FIREWALL DENGAN WEB INTERFACE MENGGUNAKAN SHELL PROGRAMMING DAN CGI PADA LINUX BERBASIS SHOREWALL DAN IP-TABLES

Skripsi

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Jurusan Teknik Informatika

Disusun Oleh :

Nama : Stephanus Warih Wibowo
NIM : 005314004

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA

2005

FIREWALL WITH WEB INTERFACE USING SHELL PROGRAMMING AND CGI ON LINUX BASED ON SHOREWALL AND IP-TABLES

A Thesis

Presented as Partial Fulfillment of the Requirements

to Obtain the *Sarjana Teknik* Degree

in Departement of Informatics Technology

by:

Stephanus Warih Wibowo
005314004

DEPARTEMENT OF INFORMATICS TECHNOLOGY
FACULTY OF ENGINEERING
SANATA DHARMA UNIVERSITY
YOGYAKARTA
2005

Halaman Persetujuan

SKRIPSI

FIREWALL DENGAN WEB INTERFACE MENGGUNAKAN SHELL PROGRAMMING DAN CGI PADA LINUX BERBASIS SHOREWALL dan IP-TABLES

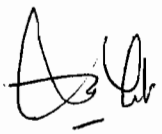
Oleh :

Nama : ST. Warih Wibowo

NIM : 005314004

Telah disetujui oleh :

Pembimbing I



(A. M. Polina, S.Kom, M.Sc.) tanggal : 7 juni 2005

Pembimbing II

(H. Agung Hernawan, S.T.) tanggal :.....

Halaman Pengesahan

SKRIPSI

FIREWALL DENGAN WEB INTERFACE MENGGUNAKAN SHELL PROGRAMMING DAN CGI PADA LINUX BERBASIS SHOREWALL dan IP-TABLES

Disusun oleh :

Nama : ST. Warih Wibowo

NIM : 005314004

Telah dipertahankan di depan Panitia Penguji
pada tanggal 18 April 2005
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

	Nama lengkap	Tanda tangan
Ketua	A.M. Polina, S.Kom., M.Sc.	.....
Sekretaris	H. Agung Hernawan, S.T.	.....
Anggota	Ds. Bambang Soelistijanto, S.T.	.....
Anggota	Albertus Agung Hadiatma, S.T.	.....

Yogyakarta,.....
Fakultas Teknik
Universitas Sanata Dharma
Dekan,

(Ir. Greg. Heliarko, S.J., S.S., B.S.T., M.A., M.Sc.)

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, 18 April 2005

Penulis

ST. Warih Wibowo

Halaman persembahan

Tugas akhir ini dipersembahkan untuk orang-orang yang banyak berperan dalam perjalanan mendapatkan gelar 'sarjana' di Universitas Sanata Dharma jogjakarta :

Bapak dan ibuku yang telah membesarkan dan mendidikku dengan penuh kasih sayang. semoga karya sederhana ini dapat menjadi salah satu bagian dari wujud tanggung jawab saya dalam menempuh pendidikan tinggi di Universitas Sanata Dharma Yogyakarta

Adikku yang selalu memberi dukungan semangat

Agustina Ariyani yang menemaniku melangkah

Teman- teman seperjuangan

Almamaterku

Halaman Motto

Tidak ada kata 'tidak mungkin' bila orang mau berusaha

Orang yang tahu "bagaimana" akan selalu mendapat pekerjaan.
Orang yang tahu "mengapa" akan selalu menjadi atasannya.

Hiduplah seperti kamu akan mati esok,
dan belajarkah seperti kamu akan hidup selamanya
(Gandhi)

ABSTRAKSI

Konfigurasi *firewalling* pada linux masih berbasis text *(text based)* dan seorang administrator harus melakukan konfigurasi pada komputer yang dijadikan sebagai komputer *firewall* tersebut secara langsung.

Pada penelitian ini dikembangkan sebuah aplikasi konfigurasi *firewall* berbasis web. Dengan menggunakan program aplikasi ini seorang administrator tidak perlu lagi mengetikkan serangkaian perintah text untuk melakukan konfigurasi *firewalling* karena program aplikasi sudah berbasis grafis, dan juga tidak perlu mendatangi komputer router karena program dapat diakses menggunakan *web browser*. Dalam pembuatan program aplikasi ini digunakan CGI *(Common Gateway Interface)* dan *Shell Programming* sebagai bahasa pemprogramannya.

Hasil ujicoba yang dilakukan menunjukkan program dapat berjalan dengan baik. Program aplikasi yang dibuat dapat diterapkan pada jaringan yang sesungguhnya. Program dapat membantu seorang administrator dalam melakukan konfigurasi *firewalling* dari jarak jauh dengan berbasis grafis menggunakan *web browser*.

ABSTRACT

Firewall configuration at Linux still base on the text (text based) and an administrator have to do the configuration at computer which taken as the computer firewall directly.

An apllication for firewall configuration using web interface is develop at this research. By using this application program, admistrator needn't type the text command for configure firewall because application program have based on graphical user interface, as well as needn't visit upon the computer router (computer firewall) because program can be accessed using web browser on other computer. CGI (Common Gateway Interface) and Shell Programming as programming language is used on this research.

The test result show aplication program can work. Aplication program made aplictable at computer network thruthfully. Program can assist an administrator in conducting firewall configuration from long distance by being on graphical user interface using the web browser.

KATA PENGANTAR

Puji syukur penulis haturkan kepada Tuhan Yang Maha Esa atas segala karunia yang diberikan, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **"FIREWALL DENGAN WEB INTERFACE MENGGUNAKAN SHELL PROGRAMMING DAN CGI PADA LINUX BERBASIS SHOREWALL dan IP-TABLES"** ini dengan baik. Penulisan ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik di Universitas Sanata Dharma pada program studi Teknik Informatika.

Selama penulisan skripsi ini penulis telah memperoleh bantuan dan bimbingan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Ibu A.M. Polina, selaku Ketua Jurusan Tenik Informatika Universitas Sanata Dharma dan sekaligus segabai dosen pembimbing I yang telah banyak membantu dan membimbing selama mengerjakan tugas akhir ini.
2. Bapak H. Agung Hernawan selaku dosen pembimbing II yang telah memberi banyak masukan dan bimbingannya dari awal pembuatan tughas akhir ini.
3. Bapak, Ibu, dan Adikku yang telah memberi dorongan baik moril maupun materi.
4. Bapak Donny yang memberikan inspirasi pada dunia Linux.
5. Bapak Belle dan Pak Dar yang mau menemani dan memberi semangat.
6. Teman-teman seperjuangan : Fito "Acong" Tatontos untuk pinjaman compie routernya, Antonius Arief atas banyak bantuannya, wisnu widodo, toro,

andhika 'pay', lukas, joe, pieter, bobo, oni 'gombloh', wawan 'suna', untuk semangatnya *'keep on fight guys'*.

7. My 97' RX. King yang menemani kemanapun aku pergi...
8. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan serta bantuannya guna penyusunan karya tulis ini

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih jauh dari kesempurnaan dan masih banyak kekurangan. Oleh karena itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi perbaikan lebih lanjut. Penulis berharap semoga Tugas akhir ini dapat bermanfaat dan berguna bagi pembaca.

Yogyakarta, Juni 2005

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Jaringan komputer atau yang dikenal dengan LAN *(Local Area Network)* merupakan sistem yang terbuka dimana setiap orang bisa masuk ke komputer milik orang lain yang terhubung di dalam jaringan atau internet. Sistem ini berarti juga bahwa tidak ada batasan bagi orang lain diluar jaringan untuk masuk ke dalam sistem. Misalnya dengan menggunakan *web browsing*, akses FTP *(File Transfer Protocol)*, dan lain sebagainya.

Pengaksesan data yang dilakukan oleh pihak yang tidak berkepentingan akan sangat merugikan. Apalagi jika data yang diakses secara ilegal tersebut disalah gunakan.

Suatu sistem jaringan perlu disertai dengan suatu utilitas pengendalian keamanan antar jaringan untuk mengatasi kejadian seperti yang digambarkan diatas. Pembatasan dapat dilakukan dengan pemblokiran terhadap akses-akses atau port tertentu *(Firewalling)*. Misalnya dengan memblokir layanan internet pada perangkat komputer yang digunakan oleh seorang operator yang tidak berkepentingan dalam hal pemakaian *internet*. Atau dapat juga dengan menutup fungsi file transfer protokol *(ftp)* untuk semua pengguna yang berada diluar jaringan lokal.

Pada sistem operasi Linux sudah terdapat fasilitas perangkat lunak yang berfungsi sebagai program pengendali keamanan antar jaringan *(firewall)*. Fasilitas tersebut berupa perangkat lunak yang sudah disertakan pada distribusi Linux

(contoh: IP-Tables, IP-Chains) dan perangkat lunak lain yang tidak disertakan pada distribusinya namun dibuat dengan kompatibilitas pada sistem operasi Linux. Contoh perangkat lunak yang belum disertakan pada distribusi Linux adalah *Shorewall*. *Shorewall* adalah sebuah program berupa file konfigurasi yang digunakan untuk mempermudah penyetingan *firewalling* berbasis fungsi IP-Tables pada Linux.

Setting yang dapat dilakukan pada perangkat lunak yang telah disebutkan diatas masih dilakukan dengan berbasis text *(text based)*. Pengesetan *firewall* secara *text based* akan sangat merepotkan karena seorang administrator harus mengetik serangkaian perintah program *(script)* secara manual dan dengan aturan penulisan tertentu.

Di lain pihak perkembangan teknologi berbasis web berkembang pesat. Banyak aplikasi-aplikasi yang memanfaatkan teknologi tersebut. Berdasarkan latar belakang tersebut diatas, maka dalam penelitian ini, diteliti penggunaan teknologi berbasis web dengan menggunakan CGI *(Common Gateway Interfaces)* yang diimplementasikan dengan *Shell Script programming* untuk menciptakan suatu utilitas sistem keamanan jaringan dibawah sistem operasi Linux. Fungsi IP-Tables yang telah disediakan digunakan sebagai fungsi dasar firewalling dengan bantuan program file konfigurasi *Shorewall*. Utilitas sistem tersebut diharapkan dapat digunakan untuk membatasi akses pada jaringan lokal dan juga antar dua buah jaringan (jaringan lokal dan jaringan global) sesuai dengan fungsi *firewall* secara antarmuka berbasis grafis *(Graphical User Interfaces)* dan dapat diakses menggunakan teknologi web.

1.2. Batasan Masalah

Karena keterbatasan waktu, tenaga dan kemampuan, penelitian ini dibatasi sebagai berikut :

1. Tidak semua tipe dan teknik *firewalling* diterapkan pada penelitian ini. Metode yang digunakan adalah tipe *packet filtering* dengan teknik *direction control.*
2. Sistem yang dibuat dibatasi untuk diimplementasikan pada jaringan lokal saja (tidak digunakan melalui jaringan global).
3. Tidak semua layanan / servis yang diberikan oleh protokol jaringan digunakan dan dibahas pada penelitian ini, seperti Ftp, Telnet, SSH, dan lainnya.
4. Penelitian ini tidak menggunakan teknologi *encryption* (enkripsi data). Tekonologi ini tidak dibahas pada penelitian ini.
5. Penelitian ini tidak membahas mengenai *routing* pada jaringan.
6. Instalasi dan cara kerja Protokol jaringan, file konfigurasi dan program bantu yang digunakan tidak dibahas dalam penelitian ini.
7. Instalasi serta *troubleshooting* pada *hardware* dan *software* yang digunakan tidak dibahas pada penelitian ini.

1.3. Rumusan Masalah

Permasalahan pada penelitian ini adalah :

a. Bagaimana mengimplementasikan teknologi berbasis web untuk membuat sebuah program konfigurasi *firewall* pada sistem operasi

berbasis Linux sehingga memberikan kemudahan dengan berbasis GUI (*Graphical User Interface*).

b. Apakah metode berbasis web dengan CGI *(Common Gateway Interface)* dan *Shell Programming* cukup mendukung untuk membuat sebuah aplikasi firewalling berbasis web *(web base)*.

1.4. Tujuan dan Manfaat Penulisan

Penelitian ini bertujuan membuat sebuah utilitas sistem keamanan jaringan berdasarkan prinsip *Firewalling*. *Firewalling* yang dimaksud adalah proses penyaringan paket data (*packet filtering*) dengan melakukan setting pada fungsi *IP-Tables* pada Linux dengan bantuan program file konfigurasi *Shorewall*.

Pembuatan utilitas berupa perangkat lunak tersebut dibuat dengan cara mengimplementasikan teknologi berbasis web pada penelitian ini. Dengan diimpementasikannya teknologi berbasis web, diharapkan penelitian ini dapat menghasilkan sebuah program keamanan jaringan yang berbasis GUI *(Graphical User Interfaces)* dan dapat diakses dengan bantuan perangkat teknologi web *(diantaranya webserver dan browser)*.

Dengan adanya perangkat lunak (program) yang akan dibuat pada penelitian ini, ada beberapa manfaat yang diperoleh. Yaitu :

1. Program dapat ditanamkan pada sebuah perangkat komputer yang selanjutnya dijadikan sebagai router sekaligus sebagai perangkat *firewall*.

2. Seorang administrator tidak perlu lagi mengetikkan serangkaian *script* program untuk melakukan setting *firewalling* pada sistem.
3. Seorang administrator dapat melakukan setting *firewalling* dari computer mana saja yang terhubung dengan router sehingga tidak perlu mendatangi PC *(Personal Computer)* yang dijadikan router.

1.5. Metodologi Penelitian

Metode yang dipergunakan untuk perancangan sistem dan pembuatan perangkat lunak, serta menyelesaikan permasalahan adalah dengan cara :

1.5.1. Studi Literatur

Kegiatan ini merupakan pendalaman materi yang berkaitan dengan pembuatan program dan sistem yang akan dibuat meliputi :

a. Mengenal prinsip *firewalling* sebagai piranti keamanan jaringan.
b. Pendalaman Linux sebagai suatu sistem operasi.
c. Pengembangan jaringan dengan menggabungkan Microsoft Windows dan Linux.
d. Mengenal kemampuan Linux dalam mendukung pemecahan permasalahan.

1.5.2. Perancangan Sistem

Setelah melalui tahap studi pustaka selanjutnya dilakukan perancangan sistem yang akan dibuat. Perancangan meliputi topologi sistem

beserta piranti-piranti yang akan digunakan, penempatan *firewall*, dan desain user interface.

1.5.3. Implementasi

Tahap ini adalah penerapan desain kedalam bentuk program dengan memanfaatkan bahasa pemprograman yang ada. Pada tahap ini digunakan *Shell script programming* dan CGI(*Common Gateway Interface*) yang ditulis dalam bahasa Perl.

Pada tahap ini juga akan dilakukan penerapan program kedalam suatu konfigurasi jaringan. Konfigurasi jaringan dibuat sedemikian rupa sehingga dapat mewakili dan mencerminkan konfigurasi jaringan yang sebenarnya. Pada tahap ini juga menjadi tahap ujicoba untuk utilitas yang telah dibuat.

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Berisi latar belakang masalah, batasan masalah, tujuan tugas akhir, rumusan masalah, metodologi penelitian, dan sistematika penulisan..

BAB II LANDASAN TEORI

Berisi mengenai penjelasan dan uraian singkat mengenai teori-teori yang berkaitan dengan topik dari tugas akhir ini. Tulisan ini menghendaki tingkat pemahaman tertentu sehingga dasar-dasar teori tidak diberikan dari dasar dan secara mendetail.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Berisi tentang langkah-langkah pembuatan perangkat lunak serta piranti yang digunakan untuk menyelesaikan masalah. Analisis berisi pendapat penulis dengan melihat dan mempertimbangkan dasar teori atas permasalahan yang ada.

Analisis yang menghasilkan model yang kemudian diwujudkan dalam bentuk desain yang selanjutnya digunakan sebagai dasar perwujudan utilitas perangkat lunak yang dikehendaki.

BAB IV IMPLEMENTASI

Menjelaskan mengenai implementasi rancangan (desain) kedalam bentuk utilitas yang nyata yang dapat dipergunakan secara langsung.

BAB V PEMBAHASAN

Berisi mengenai hasil uji coba program yang telah dibuat, kelebihan serta kekurangan program. Serta menentukan dukungan *hardware* dan *software* terhadap program yang telah dibuat.

BAB VI PENUTUP

Berisi rangkuman pembahasan dari sistem yang telah dibuat dan proses pemecahan permasalahan yang timbul pada penyelesaian tugas akhir ini. Kesimpulan dan saran penulis untuk pengembangan selanjutnya disertakan pada bagian ini.

BAB II

LANDASAN TEORI

2.1. LINUX

Linux merupakan sebuah sistem Operasi pengembangan dari UNIX. UNIX Adalah sebuah sistem operasi yang asalmulanya dikembangkan pada laboratorium Bell, AT & T. Linux padamulanya adalah proyek hobi yang dikerjakan oleh Linus Trovalds. Dalam pengerjaannya Linus Trovald memperoleh inspirasi dari minix, suatu sistem UNIX yang dikembangkan oleh Andy Tanenbaum. Versi pertama Linux Diumumkan pertama kali pada 5 oktober 1991. versi ini hanya dapat menjalankan BASH *(Bourne Again Shell)* dan GCC *(GNU C Compiler)* (Wahana, 20004).

Linux disebarluaskan secara gratis dibawah lisensi GNU *General Public LicenceII (GPL).* Hal ini berarti bahwa *source code* program juga diikutsertakan pada distribusinya. Berdasarkan lisensi GNU siapapun dapat mendapatkan program ini baik dalam bentuk *source code* nya. Hal ini menyebabkan program dapat diubah, diadaptasi, dan dikembangkan oleh siapa saja

Selain merupakan sistem operasi yang bisa didapatkan setiap orang secara gratis linux juga mempunyai beberapa keunggulan antara lain (Wahana, 2004) :

a. *Open Source*

   Linux merupakan salah satu sistem *open souce* yang berarti memberi kesempatan kepada penggunanya untuk melihat program asal, dan atau mengubahnya sesuai keperluan tanpa terkena sangsi.

b. *Freeware*

Linux merupakan program *freeware* dibawah lisensi GNU, yang memungkinkan seseorang, beberapa orang maupun sebuah instansi untuk memakai dan menyebarkan tanpa dituntut royalty oleh penciptanya.

c. Minimal Hardware

Linux dapat berjalan pada perangkat keras *(hardware)*yang sangat minimal. Minimal hardware yang dibutuhkan Linux adalah prosesor Intel 386 DX, RAM 8MB, dan kapasitas hardisk minimal 85MB.

d. *Shared Libraries*

Linux menggunakan penomoran versi *shared libraries.* Shared libraries di Linux mencantumkan versi pada file-nya, sehingga dimungkinkan untk menginstal versi terbarunya tanpa merusak keterkaitan dengan program lain.

e. Non-Fragmentasi

Non-fragmentasi membuat penggunanya mudah untuk melakukan edit dan menghapus file tanpa kuatir terjadi fragmentasi pada data atau program yang ada. Karena linux memakai sistem file ext2fs *(Second Extended File System)* yang mempunyai keunggulan reduksi fragmentasi.

f. *Bugfix*

Masalah keamanan yang menyangkut sistem operasi itu sendiri biasannya diumumkan beberapa jam saja setelah ditemukan, diikuti dengan *bugfix* (solusi masalah) nya.

g. File system 32/64bit

Linux mendukung secara penuh file system 32bit, bahkan 64 bit yang memungkinkan untuk dijadikan server baik secara terpisah maupun bersama-sama.

h. *Multi User*

Linux mendukung fungsi multi user dimana lebih dari satu orang dapat menggunakan program yang sama atau berbeda, dari satu mesin yang sama atau berbeda pada saat bersamaan, di terminal yang sama atau berbeda.

i. *Multiconsole*

Dalam satu komputer pengguna dapat melakukan login dengan nama user yang sama

j. *Firewall*

Linux dilengkapi dengan *firewall* yang digunakan untuk mengatur jalannya lalu lintas data agar lebih tahan terhadap ancaman dari luar dan dalam jaringan.

k. *Shell programable*

Shell programable memungkinkan sistem untuk menerima perintah dari pemakai *(user)* dan menjalankannya. Shell adalah salah satu antarmuka dalam linux yang bersifat CLI *(Command Line Interpreter)*.

