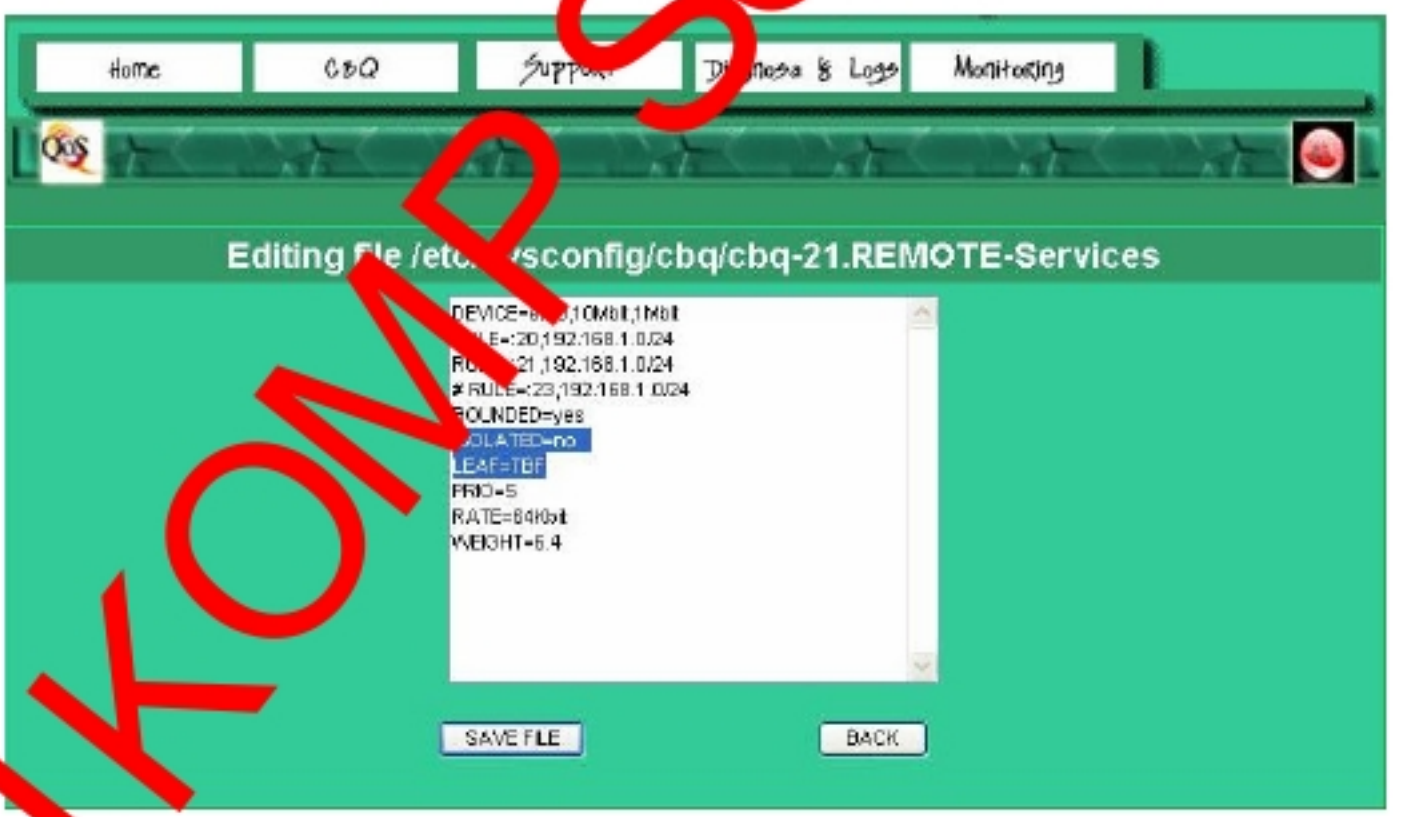
Gambar 4.5 Form Edit Class

Sedangkan tombol *icon* VIEW-HELP untuk menampilkan *manual* dari tiap-tiap fungsi dan nilai dari masing-masing parameter.

Gambar 4.6 Form Manual Parameter CBQ

Tombol *Save* Class pada form Edit Class digunakan untuk menyimpan hasil inputan dan hasil perubahan parameter kelas-kelas CBQ, sedangkan tombol *Edit Source* digunakan untuk membuat mengubah parameter kelas CBQ dengan cara langsung mengakses file kelas tersebut.

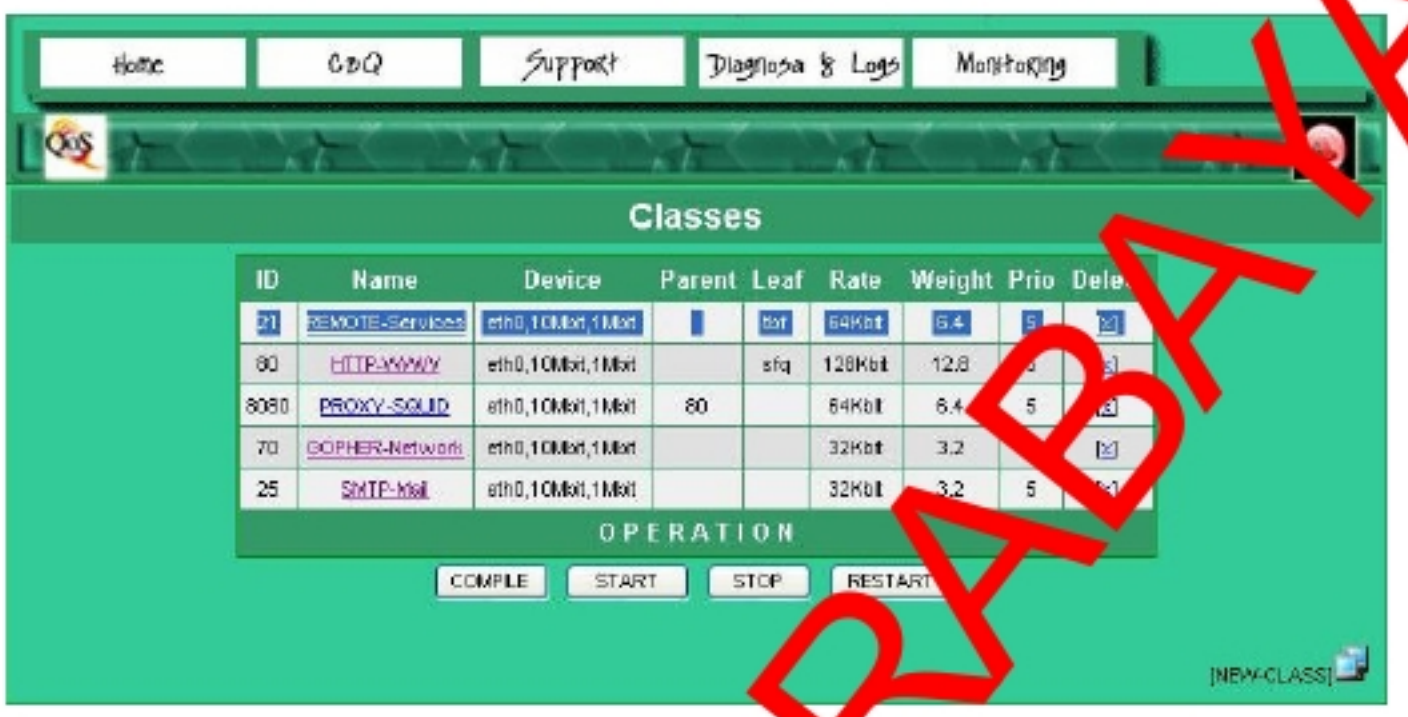
Pada form ini penguna dapat menambah juga merubah parameter-parameter lain dari kelas sesuai dengan kepentingan. Tampilan dari form Edit File seperti pada gambar 4.7 berikut :



Gambar 4.7 Form Edit File

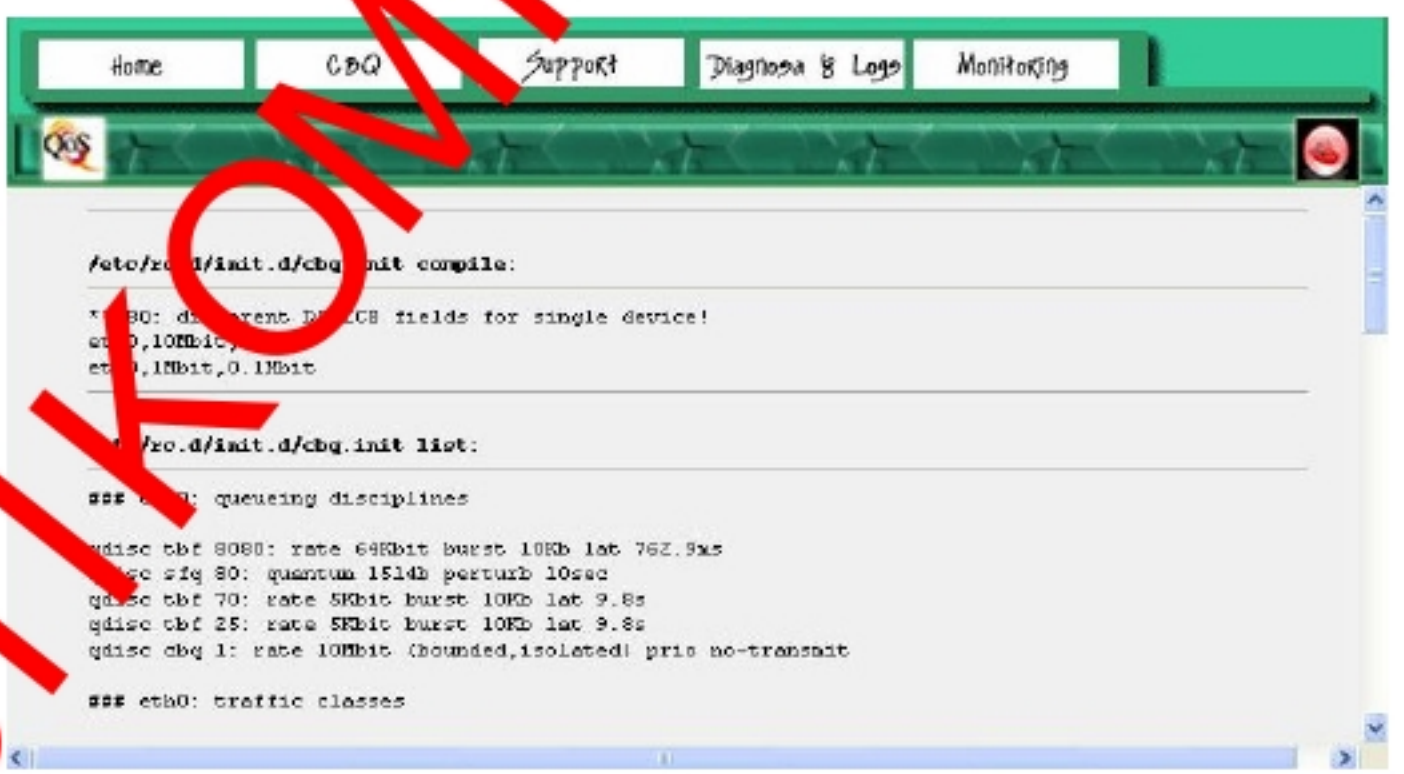
Tombol *Save File* pada form Edit File digunakan untuk menyimpan hasil perubahan yang dilakukan. Hasil dari menambahkan kelas baru dapat dilihat pada

form Classes seperti pada gambar 4.8. Daftar dari kelas-kelas CBQ diurut (*sorting*) berdasarkan ID (*Port Number*).



