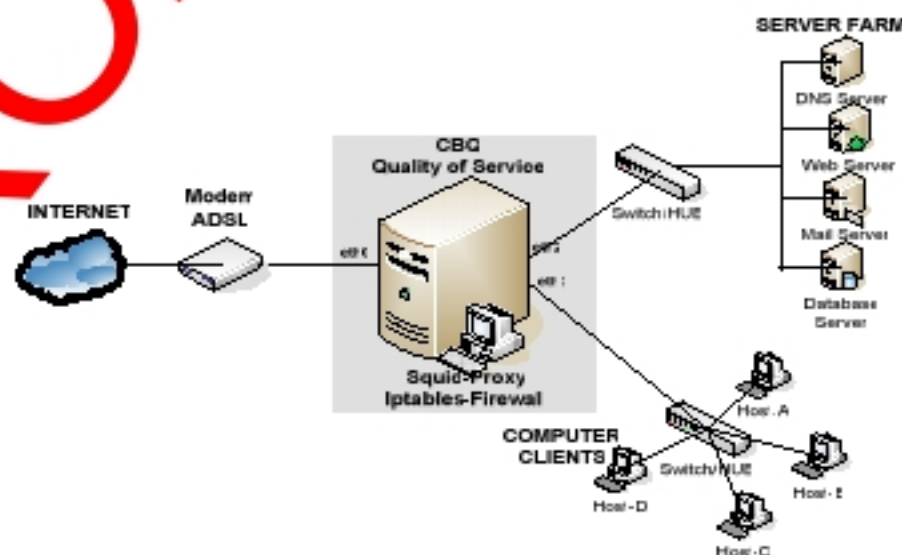


BAB III

PERANCANGAN SISTEM

3.1 Analisa Permasalahan

Manajemen bandwidth memegang peranan penting dalam mengatur jenis aplikasi yang bisa mengakses koneksi yang ada selain itu manajemen bandwidth mampu memberikan garansi kepada aplikasi yang mendapat alokasi bandwidth untuk terus mengirimkan data sesuai dengan alokasinya sekalipun terjadi kemacetan dalam jaringan bahkan dalam keadaan tertentu ketika alokasi bandwidth yang dimiliki oleh suatu aplikasi/kegiatan tidak digunakan maka dapat dialihkan sementara waktu kepada kelas yang sedang mengalami *backlog*/timbunan antrian, hal ini memberikan keuntungan mempercepat hilangnya *backlog* suatu kelas sekaligus mengoptimalkan penggunaan koneksi yang ada. CBQ (*Class Based Queuing*) sebagai implementator manajemen bandwidth yang tersedia secara gratis dan dapat dijalankan pada sistem operasi Linux. Konsep CBQ direpresentasikan dalam script (*cbq.init*) yang akan pada jaringan komputer seperti pada gambar 3.1 berikut :



Gambar 3.1 Arsitektur Sistem dan Aplikasi CBQ QoS

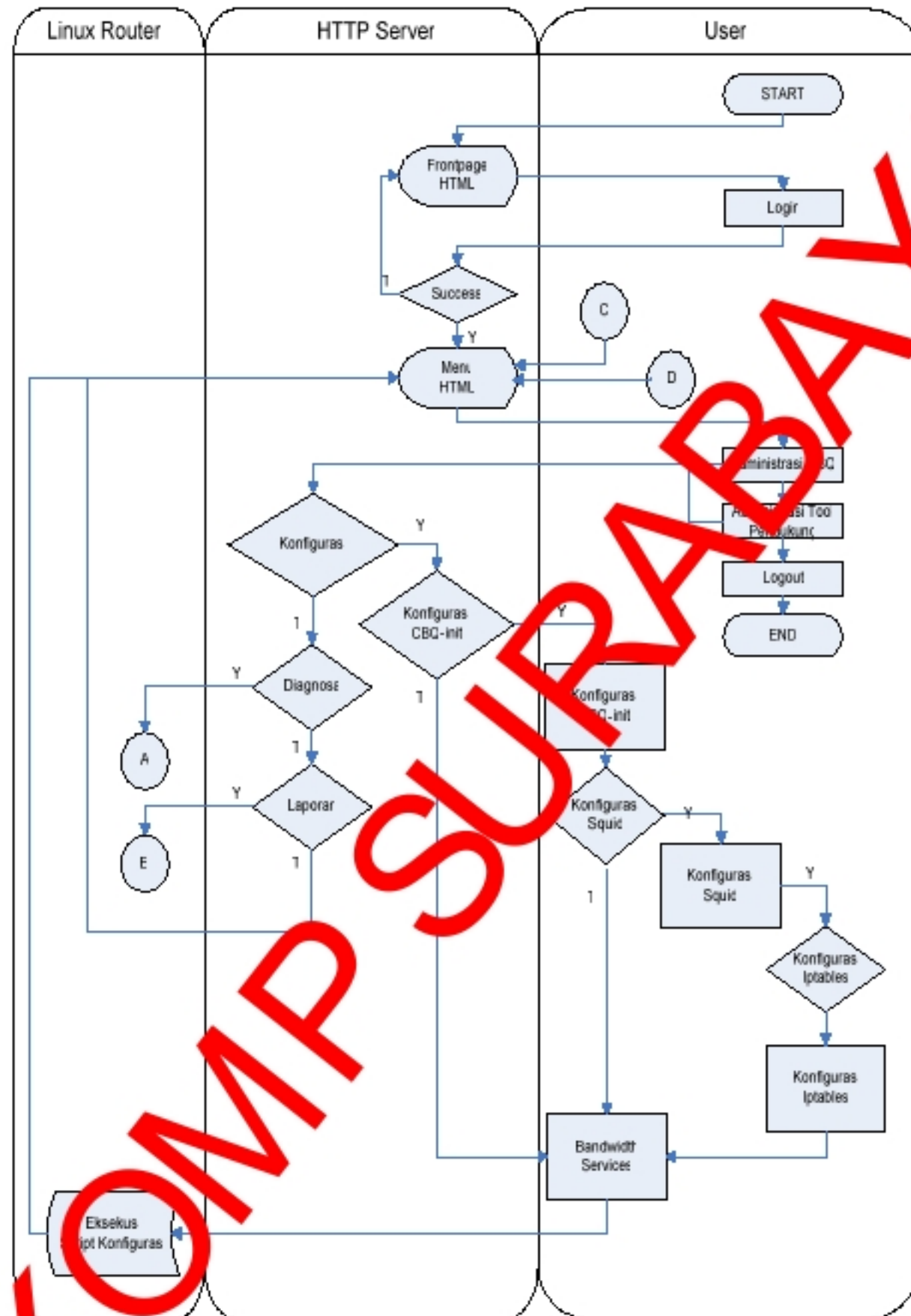
Untuk melakukan pembatasan dan pengaturan bandwidth pengelola jaringan harus masuk "login" ke mesin *router* atau *gateway* secara langsung maupun tidak langsung (*remote*) agar dapat mengkonfigurasi *cbq-init* beserta kelasnya guna membatasi maupun mengatur bandwidth untuk masing-masing layanan jaringan untuk masing-masing komputer client (*host*) pada jaringan komputer yang dikelolanya