2.2. Apache

Apache adalah sebuah *Hypertext Transfer Protocol Daemon (HTTPD) server* atau aplikasi *web server* yang digunakan untuk *connection client* dengan

menggunakan port tertentu. *(http//:old-www.aclug.org/ACLUG/members/carubak /apache1.html).*

Ketika sebuah koneksi dibangun, *web server* akan menunggu *request* dari program aplikasi *client*. Apache akan memberikan respon dengan memberikan dokumen yang tepat ketika *browser* mengirimkan perintah dengan metode GET atau POST dan mengembalikannya berupa URL dan : html text, Plain text,Gif image. Audio.

2.2.1. Direktori pada Apache

Apache merupakan aplikasi yang didistribusikan secara gratis, dan memiliki kompatibilitas terhadap terhadap berbagai macam sistem operasi. Apache mempunyai direktori-direktori yang sudah terbentuk jika instalasi selesai dilakukan pada suatu sistem operasi. Direktori-direktori tersebut adalah (pada LINUX) :

- Direktori `/etc/httpd/conf;` direktori ini berisi file-file untuk konfigurasi server seperti httpd.conf, srm.conf, dan access.conf.
- Direktori `/etc/httpd/conf/vhost;` berisi file-file untuk konfigurasi virtual host Apache.
- Direktori `/etc/httpd/common-httpd;` berisi file konfigurasi Apache advance.
- Direktori `/var/www/html;` adalah direktori default dimana file-file dokumen html akan diletakkan.

- Direktori `/var/www/cgi-bin;` pada file ini diletakkan *script* CGI yang dibuat.
- Direktori `/var/www/perl;` pada direktori ini disimpan *script* program perl yang dibuat.

2.2.2. Konfigurasi Apache

File konfigurasi yang digunakan untuk melakukan konfugurasi sesudah instalasi *(post instalation)* adalah file httpd.conf yang terdapat pada direktori `/usr/local/apache/conf.` Pada file konfigurasi tersebut terdapat beberapa baris yang merupakan setting dari Apache. Baris-baris tersebut diantaranya adalah :

- Baris yang menyatakan nama komputer yang dijadikan web server

  ```
  Server name www.skylab.com
  ```

- Baris yang menyatakan *port* (pintu gerbang) untuk komunikasi antara *client* dengan *server*.

  ```
  Port 80
  ```

- Baris yang menyatakan letak dokument HTML root. Contoh :

  ```
  Document Root "/usr/local/apache/cgi-bin".
  ```

- Baris yang menyatakan lokasi Cgi-bin (direktori file-file CGI disimpan). Contoh:

  ```
  /usr/local/apache/cgi-bin
  ```

- Baris yang menyatakan penanganan SSI (Server Side Include) oleh web server.

  ```
  AddType text/x-server-parsed-html .shtml
  ```

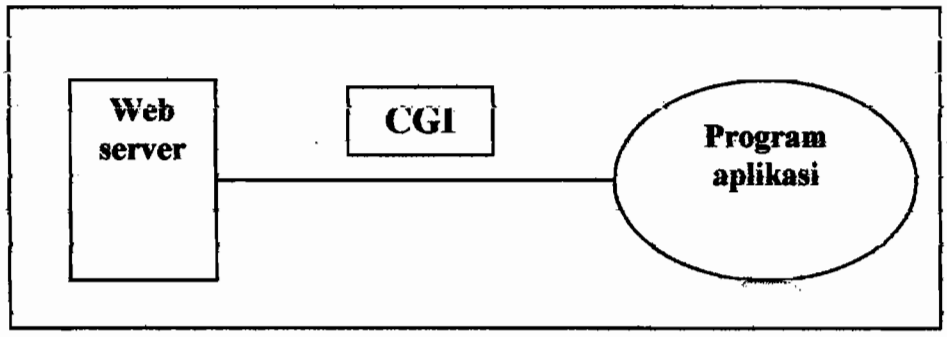
- Baris yang menyatakan mengijinkan segala fasilitas SSI pada direktori root HTML.

```
<Directory "/usr/local/apache/htdocs">
      Options All
      AllowOverride None
      Order allow, deny
      Allow from all
</Directory>
```

Setelah konfigurasi *post instalation* dilakukan maka Apache dapat dijalankan dengan mengetikkan `/usr/local/apache/bin/apachectl start`.

2.3. CGI (Common Gateway Interface)

CGI merupakan sebuah mekanisme dimana sebuah web client dapat mengeksekusi program pada web server dan menerima hasil outputnya. Mekanisme ini berupa sebuah *script. Script* yang dapat didefinisikan sebagai rangkaian dari berbagai instruksi program (*Frans,2002*). CGI menjadi perantara antara web server dengan program aplikasi.



Gambar 2.1 CGI sebagai perantara

CGI juga merupakan suatu bentuk hubungan interaktif di mana client (browser) dapat mengirimkan suatu masukan kepada server, dan server mengolah masukan tersebut serta mengembalikannya kepada client (browser) (*tutorial perl,*

www. Klik_kanan.com). Hal ini dapat dilihat pada saat digunakannya suatu mesin pencari *(search engine)*. Saat dituliskannya kata kunci dan menekan tombol maka browser akan mengirimkan kata kunci tersebut ke server. Kata kunci tersebut lalu diolah oleh server dan server mengirimkan data hasil pengolahan (yang sesuai dengan kata kunci yang dimasukkan) ke browser client. Jadi yang tampak pada browser hanya data yang sesuai dengan kata kunci yang telah dimasukkan.

CGI digunakan pada server dengan sistem operasi UNIX (beserta variantnya). Namun tidak semua server UNIX (yang merupakan versi *freeware*) mampu menangani atau meng-*handle* CGI.

Program CGI ditulis dalam bahasa yang dapat dimengerti oleh sistem misalnya C/C++, Fortran, Perl, Tcl, Visual Basic, dan lain-lain. Pemilihan bahasa yang digunakan tergantung dari sistem yang digunakan. Jika penulisan CGI menggunakan bahasa pemrograman seperti C atau Fortran maka program-program yang dibuat harus dikompile terlebih dahulu sebelum dijalankan sehingga pada server akan terdapat (kode program) *source code* dan program hasil kompilasi. Berbeda dengan penulisan CGI yang menggunakan bahasa script seperti Perl, TCL, atau Unix Shell maka hanya akan terdapat script itu sendiri (tanpa ada *source code*).

Berikut ini contoh program CGI yang akan menampilkan kata Hallo pada browser.

```
#/usr/bin/perl
print "content-type : text/html\r\n\r\n;"
print "Hallo";
```

program CGI diatas dituliskan dengan bahasa pemprograman Perl. Baris pertama berfungsi sebagai informasi bagi *web server* bahwa *script* diatas akan mengeksekusi file interpreter pada direktori tersebut. Baris yang kedua berfungsi memberikan informasi kepada web server bahwa *script* yang dirulis merupakan script dengan format HTML. Baris yang ketiga berfungsi untuk mencetak kata "Hallo".

2.3.1. Tahap- tahap proses kerja CGI

Menurut (Haryanto, 2004) tahap-tahap kerja CGI adalah sebagai berikut :

1. Aplikasi CGI, yang berupa file skrip, diletakkan pada lokasi direktori yang dapat diakses oleh web server.
2. Jika ada permintaan terhadap aplikasi dari klien, web server akan membaca skrip dan mengeksekusi program.
3. Web server memberi program ini masukan (dapat berupa masukan yang diberikan klien) dari masukan standar (*stdin*) serta beberapa *environment variable.*
4. Web server menunggu program berjalan hingga selesai lalu menangkap keluaran standar program (*stdout*). Keluaran diberikan kepada klien.

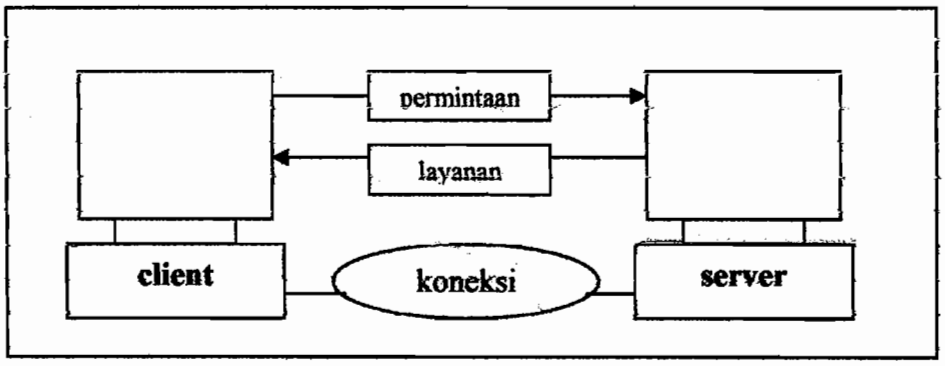
2.3.2. Konsep Client Server

CGI bekerja berdasarkan konsep *client-server*. Konsep ini adalah konsep yang terjadi pada perpindahan data antar komputer yang diterapkan pada berbagai utilitas internet.

Server adalah pemilik informasi yang menyediakan dirinya untuk memberikan *service* atau layanan (susanto,2001). Jika konsep ini terjadi pada layanan web, maka web server merupakan penyimpan data web. Jika ada komputer lain yang ingin mengakses data tersebut maka web server memberikan *service* kepada pihak yang meminta data tersebut. Hal ini terjadi tentunya jika sudah ada koneksi yang terbangun antara web server dengan komputer lain tersebut.

Sedangkan *client* adalah pihak yang meminta *service* kepada server. Komputer client dan server tidak harus berbeda. Proses client server bisa terjadi pada satu buah komputer yang sama. Hal ini terjadi ketika sebuah

komputer menyimpan data web dan data tersebut diakses oleh komputer itu sendiri.



Gambar 2.2 CGI Koneksi Client Server

2.3.3. Server side programming

Server side programming atau *client side programming* adalah metode yang digunakan ektika terjadi pengaksesan sebuah situs yang dilakukan oleh *client* ke *web-servernya.*

Server side programming terjadi ketika ada request dari *web-browser client* ke *web-server* yang mengirimkan pesan agar *web-server* mengirimkan data situs yang diminta. Yang terjadi adalah *web-server* melakukan pengecekan apakah data yang diminta *client* tersebut mengandung *server side script*, jka ada maka terjadilah *Server side programming* yaitu ketika *web-server* menerjemahkan script tersebut dengan cara memanggil program aplikasi yang dipergunakan untuk menerjemahkannya (contoh: memanggil perl interpreter, jika menggunakan bahasa perl). Script yang sudah diterjemahkan oleh *web-server* kemudian dikirim ke web-browser client .

Jika web-server tidak menemukan *server side script* maka yang terjadi adalah *client side programming* yaitu web-server langsung mengirimkan data yang diminta dan client yang melakukan proses penterjemahan.

2.4. Shell Programming

Shell adalah nama yang diberikan untuk suatu program yang berfungsi untuk menjembatani seorang pemakai *(user)* dengan sistem operasi UNIX (susanto,2001). Shell menerima setiap perintah yang diberikan oleh *user* kemudian menjlankan perintah tersebut sesuai dengan fungsinya. Dapat dikatakan shell bertindak sebagai penterjemah perintah *(command interpreter).*

Selain itu shell juga mempunyai kemampuan untuk menjalankan sekumpulan perintah UNIX dan juga shell dalam suatu file yang disebut dengan *shell script.* Dengan kata lain shell script adalah program yang ditulis dengan bahasa pemprograman shell. Kemampuan pemprograman inilah yang merupakan kemampuan UNIX membedakan dengan sistem operasi lain.

2.4.1. Jenis-jenis shell

Ada beberapa jenis *shell* yang tersedia pada UNIX diantaranya adalah (Kadir, 2001) :

1. Bourne Shell

   adalah jenis Shell yang tertua. Nama ini didasarkan atas nama penciptanya yaitu Stephen R. Bourne dari laboratorium Bell AT & T. Shell ini diperkenalkan pada akhir tahun 1970 dan dipakai sebagai shell utama pada UNIX.

2. C Shell

   diciptakan oleh Billy Joy. Shell ini dijadikan standar pada sistem unix versi Barkeley. Format perintahnya menyerupai bahasa C.

3. Korn Shell

   Korn Shell adalah shell yang diciptakan oleh David Korn di laboratorium Bell, AT & T pada tahun 1983. Shell ini kompatibel dengan Bourne shell, artinya perintah- perintah yang didukung Bourne Shell juga dapat dijalanka pada Korn shell.

2.4.2. Bash Shell

Bash Shell adalah singkatan dari *Bourne Again shell.* Bash shell merupakan *programming language,* yang memungkinkan seorang pengguna untuk membuat program secara terstruktur. Bash shell juga merupakan *command language,* hai ini memungkinkan pengontrolan eksekusi program (Susanto, 2001).

Menurut (Susanto, 2001) *s*ebagai *command language*, Bash shell memiliki karakteristik sebagai berikut:

1. Eksekusi program. Hal ini memungkinkan untuk menjalankan program secara berurutan dan paralel.
2. Pembuatan file, dapat membuat file baru atau menambah kedalam file yang sudah ada.
3. Eksekusi Program berkondisi, memungkinkan pengguna untuk menjalankan suatu program berdasarkan keberhasilan atau kegagalan program lainnya.
4. *Predefined procedures,* pengguna dapat menyusun suatu script yang mendefinisikan urutan program yang akan dieksekusi.
5. *Parameterization,* memungkinkan pengguna untuk menulis *script* menggunakan shell variabel sehingga *script* bersifat lebih umum.

Sebagai *programming language*, Bash shell memiliki karakteristik sebagai berikut:

1. *Variables,* dapat mendefinisikan variabel dan melakukan operasi terhadapnya.
2. *Structured Language Construct,* menyediakan struktur sequence, seleksi dan iterasi.
3. *Scope,* kemampuan membatasi ruang lingkup dimana variabel dan program dapat saling mengenal.
4. *Macro Substitution,* menyediakan macro dan header file.
5. *Subroutines,* kemampuan memanggil subroutin dan memungkinkan untuk membuat suatu rekursi didalamnya.

2.5. Perl

Perl adalah suatu bahasa pemprograman yang dioptimalkan untuk melakukan *parsing* (pengambilan), pencarian suatu pola karakter tertentu, pengambilan informasi pada sebuah string ataupun file text (susanto,2001).

2.5.1. Sejarah Perl

Perl dikembangkan oleh Larry wall pertama kali pada tahun 1986, ketika diminta untuk menyediakan suatu konfigurasi sistem manajemen dan kontrol untuk sebuah WAN (Wide Area Network).

Perl adalah singkatan dari *Pactical Extraction and Report language.* Perl secara bebas tersedia untuk sistem UNIX, MVS, VMS, MS/DOS/Windows, Macintosh, OS/2, Amiga dal sistem operasi lainnya. Perl tidak membatasi ukuran data yang akan diolah, karena perl dapat mengambil semua data pada suatu file menjadi sebuah *string* (kumpulan karakter) tunggal.

2.5.2. Ekspresi Reguler

Dalam Perl dikenal istilah ekspresi reguler *(regular expression)*. Ekspresi reguler digunakan terutama untuk melakukan pencarian terhadap pola tertentu untuk selanjutnya dibandingkan. Berikut ini adalah contoh penggunaan *regular expressions*.

```
#!/usr/bin/perl
$tes="Hallo";
if ($tes=~/"Hallo warih"/);{
     print "data cocok secara sensitif"; }
if ($tes=~/"Hallo warih"/i);{
     print "data cocok secara insensitif"; }
```

yang dilakukan program diatas adalah melakukan pencocokan terhadap string "hallo warih" namun perbedaan pencocokan diatur pada *expressions modifier* "/i". *Expression modifier* mengatur pencocokan dilakukan secara *case insensitive* atau *case sensitif* (pembedaan huruf kecil atau tidak).

Pada Perl hal ini dapat dilakukan, sekaligus menjadi kelebihan bahasa pemprograman ini. *Regular expression* digunakan dengan penulisan sintaks sebagai berikut :

```
$string=~/regular expression/expression modifier
```

berikut ini adalah tabel *regular expresion* yang dimiliki Perl beserta fungsinya.

Modifier	Kegunaan
I	Mencocokkan secara insensitif
M	Mencocokkan multiline yaitu tanda baris baru yang disisipkan dalam suatu kumpulan karakter

S	Mencocokkan setiap karakter kecuali baris baru
G	Mencocokkan pada seluruh data
E	Mencocokkan dan menterjemahkan
O	Mencocokkan hanya sekali
X	Mengenali spasi

Tabel II.1 regular expression Perl

2.6. Firewall

Firewall merupakan suatu cara atau mekanisme yang diterapkan baik terhadap hardware,software ataupun sistem sendiri dengan tujuan untuk melindungi,baik dengan menyaring,membatasi atau bahkan menolak suatu atau semua hubungan/kegiatan suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkupnya *(Ahmad Muammar, 2004)*. Segmen tersebut dapat merupakan sebuah workstation,server, router,atau local area network (LAN). Konfigurasi sederhananya adalah :

pc (jaringan local).═. ***firewall*** .═. **internet** (jaringan lain)

Tugas dari firewall yang utama adalah mengimplementasikan kebijakan akses keamanan jaringan atau disebut dengan *security policy*. Jika operasi tertentu tidak diinginkan oleh sebuah sistem maka firewall harus meyakinkan bahwa tindakan yang mewakili dari operasi tersebut harus gagal atau digagalkan. Dengan demikian semua akses ilegal dalam arti akses yang tidak diotorisasi oleh sistem akan ditolak.

Melakukan firewalling berarti mewajibkan semua trafik yang ada pada sistem agar dilewatkan pada firewall untuk pemberian ijin atau pemanfaatan layanan informasi. Dalam konteks ini, aliran paket data dari atau menuju firewall diseleksi

berdasarkan IP-address, nomor port, atau arahnya, dan selanjutnya disesuaikan dengan kebijakan security.

2.6.1. Karakteristik Firewall

Menurut (Muammar, 2004) Firewall Memiliki karakteristik sebagai berikut :

1. Seluruh hubungan/kegiatan dari dalam ke luar ,harus melewati firewall.Hal ini dapat dilakukan dengan cara memblok/membatasi baik secara fisik semua akses terhadap jaringan lokal, kecuali melewati firewall. Banyak sekali bentuk jaringan yang memungkinkan agar konfigurasi ini terwujud.
2. Hanya kegiatan yang terdaftar/dikenal yang dapat melewati/melakukan hubungan,hal ini dapat dilakukan dengan mengatur kebijakan pada konfigurasi keamanan lokal.
3. Firewall itu sendiri haruslah kebal atau relatif kuat terhadap serangan/kelemahan. Hal ini berarti penggunaan sistem yang dapat dipercaya dan dengan sistem yang relatif aman.

2.6.2. Teknik yang digunakan sebuah Firewall

1. Service control (kendali terhadap layanan)

   Kontrol ini bekerja berdasarkan tipe-tipe layanan yang digunakan atau diakses baik untuk kedalam ataupun keluar firewall, firewall akan melakukan cek terhadap IP Address dan juga nomor port yang digunakan baik pada protokol TCP dan UDP. Selanjutnya firewall

akan menerima dan menterjemahkan setiap permintaan akan suatu layanan sebelum mengijinkannya.

2. Direction Control (kendali terhadap arah)

   Teknik ini melakukan kontrol berdasarkan arah dari permintaan *(request)* terhadap layanan tertentu.

3. User Control

   Teknik ini melakukan kontrol berdasarkan *user* untuk dapat menggunakan suatu layanan. Dengan kata lain ada *user* yang dapat menjalankan suatu servis, sedangkan *user* yang lain tidak dapat menjalankan servis yang sama.

4. Behaviour Control

   Kontrol ini berdasarkan atas seberapa sering suatu layanan atau servis digunakan.

2.6.3. Tipe-tipe Firewall

Menurut (Muammar, 2004) firewall mempunyai tipe-tipe yang dapat digolongkan sebagai berikut :

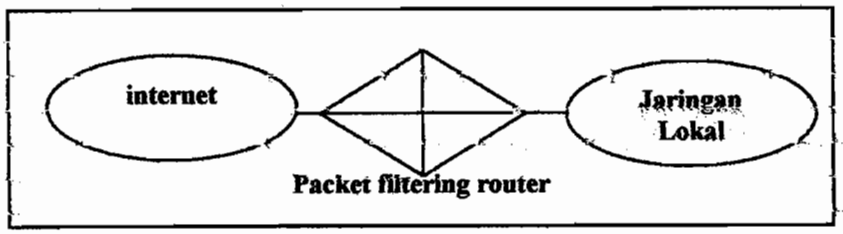
1. Packet filtering router.