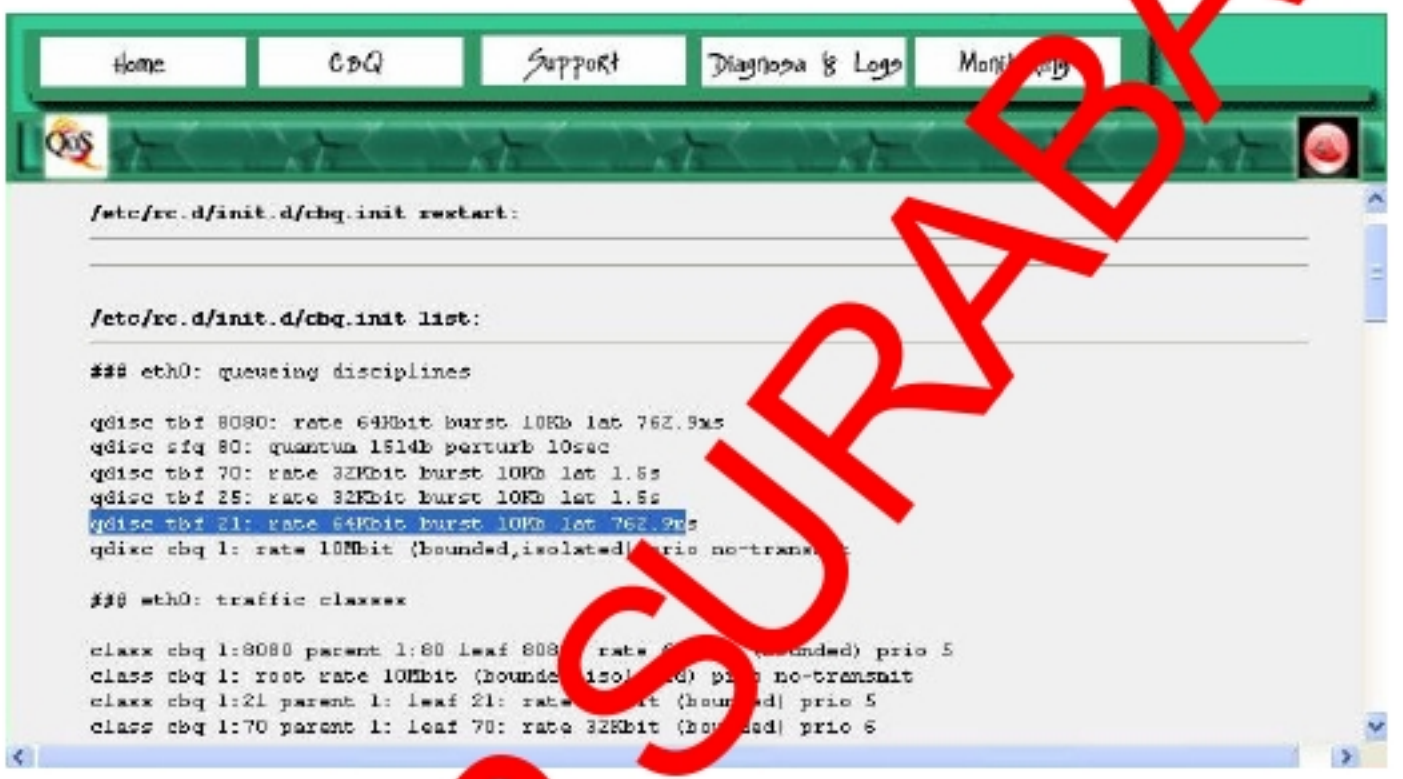
Gambar 4.8 Form Classes (setelah menambah kelas baru)

Untuk menyakinkan bahwa input parameter-parameter yang telah dilakukan adalah valid maka perlu dilakukan proses *compile*. Pada proses ini jika terdapat kesalahan menginputkan nilai parameter maka ada pesan *error* seperti pada gambar 4.9 berikut :



Gambar 4.9 Form Operation Status (*compile error*)

Jika terdapat pesan *error* seperti pada gambar 4.9 maka perlu dilakukan perubahan pada parameter kelas CBQ, dan dilanjutkan dengan proses compile lagi dan untuk mengaktifkan kelas CBQ tersebut dilakukan proses *restart*. Hasil konfigurasi parameter kelas CBQ yang baru akan terlihat setelah proses *restart* dilakukan hal ini seperti terlihat pada gambar 4.10 berikut :



Gambar 4.10 Form Operation Status (*restart*)

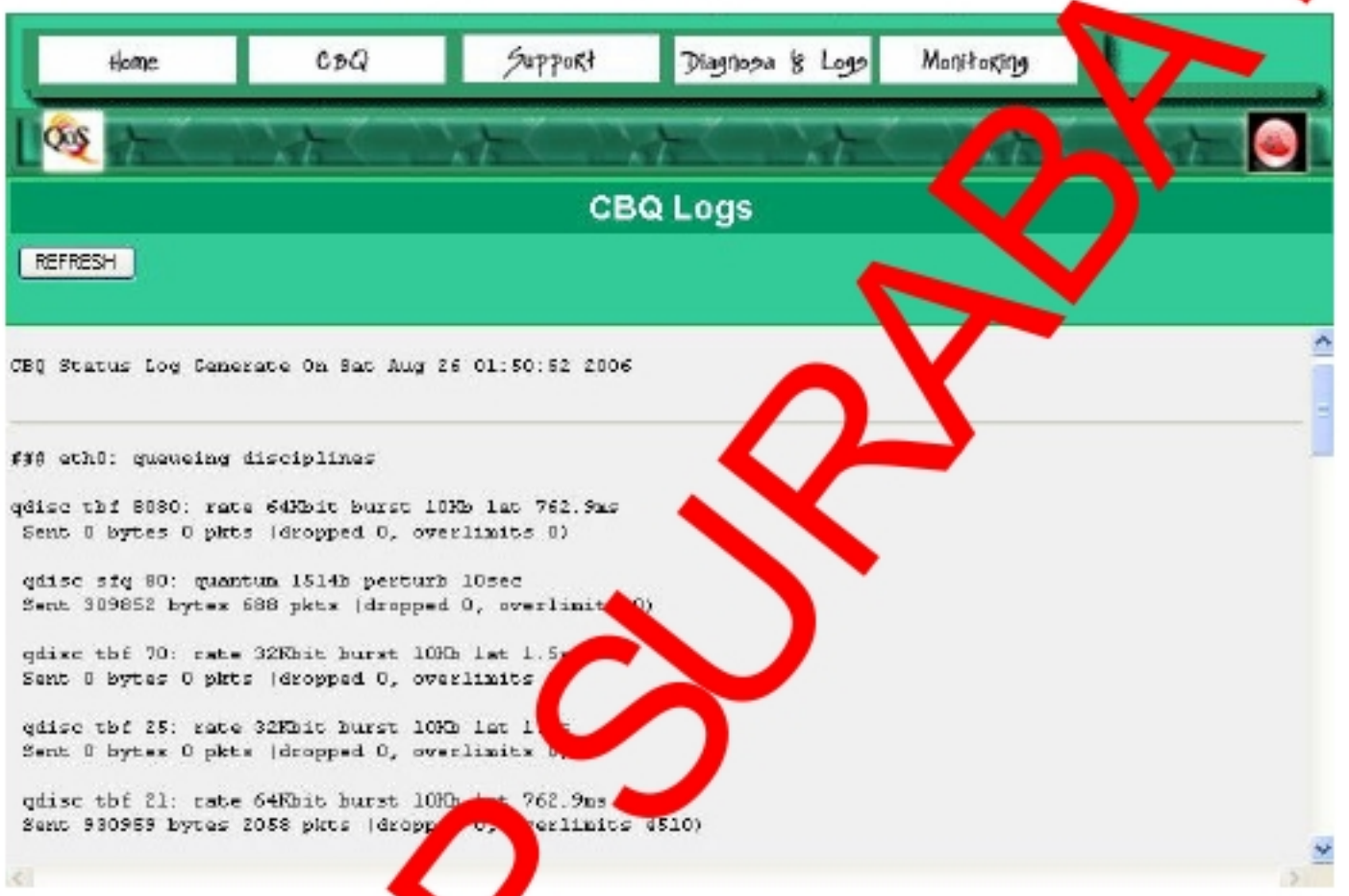
- **CBQ-TrafficTable**

Hasil perubahan yang ditunjukan dengan adanya penambahan kelas baru (*REMOTE-Services*) pada CBQ akan terlihat dengan melakukan pengiriman data melalui port 21 layanan FTP (*File Transfer Protocol*) dan port 23 layanan *Telnet* (*Remote System*).

Untuk melihat perubahan yang terjadi pada form Traffic Table misalnya dilakukan pengambilan (*download*) data pada komputer server oleh komputer *client* seperti dibawah ini pengiriman data (*dataTest.tar*) sebesar 24,7 Mb (*Mega Bytes*) atau jika dalam satuan bytes adalah 25.927.680.

- **CBQ-Logs**

Form CBQ Logs digunakan untuk melihat file *log* dari proses *cbq.init* dan hasil konfigurasi dari masing-masing kelas CBQ. Tampilan dari form CBQ Logs terlihat pada gambar 4.12 berikut :

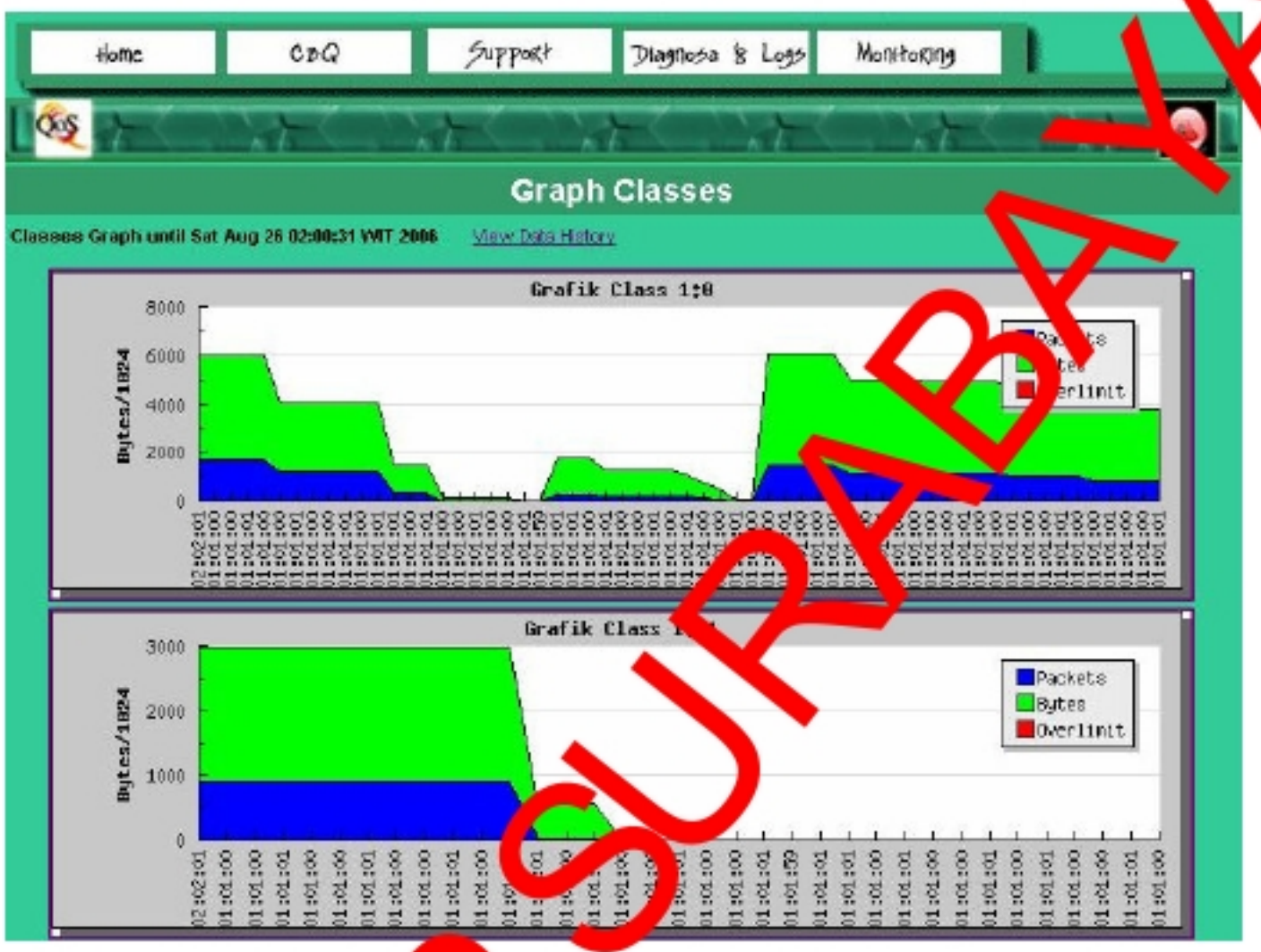


Gambar 4.12 Form CBQ Log

- **Graph Classes**

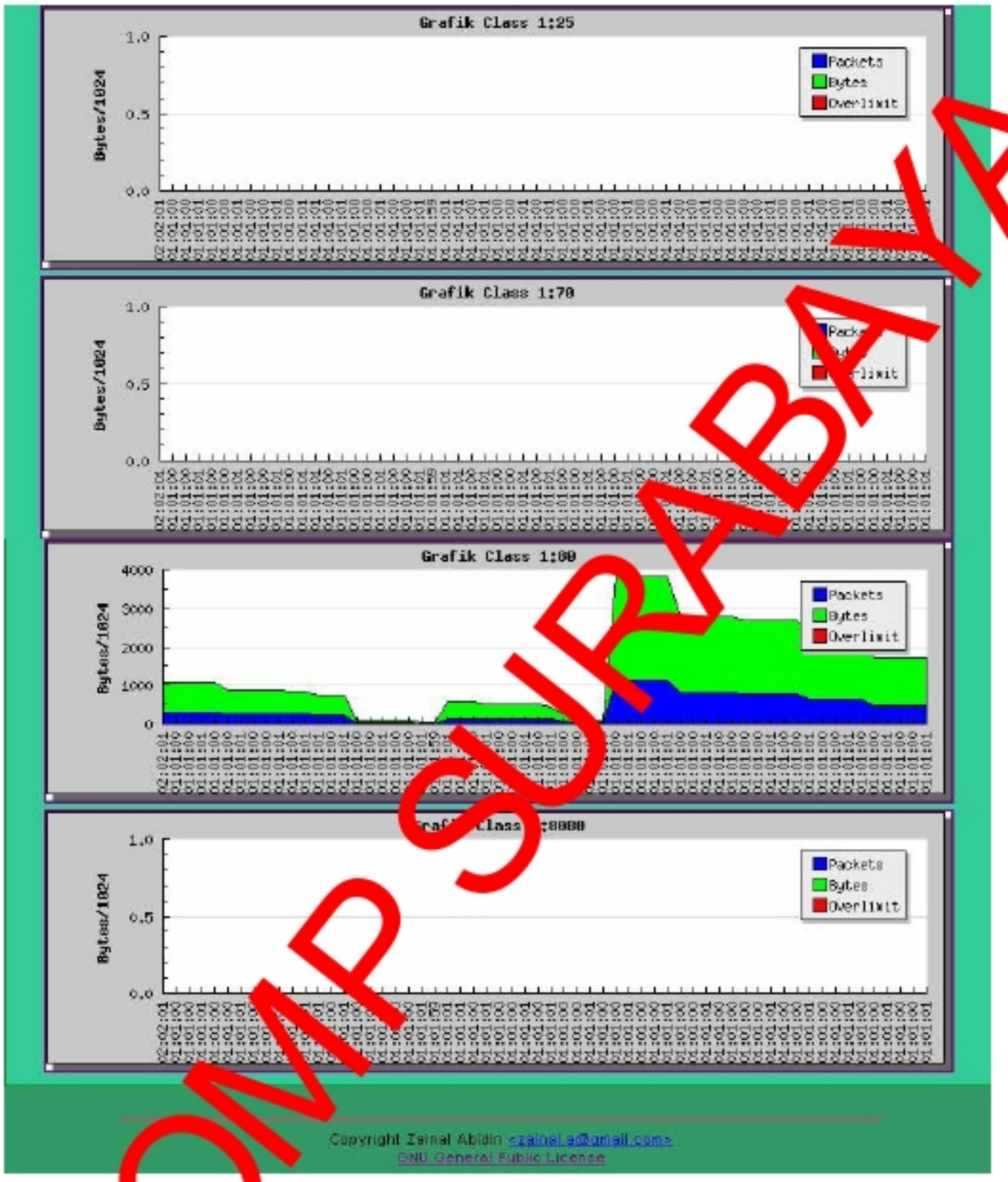
Form Graph Classes menampilkan informasi tentang traffik dari masing-masing kelas CBQ yang direpresentasikan dengan grafik. Setiap kelas CBQ yang yang dibuat secara otomatis untuk melakukan *generate* grafiknya oleh *script* sistem aplikasi. Data dari masing-masing *field* akan tersimpan pada *database* dengan interval waktu 1 menit dengan memanfaatkan fungsi *cron* (*task scheduling*) pada sistem operasi Linux, dan data-data tersebut dikenakan proses *graphic generate* untuk membuat grafik dari masing-masing kelas CBQ.Grafik yang paling utama (*class 1:0*) merupakan grafik dari kelas *parent* yang merupakan

grafik dari interface (NIC) sedangkan grafik dibawahnnya (*class 1:21*) merupakan grafik dari kelas 21 layanan REMOTE-Service.