3.1.1 Proses Perancangan Sistem

Pada aplikasi yang dibuat sebagai tugas akhir ini dalam mengkonfigurasi CBQ script dapat dilakukan via *web-based* begitu juga dengan *tool* pendukung *squid* dan *iptables*. Pengelola jaringan tidak lagi harus melakukan login langsung maupun tidak langsung (*remote*) ke mesin linux *router/gateway* untuk melakukan konfigurasi CBQ script namun hanya cukup melakukan login pada aplikasi untuk melakukan proses administrasi dan konfigurasi dari CBQ dan *tool* pendukung dari aplikasi..

Untuk menjalankan aplikasi berbasis web diperlukan sebuah web server, pemilihan apa sebagai web server karena *reliable* (handal) dan sangat mendukung sistem operasi linux dan juga dapat diletakkan pada mesin *router/gateway*. Di samping itu untuk menambah performa dari CBQ diperlukan *tool* pendukung antara lain *squid* sebagai proxy dan *iptables* sebagai *firewall* untuk mengamankannya.

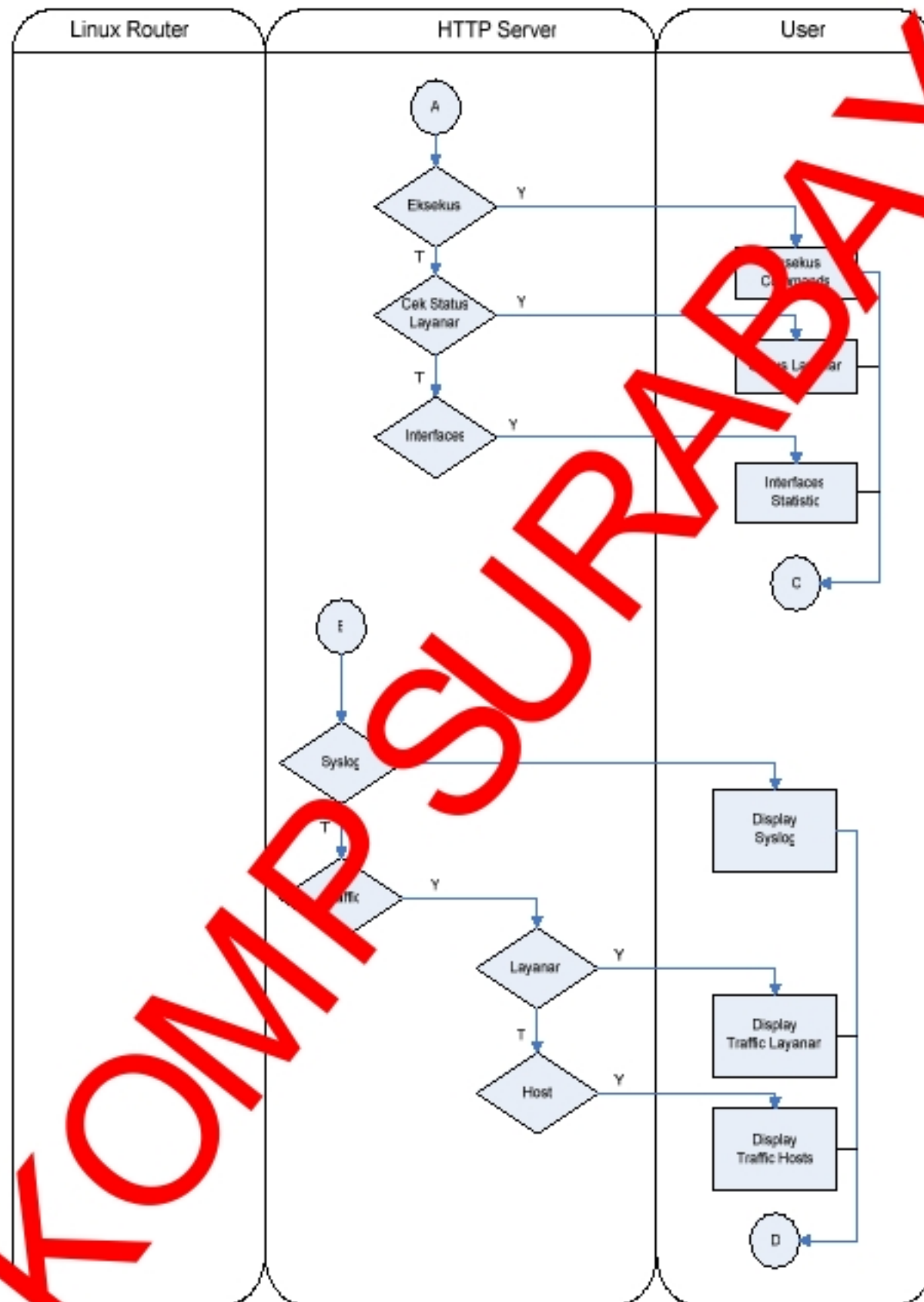
Alur dari sistem ditampilkan pada *flow diagram* pada gambar 3.2 dan 3.3, dimana pengguna (*user*) melakukan akses pada web server dimana aplikasi pembatasan dan pengaturan bandwidth diletakkan.