   Packet Filtering Router diaplikasikan dengan cara mengatur semua paket IP baik yang menuju atau melewati router. Pada tipe ini packet tersebut akan diatur apakah akan diterima dan diteruskan atau ditolak. Penyaringan paket ini di konfigurasi sesuai dengan kebijakan yang telah ditetapkan. Penyaringan paket yang akan di transfer dilakukan secara dua arah (baik dari dan ke jaringan lokal). Aturan penyaringan

(penerapan kebijakan) didasarkan pada header IP dan transport header, termasuk juga alamat awal(IP) dan alamat tujuan (IP), protokol transport yang di gunakan(UDP,TCP), serta nomor port yang digunakan. Kelebihan dari tipe ini adalah mudah untuk diimplementasikan transparan untuk pemakai dan relatif lebih cepat. Sedangkan kelemahan yang dimiliki tipe ini adalah cukup rumitnya menyeting paket yang akan di filter secara tepat. Serangan yang dapat terjadi pada tipe ini adalah :

a. IP address spoofing : yaitu adanya *intruder* (penyusup) dari jaringan luar. Hal ini dapat dilakukan dengan pemakaian ip address jaringan lokal yang telah diijinkan untuk melalui firewall.
b. Source routing attack : tipe ini melakukan penyerangan terhadap jaringan lokan firewall dengan cara tidak dianalisanya informasi routing sumber IP, sehingga memungkinkan untuk mem-*bypass* firewall.
c. Tiny fragment attack : hal ini dilakukan penyusup d engan cara membagi-bagi paket kedalam fragment-fragment kecil dan memaksa membagi informasi mengenai TCP headernya. Penyerang berharap adar paket fragment pertama saja yang diteliti oleh firewall dan yang selanjutnya akan dapat lewat dengan bebas. Hal ini dapat diatasi dengan menolak semua

paket yang berasal dari protokol TCP yang memiliki offset=1 pada IP fragment.



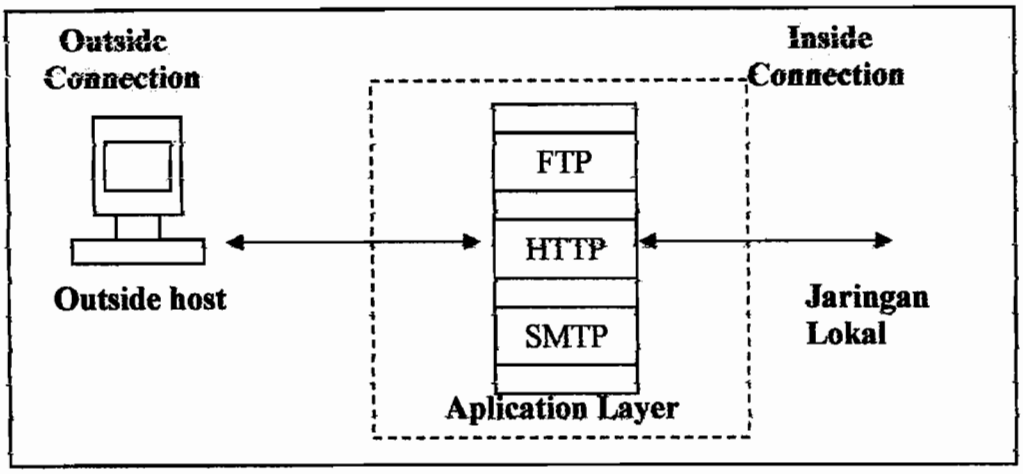
Gambar 2.3 Packet Filtering Router.

2. Application Level Gateway

Tipe ini biasa dikenal sebagai proxy server yang berfungsi untuk memperkuat/meyalurkan arus aplikasi. Tipe ini mengatur semua hubungan yang menggunakan level aplikasi.

Cara kerjanya adalah apabila ada pengguna yang menggunakan suatu aplikasi contohnya FTP untuk mengakses secara remote, maka gateway akan meminta user untuk memasukkan alamat host yang akan diakses. Saat pengguna mengirimkan user ID dan informasi lainnya yang sesuai dengan kebijakan maka gateway akan melakukan hubungan terhadap aplikasi tersebut pada remote host dan menyalurkan data pada keua titik. Dan begitu juga sebaliknya.

Kelebihan dari tipe ini adalah relatif aman daripada tipe packet filtering karena dengan penerapan tipe ini akan lebih mudah untuk memeriksa (audit) semua aliran yang masuk pada level aplikasi.

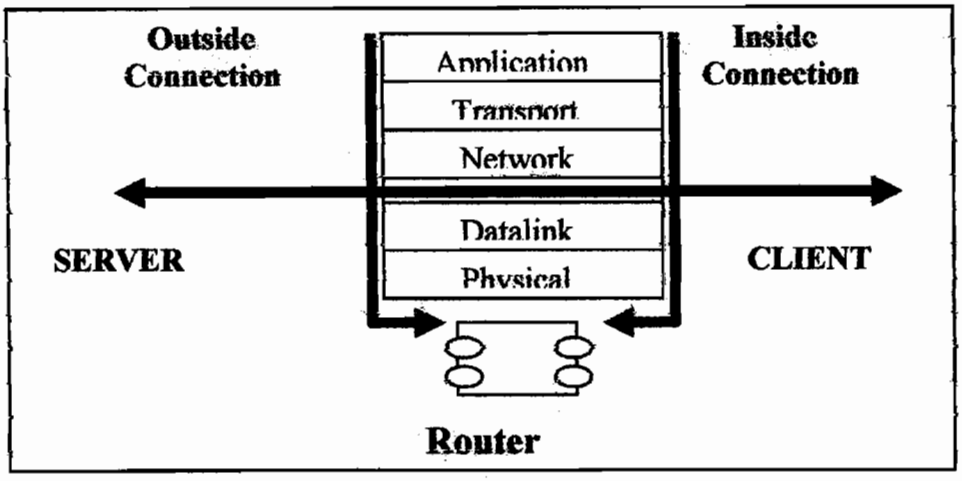


Gambar 2.4 Aplication Level Gateway

3. Statefull Inspections

Tipe ini menggabungkan fitur terbaik dari dua tipe sebelumnya. Tipe ini mempunya sebuah *engine* yang beruap aplikasi yang diletakkakn diantara 2 *layer* yaitu diantara *Layer Datalink* dan *Layer Network.* Paket data yang berasal dari *layer network* kedua arah akan diperiksa sesuai dengan kebijakan keamanan. Tipe ini dapat melihat keseluruhan isi packet sehingga dapat dibuat sebuah kebijakan keamanan dengan mempertimbangkan isi dan muatan dari paket data tersebut.

Tipe ini dapat menciptakan sebuah koneksi yang dinamis dimana sebuah servis atau *port* akan dibuka dan ditutup secara otomatis sesuai dengan kebutuhannya.

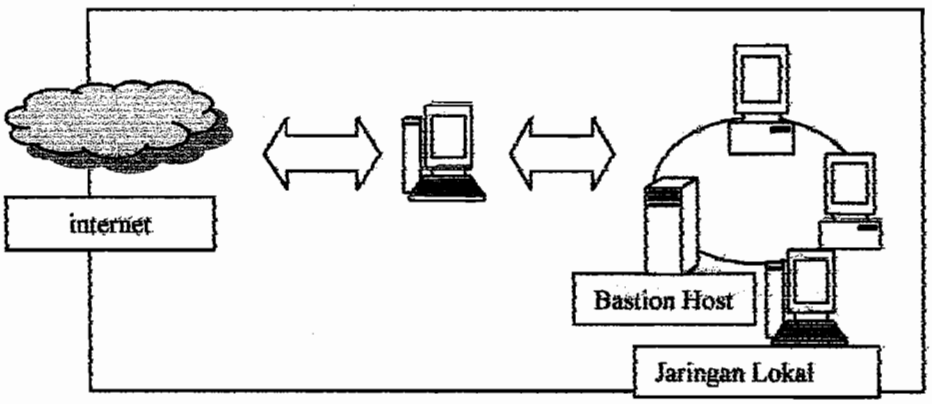


Gambar 2.5 Statefull inspection

2.6.4. Arsitektur Konfigurasi Firewall

1. *Dual Homed Gateway/DHG*

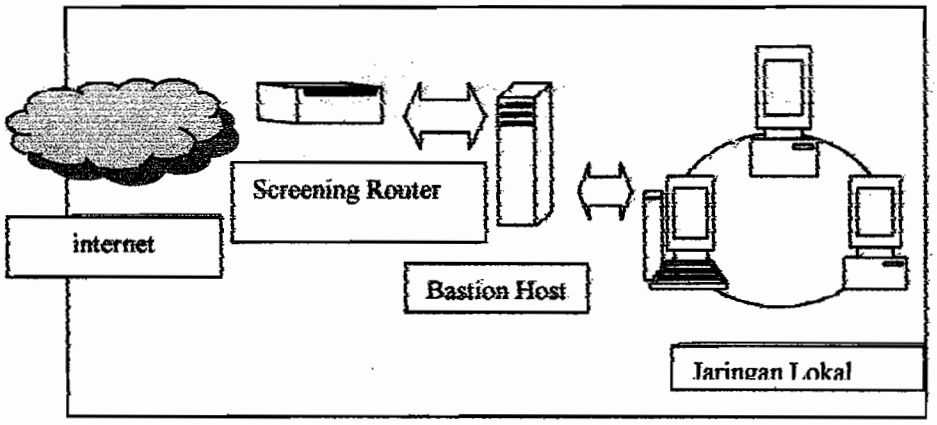
   Arsitektur ini menggunakan sebuah komputer dengan (paling sedikit) dua kartu jaringan. Kartu jaringan yang pertama terhubung dengan jaringan internet dan yang lain terhubung dengan jaringan lokal.



Gambar 2.6 Dual Homed Gateway

2. *Scereened Host Gateway/SHG*

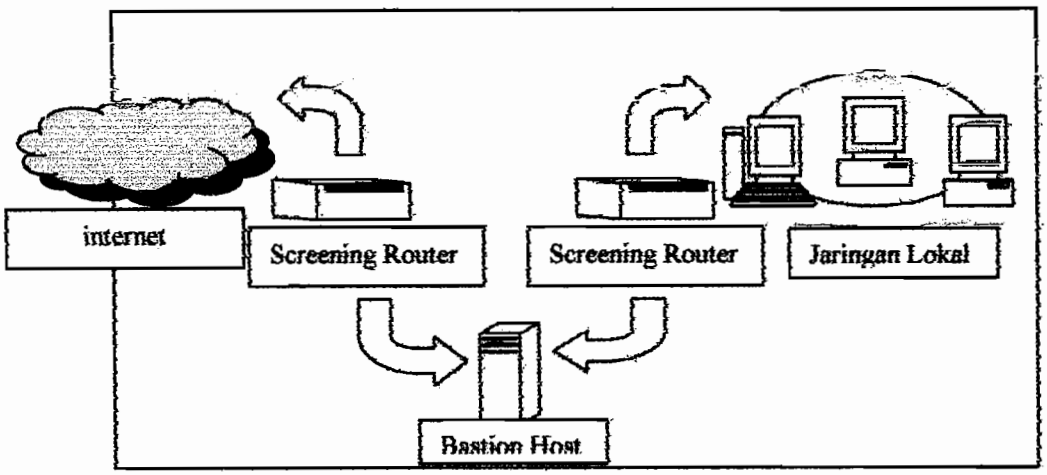
Pada arsitektur ini fungsi firewall dilakukan oleh sebuah *screening- router* dan *bastion host.* Router dikonfigurasi sedemikian rupa sehingga menolak semua *trafic* kecuali yang ditujukan ke *bastion host.* Sedangkan pada jaringan internal tidak dikenai pembatasan.



Gambar 2.7 Scereened host Gateway

5. *Screened Subnet Gateway/CCG*

Firewall dengan arsitektur ini menggunakan dua buah *screening router* dan jaringan tengah (perimeter network) antara kedua router tersebut. Kelebihan dari arsitektur ini adalah pada optimalisasi penempatan server yang dapat ditempatkan diantara kedua *screening router* (perimeter network). Arsitektur ini dapat mengatur kebijakan keamanan pada masing-masing daerah yang dibatasi oleh *screening router.*



Gambar 2.8 Scereened Subnet Gateway

2.7. Shorewall

Shore Line Firewall yang lebih dikenal dengan 'Shorewall' merupakan sebuah perangkat lunak yang digunakan untuk melakukan setting *firewalling* berbasis fungsi *IP-Tables* pada Linux. Perangkat lunak ini membuat sebuah sistem *firewall* dengan melakukan setting pada file konfigurasi yang disediakannya.

Aturan penulisan yang rumit pada fungsi IP-Tables disederhanakan pada perangkat lunak ini. Untuk menambahkan aturan kebijakan keamanan jaringan *(rules)* pada IP-Tables, seorang administrator tidak perlu menuliskan perintah IP-Tables yang rumit. Administrator hanya perlu melakukan edit file konfigurasi yang disediakan oleh Shorewall.

Shorewall membaca file konfigurasi tersebut dan dengan bantuan fungsi IP-Tables maka setting kebijakan keamanan yang telah di set pada file konfigurasi akan dijalankan sebagai fungsi IP-Tables. Shorewall dapat digunakan pada sebuah perangkat *firewall*, router, server, ataupun sistem Linux yang berdiri sendiri

(standalone). Shorewall hanya kompatibel dengan Fungsi IP-Tables dan tidak dengan fungsi *firewalling* lain (IP-Chains, IPawdm)

Program yang berupa file konfigurasi ini bebas untuk digunakan dan didistribusikan bahkan dimodifikasi ulang tanpa harus melakukan ijin untuk mendapatkan lisensi.

Untuk dapat menjalankan *Shorewall* dibutuhkan berbagai syarat pendukung yaitu :

a. Kernelversi 2.4.2 atau 2.6.6 untuk proses trafic control dapat berjalan pada versi 2.4.18 .
b. IP-Tables versi 1.2 atau lebih tinggi.
c. *iproute utility* paket ini biasanya sudah termasuk dalam distribusi Linux namun belum diinstall.
d. *Bourne Again Shell (BASH)* yang mendukung format *variabel extension.*

Shorewall mempunyai berbagai macam file konfigurasi yang digunakan untuk membuat sebuah sistem sebagai *firewall.* File-file konfigurasi tersebut diantaranya adalah:

a. `//etc/Shorewall/Shorewall.conf` file konfigurasi ini digunakan untuk melakukan setting umum yang menyangkut beberapa parameter firewall.
b. `//etc/Shorewall/params` file konfigurasi ini digunakan ketika akan dilakukan perluasan variabel dengan menggunakan variabel yang akan digunakan pada file konfigurasi lain.

c. //etc/*Shorewall*/zones File konfigurasi ini membagi sistem firewall kedalam *zone*/daerah tertentu sesuai dengan kebutuhan sistem.

d. //etc/*Shorewall*/policy file konfigurasi ini menentukan penetapan kebijakan *default* pada sistem firewall.

e. //etc/*Shorewall*/rules file ini digunakan untuk mendefinisikan kebijakan keamanan *(rules)* secara menyeluruh termasuk yang ada pada file //etc/*Shorewall*/policy

f. //etc/*Shorewall*/interfaces file ini memuat informasi deskriptif tentang kartu jaringan yang digunakan oleh sistem firewall.

g. //etc/*Shorewall*/masq file konfigurasi ini digunakan untuk melakukan konfigurasi *masquerading* atau *SNAT (Source Network Address Translation)*

h. //etc/*Shorewall*/nat digunakan untuk melakukan setting terhadap fungsi *one to one NAT (Network Address Translation).*

i. //etc/*Shorewall*/start ini merupakan file eksekusi yang digunakan untuk menjalankan *Shorewall.*

j. //etc/*Shorewall*/blacklist file ini berisi nomor IP, Subnet atau *MAC Address* yang dikenai *blacklist.*

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Sistem

Pengembangan setting *firewalling* pada Linux dapat dilakukan dengan berbagai maca m metode dan aplikasi. Untuk dapat mengembangkan sebuah program aplikasi yang dapat membantu seorang pengguna maka perlu mengetahui kebutuhan pengguna yang selanjutnya dibuat dalam sebuah perancangan sistem menggunakan model sistem.

3.1.1. Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna dimaksudkan untuk mengumpulkan kebutuhan pengguna yang akan dipenuhi pada pembuatan *software.* Kebutuhan yang harus dipenuhi aplikasi *web base firewall* yang dibuat adalah sebagai berikut :

a. Dapat melakukan fungsi *firewalling* pada linux.
b. Aplikasi yang berbasis grafis dengan user interface yang memudahkan setting *firewalling* pada linux.
c. Aplikasi yang dibuat dapat diakses dari jarak jauh dalam satu jaringan.

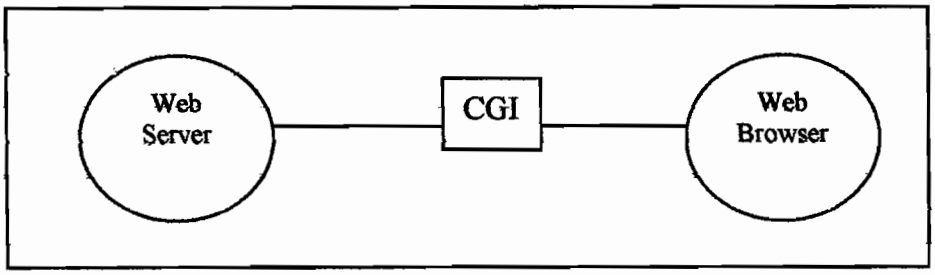
Pada penelitian ini dilakukan proses pemenuhan kebutuhan yang didefinisikan diatas dengan melakukan proses sebagai berikut :

1. Penggunaan metode berbasis web dengan CGI

Implementasi teknologi web digunakan pada pembuatan *software* ini. Implementasi teknologi web digunakan karena akan memenuhi kebutuhan akan aplikasi berbasis grafis dan dapat diakses dari jarak jauh. Administrator tidak perlu melakukan setting di komputer yang dijadikan *firewall* namun dapat dilakukan secara *remote* (jarak jauh) melalui komputer lain yang terhubung dengan komputer yang akan disetting. Proses ini dapat dilakukan dengan bantuan *webserver* (pada komputer yang akan di setting) dan *browser* (pada komputer *client/* komputer yang digunakan administrator).

Dengan metode ini *Software* yang dibuat dapat menampilkan tampilan sesuai dengan kemauan pengguna dan sesuai dengan perintahnya. Proses ini merupakan proses dinamis, yaitu proses yang menunggu masukan *(input)* dari pengguna melalui tombol-tombol dan mengembalikannya sesuai dengan perintah/fungsi tombol yang dimaksud oleh pengguna. Proses dinamis ini melibatkan peranan komputer *client* dan *server* dengan bantuan *browser* dan *web server*. *Software* pada *client* akan melakukan permintaan dengan mengirimkan *script* ke *web server* dan *web server* akan mengirimkan hasilnya ke komputer *client.*

Proses tersebut tidak dapat dilakukan secara langsung. Proses dinamis tersebut membutuhkan perantara antara *web browser* dan *web server*. Pada penelitian ini digunakan CGI *(Common Gateway Interface)* sebagai perantara antara *web browser* dan *web server*. CGI merupakan *Gateway* antara kedua aplikasi tersebut.



Gambar 3.1 CGI sebagai aplikasi Gateway

CGI digunakan pada penelitian ini karena *web server* ataupun *web browser* tidak mengenali *script* IP-Tables sebagai *software firewall.* CGI berperan sebagai penghubung bahasa *software* IP- Tables sehingga *script* IP-Tables dapat digunakan pada komunikasi antara *web browser* dan *web server.*

2. Penggunaan Shorewall

Pada penelitian ini digunakan file konfigurasi *Shorewall* untuk memudahkan penulis dalam melakukan pemenuhan kebutuhan terhadap setting *firewalling* berbasis *IP-Tables.* Fungsi *Shorewall* sebenarnya adalah sebagai penjembatan antara *Software* dan fungsi *IP-Tables* itu sendiri. Dalam menjalankan perintah IP-Tables, *Shorewall* membaca file konfigurasi yang sudah ditambah dengan sintaks khusus yang mewakili perintah IP-Tables. Selanjutnya *Shorewall* mengkonfersi file kofigurasi tersebut ke dalam *script IP-Tables* dan kemudian menjalankannya.