Gambar 4.13 Form CBQ Graph

Dari gambar 4.13 dan 4.14 (lanjutan) terlihat jumlah *bytes* dari kelas 21 (*REMOTE-Service*) adalah 3000 (*bytes/1024*) yang merupakan nilai rata-rata dari data *bytes* yang diambil dari database yang merupakan nilai aktual dari bandwidth untuk layanan FTP (*File Transfer Protocol*), sedangkan untuk nilai dari *field packet* adalah 1000 bytes. Perlu diketahui untuk nilai dari masing-masing field merupakan hasil dari pengumpulan data menggunakan fungsi SNMP (*Simple Network Management Protocol*) dengan OID .1.3.6.1.4.1.18756.1.1.35.2.0.1 (*qos Object Identifier*).



Gambar 4.14 Form CBQ Graph (lanjutan)

Jika sebuah kelas layanan dihapus (*delete*) maka grafik dari kelas tersebut tidak ditampilkan lagi pada form Graph CBQ namun data dari kelas tersebut akan tetap tersimpan pada *database*, hal ini berguna sebagai data *history* yang dapat digunakan sebagai analisa untuk mengetahui pemakaian *bandwidth* dari kelas tersebut. Untuk melihat data *history* dari masing-masing kelas pengguna dapat

melihat pada form CBQ data *history* seperti pada gambar 4.15. Pengguna dapat melakukan seleksi berdasarkan tanggal untuk melihat data *history*. Tampilan dari form CBQ data *history* adalah sebagai berikut :

Gambar 4.15 Form CBQ Data History

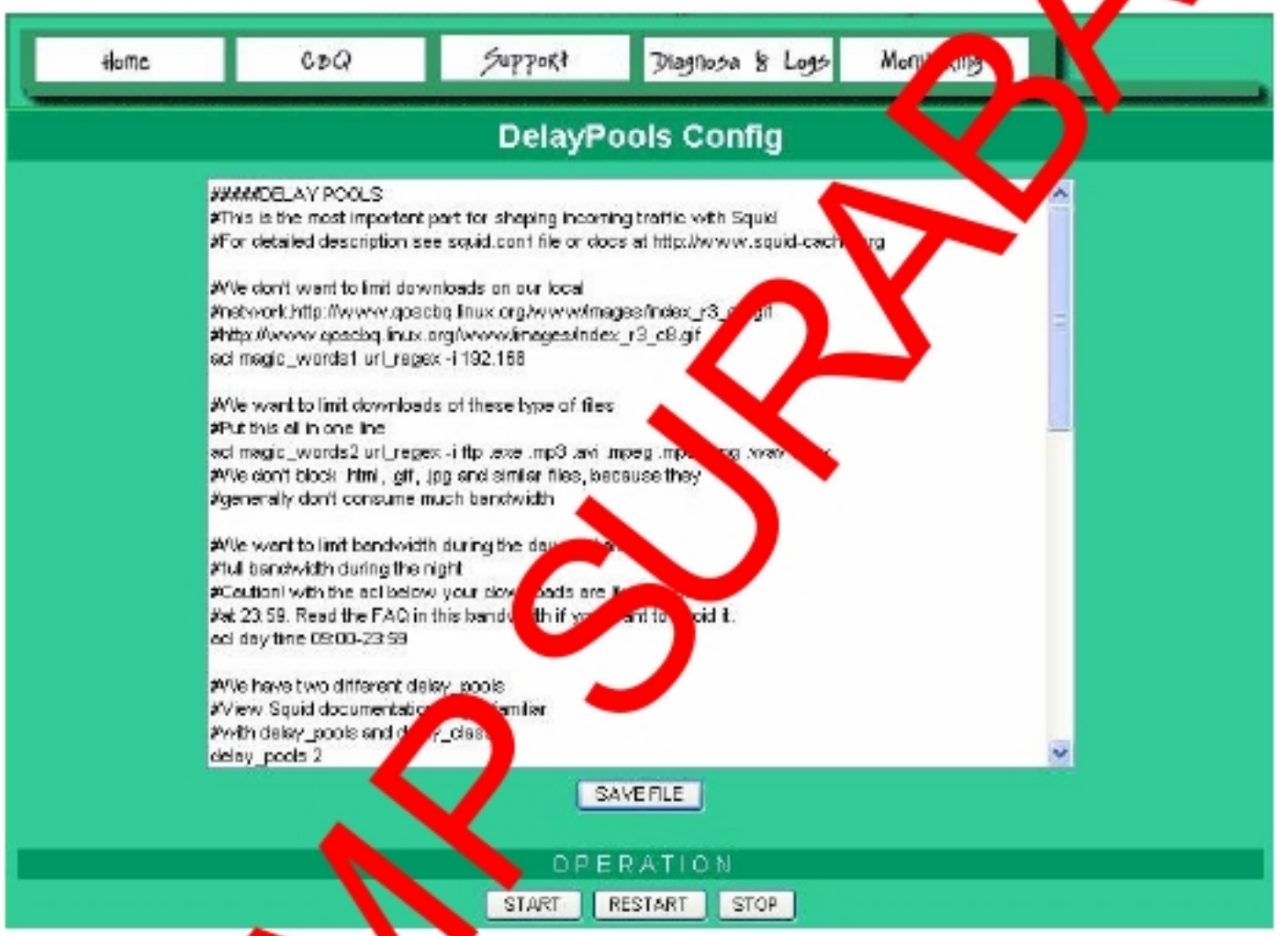
B. Menu Support

Form-form yang terdapat pada menu CBQ antara lain sebagai berikut :

- **DelayPools Config**

Sebagai aplikasi pendukung kinerja dari aplikasi pengaturan dan pembatasan *bandwidth* dangan CBQ khususnya pada layanan HTTP (*Hypter Text Transfer Protocol*) *squid* dengan fitur *delaypools* dapat dikonfigurasi pada form DelayPools Config seperti pada gambar 4.16. Parameter penting dari *delaypools* adalah *acl magic*, pada parameter ini pengguna dapat mendefinisikan ekstension dari file-file yang akan dibatasi (*limit*) pemakaian *bandwidth*-nya misalnya *.mp3 yang berarti melakukan pembatasan pemakaian bandwidth untuk proses download

file-file yang berekstension *.mp3 (*audio file*) disamping parameter-paramter penting lainnya. Tombol *Save File* digunakan untuk menyimpan hasil perubahan dari parameter-parameter *delaypools* squid, sedangkan untuk tombol tombol *operation* digunakan untuk menjalankan dan menghentikan proses squid. Tampilan dari form DelayPools Config sebagai berikut :

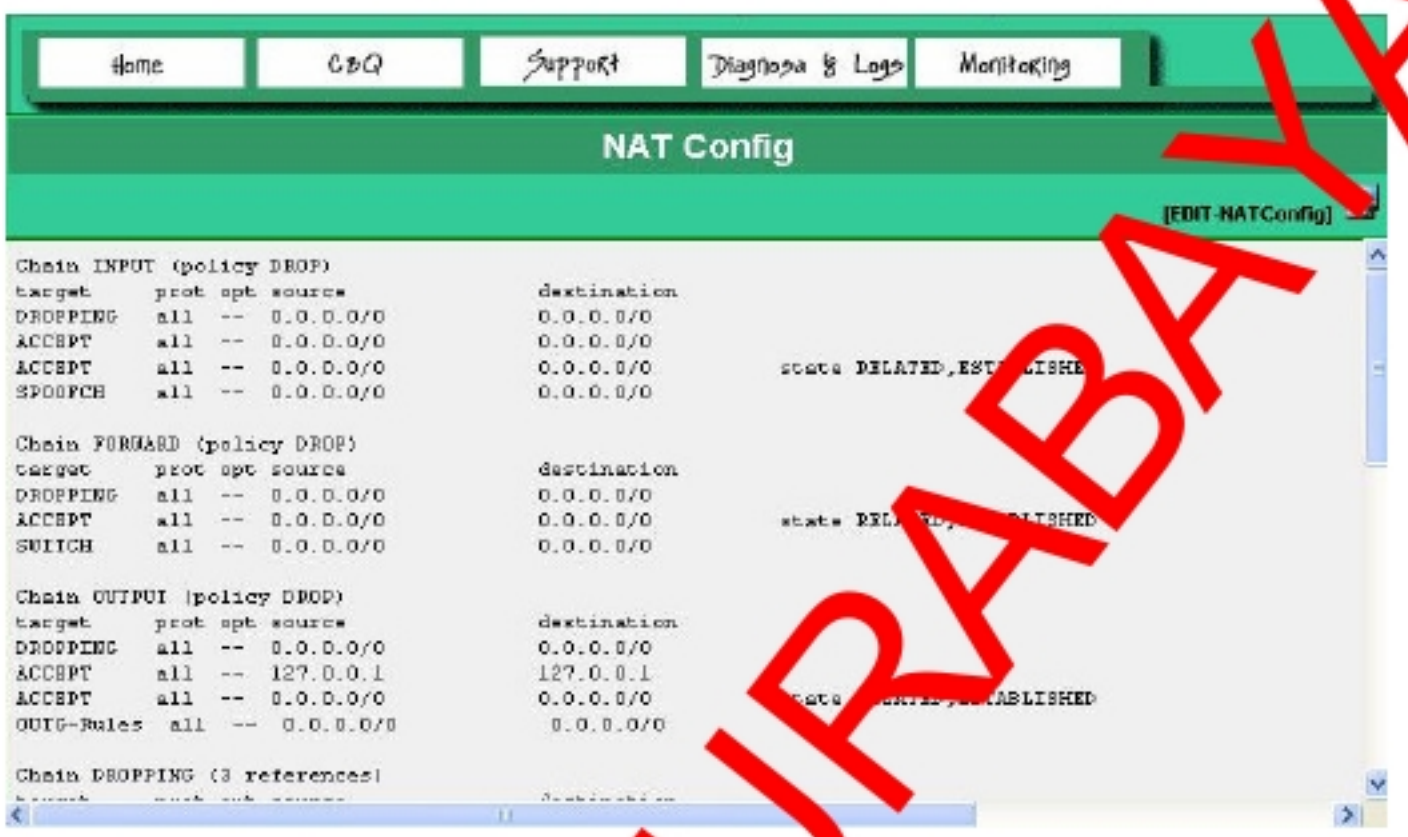


Gambar 4.16 Form Squid DelayPools Config

- **NAT Config**

Form NAT (*Network Address Translation*) digunakan untuk mengubah konfigurasi *iptables* yang berfungsi *router/gateway* yang berada pada perimeter terluar dari struktur jaringan komputer dimana aplikasi pembatasan dan pengaturan *bandwidth* dengan CBQ diinstall. Disamping itu kegunaan dari NAT iptables adalah untuk mengubah *IP Public* menjadi *IP Private* agar komputer

client dapat melakukan koneksi internet. Tampilan dari form NAT Config seperti pada gambar 4.17 berikut :



Gambar 4.17 Form Iptables NAT Config

- **Managemet Support**

Managemet support merupakan sub-menu dengan beberapa form antara lain form Network Management, form Resolv Config, dan form Change Password. Form Network Management digunakan untuk melihat data alamat IP (*Internet Protocol*) dari sistem dan juga untuk mengubah alamat IP. Sedangkan form Resolv Config digunakan untuk mengkonfigurasi file */etc/resolv* yang berguna untuk mendefinisikan *domain, search, nameserver* agar sistem dapat bertindak sebagai *DNS-Client*.