Gambar 3.2 Flow Diagram Sistem yang akan dibuat

Pengguna akan melakukan konfigurasi dan administrasi dari aplikasi dan jika sistem berbasis web, konfigurasi dan adminstrasi dapat dilakukan pada CBQ script dan juga *tool* pendukung yakni squid dan iptables dan untuk mengetahui konfigurasi yang dilakukan sesuai dengan yang diinginkan pengguna akan

melakukan diagnosa dan juga melihat file *log* dari konfigurasi dan administrasi yang dilakukan dengan mengeksekusi perintah via-web



Gambar 3.3 Flow Diagram Sistem yang akan dibuat (lanjutan)

Setelah melakukan administrasi sistem dan konfigurasi dilakukan dan aplikasi serta sistem sudah berjalan maka pengguna dapat memantau (*monitoring*) layanan-layanan yang digunakan oleh pengguna pada komputer client (*host*). Hal-

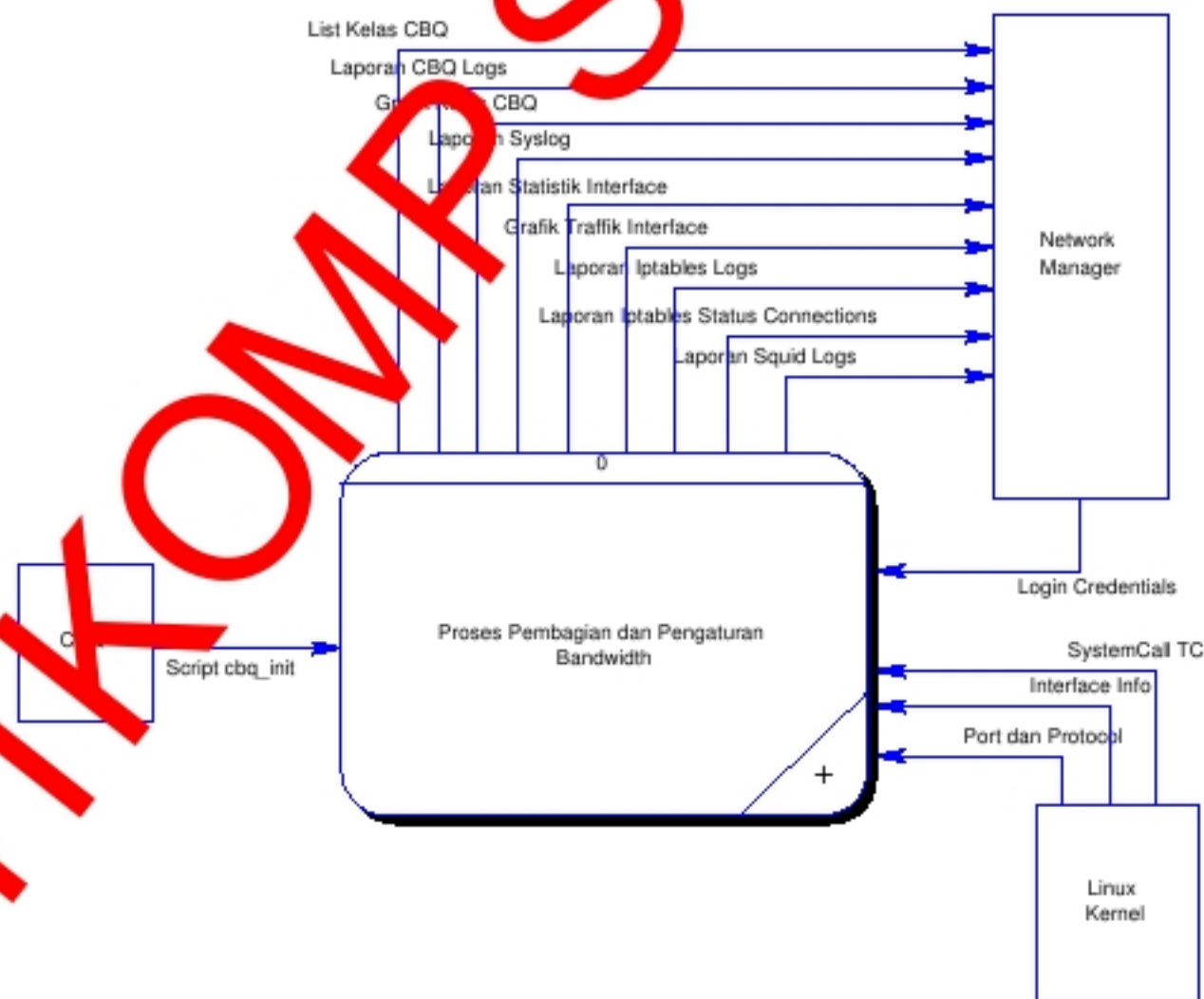
hal yang dapat dipantau yakni besarnya traffic penggunaan bandwidth jaringan, besarnya traffic penggunaan bandwidth tiap kelas layanan CBQ apakah sesuai dengan administrasi dan konfigurasi yang dilakukan sebelumnya.

3.2 Data Flow Diagram (DFD)

Langkah selanjutnya dalam perancangan sistem adalah pembuatan data flow diagram. DFD adalah representasi grafik dari suatu sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran aliran data diantara komponen-komponen, dan asal, tujuan juga penyimpanan dari data tersebut.

3.2.1 Context Diagram

Berikut ini adalah gambar *Context diagram* dari aplikasi pembatasan dan pengaturan bandwidth dengan Class Based Queueing (CBQ).