Penggunaan file konfigurasi ini dimaksudkan untuk menjembatani antara Program yang dibuat dengan fungsi. *Shorewall* sangat membantu terutama dalam hal penulisan sintaks *IP-Tables.* Hal ini bisa terjadi karena sebuah *script* perintah yang rumit dalam *IP-Tables* dapat diganti dengan sebuah variabel setelah mendefinikannya terlebih dahulu pada *Shorewall.*

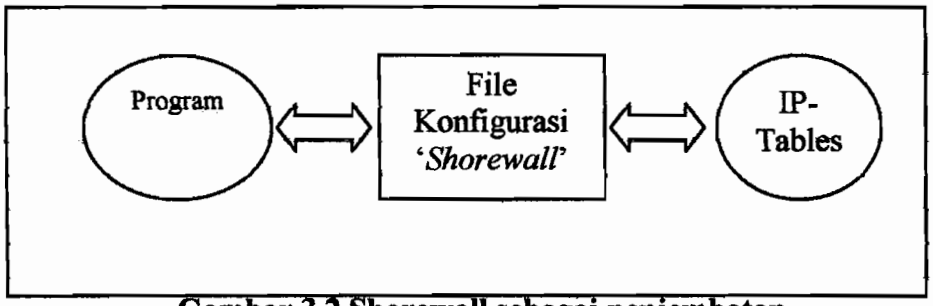
Selanjutnya variables yang telah didefinisikan sebagai *script* IP-Tables tersebut dapat digunakan dalam file konfigurasi lain pada *Shorewall.* Misalnya saja sebuah *script* IP-Tables berikut ini :

```
Iptables -I  INPUT  -p tcp  -s 192.168.0.1/32  -d 0/0 -j DROP
```

Script ini berfungsi untuk melakukan penutupan protokol TCP jika akses dilakukan dari komputer dengan IP 192.168.0.1 dengan prefix 32 dan ditujukan ke semua alamat. S*cript* tersebut dapat dituliskan kedalam file konfigurasi `//etc/Shorewall/rules` *Shorewall* dengan mudah yaitu :

```
LOC  ALL  TCP  DROP
```

Tidak semua file konfigurasi yang ada pada *Shorewall* digunakan pada pembuatan penelitian ini. Penelitian ini hanya menggunakan file konfigurasi yang diperlukan untuk pengimplementasian fungsi *software* yang akan dibuat.

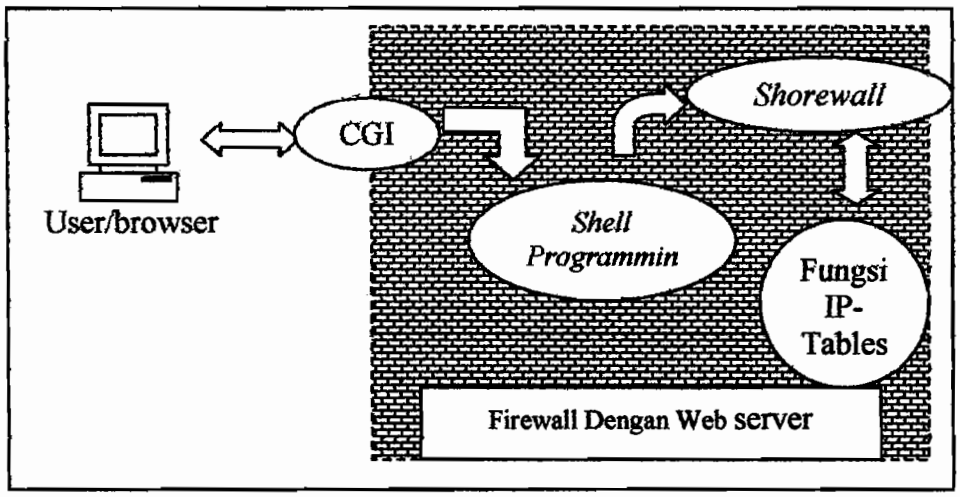


Gambar 3.2 Shorewall sebagai penjembatan

Dengan menerapkan aturan-aturan diatas maka lalu-lintas koneksi jaringan akan dapat dikontrol untuk mencapai tingkat keamanan jaringan yang diharapkan. *Firewall* dapat terealisasi menggunakan fungsi *IP-Tables* dengan bantuan *Shorewall.*

3.1.2. Alur Sistem

Pembuatan *software* dengan menggunakan berbagai fungsi dan konfigurasi mempunyai alur sebagai berikut:



Gambar 3.3 Alur Sistem

Software yang akan dibuat akan ditanamkan / diinstall pada sebuah komputer yang dijadikan sebagai *firewall* pada sistem dan sekaligus dijadikan sebagai PC router. Pada komputer tersebut juga terdapat sebuah *web server* aktif yang digunakan sebagai server *software* yang akan dibuat sehingga *software* akan dapat diakses oleh pihak pengguna dengan bantuan *browser.*

Web server akan menterjemahkan permintaan dari *software* yang sedang digunakan oleh pengguna (administrator) dan mengembalikannya berupa keluaran yang berbentuk antarmuka grafis. Jika permintaan tersebut berupa sebuah konfigurasi IP-Tables maka server akan menterjemahkannya kedalam bahasa file konfigurasi *Shorewall*, dan kemudian *Shorewall* akan mengubahnya ke dalam *script* IP-Tables. Fungsi IP-Tables akan

mengeksekusi *script* tersebut. Setelah dilakukan restart pada fungsi IP-Tables.

3.1.3. Model Sistem (UML)

Cara kerja sistem yang akan dibuat digambarkan dengan menggunakan notasi UML *(Unified Model Language)*. Dalam penggambaran sebuah sistem, notasi ini mempunyai dua buah diagram yang akan digunakan pada permodelan ini yaitu *use case diagram* dan *squence diagram.*

Use case diagram merupakan sebuah diagram bernotasi yang menggambarkan cara kerja sistem dalam berinteraksi dengan pengguna sistem. Dalam sebuah sistem dapat terdiri dari beberapa pengguna yang selanjutnya disebut sebagai *aktor.*

Squence diagram menggambarkan bagaimana sistem memproses setiap tahapan fungsi yang berjalan pada sistem tesebut.

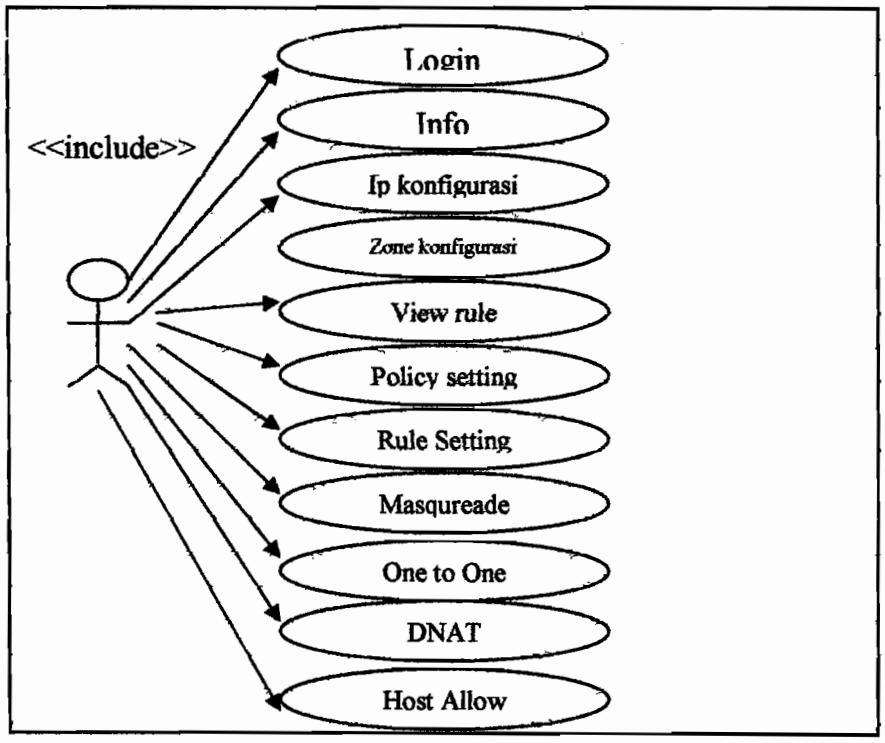
a. Use Case Diagram

Use Case Diagram merupakan permodelan dari tiap-tiap skenario (perancangan) fungsi pada sistem. Setiap skenario fungsi pada sebuah sistem digambarkan dengan sebuah *use case.* Sedangkan orang yang mengakses atau menggunakan sistem digambarkan dengan seorang *aktor*.

Pada Use case diagram ini hanya terdapat seorang *aktor* yang menggunakan sistem. *Aktor* tersebut adalah administrator (Admin). Admin adalah pemakai *software* yang merupakan seorang administraror (pengelola jaringan). *Software* yang akan dibuat hanya

dapat diakses oleh *aktor* ini. Hal ini karena *software* merupakan program setting keamanan jaringan. Jadi satu-satunya orang yang berhak mengakses *software* yang akan dibuat adalah seorang administrator.

Software yang akan dibuat didesain untuk digunakan oleh *aktor* ini (administrator). Untuk dapat menggunakan software yang akan dibuat seorang administrator harus mempunyai pengguna *name* dan *password* (kata kunci). *User name* dan *password* akan divalidasi pada proses *login* dengan kepentingan pembatasan terhadap pemakaian *software*. Setelah proses *login* berhasil, maka admin dapat menggunakan menu-menu yang disediakan *software* yang akan dibuat untuk melakukan setting *firewalling*.



Gambar 3.4 Use Case Diagram

Pada *use case diagram* tersebut, masing-masing *use case* mempunyai fungsi yang berbeda-beda. Fungsi dari masing-masing *use case* adalah sebagai berikut :

1. *use case* Login

   use case ini merupakan proses login bagi seorang pengguna *software* yaitu administrator. Untuk melakukan pengaksesan *software*, seorang administrator harus melewati proses ini terlebih dahulu.

2. *use case* Info

   use case ini memberikan informasi dari sistem yang dipakai oleh *software* yang akan dibuat. Informasi tersebut adalah berupa versi komponen *software* yang digunakan beserta hardware lainnya yang digunakan.

3. *use case* IP konfigurasi

   use case ini digunakan untuk melakukan proses konfigurasi terhadap kartu jaringan yang digunakan oleh *software* yang akan dibuat. *Admin* dapat melakukan konfigurasi dengan memasukkan nomor IP konfigurasi lainnya pada use case ini.

4. *Use case* Host Allow

   Use case ini merupakan proses pemberian ijin kepada sebuah host atau maksimal 3 yang dapat menjalankan program yang akan dibuat.

5. *use case* Zone konfigurasi

   use case ini merupakan proses setting *zone* atau wilayah yang ada pada sistem yang digunakan berdasarkan koneksi dari kartu jaringan yang digunakan. Misalnya kartu jaringan yang terhubung dengan internet diberi nama dengan 'net' dan sebagainya.

6. *use case* View Rule

   use case ini merupakan proses pemberian informasi mengenai *rule* yang telah di terapkan pada sistem.

7. *use case* Policy Setting

   use case ini merupakan proses setting terhadap keadaan default program. Jika rule atau aturan tidak diisi maka system akan menjalankan keadaan ini.

8. *use case* Rule Setting

   use case ini merupakan proses setting terhadap *rules* yang akan diterapkan pada sistem.

9. *use case* Masqureade

   use case ini merupakan suatu proses *Netwok Address Translation (NAT)*. Fungsi ini digunakan untuk melakukan *masqureade* atau translasi sekelompok alamat IP lokal pada suatu subnet (kelas jaringan) ke dalam suatu nomor IP public.

10. *use case* One to One

use case ini merupakan proses translasi satu alamat IP lokal ke dalam satu alamat IP global.

Fungsi traslasi dengan tipe *one to one* di berikan pada fungsi ini.

11. *use case* DNAT

    Use case merupakan fungsi yang melakukan proses *DNAT (Destination Network Address Translation)*. Proses ini melakukan transalsi terhadap alamat IP external yang akan mengakses sebuah alamat IP lokal dengan *port* tertentu.

12. *use case* Restart : *use case* ini merupakan proses *restart* sistem (*software*) untuk dapat menjalankan *rule* yang telah di set pada masing-masing *zone*.

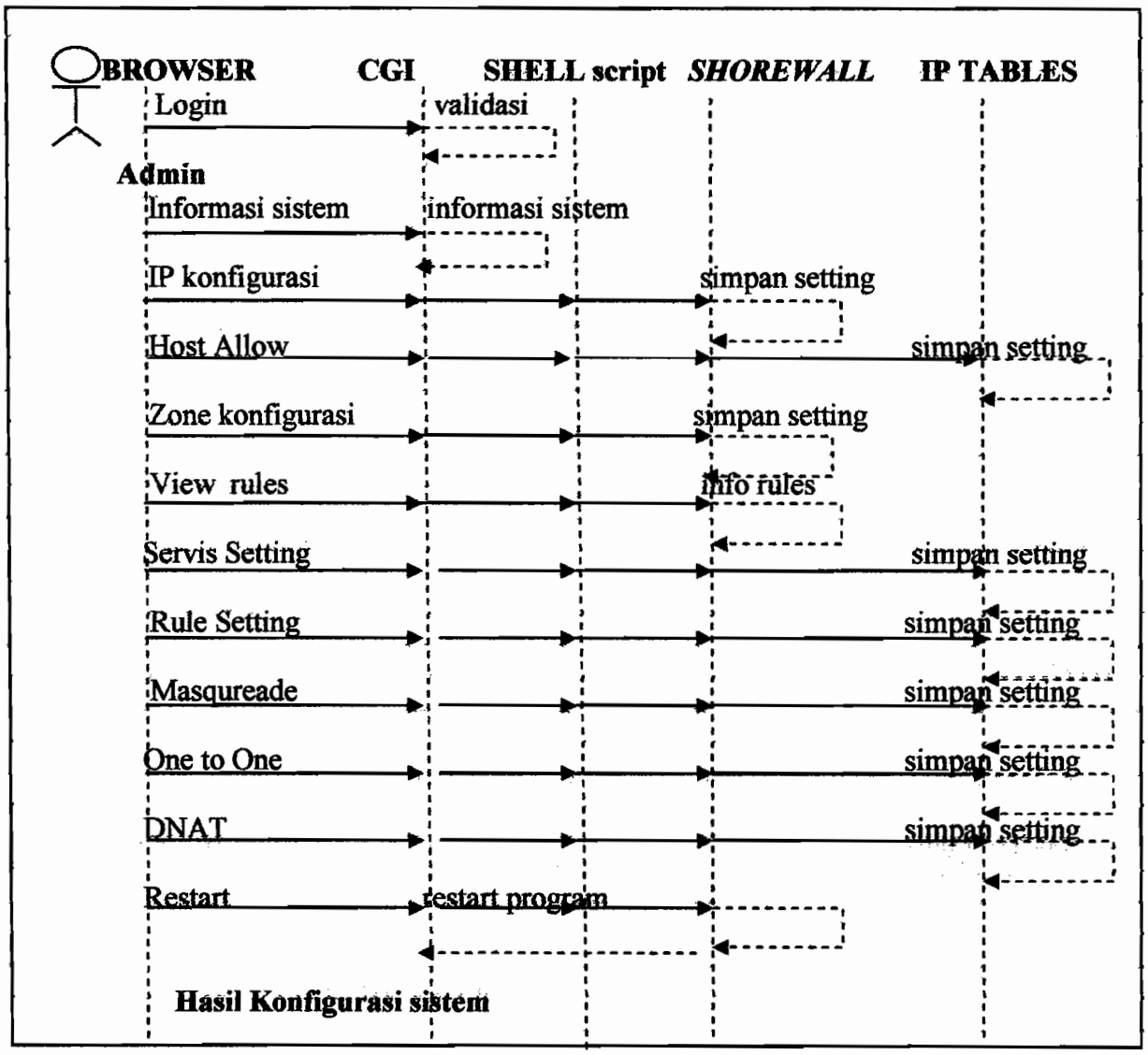
b. Squence Diagram

Diagram ini merupakan diagram permodelan yang memodelkan cara kerja dari sistem *software* yang akan dibuat. Diagram ini juga menggunakan permodelan *aktor* yang sidah didefinisikan pada diagram sebelumnya *(use case diagram)*.

Squence diagram menjelaskan tahap demi tahap proses yang terjadi untuk menghasilkan suatu fungsi di dalam sebuah *use case*. Pada penelitian ini proses yang terjadi dihasilkan dari proses kerja berbagai program dan file konfigurasi yang digunakan.

Squence diagram akan menggambarkan tahap-tahap proses pembentukan suatu fungsi pada *use case*. Tahap-tahap tersebut

melewati beberapa proses dari penggunaan program aplikasi dan file konfigurasi yang digunakan pada penelitian ini yaitu : CGI, *Shorewall* dan IP Tables.



Gambar 3.5 Squence diagram

Dari *squence diagram* diatas dapat dilihat bahwa sebuah fungsi pada *use case* tidak selalu dihasilkan oleh IP-Tables, *Shorewall*, ataupun CGI. Masing masing fungsi pada *use case*

dihasilkan dari proses kerja yang dilakukan oleh program aplikasi ataupun file konfigurasi yang berbeda- beda.

1. Proses login dihasilkan dari validasi yang dilakukan oleh CGI.
2. Fungsi informasi sistem yang diberikan oleh *software* yang akan dibuat juga dihasilkan dari proses yang dilakukan oleh CGI.
3. Fungsi IP konfigurasi dan Zone konfigurasi dihasilkan dari proses yang dilakukan oleh file konfigurasi *Shorewall.*
4. Fungsi setting dari program *firewall* yang akan dibuat yaitu : View rule, Servis setting, Rule setting, dihasilkan oleh proses IP-Tables sebagai *software* utama setting *firewalling* pada LINUX.
5. Fungsi setting NAT *(Network Address Translation)* yaitu: Masqureade, One to One, dan DNAT juga dihasilkan oleh proses yang dilakukan IP-Tables.
6. Fungsi restart akan menghasilkan setting *firewalling* pada *software* yang akan dibuat dihasilkan oleh *Shorewall* yang melakukan eksekusi terhadap setting yang telah dilakukan terhadap file konfigurasi yang ada didalamnya.

Penggunaan bebagai program apilkasi dan file konfigurasi diatas diharapkan dapat membantu dalam mewujudkan suatu *software* untuk memenuhi tujuan penelitian ini.

Pada tahap selanjutnya *Use case* diimplementasikan menjadi sebuah menu yang mempunyai fungsi tertentu pada *software* yang akan dibuat. Menu tersebut akan menampilkan suatu bentuk grafis dari file konfigurasi Shorewall yang diharapkan dapat membantu administrator untuk melakukan setting *firewalling* berbasis *IP-Tables.*

3.2. Perancangan Sistem

Tahap selanjutnya dari penelitian ini adalah perancangan piranti yang digunakan oleh sistem yang akan dibangun. Piranti ini berupa perangkat keras *(hardware)*, perangkat lunak *(software)*. Arsitektur *firewall* juga akan diterapkan pada pembuatan sistem *firewall* ini.

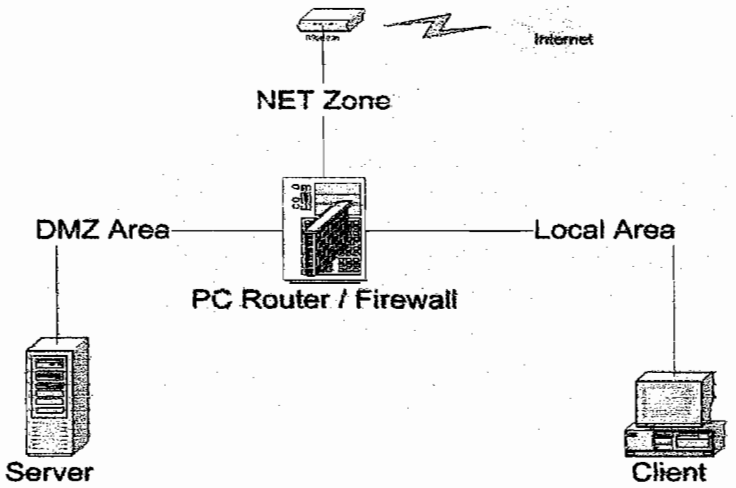
3.2.1. Konfigurasi Jaringan

Pada penelitian ini digunakan arsitektur *Screened Subnet Gateway.* Konfigurasi ini menggunakan sebuah komputer yang dijadikan sebagai *PC Router* untuk melakukan tugas sebagai *screening router.* PC (*Personal Computer*) ini juga digunakan sebagai perangkat dimana *firewall* ditanamkan (diinstall).