Sedangkan form Change Password digunakan untuk mengubah/mengganti password dari aplikasi ini, tampilan dari form Change Password terlihat pada gambar 4.18 berikut :

Gambar 4.18 Form Change Passwo

C. Menu Diagnosa & Logs

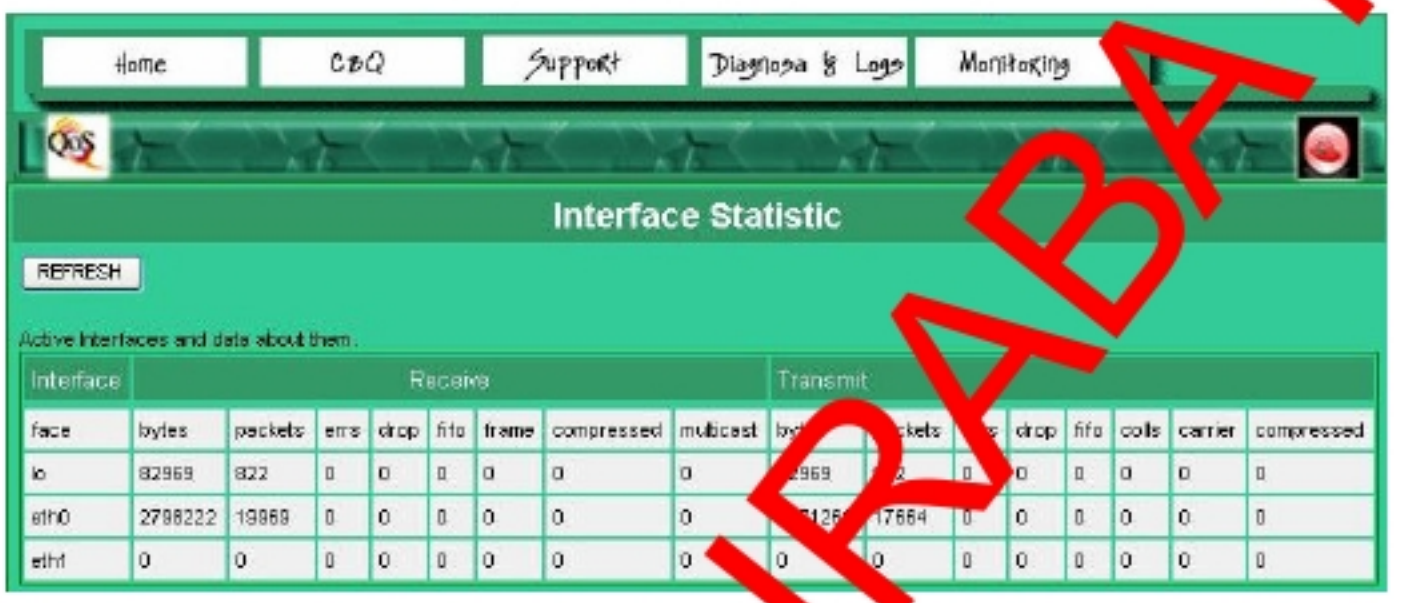
Form-form yang terdapat pada menu CBQ antara lain sebagai berikut :

- **Interface Info & Statistik**

Gambar 4.19 Form Interface Info

Form Interface Info seperti pada gambar 4.19 menampilkan informasi dari interface (NIC) yang terdapat pada sistem informasinya antara lain *IP Address*,

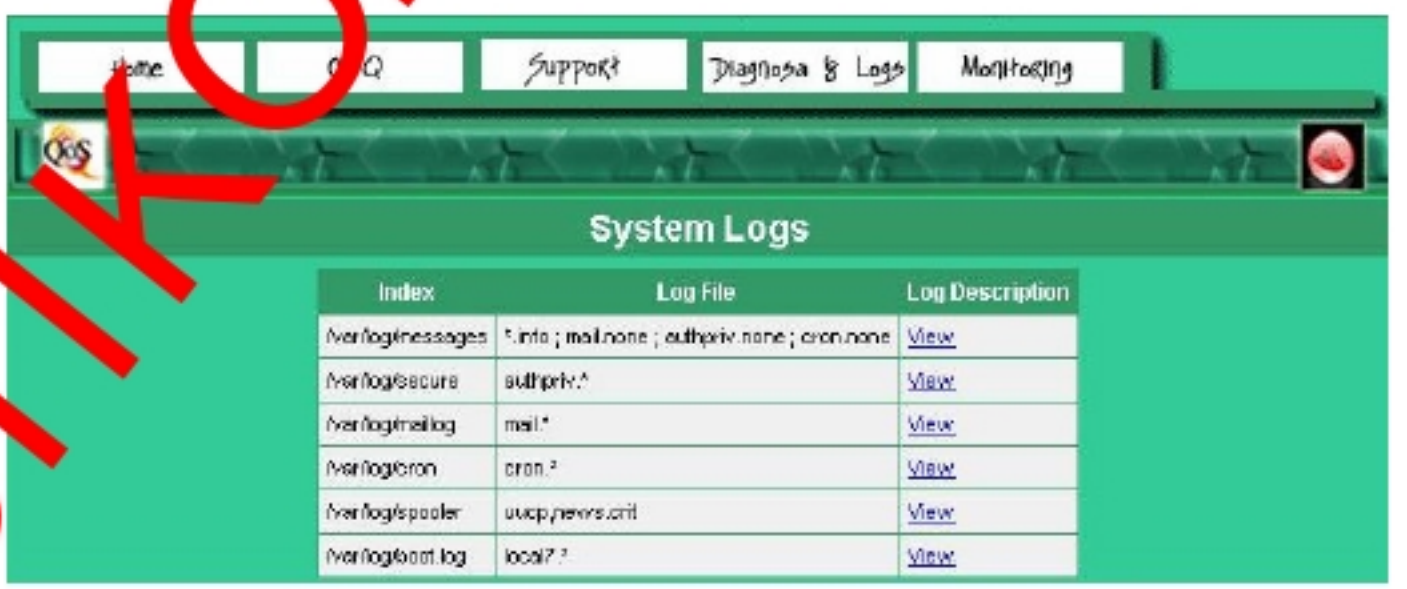
Netmask, *IP Gateway*, *MAC Address*, MTU (*Maximum Transfer Unit*) dan sebagainya. Sedangkan form Interface Statistic berisi informasi besarnya pengiriman (*transmiting*) dan penerimaan (*receiving*) dari interface seperti terlihat pada gambar 4.20 berikut :



Gambar 4.20 Form Interface Statistik

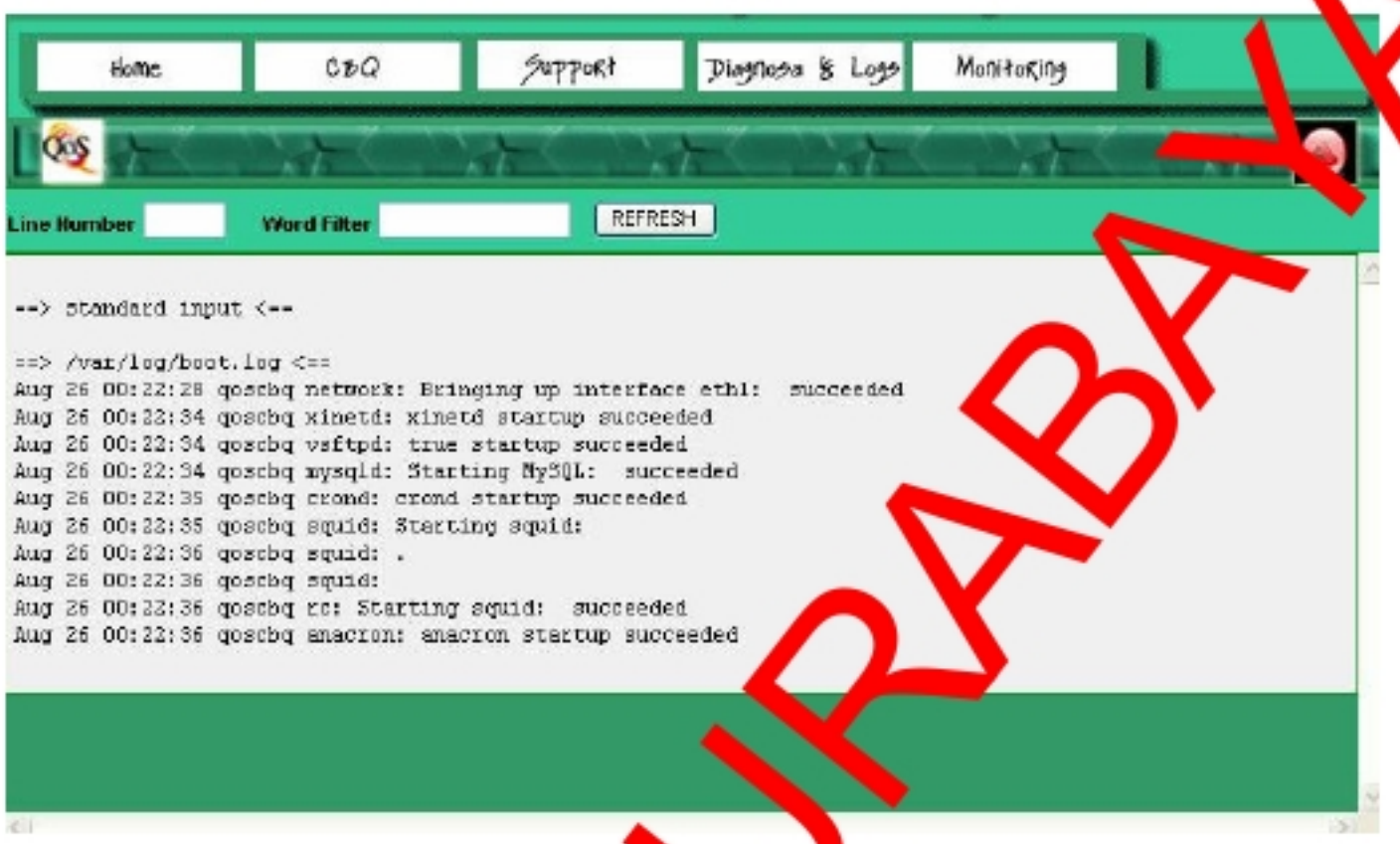
- **System Logs**

Untuk melihat informasi log dan statistik dari sistem, terdapat form System Logs dan Detail System Logs. Pada form System Log terdapat daftar *file log* sistem pada sistem operasi Linux RedHat 9.0, daftar file log tersebut diambil dari file */etc/syslog.conf*. Berikut tampilan dari form System Log :



Gambar 4.21 Form System Logs

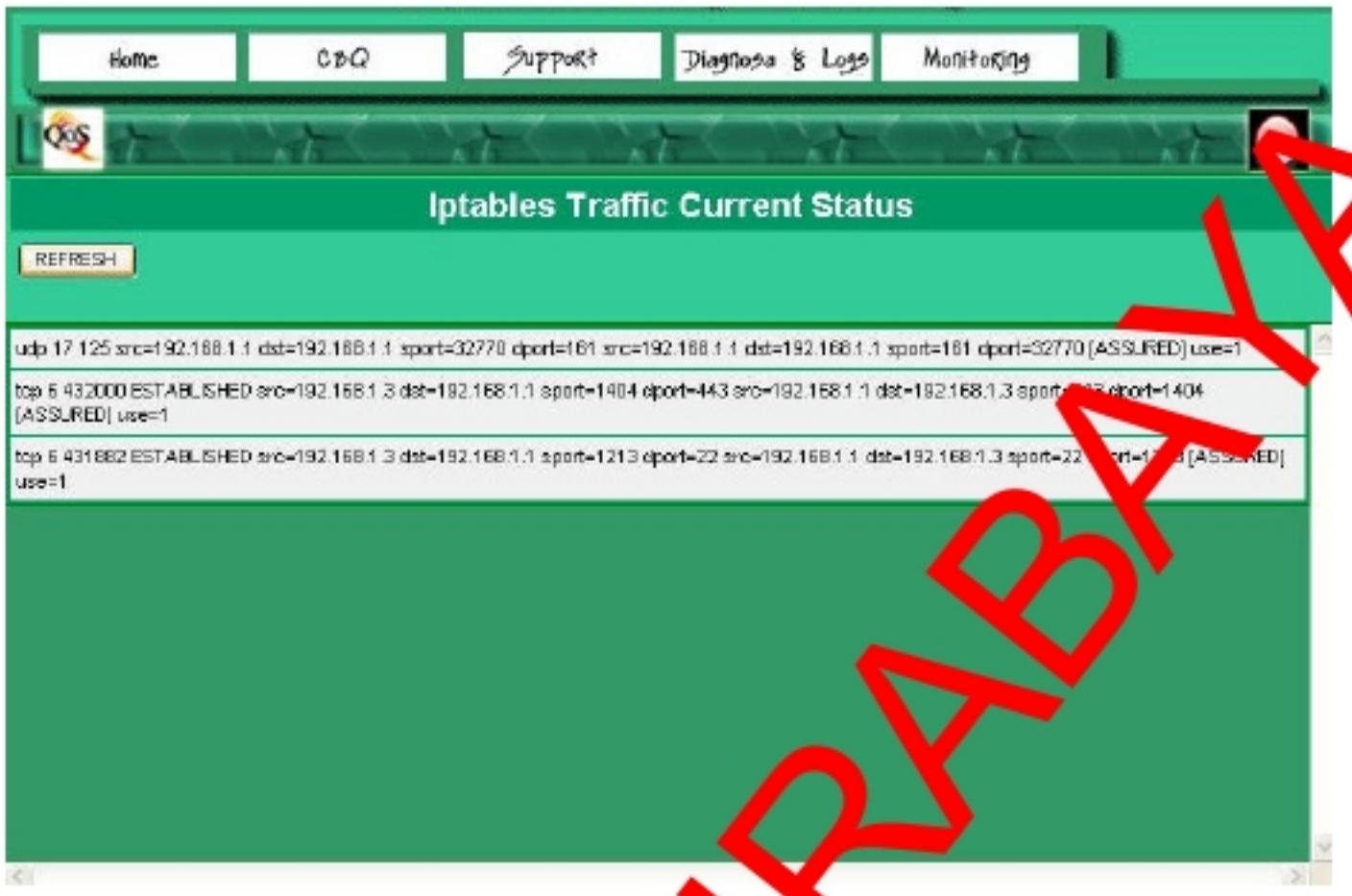
Sedangkan detail dari masing-masing file system logs terdapat pada form Detail System Log seperti terlihat pada gambar 4.22 berikut :