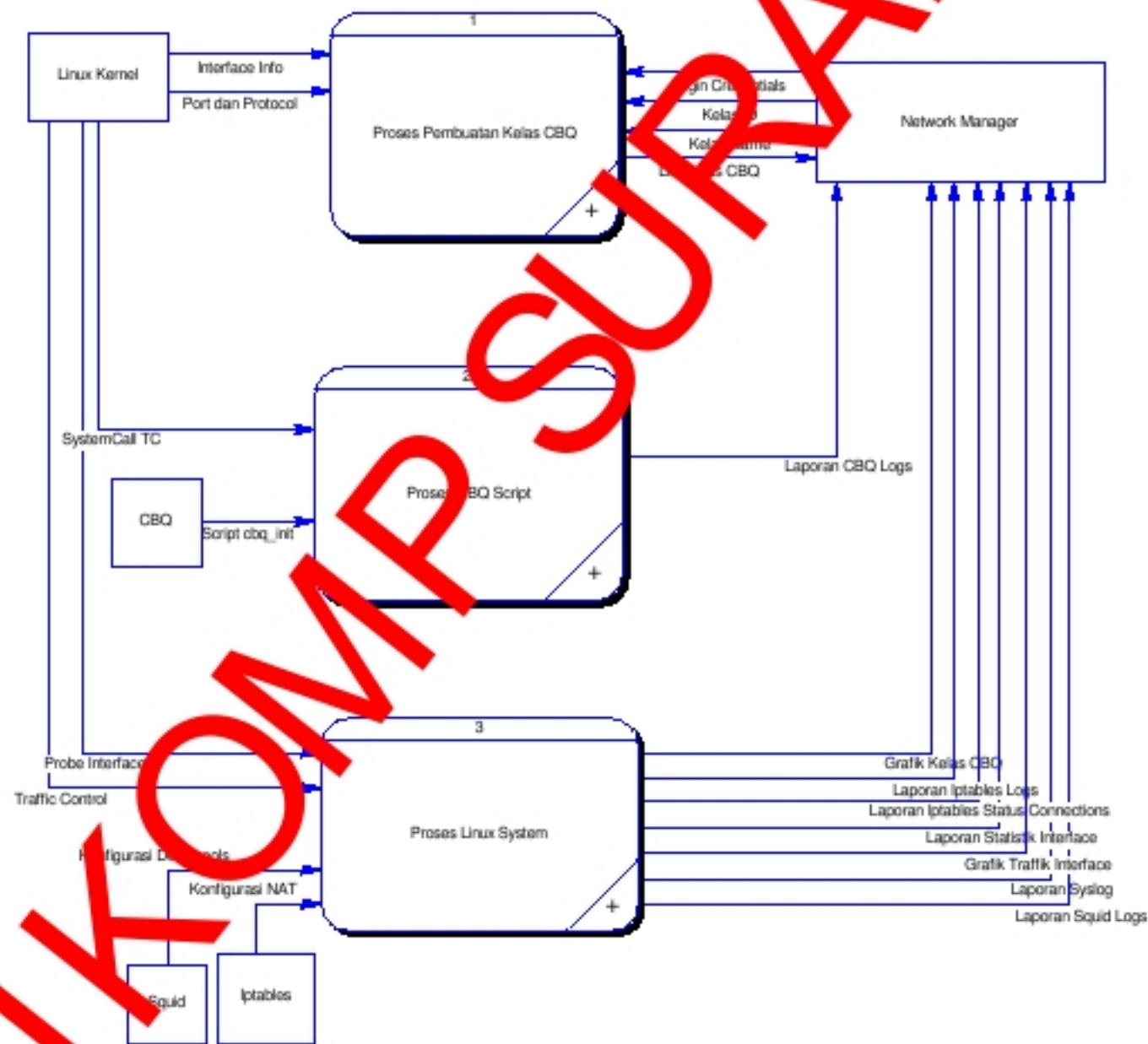


Gambar 3.4 Context Diagram

Context Diagram merupakan level paling dari suatu DFD. Pada Gambar 3.4 dari *Context Diagram* terdapat tiga buah entity yang berperan dalam aplikasi pembatasan dan pengaturan bandwidth yakni *Network Manager*, *CBQ*, dan *Linux Kernel*.

3.2.2 DFD Level 0 Proses Pembatasan dan Pengaturan Bandwidth

Berikut ini adalah DFD level 0 dari proses aplikasi pembatasan dan pengaturan bandwidth :



Gambar 3.5 DFD Level 0 Proses Pengaturan dan Pembatasan Bandwidth

Seperti yang terlihat pada gambar 3.5, DFD level 0 memiliki tiga proses yang mempunyai peran sendiri-sendiri dalam proses pengaturan dan pembatasan

bandwidth. Berikut ini adalah proses-proses yang terdapat pada DFD level 0 antara lain :

1. Proses Pembuatan Kelas CBQ

Proses Pembuatan Kelas CBQ oleh *Network Manager* setelah melakukan login pada aplikasi proses menerima inputan data CBQ kelas *ID* dan kelas *Name*, disamping itu *Network Manager* sebagai pengguna menerima output berupa daftar dari kelas-kelas CBQ yang berupa file-file kelas yang tersimpan pada directory *"/etc/sysconfig/cbq/"* sistem operasi linux. Proses juga menerima inputan dari Linux Kernel berupa informasi interface/NIC (*Network Interface Card*). Inputan lain adalah data *Port (Port Number & Port Name)* dan *Protocol* yang berguna untuk menentukan layanan jaringan yang akan dilakukan pembatasan dan pengaturan bandwidth-nya.

2. Proses CBQ Script

Proses CBQ Script merupakan proses mengeksekusi script *cbq.init* oleh linux kernel dalam hal ini *System Call TC (tc command)*. Proses ini dilakukan untuk mengeksekusi file-file kelas CBQ yang diinputkan dan untuk mengetahui hasil dari proses eksekusi ini terdapat laporan CBQ log sehingga network manager dapat mengetahui hasil konfigurasi yang dilakukan pada file-file kelas CBQ.