Personal Komputer yang berada diantara dua buah jaringan ini menggunakan 3 buah kartu jaringan. Kartu jaringan yang pertama terhubung dengan jaringan lokal dan sekaligus melakukan tugas *Screening router* yang pertama, dan kartu jaringan yang kedua akan diasumsikan terhubung dengan jaringan global (internet) sekaligus melakukan fungsi *screening router* yang

kedua. Kartu jaringan yang ketiga dihubungkan dengan daerah server (DMZ / *Demiliterized Zone*) yang berada pada *perimeter network.*

Masing masing kartu jaringan dapat diset dengan kebijakan aturan keamanan jaringan *(rules)* yang disesuaikan dengan fungsi masing –masing daerah *(zone)*.



Gambar 3.6 Konfigurasi jaringan

3.2.2. Konfigurasi Hardware dan Software

Untuk mengembangkan sebuah konsep *firewalling* dengan *software* dibuat, dibutuhkan perangkat lunak (*software)* dan perangkat keras (*Hardware*) yang mendukung.

Perangkat Keras yang digunakan untuk menerapkan *software* yang akan dikembangkan harus memenuhi kebutuhan minimal sebagai berikut :

Prosesor : dengan kecepatan setara 166 MHZ.

Memori : 32 Mb

Vga : 2 Mb

Hardisk : 2 Gb

Kartu jaringan : 3 Buah kartu jaringan 10 kbps

Perangkat lunak yang digunakan dalam pembuatan sistem *firewall* berupa Sistem operasi dan *software* aplikasi lannya yang terdiri dari:

a. Linux Redhat 9.0 sebagai sistem operasi yang akan digunakan sebagai *firewall.*
b. IP-Tables sebagai program dasar setting *firewall* pada Linux.
c. *Shorewall.* Penggunaan File Konfigurasi ini dimaksudkan untuk memudahkan penulisan *script* IP-Tables pada pembuatan *software.*
d. Apache Web Server. Digunakan sebagai *web server* dari aplikasi *software* yang akan dibuat.
e. Perl. Digunakan sebagai bahasa pemprograman dalam penulisan CGI.

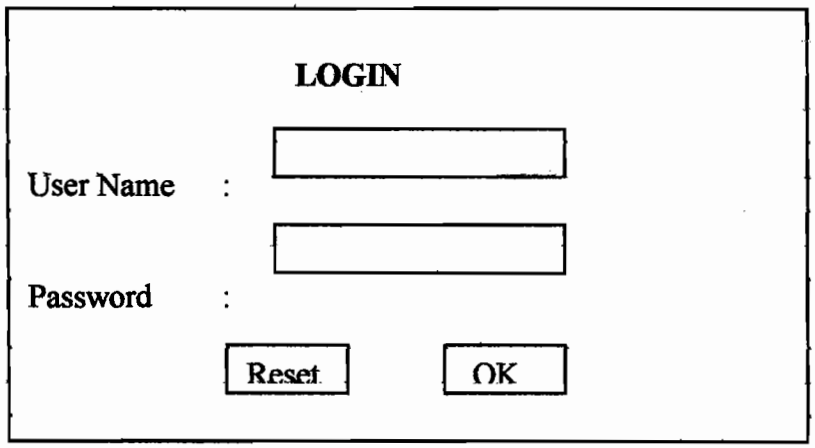
3.2.3. Antarmuka (User Interface)

Pemenuhan terhadap tujuan pembuatan sistem yaitu mempermudah setting *firewall,* dilakukan dengan cara menciptakan sebuah *software* yang mempunyai antarmuka dengan bentuk tampilan grafis (*GUI/Graphical User Interface*).

Adapun perancangan antarmuka pada *software* yang akan dibuat adalah sebagai berikut :

3.2.3.1. Perancangan Antarmuka login Administrator

Antarmuka ini berfungsi sebagai proteksi terhadap pengguna yang tidak diijinkan mengakses *software*. Satu-satunya pengguna yang diijinkan untuk mengakses *software* ini adalah administrator yang mempunyai *user name* dan *password* yang akan di validasi untuk diijinkan menggunakan *software*.



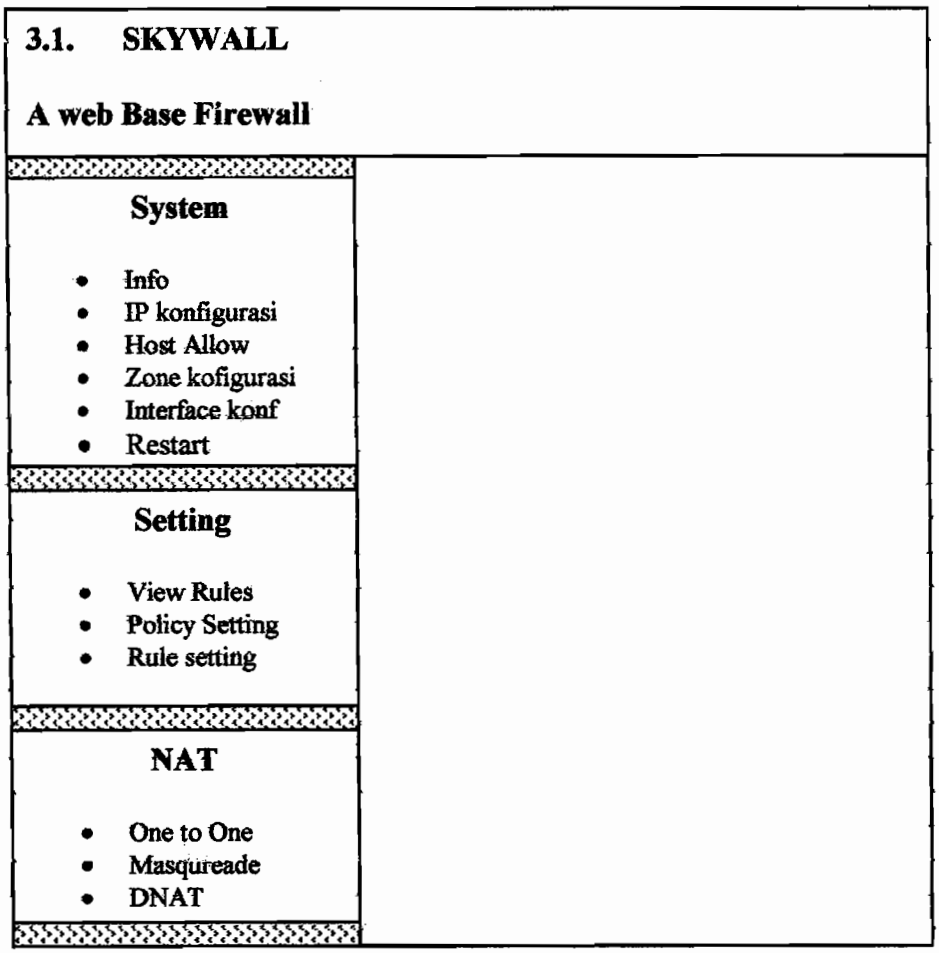
Gambar 3.7 Perancangan Halaman Login

Administrator harus memasukkan *user name* dan *password* pada form isian yang disediakan untuk dapat menggunakan *software*.

3.2.3.2. Perancangan Antarmuka Halaman Index

Jika Administrator diijinkan untuk mengakses *software*, maka administrator akan masuk pada halaman ini. Pada halaman ini terdapat tiga kelompok menu utama yang dikelompokkan berdasarkan fungsinya masing masing..

Berikut ini adalah gambar halaman index yang merupakan halaman pertama, yang sekaligus mengelompokkan fungsi-fungsi yang ada ke dalam kelompok menu.



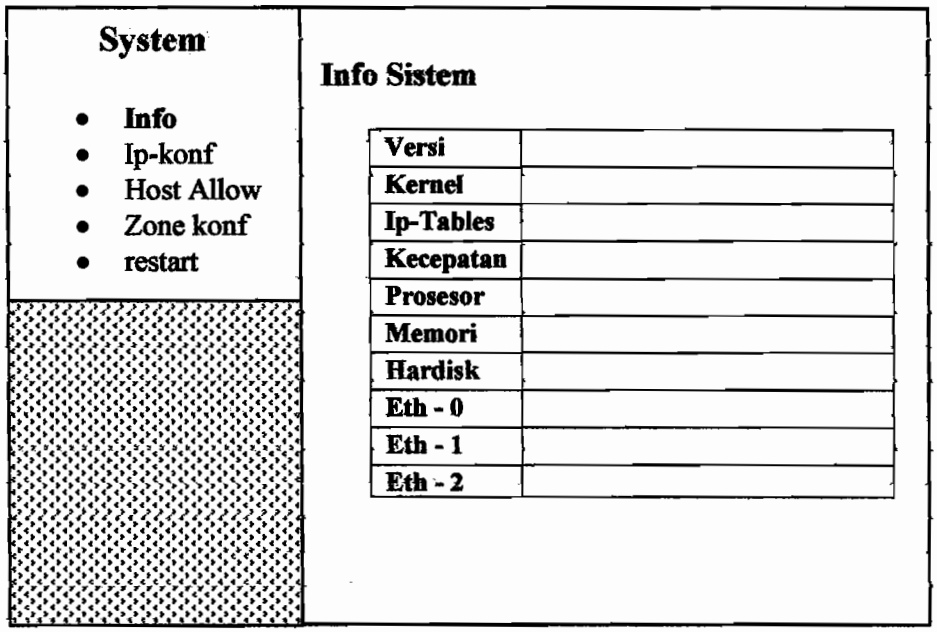
Gambar 3.8 Perancangan Halaman Index

3.2.3.3. Perancangan Antarmuka Kelompok menu Sistem

Pada Kelompok menu ini terdapat 3 Menu pilihan yang masing-masing akan menampilkan isi sesuai dengan fungsi pilihan Menu yang ditekan. Menu tersebut adalah Info, IP Konfigurasi, Zone Konfigurasi dan Restart

3.2.3.4. Perancangan Antarmuka Menu Info

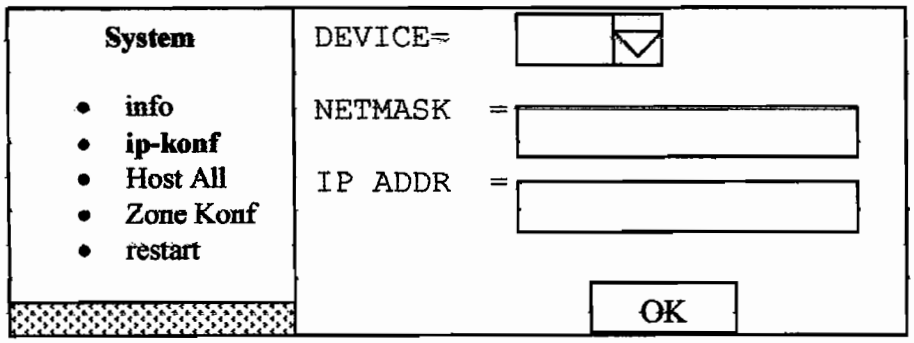
Pada Menu ini menampilkan sebuah tabel yang menginformasikan dimana *software* ditanamkan *(diinstall)*. Menu ini akan menginformasikan sistem operasi yang digunakan, versi kernel, versi IP-Tables dan *hardware* yang digunakan oleh komputer tersebut.



Gambar 3.9 Perancangan Antarmuka Menu Info

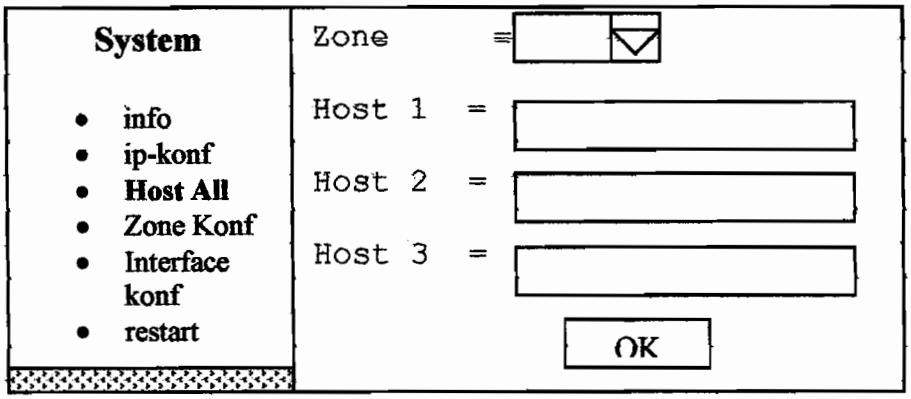
3.2.3.5. Perancangan Antarmuka Menu IP Konfigurasi

Menu ini menampilkan *form* konfigurasi untuk masing masing kartu jaringan yang digunakan. Setting tersebut terdiri dari nama kartu jaringan *(device)*, Alamat IP, Subnet, Broadcast, Gateway dan onboot. Administrator dapat melakukan setting kartu jaringan pada halaman ini dengan memasukkan alamat IP digunakan oleh masing-masing *device* pada form input yang disediakan.



Gambar 3.10 Perancangan Antarmuka Menu IP Konfigurasi

3.2.3.6. Perancangan Antarmuka Menu Host Allow



Gambar 3.11 Perancangan Antarmuka Menu Host Allow

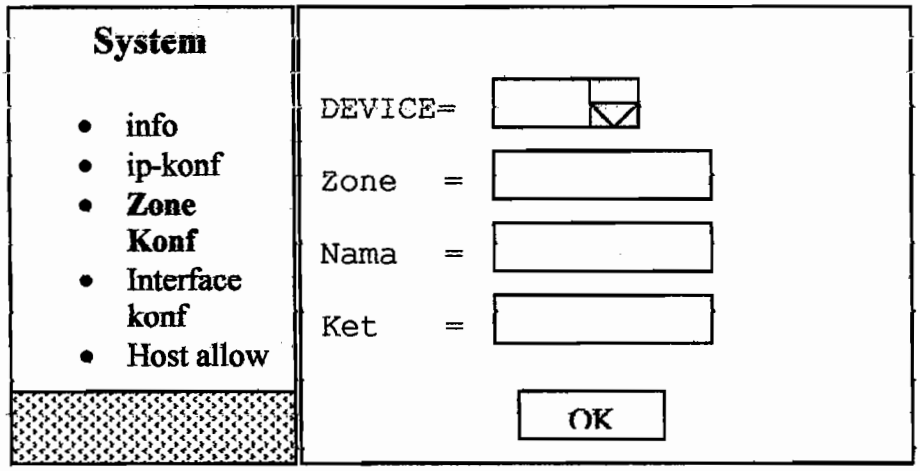
Menu Ini akan menampilkan form yang berisi inputan untuk host yang diijinkan mengakses program yang akan dibuat.

3.2.3.7. Perancangan Antarmuka Menu Zone konfigurasi

Pada halaman ini Administrator dapat melakukan pendefinisian *zone* (daerah)sesuai dengan hubungan kartu jaringan yang digunakan. Misalnya kartu jaringan yang pertama terhubung dengan internet maka *zone* dari kartu ini (eth-0) dapat dinamai dengan NET *zone*. Kartu yang terhubung dengan lokal area dapat dinamai

dengan LOC dan kartu yang terhubung dengan *Demiliterized Zone* dinamai dengan DMZ atau dapat diganti dengan nama-nama lain.

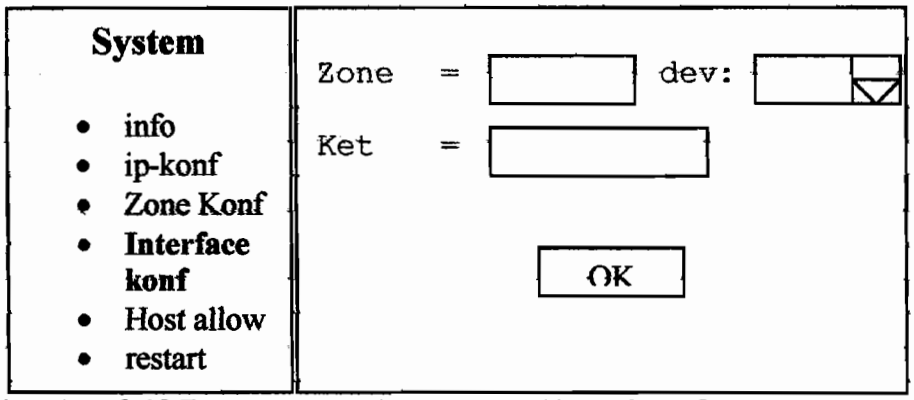
Hal ini dilakukan untuk mempermudah setting *firewall* dengan melakukan definisi area *(zone)* jaringan sekaligus memberikan variabel terhadap *device* kartu jaringan yang digunakan.



Gambar 3.12 Perancangan Antarmuka Menu Zone Konfigurasi

3.2.3.8. Perancangan Antarmuka Menu Interface Konfigurasi

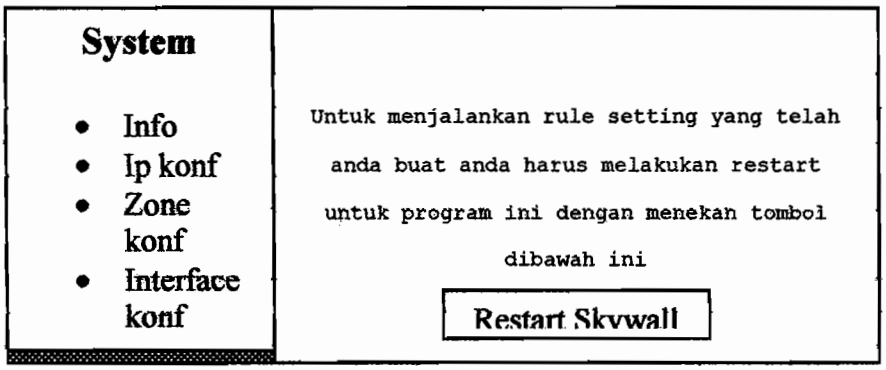
Menu ini akan menampilkan form pengaturan penggunaan kartu jaringan yang akan didefinisikan sebagai zone.



Gambar 3.13 Perancangan Antarmuka Menu interface konfigurasi

3.2.3.9. Perancangan Antarmuka Menu Restart

Menu ini mempunyai tombol yang berfungsi untuk melakukan restart terhadap *software*. Menu *Restart* berfungsi untuk mengeksekusi file konfigurasi yang telah dikonfigurasi dari menu setting dan menu sistem yang dilakukan oleh administrator.



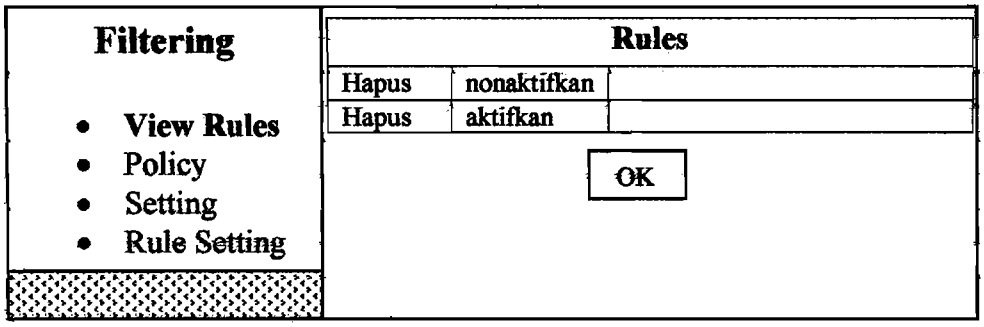
Gambar 3.14 Perancangan Antarmuka Menu Restart

3.2.3.10. Perancangan Antarmuka Kelompok Menu Setting

Kelompok menu setting merupakan menu inti pada *software* yang akan dibuat. Kelompok Menu Setting memiliki 3 buah Menu yaitu : View rule, Policy setting, dan Rule setting. Berikut ini adalah perancangan antarmuka dari masing-masing menu beserta fungsinya :

3.2.3.11. Perancangan Antarmuka Menu View Rule

Menu ini berfungsi untuk melihat konfigurasi yang sudah terjadi atau sedang dijalankan.