Gambar 4.22 Form Detail System Logs

- **Iptables Logs & Current Status**

Pada form Iptables Current Status berisi informasi tentang aplikasi pendukung dari CBQ yakni iptables, dimana terdapat informasi tentang trafik jaringan komputer yang sedang aktif. Tampilan dari form tersebut terlihat pada gambar 4.23 berikut :



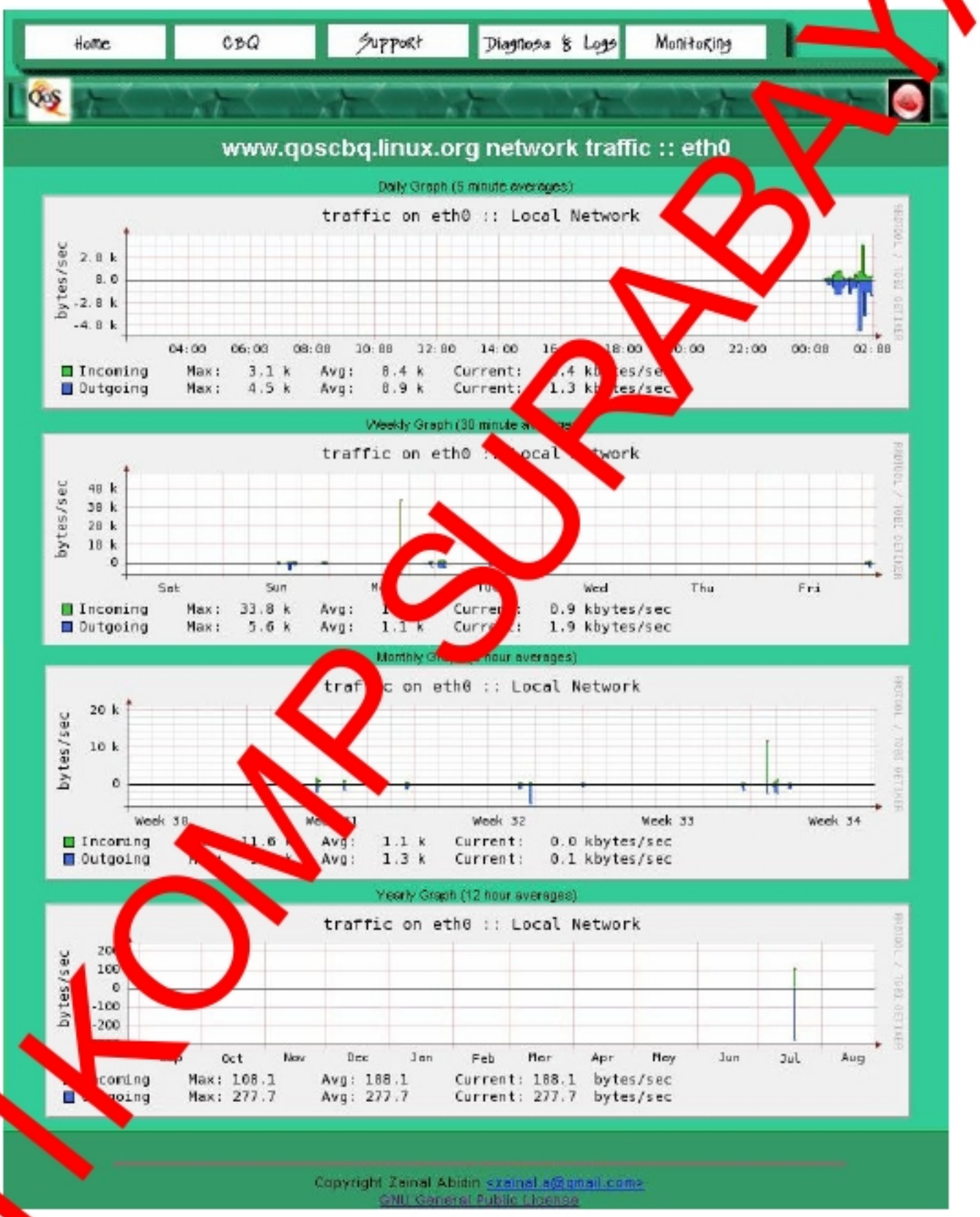
Gambar 4.23 Form Iptables Traffic Current Status

D. Menu Monitoring

- **Grafik Interface Traffic**

Form Grafik Interface seperti pada gambar menampilkan representasi grafik dari masing-masing interface yang terdapat pada sistem, misalnya *eth0*, *eth1*, dan *lo*. Masing-masing interface menampilkan informasi traffik masuk (*incomming*) dan keluar (*outgoing*) dengan beberapa interval waktu yang antara lain informasi traffik 5 menit yang lalu, 30 menit yang lalu, 2 jam yang lalu, 12 jam yang lalu. Dari interval-interval waktu diatas akan diambil nilai rata-rata (*average*) untuk beberapa waktu tertentu antara lain rata-rata per hari (*daily*), rata-rata per minggu (*weekly*), rata-rata per bulan (*monthly*), dan rata-rata per tahun (*yearly*). Hal ini berguna untuk melakukan analisa besarnya pemakaian koneksi jaringan oleh para pengguna jaringan sehingga *network manager* dapat

mengambil keputusan untuk melakukan pembatasan dan pembagian bandwidth yang sesuai dengan kebutuhan dan kondisi dari koneksi jaringan komputer. Berikut tampilan dari form Traffik Interface :



Gambar 4.24 Form Grafik Interface

Pada form Grafik interface disamping rata-rata (*average*) dari kecepatan koneksi terdapat nilai paling rendah (*min value*), paling tinggi (*max value*), dan nilai rata-

rata (*avg value*) dalam satuan *kilo bytes*, dan nilai *current* (*current value*) merupakan besarnya kecepatan koneksi interface pada saat itu dalam satuan *kilobytes per seconds* (kbps).

- **Grafik Load Average**

Berikut ini adalah tampilan dari form Grafik Load Averag

Gambar 4.25 Form Grafik Load Average

Informasi grafik *load average* seperti terlihat pada gambar 2.26 merupakan informasi *uptime* (lama waktu beroperasi) dari sistem yang terdiri dari

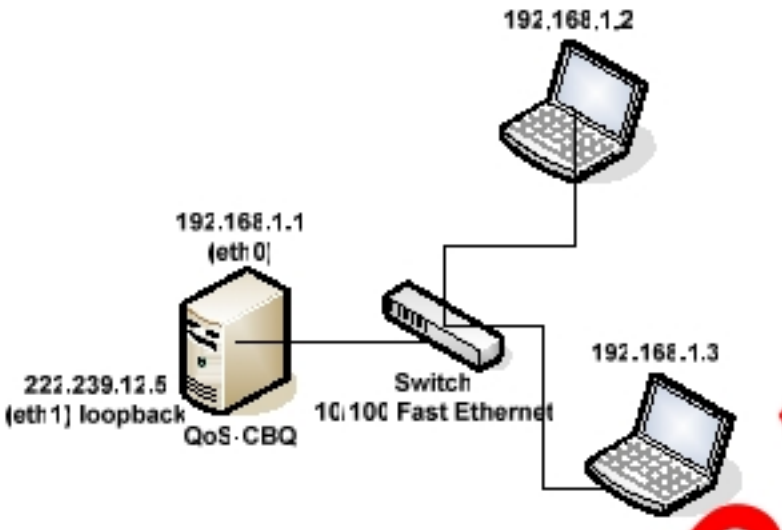
penggunaan CPU (*Central Processing Unit*) dan perangkat hardware seperti *processor* yang digunakan oleh proses-proses yang berjalan pada sistem. Informasi ini dibagi menjadi tiga interval waktu yang *load average* per 1 menit, per 5 menit dan per 15 menit.

4.3 Uji dan Evaluasi

Untuk mengetahui sejauh mana aplikasi ini melakukan pembatasan dan pengaturan penggunaan bandwidth dengan CBQ (*Class Based Queueing*), peneliti melakukan uji dan evaluasi dengan membuat beberapa percobaan implementasi dengan 3 skenario besar yaitu skenario dengan beberapa kelas, skenario dengan beberapa parameter dan skenario dengan *topologi* berbeda.

Untuk melakukan uji dan evaluasi ini desain jaringan komputer yang digunakan adalah seperti pada gambar 4.20 dimana terdapat dua sebuah komputer *server* dengan menggunakan sistem operasi Linux Redhat 9.0 yang sudah diinstall aplikasi pengaturan dan pembatasan bandwidth dengan CBQ (*Class Based Queueing*) dengan alamat IP 192.168.1.1 untuk interface 1 (*eth0*) dan 222.239.12.5 untuk interface 2 (*eth1*) yang didesain *loopback* dengan IP Gateway 192.168.1.4.

Sedangkan untuk dua komputer yang bertindak sebagai *client* dengan alamat IP masing-masing 192.168.1.2 dan 192.168.1.3 dan perangkat jaringan yang menghubungan komputer server dan client adalah *switch 10/10 Fast Ethernet*. Berikut gambar dari desain jaringan yang digunakan untuk implementasi sistem :



Gambar 4.26 Desain Jaringan Uji dan Evaluasi

Dari skenario-skenario tersebut akan dilakukan beberapa analisa. Analisa-analisa tersebut antara lain :

a. Analisa *Throughput*

Analisa ini dilakukan untuk mengetahui besarnya pemakaian *bandwidth* aktual yang digunakan oleh sebuah komputer client untuk mengakses layanan setelah dilakukan pembatasan dan pengaturan *bandwidth* sebagai akibat dari efek (*ripple*) dari algoritma *link sharing* yang terdapat pada CBQ dengan menganalisa grafik yang diperoleh dari CBQ Glass dan grafik yang diperoleh dari sisi komputer client. Untuk mengetahui grafik dari sisi komputer client peneliti menggunakan aplikasi NetMeter versi 3.2 (http://www.hootech.com/NetMeter/). Satuan waktu analisa menggunakan waktu t (dalam detik) untuk melihat besarnya *throughput* traffik *upload* dan *download* yang dihasilkan. (Floyd, Sally, Jacobson, Van. 1998:5)

b. Analisa waktu pemrosesan pesan dalam bandwidth oleh CBQ

Analisa ini dilakukan dengan menggunakan rumus perhitungan waktu pemrosesan pesan didalam bandwidth yang terdapat pada CBQ. (Floyd, Sally,1998:15).

Rumus perhitungan yang digunakan adalah sebagai berikut :

$$Tm = \frac{packet\_size \times 8}{line\_speed} \quad X\ 1000ms\ (mili\ seconds)$$

$$Tbm = Tb - Tm$$

Dimana :

Packet Size; merupakan besarnya paket data yang diproses

Line Speed; kecepatan/*rate* yang digunakan untuk mengirimkan paket data

Tm; waktu pengiriman pesan tanpa CBQ

Tb; waktu pengiriman pesan melalui CBQ

Tbm; waktu pemrosesan pesan dalam CBQ

Analisa waktu pemrosesan pesan dalam bandwidth oleh CBQ hanya berlaku pada skenario topologi (Floyd, Sally,1998:15)

c. Analisa *Loss Datagram*

Untuk menganalisa *loss datagram* (paket datagram yang hilang) didapat dari pengaktifan aplikasi *Ping* (*ping command*) untuk memeriksa konektifitas komputer client pada saat menggunakan bandwidth secara bersama dengan komputer client lainnya. Paket *Ping* sebanyak 30 x 64 *bytes* dikirimkan. Pada parameter *ping* tersebut terdapat nilai *loss* dalam ukuran bytes dan juga dalam persentase (%). Dari nilai tersebut dapat dianalisa berapa *loss datagram* pada saat sebuah komputer client sedang menggunakan bandwidth untuk mengakses atau mengirimkan paket data dengan aplikasi layanan jaringan. (Floyd, Sally,1998:15)

d. Analisa *Respon Time* dan *Round Trip Time* (rtt)

Untuk menganalisa *respon time* dan *round trip time* nilai parameter yang diperlukan untuk menganalisa didapat dari pengaktifan aplikasi *Ping* (*ping command*) untuk memeriksa konektifitas komputer client pada saat menggunakan bandwidth secara bersama dengan komputer client lainnya. Paket *Ping* sebanyak

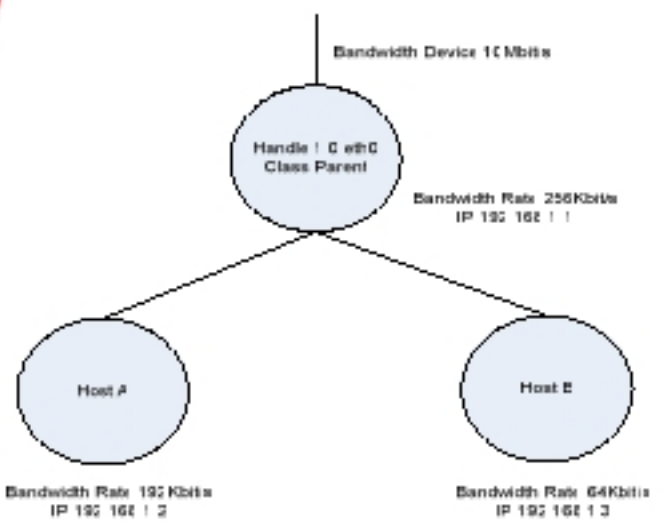
30 x 64 *bytes* dikirimkan, dan dilakukan sebanyak 3 kali untuk mendapatkan nilai *round trip time* dan *respon time* masing-masing komputer client. (Floyd, Sally. 1998:15)

4.3.1 Skenario Kelas

Pada skenario pembatasan dan pengaturan bandwidth dilakukan pada kelas-kelas cabang berdasarkan alamat IP (*Internet Protocol*) dan berdasarkan *Port Number*. Terdapat dua buat alamat IP dari dua komputer client dan untuk *Port Number* terdapat beberapa *port* yang mendefinisikan layanan dari aplikasi jaringan yang akan dilakukan pembatasan dan pengaturan bandwidth dengan CBQ. Sebagai asumsi jumlah bandwidth keseluruhan yang akan dilakukan pembatasan dan pembagian dalah 256*Kbit/second*.