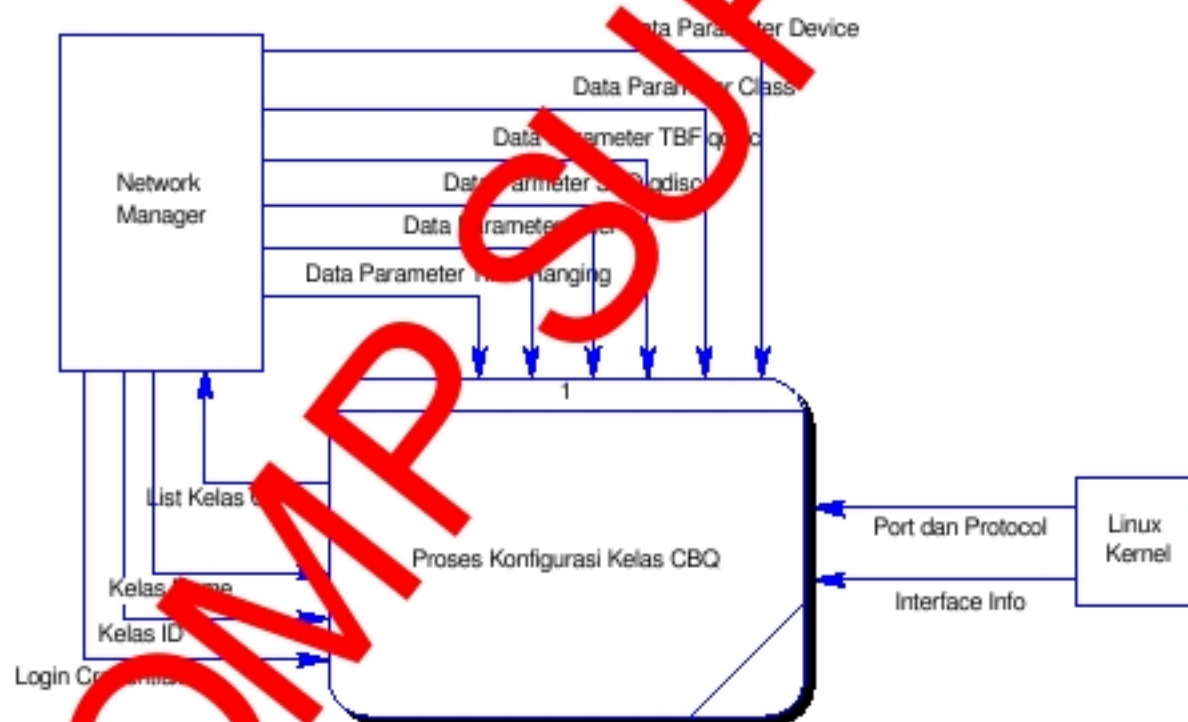
3. Proses Linux System

Pada proses ini entiti linux kernel memberikan inputan berupa probe (infomasi module) interface/NIC (*Network Interface Card*) dan mekanisme traffic control. Traffic control akan diproses dengan inputan hasil eksekusi dari *script cbq.init* dari proses CBQ script sebelumnya untuk menghasilkan laporan traffik dan grafik dari kelas CBQ dan interface jaringan. Data dari entiti linux kernel akan bergun

untuk proses dari aplikasi pendukung yakni squid dan iptables yang memberikan inputkan kepada proses berupa konfigurasi *delaypool* squid dan konfigurasi NAT (*Network Translation Address*) iptables. Dan untuk mengetahui hasil dari konfigurasi aplikasi ada laporan berupa log dari squid dan iptables sebagai keluaran kepada entiti network manager. Dan laporan dari proses sistem linux ini berupa laporan Syslog (*System Log*).

3.2.3 DFD Level 1 Proses Pembuatan Kelas CBQ

Berikut ini adalah DFD level 1 dari proses pembuatan kelas pada aplikasi pembatasan dan pengaturan bandwidth :



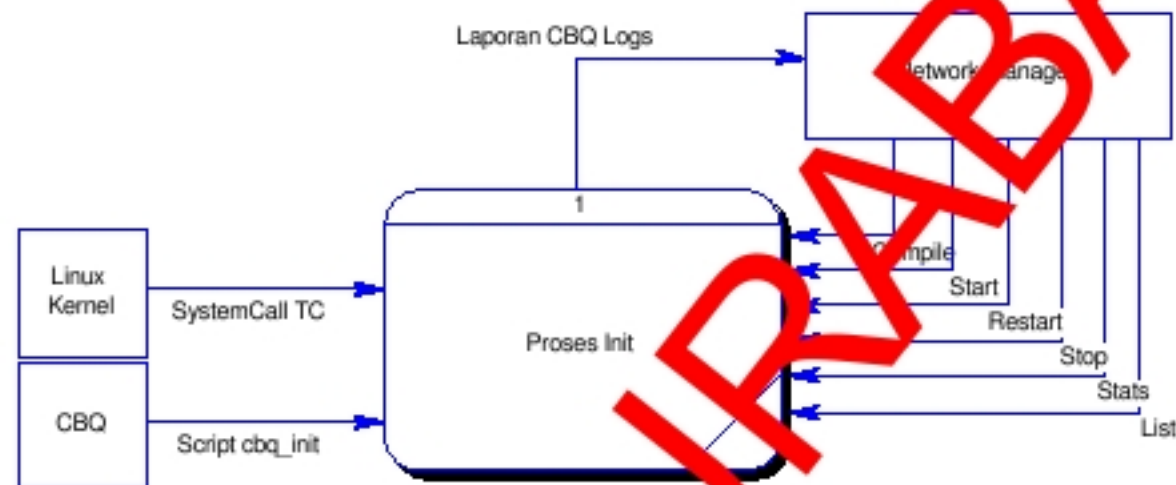
Gambar 3.6 DFD Level 1 Proses Pembuatan Kelas CBQ

Seperti terlihat pada gambar 3.6 DFD level 1 dari proses pembuatan kelas CBQ memiliki sebuah proses yaitu proses konfigurasi kelas CBQ. Proses ini merupakan proses lanjutan dari proses pembuatan kelas CBQ dimana setelah entiti Network Manger menginputkan data kelas *ID* dan kelas *Name* dilanjutkan dengan menginputkan data pada masing-masing parameter yang dimiliki oleh CBQ pada

kelas disesuaikan dengan kebutuhan dan pemakaian oleh pengguna jaringan komputer.

3.2.4 DFD Level 1 Proses CBQ Script

Berikut ini adalah DFD level 1 dari proses CBQ script pada aplikasi pembatasan dan pengaturan bandwidth :