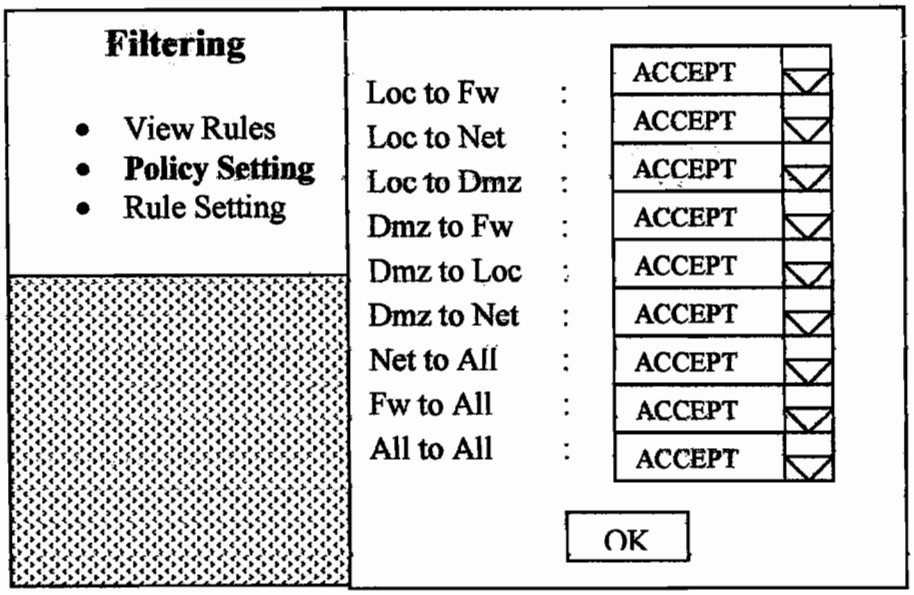


Gambar 3.15 Perancangan Antarmuka Menu View rule

Menu ini menampilkan kebijakan keamanan jaringan *(rule)* yang berjalan pada sistem. Dari Menu ini dapat dilihat tindakan *(action)* yang dilakukan terhadap suatu alamat IP, *protokol* ataupun *port* yang ada pada sistem. Dengan melihat tampilan ini maka seorang administrator dapat menentukan *rule* yang harus ditambahkan ataupun dikurangi pada sistem *firewall.* Admin juga dapat hanya mengaktifkan atau menonaktifkan rule yang sudah ada pada halaman ini.

3.2.3.12. Perancangan Antarmuka Menu Policy Setting

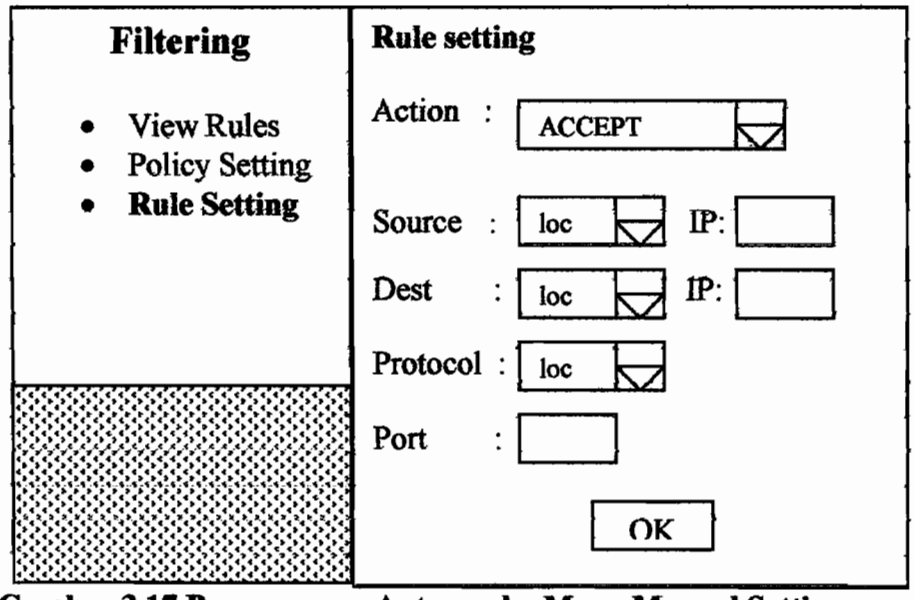
Pada menu ini disediakan setting untuk keadaan default bagi system. Keadaan default akan dijalankan jika tidak ada rule yang harus dijalankan. *Form* pada menu ini menyediakan policy untuk masing masing keadaan sesuai dengan implementasi dengan 3 zones.



Gambar 3.16 Perancangan Menu Policy Setting

3.2.3.13. Perancangan Antarmuka Menu Rule Setting

Pada Menu ini dapat didefinisikan secara detail *rule* yang akan di set pada alamat IP dan protokol pada *zone* tertentu sesuai dengan kebijakan sistem. *Form list* untuk *action, source* dan *destination, protocol* dan *port* disediakan pada halaman ini



Gambar 3.17 Perancangan Antarmuka Menu Manual Setting

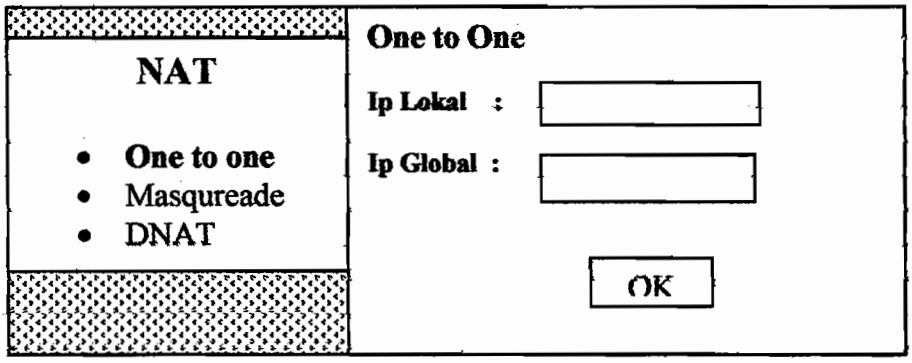
Pendefinisian alamat IP asal dan tujuan paket data dapat dilakukan dengan mengisi *form input* yang telah disediakan pada Menu ini. Dengan pendefinisian secara rinci, maka setting yang dilakukan pada sistem dapat lebih detail. Karena asal dan tujuan alamat IP, dan Port, dan *zone* dapat diset sesuai dengan kebijakan keamanan sistem.

3.2.3.14. Perancangan Antarmuka Kelompok Menu NAT

Kelompok menu ini berfungsi untuk melakukan NAT *(Network Address Translation)* atau traslasi / penterjemahan terhadap alamat IP Lokal ke alamat IP global dan sebaliknya. Hal ini terjadi karena alamat IP lokal tidak dapat digunakan untuk berkomunikasi pada jaringan global (internet). NAT pada Kelompok menu ini dibagi menjadi 3 Menu yaitu : One to One, Masqureade, dan DNAT.

3.2.3.15. Perancangan Antarmuka Menu One to One

Menu ini mempunyai fungsi untuk melakukan translasi satu alamat IP lokal ke dalam satu alamat IP global. Menu ini memberikan tampilan berupa *form input* untuk alamat IP lokal yang akan di akan translasi dan *form input* IP global yang digunakan untuk translasi beserta dengan *subnet mask* nya.

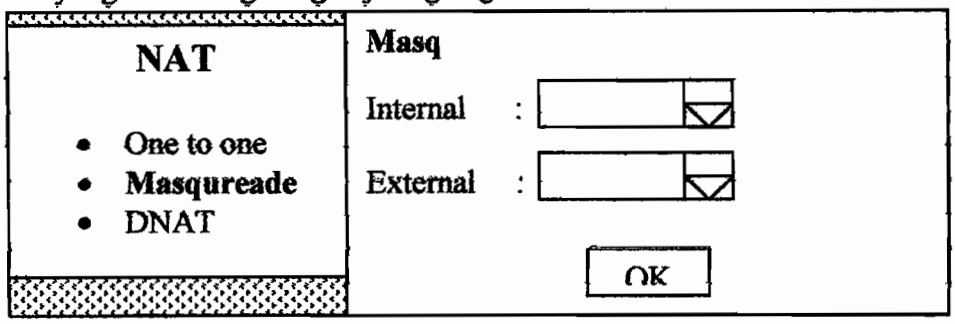


Gambar 3.18 Perancangan Antarmuka Menu One to One

Untuk melakukan translasi alamat IP secara *One to One*, seorang administrator tinggal memasukkan alamat IP lokal yang akan di traslasi dan alamat IP global beserta prefix-nya pada *form input* yang ada di halaman ini. Konfigurasi akan disimpan ketika tombol 'OK' ditekan.

3.2.3.16. Perancangan Antarmuka Menu Masqureade

Menu ini berfungsi untuk melakukan proses *Network Address Translation* (NAT) dengan metode *Masqureade*. Fungsi pada Menu ini akan dapat melakukan translasi alamat IP dalam satu *subnet mask* menggunakan alamat IP yang digunakan oleh sebuah kartu jaringan yang terhubung dengan jaringan global.



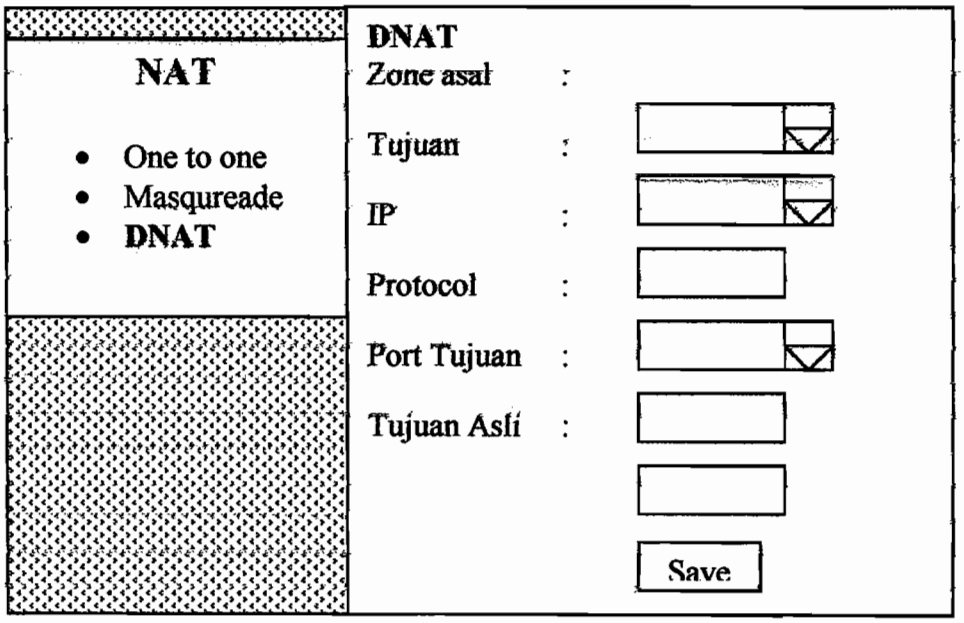
Gambar 3.19 Perancangan Antarmuka Menu Masqureade

Untuk melakukan fungsi NAT dengan *masqeureade,* administrator dapat menentukan *interface* mana yang akan dikenai *masquerade* dan *interface* mana yang digunakan sebagai alamat global pada halaman. Setting akan dijalankan seteleh disimpan pada file konfigurasi dengan menekan tombol 'OK'.

3.2.3.17. Perancangan Antarmuka Menu DNAT

Fungsi dari Menu ini adalah untuk melakukan proses DNAT *(Destination Network Address Translation).* Sebuah alamat IP yang akan masuk ke jaringan lokal dengan kepentingan akses tertentu harus melewati proses ini. Kebijakan keamanan dari proses ini dapat dilakukan pada Menu ini dengan *form* yang telah disediakan.

Pada Menu ini administrator dapat melakukan setting DNAT dengan menentukan asal paket data. Hal ini dapat dilakukan dengan memilih pada *form list* yang sudah disediakan. Tujuan akses yang diijinkan dapat ditentukan dengan mengisikan *zone,* alamat IP dan port yang diijinkan diakses dari luar jaringan lokal pada *form input* yang berupa *text field* pada halaman ini. Selanjutnya setting dapat disimpan pada file konfigurasi dengan menekan tombol OK. Konfigurasi akan dijalankan setelah dilakukan *restart* pada *software.*



Gambar 3.20 Perancangan Antarmuka Menu DNAT

BAB IV

IMPLEMENTASI

Implementasi merupakan tahap pengkodean dari bentuk perancangan. Pada bab sebelunya sudah dijelaskan mengenai perancangan program *firewall* berbasis web sebagai program bantu setting *firewall* pada Linux. Pada bab ini akan dijelaskan implementasi dari perancangan yang sudah dilakukan.

4.1. Lingkungan Implementasi

Untuk dapat menerapkan program yang akan dibuat ke dalam suatu sistem jaringan maka perlu didefinisikan lingkungan yang mendukung implementasi program.

4.1.1. Lingkungan Perangkat Lunak (Software)

Perangkat-lunak yang digunakan di dalam sistem ini adalah sebagai berikut:

a.	Sistem Operasi Router	: Linux Redhat 9.0
b.	Web server	: Apache versi 2.0
c.	Sistem Operasi Host (client)	: Linux atau Windows 98 keatas
d.	Pengelolaan Firewalling	: Shorewall 2.1.9 berbasis IP- Tables
e.	Bahasa Pemrograman	: Perl 5.0 , Shell Scripting

4.1.2. Lingkungan perangkat keras (Hardware)

Komputer Router yang sesuai untuk implementasi program ini pada sebiah sistem harus memenuhi kebutuhan minimal berikut ini:

a. Prosesor dengan kecepatan 166Mhz

b. Memori minimal 64 MB.

c. Hard disk minimal 2 GB.

d. 3 buah kartu jaringan.

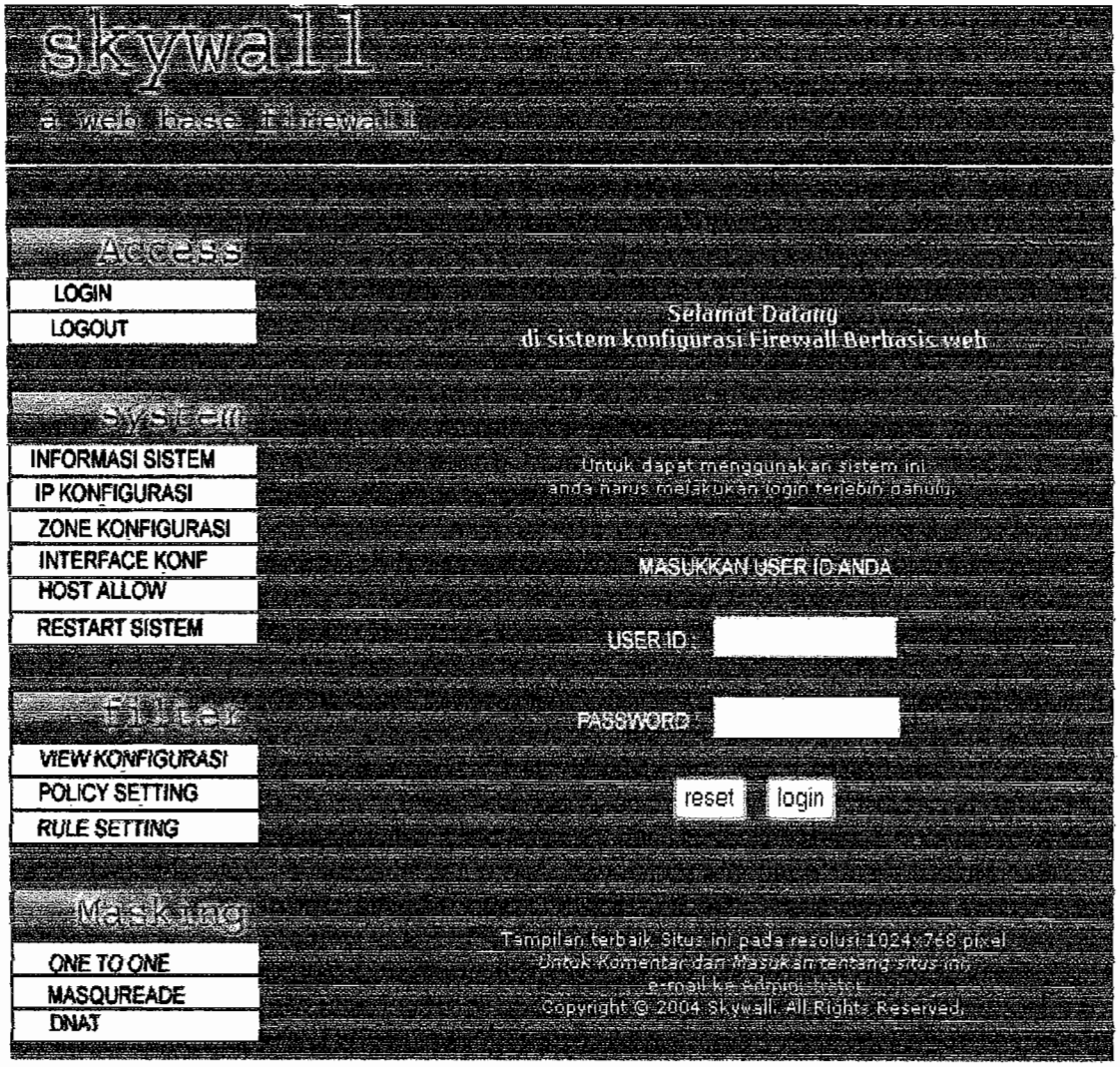
4.2. Karakteristik Pengguna

Program ini dibuat untuk memudahkan seorang administrator jaringan untuk melakukan setting *firewalling* pada linux Redhat 9.0. Namun untuk menggunakan program ini, seorang administrator harus memenuhi persyaratan berikut ini:

1. Memahami pengoperasian sistem operasi Linux.
2. Memahami konsep firewalling.
3. Mengerti setting file konfigurasi Shorewall secara manual.

4.3. Implementasi Antarmuka

Antarmuka adalah tampilan yang berinteraksi langsung dengan pengguna. Pada program ini dibuat antarmuka halaman index (halaman yang pertama diakses) dengan menu-menu yang dapat digunakan untuk melakukan setting *firewalling* sesuai dengan fungsi menu yang disajikan. Tampilan antarmuka halaman index adalah sebagai berikut.



Gambar 4.1 Halaman Index

Pada halaman ini terdapat 3 kelompok menu yang masing-masing mempunyai fungsi sebagai berikut :

1. Kelompok Menu Access

   Kelompok menu ini berisi menu yang mengatur keamanan program dari segi akses. Kelompok menu ini mempunyai menu sebagai berikut.

   a. Menu login

      Menu ini berfungsi untuk melakukan proses login untuk mendapatkan akses penggunaan program.

b. Menu logout

Menu ini untuk melakukan logout program agar siapapun yang melakukan akses program, harus melakukan login kembali.

2. Kelompok Menu System

Kelompok menu ini mempunyai 6 menu yang berhubungan dengan konfigurasi sistem dari berbagai setting. Menu-menu tersebut adalah sebagai berikut :

a. Menu Informasi sistem

Menu ini untuk mengetahui informasi mengenai komputer router.

b. Menu IP konfigurasi

Menu ini digunakan untuk melakukan setting IP-address kartu jaringan yang digunakan oleh komputer router.

c. Menu Zone konfigurasi

Menu ini digunakan untuk mendefinisikan zone sistem.

d. Menu Interface konfigurasi

Menu ini digunakan untuk mendefinisikan interface yang digunakan untuk zone pada sistem.

e. Menu Host Allow

Menu ini digunakan untuk mendefinisikan host yang diijinkan untuk mengakses program.

f. Menu Restart Sistem

Menu ini digunakan untukmelakukan restart program.