A. Dengan dua cabang (*leaf*) kelas berdasarkan *IP Address*

Skenario ini pembagian bandwidth dibagi untuk dua alamat IP dari komputer client, untuk IP 192.168.1.2 sebesar 192*Kbit/second* dan untuk IP 192.168.1.3 sebesar 64*Kbit/second*. Dari jumlah alokasi bandwidth pada kedua buah alamat IP tersebut adalah 256*Kbit/second* yang merupakan kelas *parent* (*handle 1:0 eth0*). Gambaran dari skenario ini terlihat pada gambar dibawah ini :



Gambar 4.27 Skenario 2 Leaf Kelas Berdasarkan IP Address

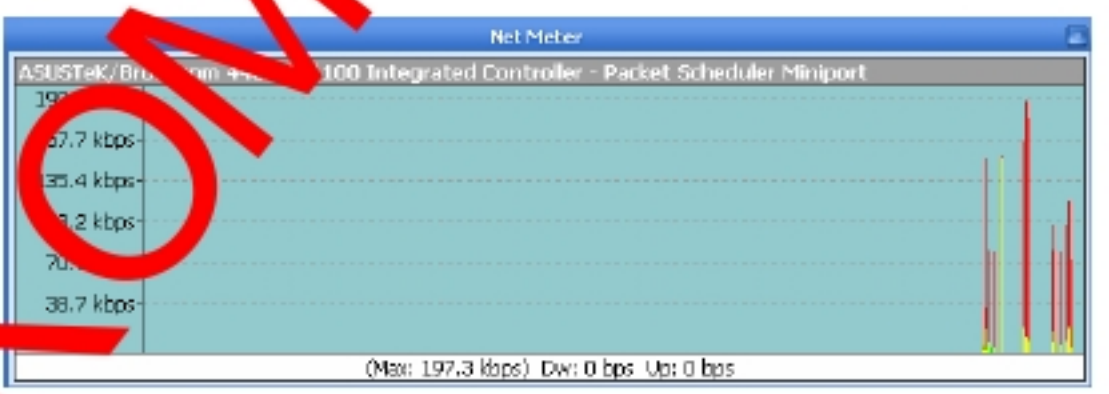
Dan berikut konfigurasi masing-masing kelas *leaf* berdasarkan alamat IP yang dilakukan pada aplikasi pembatasan dan pengaturan bandwidth dengan CBQ pada sistem operasi linux berbasis web adalah sebagai berikut :

Gambar 4.28 Konfigurasi Kelas CBQ dengan 2 leaf kelas IP Address

Dari skenario diatas dapat dilakukan uji dan analisa sebagai berikut :

- Throughput

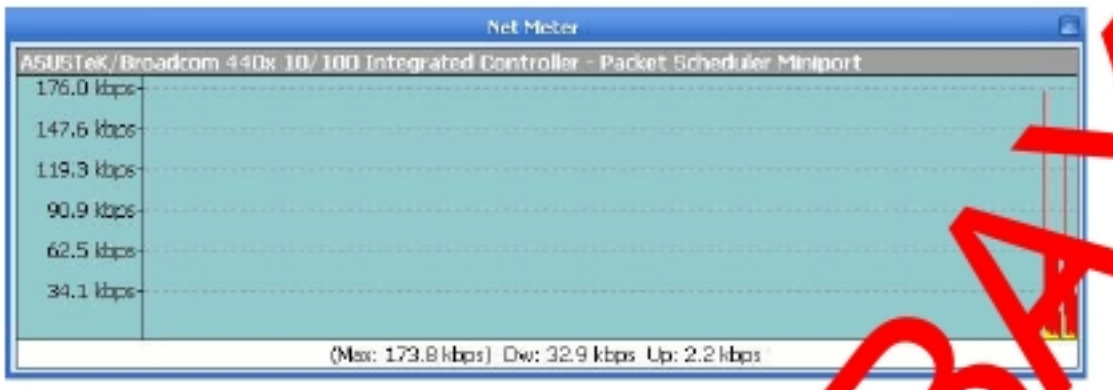
Dari grafik pada gambar 4.29, yang diperoleh dari komputer client dengan IP 192.168.1.2 terlihat kecepatan maksimum (*max*) bandwidth aktual adalah 197,3 kbps hal ini berarti terjadi kebocoran bandwidth sebesar 5,3 kbps dari bandwidth yang dialokasikan sebesar 192kbps.



Gambar 4.29 Grafik Skenario 2 Leaf Kelas IP 192.168.1.2

Sedangkan grafik yang diperoleh dari komputer client dengan IP 192.168.1.3 dengan mengakses layanan yang sama terlihat diatas 62.5 kbps berarti tidak terjadi kebocoran bandwidth namun terdapat traffik loncatan (*spike*) hal ini disebabkan adanya parameter *leaf* yang menyebabkan kelas IP 192.168.1.3 dapat

meminjam bandwidth *idle* dari kelas IP 192.168.1.2. Berikut tampilan grafiknya adalah sebagai berikut :



Gambar 4.30 Skenario 2 Leaf Kelas IP 192.16[illegible]1.3

- Loss Datagram

Pada IP 192.168.1.2 dan 192.168.1.3 setela[illegible] dilakukan pengiriman paket ping sebanyak 30 kali masing-masing sebesar 64 *bytes* hasil yang diperoleh adalah tidak terjadi *loss datagram* berarti [illegible]mua paket yang diterima dengan total jumlah yang sama seperti dikirim oleh komputer server.

- Respon Time dan Round Trip Time

Jumlah *respon time* (waktu tanggap) dan *round trip time* (rtt) untuk IP 192.168.1.2 ketika dikirim 30 paket ping yang masing-masing berjumlah 64 byte maka jumlah respon time adalah 28998ms dan round trip time adalah min/avg/max/mdev = 0,201/0,207/0,272/0,021 *miliseconds*, sedangkan untuk IP 192.168.1.3 jumlah respon time adalah 43013ms dan *round trip time* adalah min/avg/max/mdev = 0,201/4,072/170,181/25,33 *miliseconds*

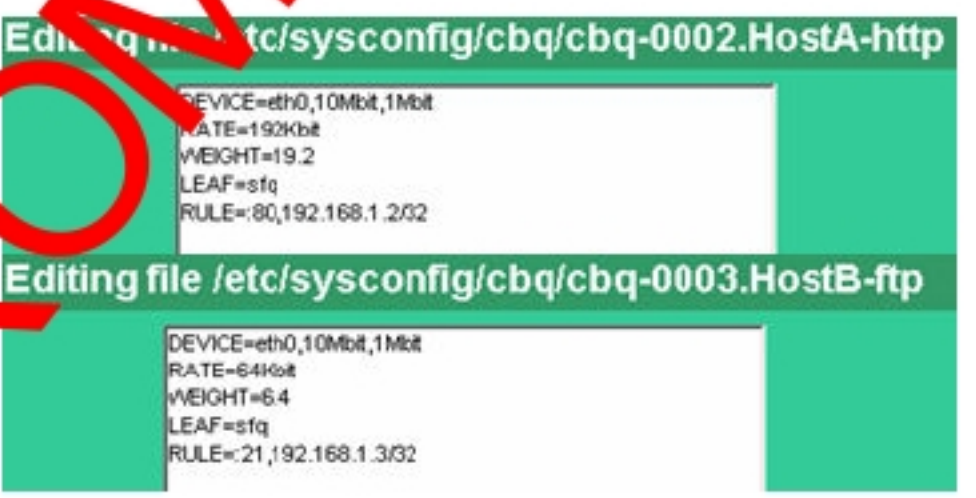
B. Dengan dua *leaf* kelas berdasarkan *Port Number*

Pada skenario ini pembagian dan pembatasan bandwidth dengan mengalokasikan bandwidth berdasarkan pada nomor *port* yakni port 80 yang merupakan layanan HTTP (*Hyper Text Transfer Protocol*) dan port 21 yang merupakan layanan FTP

(*File Transfer Protocol*), dengan alokasi bandwidth masing-masing sebesar 192Kbit/s untuk port 80 dengan alamat IP 192.168.1.2 dan port 64Kbit/s untuk port 21 dengan alamat IP 192.168.1.2. Berikut gambaran skenario dengan dua *leaf* kelas (kelas cabang) :

Gambar 4.31 Skenario Dua Leaf Kelas Berdasarkan Port Number

Sedangkan konfigurasi masing-masing kelas *leaf* berdasarkan Port Number adalah sebagai berikut :

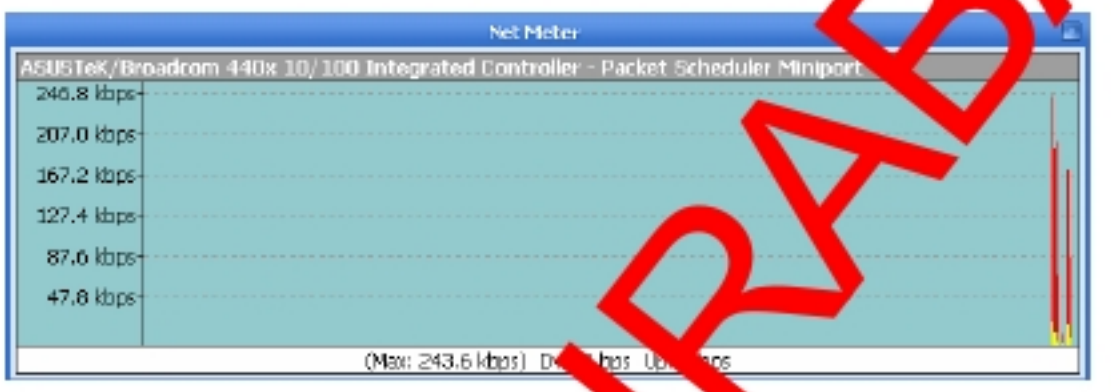


Gambar 4.32 Konfigurasi Kelas CBQ dengan 2 leaf kelas Port Number

Dari skenario diatas dapat dilakukan pengujian dan analisa sebagai berkut :

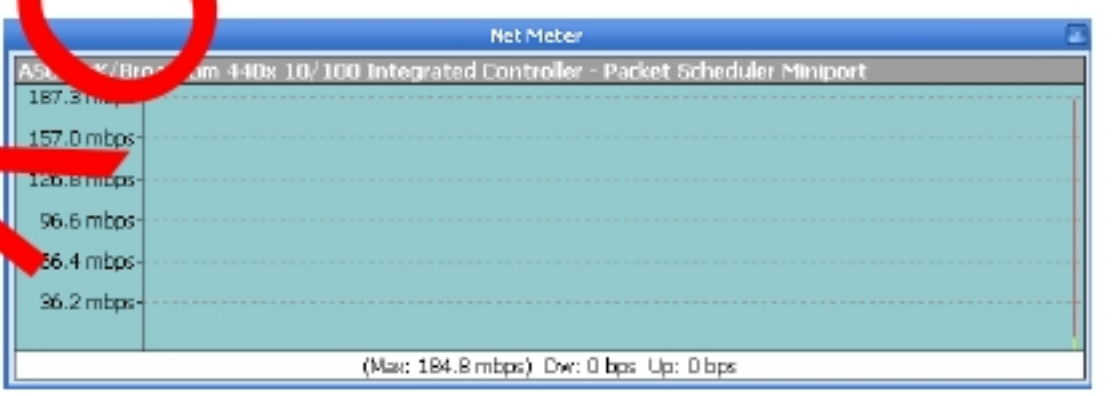
- Throughput

Dari grafik pada gambar 4.33, yang diperoleh dari komputer client dengan IP 192.168.1.2 terlihat kecepatan maksimum (*max*) bandwidth aktual adalah 243,6 kbps hal ini berarti terjadi kebocoran bandwidth sebesar 51,6 kbps dari bandwidth yang dialokasikan sebesar 192kbps hal ini disebabkan layanan pada IP 192.168.1.3 dalam keadaan *idle* maka layanan 80 dapat meminjam bandwidth.