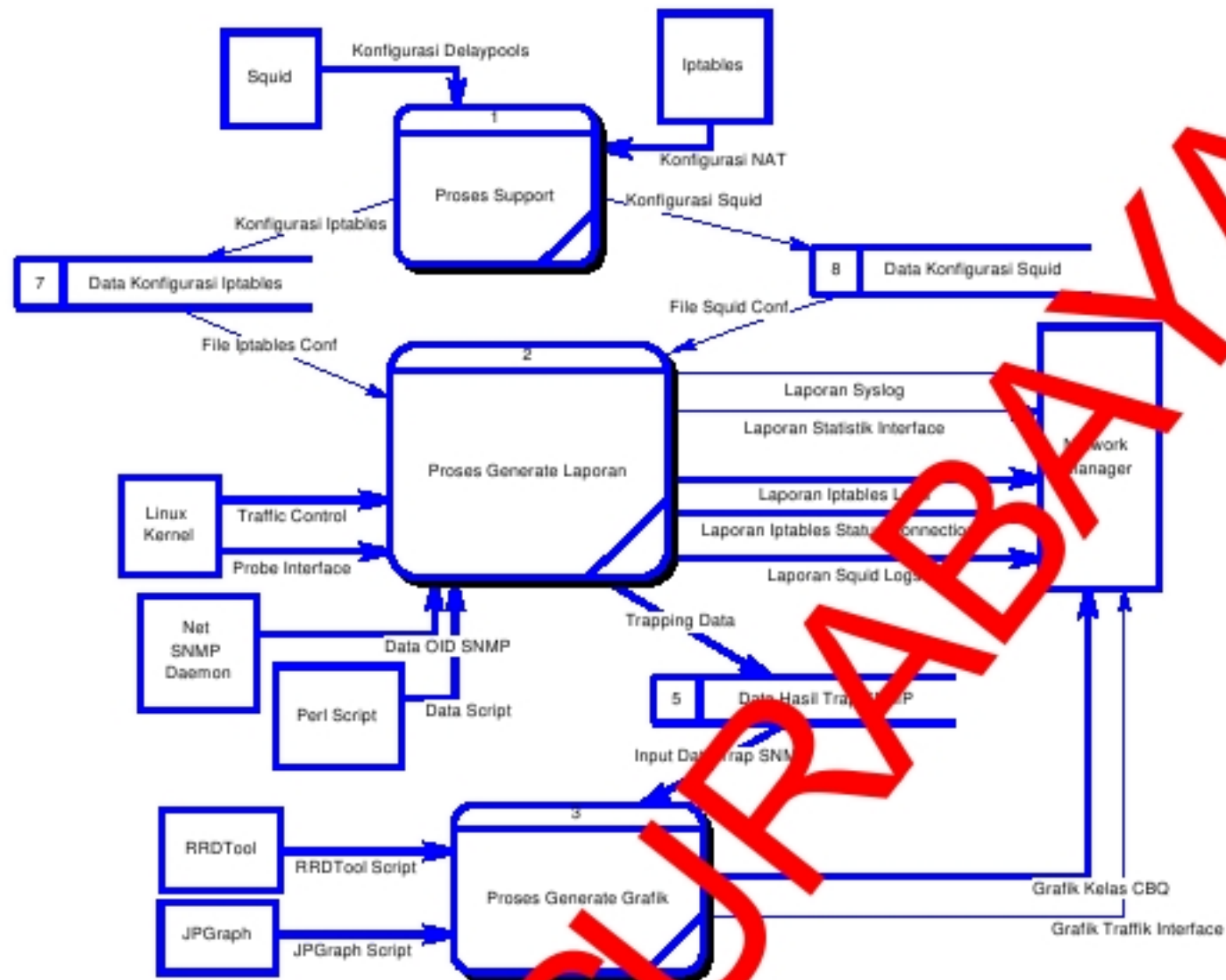


Gambar 3.7 DFD Level 1 Proses CBQ Script

Seperti terlihat pada gambar 3.7 DFD level 1 dari proses CBQ script memiliki sebuah proses yakni proses init. Proses ini untuk mengetahui apakah konfigurasi pada penempatan masing-masing kelas CBQ sudah benar hal ini dapat dilakukan dengan menginputkan data *compile*, disamping itu untuk melihat status dan daftar dari kelas CBQ setelah dieksekusi oleh systemcall tc pada kernel sistem operasi linux dan juga untuk menjalankan serta menghentikan proses init tersebut.

3.2.5 DFD Level 1 Proses Linux System

Berikut ini adalah DFD level 1 dari proses Linux System pada aplikasi pembatasan dan pengaturan bandwidth :



Gambar 3.8 DFD Level 1 Proses Linux System

Seperti yang terlihat pada gambar 3.8, DFD level 1 proses Linux System memiliki tiga proses yang mempunyai peran sendiri-sendiri dalam proses pengaturan dan manajemen bandwidth. Berikut ini adalah proses-proses yang terdapat pada DFD level 1 proses Linux System antara lain :

1. Proses Support

Proses support merupakan proses penyimpanan konfigurasi aplikasi pendukung CBQ. Squid dengan parameter *delaypools*-nya dan iptables dengan konfigurasi NAT (*Network Translation Address*).

2. Proses Generate Laporan

Proses Generate Laporan merupakan proses pembuatan laporan dengan memanfaatkan protokol SNMP (*Simple Network Management Protocol*) pada

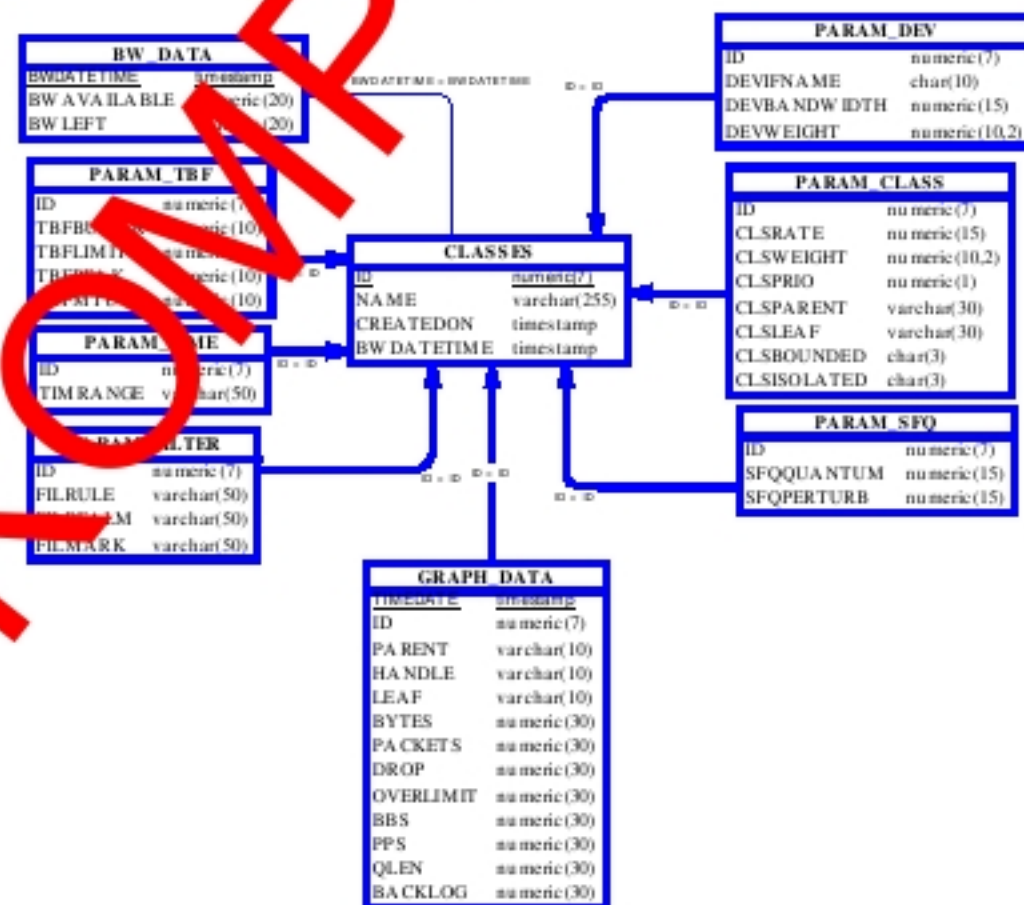
aplikasi Net-SNMP dan untuk mengambil data (*trapping data*) secara real-time dengan dibantu oleh script Perl untuk memilah data-data tersebut menjadi sebuah informasi laporan kepada network manager.