3. Kelompok Menu Filter

   Kelompok menu ini berisi menu setting aturan *security policy* yang merupakan inti dari proses *firewalling.*

   a. Menu View Konfigurasi

      Menu ini digunakan untuk melihat konfigurasi yang sudah dijalankan dan melakukan proses hapus, mengaktifkan, dan menonaktifkan baris konfigurasi.

   b. Menu Policy setting

      Menu ini digunakan untuk melakukan setting *policy* (keadaan default program).

   c. Menu Rule Setting

      Menu ini digunakan untuk melakukan setting aturan firewalling.

4. Kelompok Menu Masking

   Kelompok menu ini terdiri dari menu setting *Network Address Translation (NAT)* untuk kepentingan yang berbeda. Menu-menu tersebut adalah sebagai berikut :

   a. Menu One to One

      Menu ini digunakan untuk setting NAT dengan metode *one to one.*

   b. Menu Masqureade

      Menu ini digunakan untuk setting NAT dengan metode *masqureade.*

   c. Menu DNAT

      Menu ini digunakan untuk setting *Destination NAT (DNAT).*

Menu-menu pada halaman ini tidak dapat diakses sebelum dilakukan login terlebih dahulu. Hal ini dilakukan unuk menjaga keamanan terhadap pemakaian program. Untuk dapat melakukan proses tersebut perlu dilakukan pengecekan nilai suatu file yang akan di set dengan nilai 1 pada saat login sudah dilakukan dan benar. Proses ini dilakukan dengan kode program sebagai berikut :

```
$hai=`cat skywall`;
```

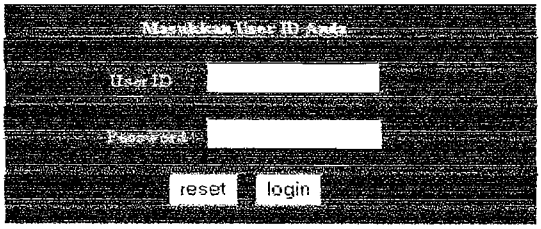
Jika variabel $hai bernilai 1 maka form akan ditampilkan jika keadaan tidak terpenuhi maka tampilan peringatan akan muncul dan menu tidak dapat diakses. Tampilan peringatan tersebut adalah sebagai berikut :

Gambar 4.2 Antartmika peringatan

Antarmuka untuk masing-masing menu diuraikan sebagai berikut :

4.3.1. Implementasi antarmuka Login

Antarmuka ini digunakan untuk melakukan login untuk dapat menggunakan program. ID pengguna program dapat dituliskan ke *text field* yang pertama dan password dapat dituliskan pada *text field* yang kedua kemudian dilanjutkan dengan menekan tombol login.



Gambar 4.3 Antarmuka Login

Jika password dan User ID cocok maka akan muncul tampilan sebagai berikut :

Gambar 4.4 Antarmuka Login sukses

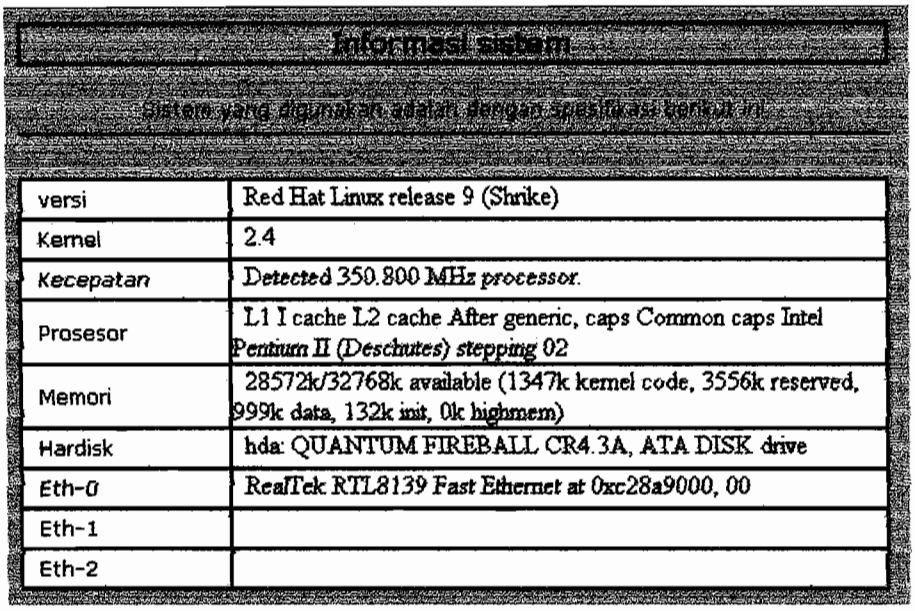
4.3.2. Implementasi antarmuka Logout

Menu logout digunakan untuk mengakhiri proses login sehingga menu program tidak bisa diakses kembali dan harus melakukan proses login terlebih dahulu. Jika proses logout berhasil maka akan muncul tampilan sebagai berikut :

Gambar 4.5 Antarmuka Logout sukses

4.1.1 Implementasi antarmuka Informasi Sistem

Menu Informasi sistem akan menampilkan antarmuka informasi sistem dari komputer router.

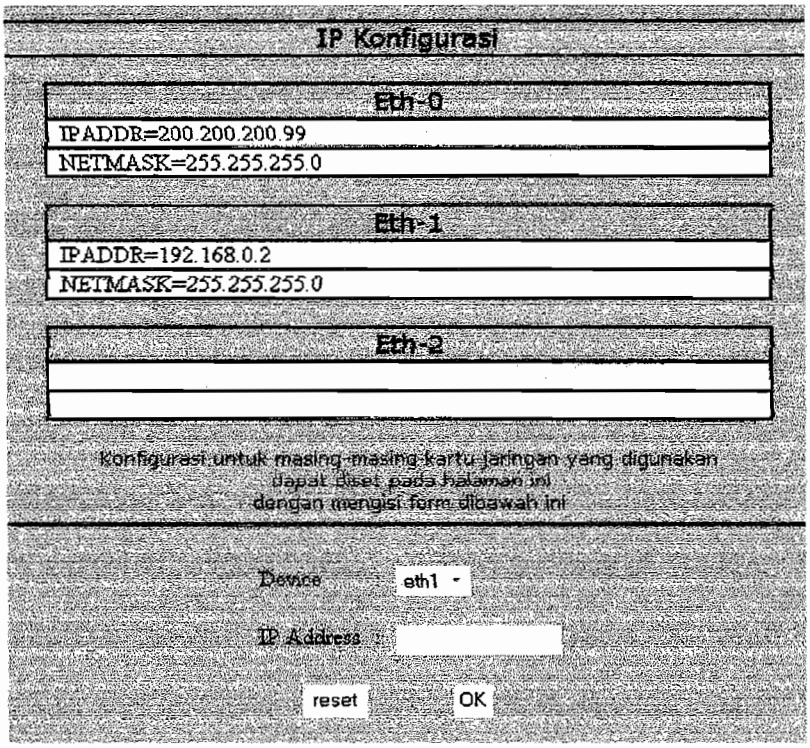


Gambar 4.6 Antarmuka Informasi Sistem

Informasi yang ditampilkan adalah berupa distro Linux yang digunakan beserta versinya, versi kernel, kecepatan prosesor, kapasitas memory yang digunakan, hardisk yang digunakan, dan kartu jaringan yang digunakan oleh komputer router.

4.3.3. Implementasi antarmuka IP Konfigurasi

Menu IP konfigurasi akan menampilkan antarmuka IP konfigurasi dimana pengguna dapat melakukan konfigurasi alamat IP pada komputer router melalui antarmuka ini.

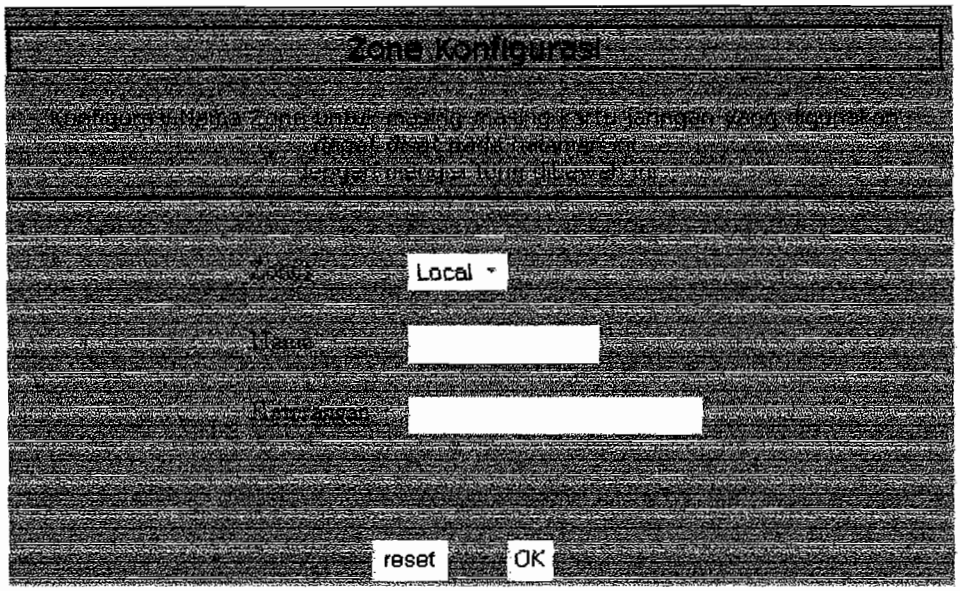


Gambar 4.7 Antarmuka IP Konfigurasi

Konfigurasi sebelumnya ditampilkan pada tabel Eth-0, Eth 1, dan Eth2. setting alamat IP-address baru dapat di inputkan dalam *text filed* IP-address dan kartu jaringan yang akan di setting dapat dipilih melalui *jump menu* Device yang disediakan.

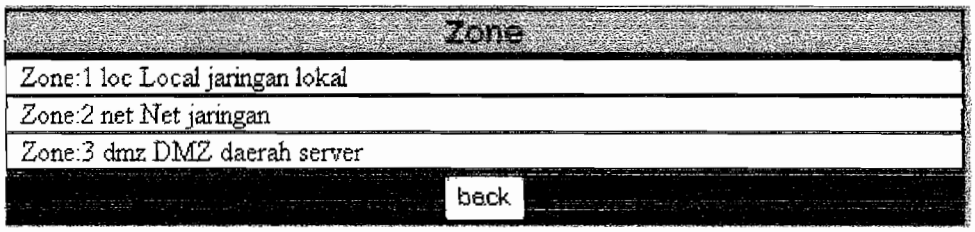
4.3.4. Implementasi antarmuka Zone Konfigurasi

Menu Zone konfigurasi akan menampilkan antarmuka setting zone yang akan dikenakan pada komputer router.



Gambar 4.8 Antarmuka Zone konfigurasi

Zone yang akan diset dapat dipilih dengan *jump menu* Zones dan dapat dinamai dengan memasukkannya pada *text field* Nama. Dan keterangan untuk zone yang dipilih dapat diinputkan pada *text field* Keterangan. Jika konfigurasi sudah dilakukan maka dapat dilanjutkan dengan menekan tombol OK untuk menyimpan konfigurasi.

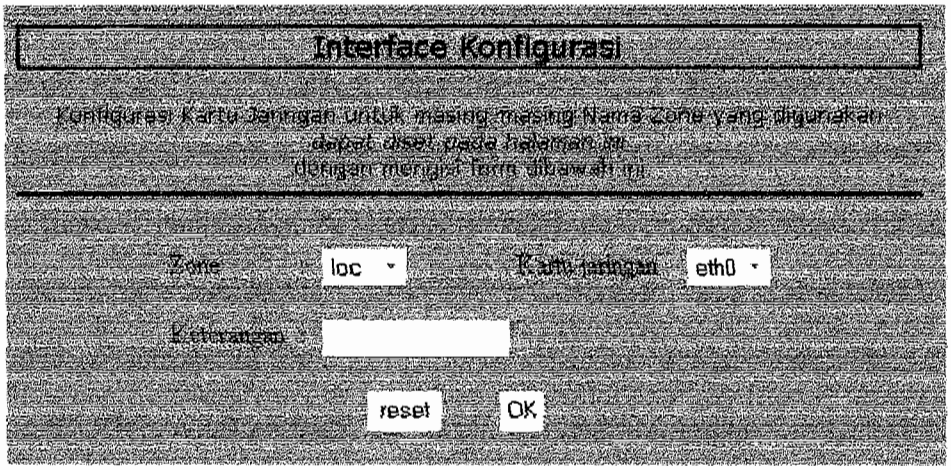


Gambar 4.9 Antarmuka hasil zone konfigurasi

Setting yang dilakukan terakhir kali akan ditampilkan pada baris paling atas dari tabel antarmuka.

4.3.5. Implementasi antarmuka Interface Konfigurasi

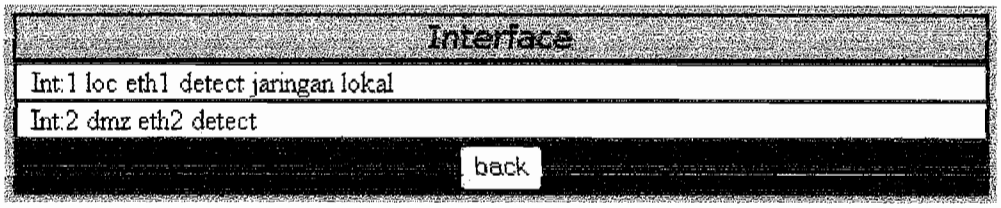
Menu interface konfigurasi akan menampilkan antarmuka setting kartu jaringan yang akan digunakan untuk implementasi setting zone.



Gambar 4.10 Antarmuka Interface konfigurasi

Pada antarmuka ini dapat dilakukan setting kartu jaringan mana yang akan dipergunakan untuk zone tertentu dengan memilih *jump menu* zone dan Kartu jaringan yang tersedia. Sedangkan *text field* Keterangan ditambahkan untuk memasukan keterangan tambahan untuk konfigurasi ini.

Jika konfigurasi sudah dilakukan dan ditekan tombol OK maka akan muncul tabel hasil konfigurasi sebagai berikut

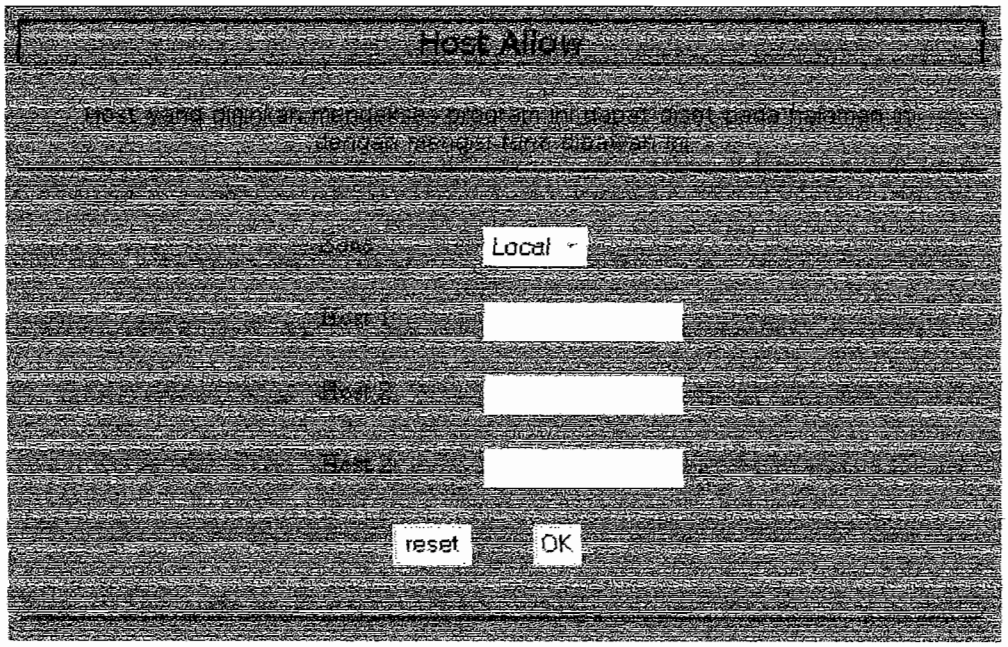


Gambar 4.11 Antarmuka hasil Interface konfigurasi

4.3.6. Implementasi antarmuka Host Allow

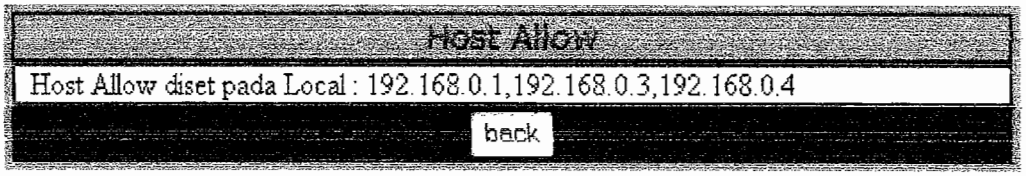
Menu Host Allow akan menampilkan antarmuka yang dapat digunakan untuk melakukan setting *host* mana saja yang diijinkanuntuk mengakses program ini. Alamat IP *host* yang diijinkan dapat dimasukkan pada *text field* yang disediakan. Dan menekan tombol OK untuk menyimpan

konfigurasi. *Host* yang diijinkan adalah maksimal 3. Berikut ini antarmuka menu Host allow :



Gambar 4.12 Antarmuka Host Allow konfigurasi

Jika setting untuk *host allow* sudah dilakukan dan tombol OK ditekan maka setting akan disimpan dan akan muncul tampilan dibawah ini sesuai dengan input yang dilakukan.

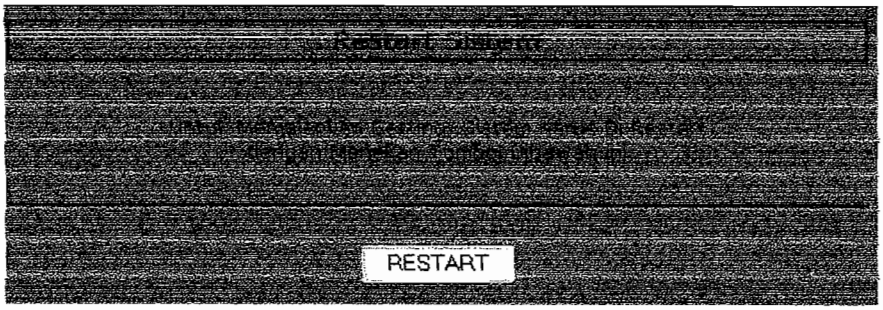


Gambar 4.13 Antarmuka hasil Host allow konfigurasi

Setting akan dijalankan setelah dilakukan restart terhadap program.

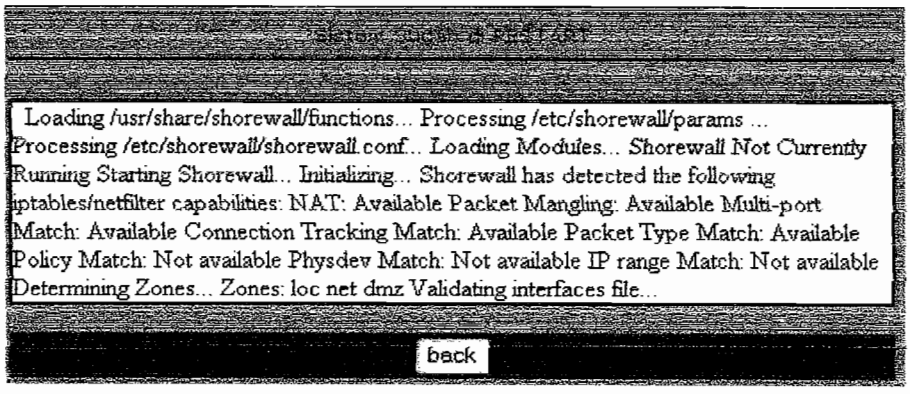
4.3.7. Implementasi antarmuka Restart Sistem

Menu Restart sistem digunakan untuk melakukan restart program dan menjalankan konfigurasi yang sudah diset pada program. Antarmuka untuk melakukan Restart program adalah sebagai berikut.