Gambar 4.33 Skenario 2 Leaf Kelas Port 80 IP 192.168.1.2

Sedangkan grafik yang diperoleh dari komputer client dengan IP 192.168.1.3 dengan mengakses layanan yang sama terlihat diatas 64 kbps berarti tidak terjadi kebocoran bandwidth namun terdapat traffik loncatan (*spike*) hal ini disebabkan adanya parameter *leaf* yang menyebabkan kelas IP 192.168.1.3 dapat meminjam bandwidth *idle* dari kelas IP 192.168.1.2. Berikut tampilan grafiknya adalah sebagai berikut :



Gambar 34 Skenario 2 Leaf Kelas Port 21 IP 192.168.1.3

- Loss datagram

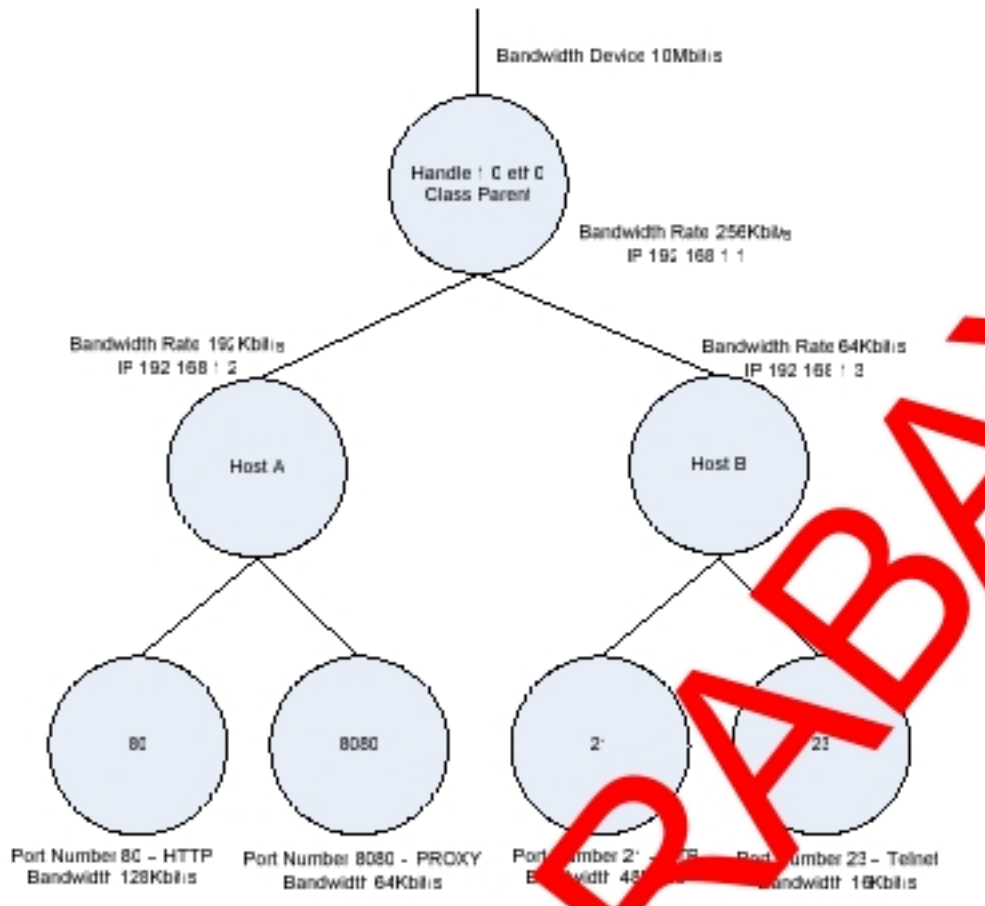
Pada IP 192.168.1.2 dan 192.168.1.3 setelah dilakukan pengiriman paket ping sebanyak 30 kali masing-masing sebesar 64 bytes hasil yang diperoleh adalah tidak terjadi loss datagram berarti semua paket ping diterima dengan total jumlah yang sama seperti dikirim oleh komputer server

- Respon time dan Round Trip Time

Jumlah *respon time* (waktu) dan *round trip time* (rtt) untuk IP 192.168.1.2 ketika dikirim 30 paket ping yang masing-masing berjumlah 64 byte maka jumlah respon time adalah 28995ms dan round trip time adalah min/avg/max/mdev = 0,199/0,207/0,271/0,023 *miliseconds*, sedangkan untuk IP 192.168.1.3 jumlah respon time adalah 29008ms dan round trip time adalah min/avg/max/mdev = 0,200/0,251/0,849/0,150 *miliseconds*

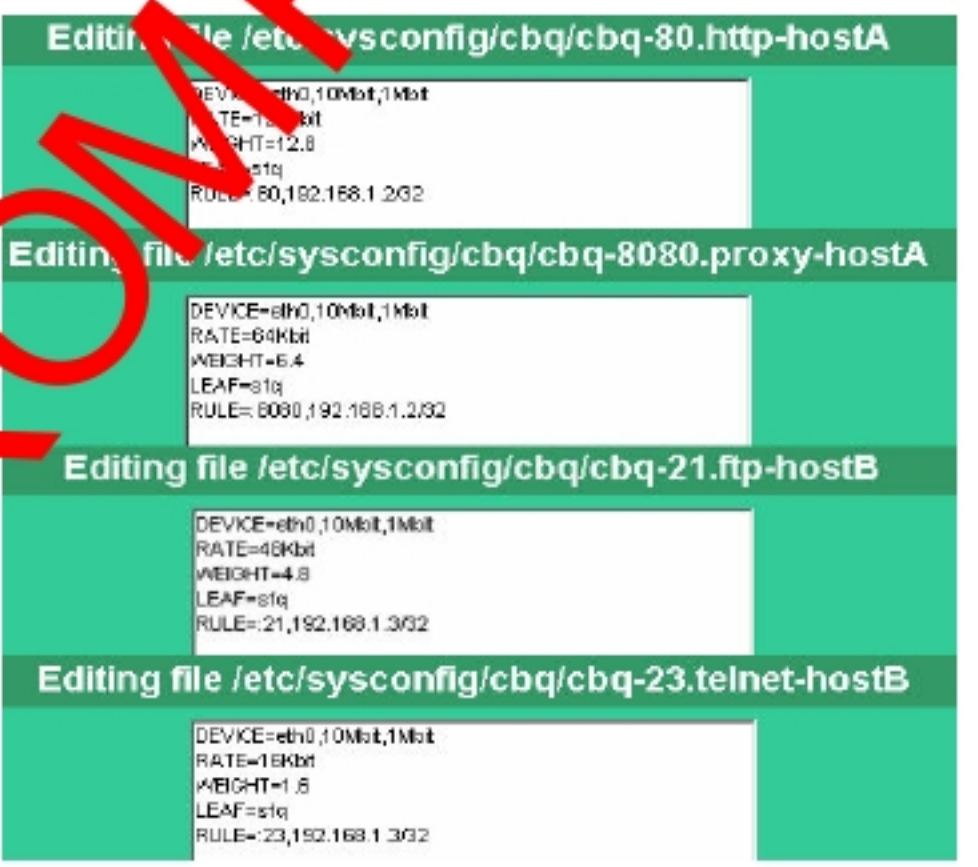
C. Dengan empat *leaf* kelas berdasarkan *Port Number*

Pada skenario ini pembagian dan pembatasan bandwidth dengan mengalokasikan bandwidth berdasarkan pada nomor *port* yakni port 80 dan port 21, dimana port 80 (*class 80*) merupakan layanan HTTP (*Hyper Text Transfer Protocol*) dengan alokasi bandwidth sebesar 192Kbit/s dan port 21 (*class 21*) yang merupakan layanan FTP (*File Transfer Protocol*) dengan alokasi bandwidth sebesar 64Kbit/s. Berikut gambaran skenario dengan dua *leaf* kelas (kelas cabang) :



Gambar 4.35 Skenario Empat Leaf Kelas Berdasarkan Port Number

Konfigurasi masing-masing kelas leaf berdasarkan Port Number yang dibuat pada aplikasi pembatasan dan pengaturan bandwidth dengan CBQ adalah sebagai berikut :

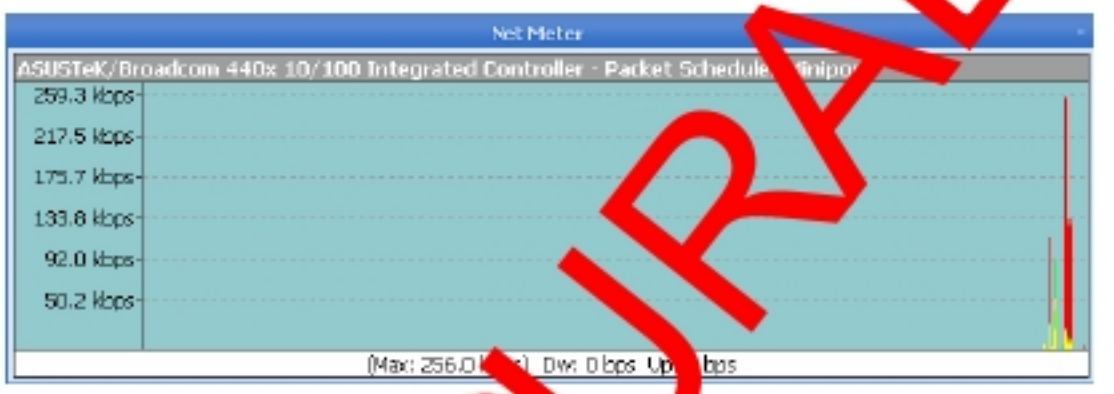


Gambar 4.36 Konfigurasi Kelas CBQ dengan 4 leaf kelas Port Number

Dari skenario diatas dapat dilakukan pengujian dan analisa sebagai berkut :

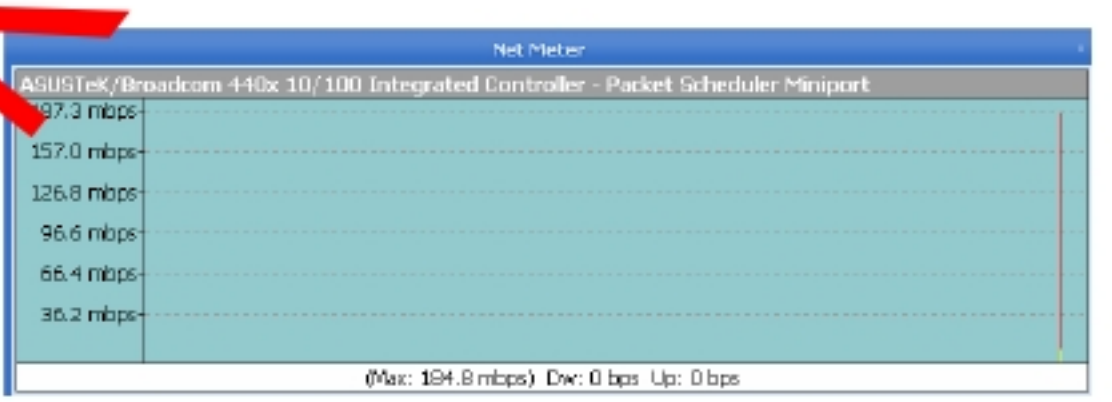
- Throughput

Dari grafik pada gambar 4.33, yang diperoleh dari komputer client dengan IP 192.168.1.2 terlihat bandwidth aktual adalah 133,8 kbps hal ini berarti terjadi kebocoran bandwidth sebesar 5,8 kbps dari bandwidth yang dialokasikan sebesar 128 kbps hal ini disebabkan layanan pada IP 192.168.1.3 dalam keadaan *idle* maka layanan 80 dapat meminjam bandwidth.



Gambar 4.37 Skenario 4 Leaf Kelas Port 80 IP 192.168.1.2

Sedangkan grafik yang diperoleh dari komputer client dengan IP 192.168.1.3 dengan mengakses layanan yang sama terlihat diatas 184 kbps berarti tidak terjadi kebocoran bandwidth namun terdapat traffik loncatan (*spike*) hal ini disebabkan adanya parameter *leaf* yang menyebabkan kelas IP 192.168.1.3 dapat meminjam bandwidth *idle* dari kelas IP 192.168.1.2. Berikut tampilan grafiknya adalah sebagai berikut :



Gambar 4.38 Skenario 4 Leaf Kelas Layanan 21 IP 192.168.1.3

- Loss datagram

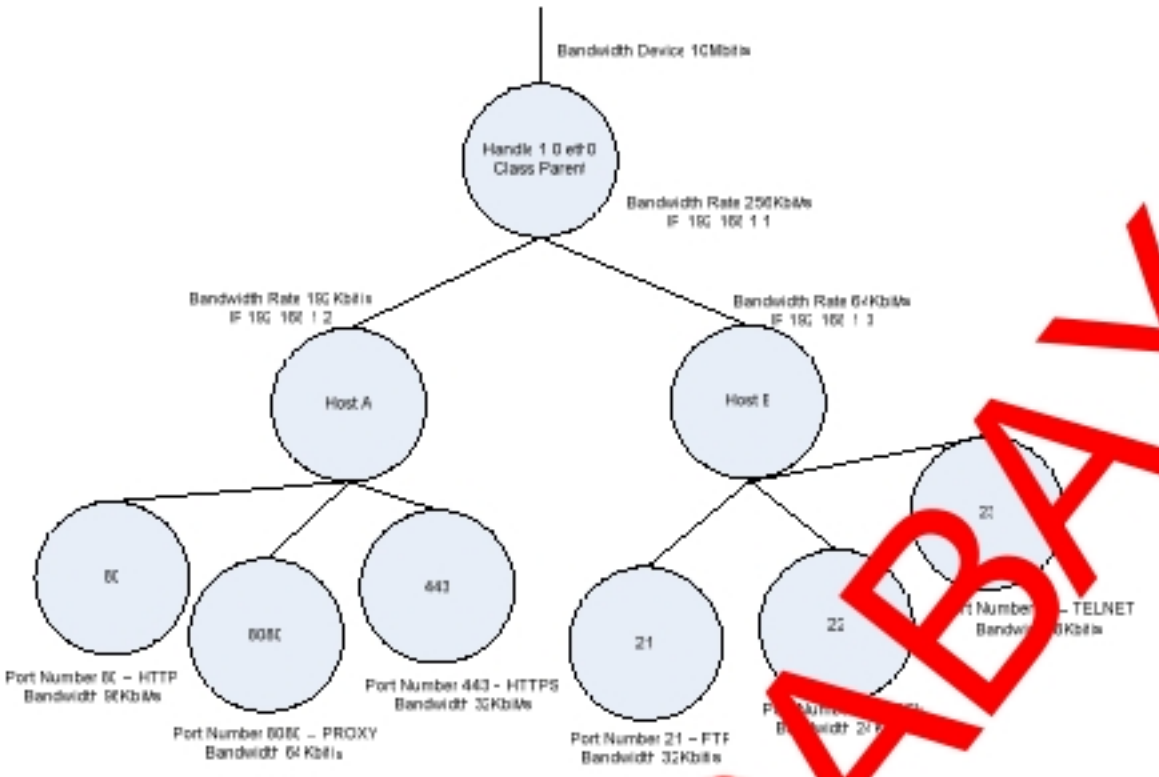
Pada IP 192.168.1.2 dan 192.168.1.3 setelah dilakukan pengiriman paket ping sebanyak 30 kali masing-masing sebesar 64 Bytes hasil yang diperoleh adalah tidak terjadi loss datagram berarti semua paket ping diterima dengan total jumlah yang sama seperti dikirim oleh komputer server

- Respon Time dan Round Trip Time

Jumlah *respon time* (waktu) dan *round trip time* (rtt) untuk IP 192.168.1.2 ketika dikirim 30 paket ping yang masing-masing berjumlah 64 byte maka jumlah respon time adalah 28995ms dan round trip time adalah min/avg/max/mdev = 0,199/0,206/0,273/0,016*miliseconds*, sedangkan untuk IP 192.168.1.3 jumlah respon time adalah 28995ms dan round trip time adalah min/avg/max/mdev = 0,148/0,202/0,254/0,016 *miliseconds*.