3. Proses Generate Grafik

Proses Generate Grafik adalah proses membuat grafik dari data-data yang diinputkan dari hasil trapping data SNMP. Tool yang digunakan untuk membuat grafik adalah *RRDTool* dan *JPGraph*. Grafik yang dihasilkan adalah grafik trafik interface dan grafik trafik masing-masing kelas CBQ.

3.3 Entity Relation Diagram (ERD)

ERD digunakan untuk menginterpretasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan database. Berikut ini adalah ERD dari sistem pembatasan dan pengaturan bandwidth dengan CBQ (*Class Based Queueing*) pada sistem operasi Linux yang akan dibuat :



Gambar 3.9 ERD Sistem Pengaturan dan Pembatasan Bandwidth

Pada Gambar 3.9 merupakan ERD yang digunakan untuk membuat sistem pembatasan dan pembagian bandwidth dengan CBQ pada sistem operasi linux berbasis web. ERD pada sistem ini terdapat 9 entity yang nantinya merupakan tabel-tabel database dari sistem yang dibangun.

3.4 Struktur Database

Struktur database yang digunakan dalam sistem ini berdasarkan ERD yang telah dibuat dan akan digunakan dalam program aplikasi.

1. Tabel Data Bandwidth

Nama tabel : BW_DATA

Fungsi : Untuk menyimpan data jumlah bandwidth

Tabel 3.1 Tabel BW_DATA

Field	Type	Null	Key
BwDateTime	Date Time unsigned	NOT NULL	PRI
BwAvailable	Numeric(20)		
BwLeft	Numeric(20)		

2. Tabel Data CBQ Class parameter

Nama tabel : CLASSES

Fungsi : Untuk menyimpan kelas-kelas CBQ yang dibuat

Tabel 3.2 Tabel CLASSES

Field	Type	Null	Key
ID	Numeric(7) unsigned	NOT NULL	PRI
Name	Varchar(255)		
CreatedOn	Date Time		
BwDateTime	Date Time		FK

Tabel 3. 5 Tabel PARAM_TBF

Field	Type	Null	Key	Default
ID	Numeric(7) unsigned NOT NULL		PRI, FK	
TbfBuffer	Numeric(10)			1
TbfLimit	Numeric(10)			15
TbfPeak	Numeric(10)			
TbfMTU	Numeric(10)			1500

6. Table Data CBQ SFQ qdisc parameter

Nama tabel : PARAM_SFQ

Fungsi : untuk menyimpan parameter-parameter SFQ qdisc CBQ

Tabel 3. 6 Tabel PARAM_SFQ

Field	Type	Null	Key	Default
ID	Numeric(7) unsigned NOT NULL		PRI, FK	
SfqQuantum	Numeric(10)			
SfqPerturb	Numeric(10)			10

7. Tabel Data CBQ Filter parameter

Nama tabel : PARAM_FILTER

Fungsi : Untuk menyimpan parameter-parameter Filter CBQ

Tabel 3. 7 Tabel PARAM_FILTER

Field	Type	Null	Key	Default
ID	Numeric(7) unsigned NOT NULL		PRI,FK	
FilterName	Varchar(50)			
FilterM	Varchar(50)			
FilterMark	Varchar(50)			5

8. Tabel Data CBQ Time Ranging parameter

Nama tabel : PARAM_TIME

Fungsi : Untuk menyimpan parameter Time Ranging CBQ

Tabel 3. 8 Tabel PARAM_TIME

Field	Type	Null	Key	Default
ID	Numeric(7) unsigned NOT NULL		PRI,FK	
TimRange	Varchar(50)			

9. Tabel Data Graph

Nama tabel : GRAPH_DATA

Fungsi : untuk menyimpan data-data yang berguna untuk membuat grafik trafik yang dihasilkan oleh CBQ

Tabel 3. 9 Tabel GRAPH_DATA

Field	Type	Null	Key
ID	Numeric(7) unsigned NOT NULL		PRI,FK
Parent	Varchar(10)		
Handle	Varchar(10)		
Leaf	Varchar(10)		
Bytes	Numeric(30)		
Packets	Numeric(30)		
Drop	Numeric(30)		
Overlimit	Numeric(30)		
BBS	Numeric(30)		
PPS	Numeric(30)		
Qlen	Numeric(30)		
Backlog	Numeric(30)		

3.5 Desain Input Output

Desain input output adalah suatu desain yang memberikan informasi tentang rancangan jalannya suatu sistem. Dengan desain input output ini dapat memudahkan dalam memahami jalannya suatu sistem. Sebelum desain input output terdapat diagram menu yang menjelaskan alur dari menu-menu serta fungsi yang terdapat pada aplikasi pembatasan dan pengaturan bandwidth.

3.5.1 Diagram Menu

Menu diagram pada aplikasi pembatasan dan pembagian bandwidth ini dibagi menjadi empat bagian, Menu CBQ, Menu Support, Menu Diagnostika-Log, dan Menu Monitoring. Berikut gambar diagram menu dari aplikasi pembatasan dan pengaturan bandwidth dengan CBQ (Class Based Queue) pada sistem operasi linux berbasis web.



Gambar 3.10 Diagram Menu

Menu CBQ merupakan menu utama dari aplikasi dimana dalam menu ini terdapat tiga sub-menu yakni CBQ-init, CBQ-Traffic, CBQ-Logs, sub-menu CBQ-init untuk membuat kelas dan merubah dan menghapus kelas itu sendiri, disamping itu mengubah kelas bisa melalui web atau langsung mengakses file kelas tersebut berbasis web. Disamping itu terdapat tombol operasi untuk

menjalankan `cbq.init` dan menghentikannya (*start, restart, stop*). Menu Support merupakan menu untuk melihat dan memantau aplikasi pendukung dari CBQ yakni `squid-proxy` dan `iptables-firewall`. Menu Diagnosa dan Logs untuk memantau kinerja dari interface dan juga melihat file history dari sistem. Sedangkan Menu Monitoring adalah untuk memonitoring sistem yang menjalankan aplikasi dan juga informasi lain yang direpresentasikan dengan grafik.

3.5.2 Desain Input

Pada tiap-tiap program menggunakan desain input sistem berbentuk form yang setiap proses didalam sistem menggunakan form yang berbeda. Desain input pada aplikasi dibagi hanya terdapat pada Menu CBQ sedangkan menu-menu yang lain berupa output. Berikut desain input dari aplikasi pembagian dan pembatasan bandwidth menggunakan CBQ pada sistem operasi linux;

a. Desain Form Classes

Desain form untuk menampilkan semua daftar kelas `cbq` yang dibuat beserta parameternya seperti digambarkan pada gambar 3.11 berikut :

CLASSES										
ID	Name	Device	Parent	Leaf	Rate	Weight	Pric	Boundec	Isolatec	Delete
										x
										x
										x

Gambar 3.11 Desain Form Classes

Untuk penjelasan tiap-tiap item yang ada pada form adalah sebagai berikut:

- a. *Id*; merupakan indentifikasi unik dari sebuah kelas CBQ
- b. *Name*; merupakan nama dari sebuah kelas CBQ
- c. *Device*; merupakan nama dari perangkat / NIC (*Network Interface Card*) yang akan dilakukan pembagian dan pembatasan bandwidth
- d. *Parent*; merupakan parent dari kelas
- e. *Leaf*; merupakan leaf atau cabang dari kelas
- f. *Rate*; merupakan besarnya kapasitas ukuran bandwidth yang diberikan kepada sebuah kelas CBQ
- g. *Weight*; hasil dari pembagian rate dibagi 10 yang merupakan default
- h. *Prio*; prioritas yang rangenya antara 1 s/d 8
- i. *Bounded*; mengidentifikasikan sebuah kelas diperbolehkan untuk meminjamkan bandwidth-nya kepada kelas yang lain
- j. *Isolated*; mengidentifikasi apakah sebuah kelas diisolasi agar tidak meminjamkan bandwidth kepada kelas yang lain
- k. *Delete*; untuk menghapus sebuah kelas

b. Desain Form New Class

Desain form New Class sebagai mana digambarkan pada gambar 3.12 digunakan untuk membuat sebuah kelas CBQ

NEW CLASS	
Class ID	<input type="text"/>
Class Name	<input type="text"/>
<input type="button" value="OK"/>	<input type="button" value="CANCEL"/>

Gambar 3.12 Desain Form New Class

c. Desain Form Edit Class

Desain form Edit Class gambar 3.13 digunakan untuk memodifikasi parameter-parameter dari sebuah kelas CBQ, penjelasan parameter seperti terlihat pada penjelasan pada gambar 3.11 diatas.

Gambar 3.13 Desain Form Edit Class

d. Desain Form Edit File Class

Desain form Edit File Class sebagai mana digambarkan pada gambar 3.14 digunakan untuk memodifikasi langsung file dari sebuah kelas CBQ.

Gambar 3.14 Desain Form Edit Class

3.5.3 Desain Output

Desain output terdapat pada Menu Diagnosa-Logs dan Menu Monitoring, dapat diuraikan sebagai berikut:

a. Desain Form Traffic Table

Desain form traffic table sebagai mana digambarkan pada gambar 3.15 digunakan untuk menampilkan daftar traffic dari masing-masing kelas CBQ yang diambil dari SNMP (*Simple Network Management Protocol*) Linux+IOS.

Parent	Handle	Leaf	Bytes	Packets	Drop	Over	Init	BBS	PPS	QID	Backlog

Gambar 3.15 Desain Form Traffic Table

b. Desain Form CBQ Logs

Desain form CBQ Logs sebagai mana digambarkan pada gambar 3.16 digunakan untuk menampilkan history dari CBQ script (*cbq.init*).

Gambar 3.16 Desain Form CBQ Logs

c. Desain Form System Logs

Desain form System Logs sebagai mana digambarkan pada gambar 3.17 digunakan untuk menampilkan seluruh file log (*history*) dari sistem.

SYSTEM LOGS		
Index	Log File	Log Description
		View
		View
		View
		View
		View

Gambar 3.17 Desain Form System Logs

d. Desain Form Log Detail

Desain form Log Detail sebagai mana digambarkan pada gambar 3.18 digunakan untuk menampilkan detail dari masing-masing file log.

LOG DETAILS		
Line Number	<input type="text"/>	View Filter <input type="text"/>
<input type="button" value="REFRESH"/>		
<input type="button" value="BACK"/>		

Gambar 3.18 Desain Form Log Detail

e. Desain Form Interface Info

Desain form Interface Info sebagai mana digambarkan pada gambar 3.19 digunakan untuk menampilkan diagnosa dari interface / NIC.

Gambar 3.19 Desain Form Interface Info

f. Desain Form Interface Statistic

Desain form Interface Statistic sebagai mana digambarkan pada gambar 3.20 digunakan untuk melihat statistik dari masing-masing interface (NIC).

Interface	Receive	Transmit

Gambar 3.20 Desain Form Interface Statistic

g. Desain Form Squid Traffic Current Status

Desain form Squid Traffic Current Status sebagai mana digambarkan pada gambar 3.21 digunakan untuk melihat traffic status yang terjadi dari pendukung aplikasi squid proxy.

Gambar 3.21 Desain Form Squid Traffic Current Status

h. Desain Form Squid Logging

Desain form Squid Logging sebagai mana digambarkan pada gambar 3.22 digunakan untuk melihat file history dari aplikasi pendukung squid-proxy.

Gambar 3.22 Desain Form Squid Logging

i. Desain Form Iptables Traffic Current Status

Desain form Iptables Traffic Current Status sebagai mana digambarkan pada gambar 3.23 digunakan menampilkan traffic status yang terjadi dari aplikasi pendukung iptables-firewall.

Gambar 3.23 Desain Form Iptables Traffic Current Status

j. Desain Form Iptables Logging

Desain form Iptables Logging sebagai mana digambarkan pada gambar 3.24 digunakan untuk menampilkan file log history dari aplikasi pendukung iptables-firewall.

Gambar 3.24 Desain Form Iptables Logging