D. Dengan enam *leaf* kelas berdasarkan *Port Number*

Pada skenario ini pembagian dan pembatasan bandwidth dengan mengalokasikan bandwidth berdasarkan pada nomor *port* yakni port 80 dan port 21, dimana port 80 (*class 80*) merupakan layanan HTTP (*Hyper Text Transfer Protocol*) dengan alokasi bandwidth sebesar 192Kbit/s dan port 21 (*class 21*) yang merupakan layanan FTP (*File Transfer Protocol*) dengan alokasi bandwidth sebesar 64Kbit/s. Berikut gambaran skenario dengan dua *leaf* kelas (kelas cabang) :



Gambar 4.39 Skenario Enam Leaf Kelas Berdasarkan Port Number

Sedangkan konfigurasi kelas *leaf* pada aplikasi pembatasan dan pembagian bandwidth dengan CBQ adalah sebagai berikut :

Gambar 4.40 Konfigurasi Kelas CBQ dengan 6 kelas Port Number

Dari skenario diatas dapat dilakukan pengujian dan analisa sebagai berkut :

Skenario ini hampir sama dengan skenario dengan 4 leaf kelas namun perbedaannya terletak pada loss datagram dimana pada skenario 3 leaf kelas datagram yang diperoleh adalah 3% sedangkan pada skenario ini terjadi *loss datagram* sebesar 10% untuk IP 192.168.1.2 dan IP 192.168.1.3 hal ini dikarenakan ketidakmampuan komputer client menerima datagram ang dikirimkan melewati CBQ secara simultan dengan beberapa port dibuka untuk menerima layanan. Dan jumlah respon time (time) dan round trip time (rtt) untuk IP 192.168.1.2 ketika dikirim 30 paket ping yang masing-masing berjumlah 64 byte maka jumlah respon time adalah 40326ms dan round trip time adalah min/avg/max/mdev = 440,087/360,017/380,182/15,341*miliseconds*, sedangkan untuk IP 192.168.1.3 jumlah respon time adalah 40278ms dan round trip time adalah min/avg/max/mdev = 950,089/1090,212/1150,100/48,719 *miliseconds*

4.3.2 Skenario Parameter

Skenario ini menunjukan penggunaan parameter yang mematikan fungsi link sharing berarti kelas kelas CBQ yang sedang melayani antrian hanya dapat menggunakan bandwidth sebatas yang dialokasikan kepadanya. Bandwidth pada kelas CBQ yang idle tidak dapat digunakan untuk meningkatkan rate bandwidth kelas yang sedang menerima antrian pesan.

Dengan parameter "tidak meminjamkan tidak *bandwidth idle*"

Skenario ini pada dasarnya membagi alokasi bandwidth berdasarkan alamat IP (*Internet Protocol*), dengan alokasi bandwidth sebagai berikut IP 192.168.1.2 dengan bandwidth 192Kbit/s dan IP 192.168.1.3 dengan bandwidth 64Kbit/s. Pada gambar 4.35 terlihat pada masing-masing kelas parameter *LEAF* diset *tbf*,

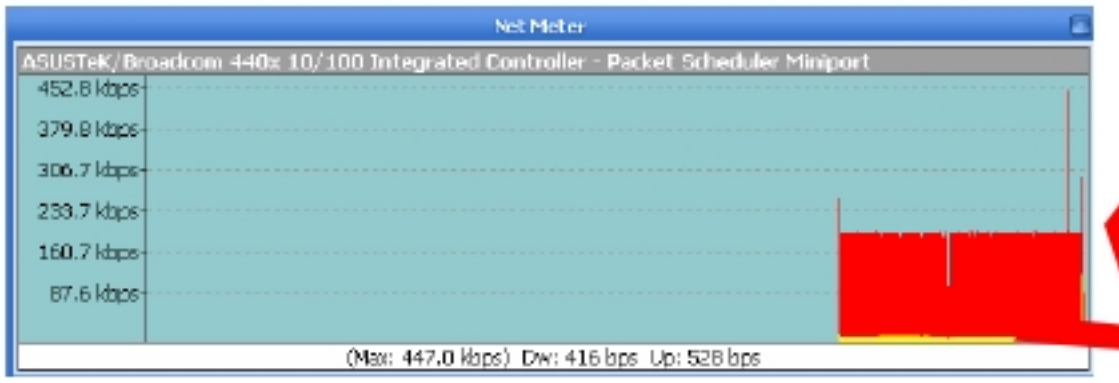
parameter *BOUNDED* diset *no*, serta parameter *ISOLATED* diset *yes*, hal ini menerangkan bahwa kelas ini tidak akan meminjamkan bandwidthnya kepada kelas yang lain meskipun pada kondisi tertentu sebuah kelas jika tidak digunakan maka tedapat bandwidth *idle* yakni bandwidth yang alokasikan tidak terpakai. Berikut ini adalah konfigurasi dari masing-masing kelas CBQ :

Gambar 4.41 Konfigurasi Kelas CBQ Tidak Meminjamkan Bandwidth Idle

Dari skenario diatas dapat dilakukan pengujian dan analisa sebagai berkut :

- Throughput

Dari grafik pada gambar 4.42, yang diperoleh dari komputer client dengan IP 192.168.1.2 terlihat bandwidth aktual sekitar 192 kbps hal ini berarti tidak terjadi kebocoran bandwidth 192 kbps hal ini disebabkan kelas tidak dapat meminjamkan bandwidthnya dari kelas/layanan yang lain dari IP 192.168.1.3 yang merupakan efek dari parameter BOUNDED dan ISOLATED.



Gambar 4.42 Skenario Tidak Meminjam Bandwidth IP 192.168.

Sedangkan grafik yang diperoleh dari komputer client dengan IP 192.168.1.3 dengan mengakses layanan yang sama terlihat diatas 64 kbps berarti tidak terjadi kebocoran bandwidth hal ini merupakan efek dari parameter BOUNDED dan ISOLATED yang menyebabkan kelas tidak dapat meminjam bandwidth dari kelas yang lain. Berikut tampilan grafiknya adalah sebagai berikut :

Gambar 4.43 Skenario Tidak Meminjam Bandwidth IP 192.168.1.3

- Loss Datagram

Pada skenario ini terjadi *loss datagram* sebesar 3% untuk IP 192.168.1.2 dan IP 192.168.1.3 hal ini dikarenakan tidak ada peminjam meminjam bandwidth idle antara kedua kelas IP tersebut

- Respon Time dan Round Trip Time

Jumlah respon time (time) dan round trip time (rtt) untuk IP 192.168.1.2 ketika dikirim 30 paket ping yang masing-masing berjumlah 64 byte maka jumlah respon time adalah 30306ms dan round trip time adalah min/avg/max/mdev =

340,087/360,017/380,182/13,341*miliseconds*, sedangkan untuk IP 192.168.1.3 jumlah respon time adalah 30278ms dan round trip time adalah min/avg/max/mdev = 950,089/1090,712/1150,100/48,719 *miliseconds*

4.3.3 Skenario Topologi

Secara *default* sebuah komputer dengan sistem operasi linux dengan menggunakan aplikasi *iptables* yang diset parameternya sehingga dapat berfungsi sebagai *router (Linux PC Router).* Sedangkan untuk melakukan fungsi sebagai *bridge (Linux PC Bridge*) diperlukan paket tambahan (http://bridge.sourceforge.net) yang harus diinstall (*kompilasi*) pada linux *kernel*. Dengan memanfaatkan kedua fungsi ini penelitian melakukan uji dengan dua topologi yakni komputer server berfungsi sebagai *router* dan sebagai *bridge*. Skenario dengan topologi jaringan yang berbeda, mengalokasikan bandwidth berdasarkan alamat IP 192.168.1.2 mendapat alokasi *bandwidth* sebesar 32 kbps dan IP 192.168.1.2 mendapat alokasi *bandwidth* sebesar 16 kbps (*kilo bytes per second*) dan pesan yang dikirim dari server sebesar 151 Kbytes.

A. Sebagai *Router*

Dari hasil uji dengan topologi *router* analisa yang dapat sebagai berikut :

- Throughput

Bandwidth aktual (*throughput*) pada IP 192.168.1.2 sesuai dengan *rule* pembatasan bandwidth pada kelas CBQ yakni rate adalah 32kbps, sedangkan *throughput* pada IP 192.168.1.3 tidak sesuai dengan definisi *rule* pada kelas CBQ yakni 18.5kbps.

- Waktu pemrosesan pesan dalam bandwidth

Tabel 4.1 Waktu Pemrosesan Pesan Topologi Router

IP Address	t*Router* (detik)	t*non-router* (detik)	t(*x*) (detik)	t*delay-router* (detik)	T*bm* (detik)
192.168.1.2	36.4	35.634	0.7658	0.0002	0.7656
192.168.1.3	38.2	37.28	0.9160	0.0002	0.9158

Dimana rumus perhitungan yang digunakan untuk menghitung waktu pemrosesan pesan pada topologi *router* adalah sebagai berikut (Floyd, Sally,1998:17)

Waktu proses pesan pada PC-Router t(x) = t*delay-Router* + T*bm*

= 0.0002 detik + T*bm*

Waktu proses pesan melalui PC-Router = t*Router* + T*bm*

Waktu proses pesan tanpa PC-Router = t*non-Router*

$$= \frac{packet\_size \times 8}{line\_speed} \times 1000 \text{ ms}$$

t(x) = t*Router* – t*non-Router*

- Loss Datagram

Pesan yang dikirim ke komputer server melalui CBQ diterima seluruhnya oleh komputer client sebanyak 105 *datagram*, hal ini berarti tidak terjadi *loss datagram* pada saat uji skenario pada topologi sebagai *router*.

B. Sebagai *Bridge*

Dari hasil uji dengan topologi *bridge* analisa yang dapat sebagai berikut :

- Throughput

Pola *throughput* yang dihasilkan pada skenario ini hampir sama dengan skenario menggunakan router. Dimana throughput pada IP 192.168.1.2 sesuai dengan alokasi bandwidth yang diberikan yakni 32 kbps, sedangkan pada IP 192.168.1.3 tidak sesuai dengan rule yakni rata-rata adalah 18.5 kbps hal ini membuktikan

terjadi kebocoran pemakaian bandwidth dengan skenario sebagai bridge pada salah satu komputer client.

- Waktu pemrosesan pesan dalam bandwidth

Tabel 4.2 Waktu Pemrosesan Pesan Topologi Bridge

IP Address	t*Bridge* (detik)	t*non-bridge* (detik)	t(*y*) (detik)	t*delay-bridge* (detik)	T*bm* (detik)
192.168.1.2	36.4	35.634	0.7658	0.0001	0.7657
192.168.1.3	38.2	37.28	0.9160	0.0001	0.9159

Dimana rumus perhitugan yang digunakan untuk menghitung pemrosesan pesan pada tolopolgi *bridge* adalah sebagai berikut (Floyd Sally, 1998:18) :

Waktu proses pesan pada PC-Bridge (ty) = *delay-Bridge* + T*bm*

= 0.0001 detik + T*bm*

Waktu proses pesan melalui PC-Bridge = t*Bridge* + T*bm*

Waktu proses pesan tanpa PC-Bridge = t*non-Bridge*

$$= \frac{packet\_size \times 8}{line\_speed} \times 1000 \text{ ms}$$

t(y) = t*Bridge* – t*non-Bridge*

- Loss Datagram

Pesan yang dikirim ke komputer server melalui CBQ diterima seluruhnya oleh komputer client sebanyak 105 *datagram*, hal ini berarti tidak terjadi *loss datagram* pada saat uji skenario pada topologi sebagai *bridge* hal ini sama seperti skenario topologi sebagai *router*.