Gambar 4.14 Antarmuka Restart Sistem

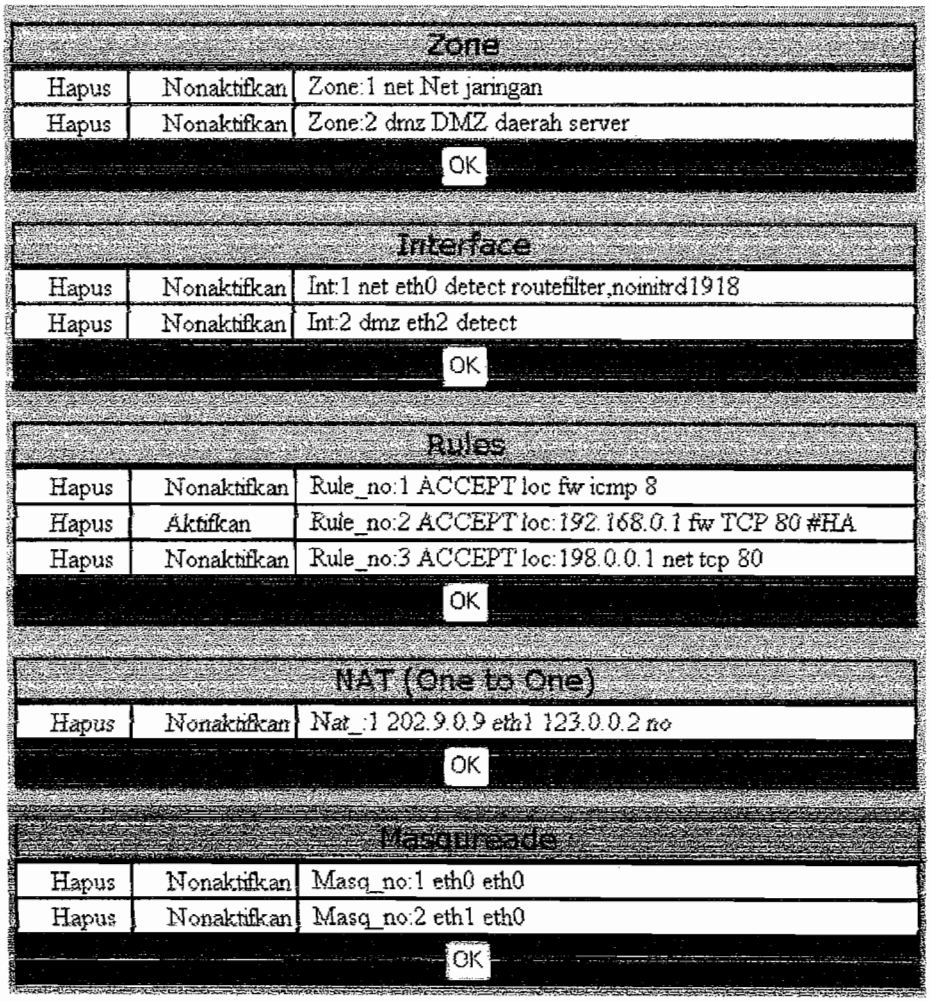
Dengan menekan tombol Restart maka program akan direstart dan konfigurasi yang telah diset akan diljalankan. Jika ada kesalahan atau kekurangan dalam konfigurasi maka akan ditampilkan kesalahan yang terjadi seperti tampak pada antarmuka berikut



Gambar 4.15 Antarmuka hasil Restart Sistem

4.3.8. Implementasi antarmuka View Konfigurasi

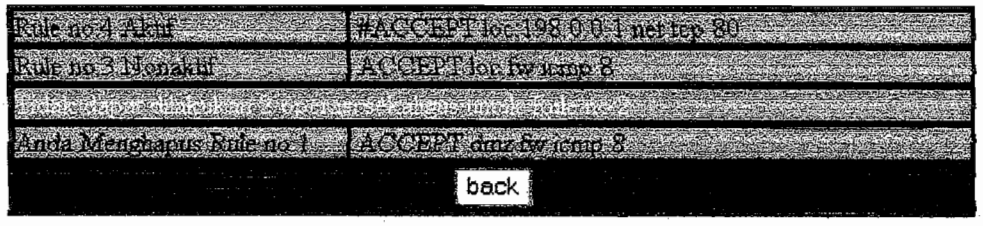
Menu view konfigurasi akan menampilkan seluruh setting konfigurasi yang sudah dilakukan pada semua file konfigurasi yang digunakan oleh Shorewall untuk melakukan proses *firewalling*. Pada antarmuka ini juga disertai fungsi penghapusan baris konfigurasi, pengaktifan baris konfigurasi yang tidak aktif atau sebaliknya yaitu pe-nonaktifan baris konfigurasi yang masih aktif. Antarmuka View konfigurasi tersebut adalah sebagai berikut :



Gambar 4.16 Antarmuka View Konfigurasi

Fasilitas *Checkbox* ada pada tiap-tiap baris konfigurasi pada semua file konfigurasi. Proses penghapusan baris konfigurasi akan dilakukan jika c*heckbox* hapus ditandai. *Checkbox* aktifkan akan otomatis muncul disamping baris konfigurasi yang tidak aktif. Dan sebaliknya *Checkbox* nonaktifkan akan otomatis muncul disamping baris konfigurasi yang aktif.

Administrator dapat melakukan penghapusan, pengaktifan, atau pe-nonaktifkan baris konfigurasi dengan fasilitas *checkbox* yang disediakan, tetapi tidak diijinkan untuk melakukn 2 operasi sekaligus untuk 1 baris konfigurasi. Tabel konfigurasi yang dilakukan terhadap baris konfigurasi untuk 1 file konfigurasi akan ditampilkan setelah tombol OK ditekan. Contoh antarmuka yang ditampilkan setelah dilakukan operasi terhadap baris konfigurasi pada file konfigurasi rules adalah sebagai berikut :



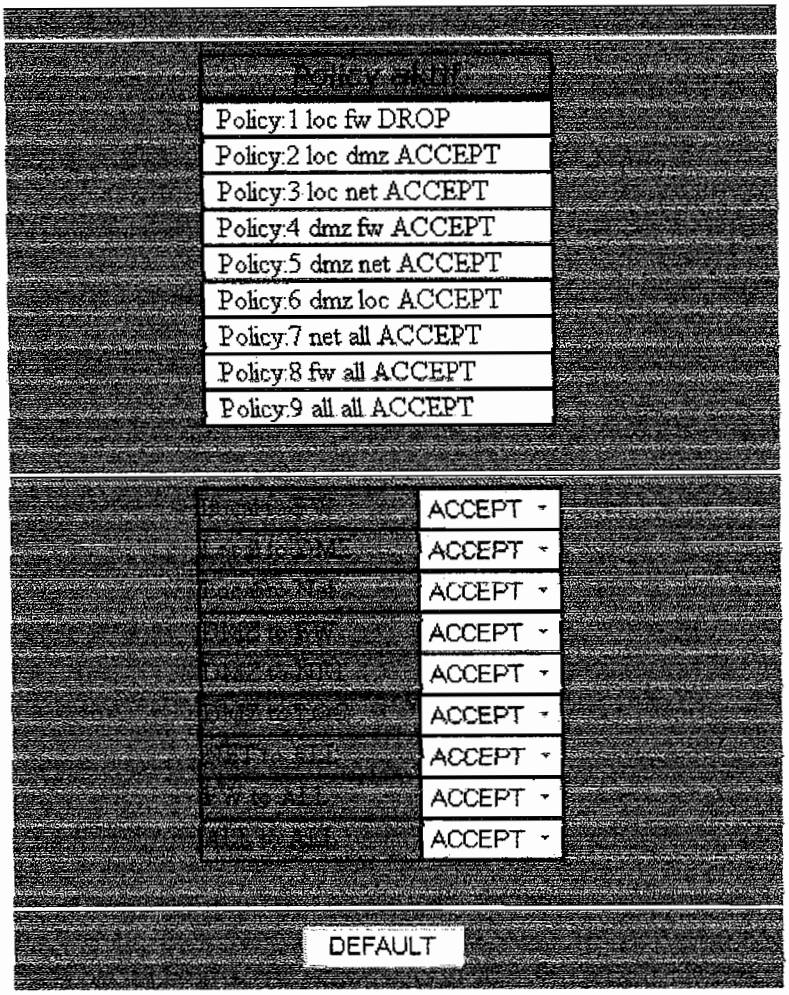
Gambar 4.17 Antarmuka Hail operasi terhapad baris konfigurasi

Tombol OK juga sekaligus menyimpan hasil operasi terhadap baris konfigurasi ke dalam file konfigurasi yang baru yang akan dijalankan setelah program di restart.

4.3.9. Implementasi antarmuka Policy Setting

Menu Policy setting akan menampilkan antarmuka yang digunakan untuk melakukan setting policy (keadaan default) pada program. Keadaan

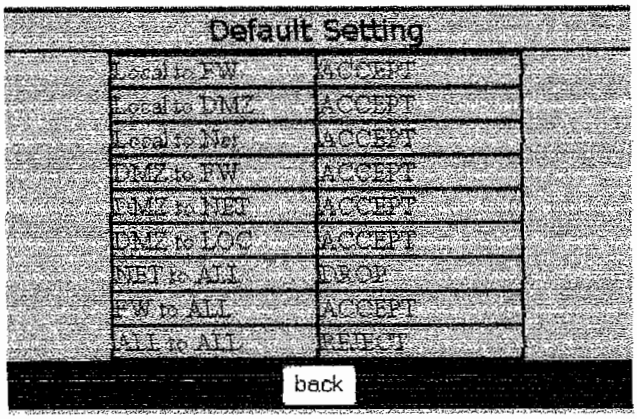
default sebelumnya ditampilkan juga dengan sebuah tabel. Berikut ini antarmuka yang digunakan untuk melakukan setting *policy*.



Gambar 4.18 Antarmuka Policy konfigurasi

Kebijakan untuk 9 keadaan dari aliran data yang menggunakan 3 kartu jaringan dan 3 zona dapat diset dengan memilih *jump menu* disamping masing – masing keadaan. Dan kebijakan *default* sebelumnya dapat dilihat pada tabel policy aktif diatasnya.

Jika tombol 'default' ditekan maka konfigurasi akan disimpan dan akan ditampilkan hasil dari konfigurasi yang dilakukan dengan antarmuka berikut.



Gambar 4.19 Antarmuka hasil Policy konfigurasi

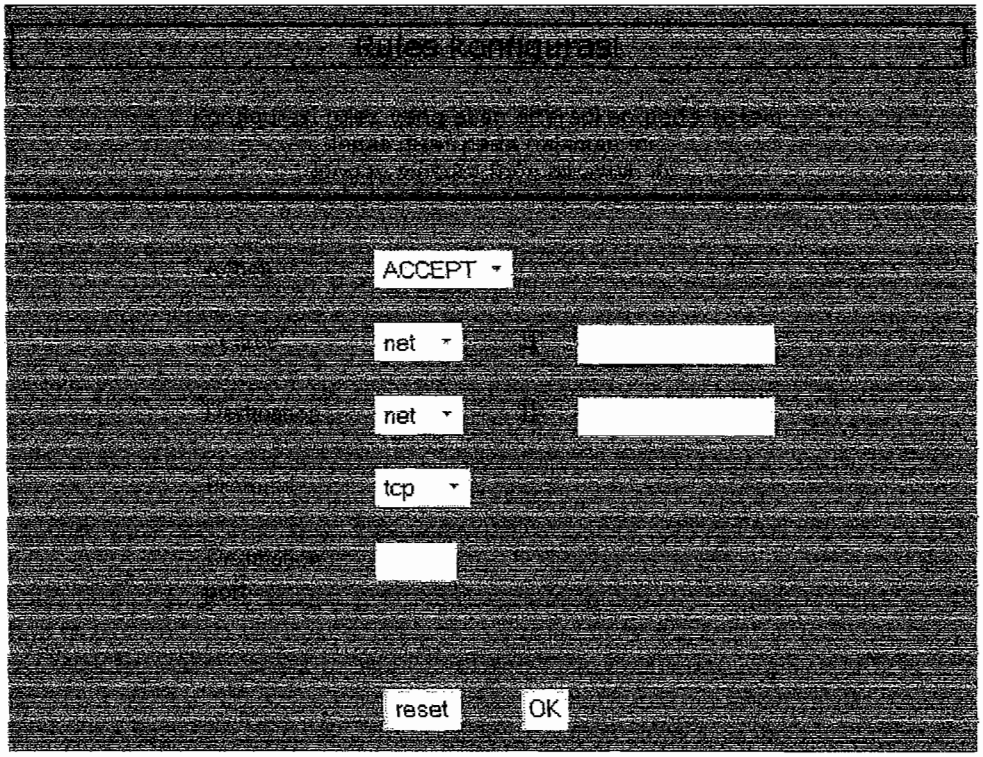
Konfigurasi *Policy* yang baru akan dijalankan setelah dilakukan restart terhadap program.

4.3.10. Implementasi antarmuka Rule Setting

Menu Rule setting akan menampilkan antarmuka yang berfungsi untuk melakukan setting terhadap konfigurasi *rules* yang merupakan inti dari proses *firewalling.*

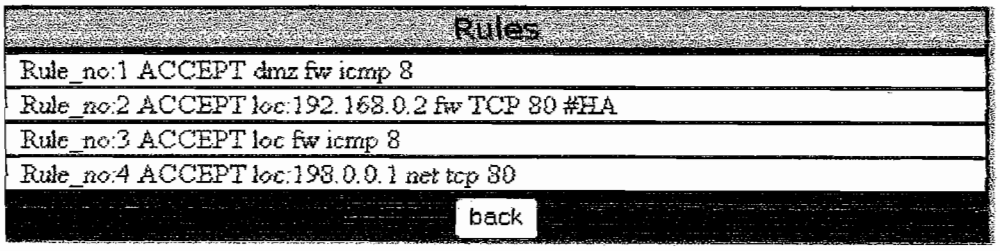
Security policy yang akan dikenakan terhadap suatu paket aliran data yang masuk atau melewati router dapat diset pada antarmuka ini dengan memilih *jump menu* Action. Asal paket data dapat ditentukan dengan memilih *jump menu* Source dan alamat IP asalnya dapat diisikan pada *text field* IP. Begitu juga dengan tujuannya, dapat digunakan *jump menu* Destination dan *text field* IP. Protokol yang digunakan paket data dapat ditentukan dengan memilih pada *jump menu* Protocol. Port tujuannya juga ditentukan dengan mengisi *text field* Destination Port.

Antarmuka menu Rule Setting adalah sebagai berikut :



Gambar 4.20 Antarmuka Rules konfigurasi

Jika konfigurasi sudah diset dan disimpan dengan tombol OK, maka akan ditampilkan tabel sesuai dengan inputan konfigurasi yang dilakukan. Setting konfigurasi yang dilakukan terakhir kali ditampilkan pada baris paling atas.

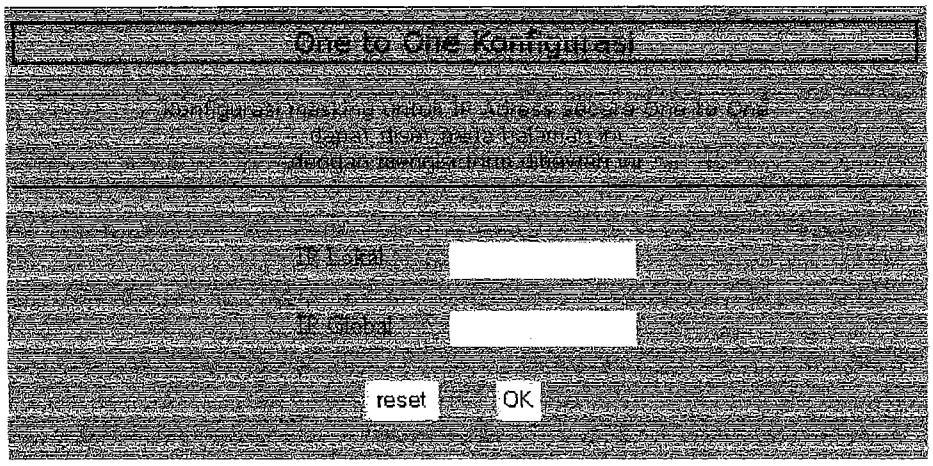


Gambar 4.21 Antarmuka hasil Rules konfigurasi

Konfigurasi *rules* diatas akan dijalankan setelah dilakukan restart terhadap program.

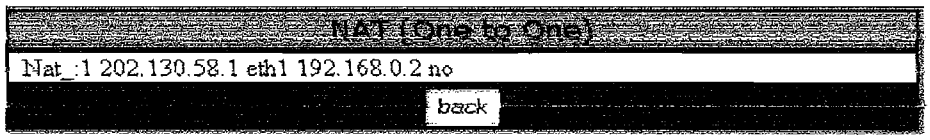
4.3.11. Implementasi antarmuka One to One

Menu One to One akan menampilkan antartmuka setting NAT secara *one to one*. Setting konfigurasi untuk NAT secara one to one dapat dilakukan dengan mengisi *text field* IP Lokal dengan alamat IP lokal yang akan dikenai NAT. sedangkan alamat IP global yang akan digunakan untuk NAT dapat diinputkan pada *text filed* IP Global. Antarmuka menu One to One adalah :



Gambar 4.21 Antarmuka One toOne konfigurasi

Jika tombol Ok ditekan maka konfigurasi akan disimpan dan akan muncul tampilan dibawah ini sesuai dengan hasil input yang dilakukan.



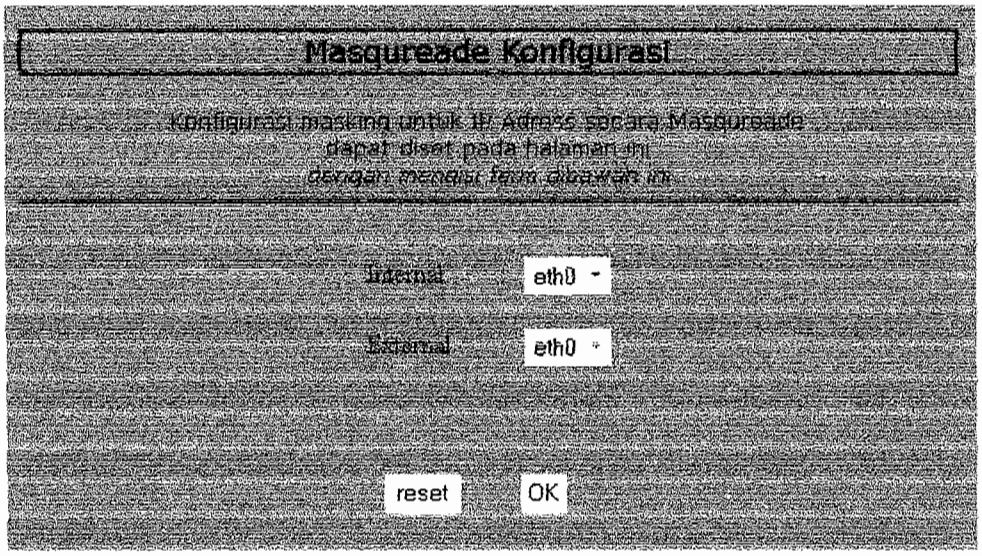
Gambar 4.22 Antarmuka hasil One to One konfigurasi

Konfigurasi diatas akan dijalankan setelah dilakukan restart terhadap program.

4.3.12. Implementasi antarmuka Masqureade

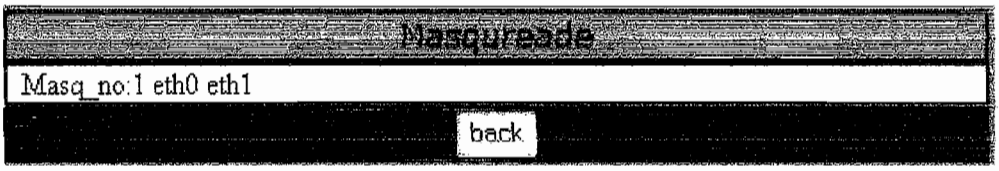
Menu Masqureade akan menampilkan antarmuka setting masqureade pada program. Setting dapat dilakukan dengan memilih kartu jaringan yang

akan dikenai *masqureade* (kartu jaringan yang terhubung dengan jaringan lokal atau DMZ) dengan *jump menu* Internal. Dan kartu jaringan yang akan digunakan untuk masqureade dapat dipilih dengan *jump menu* external



Gambar 4.23 Antarmuka Masqureade konfigurasi

Jika tombol OK ditekan setting akan disimpan dan akan muncul tampilan dibawah ini.



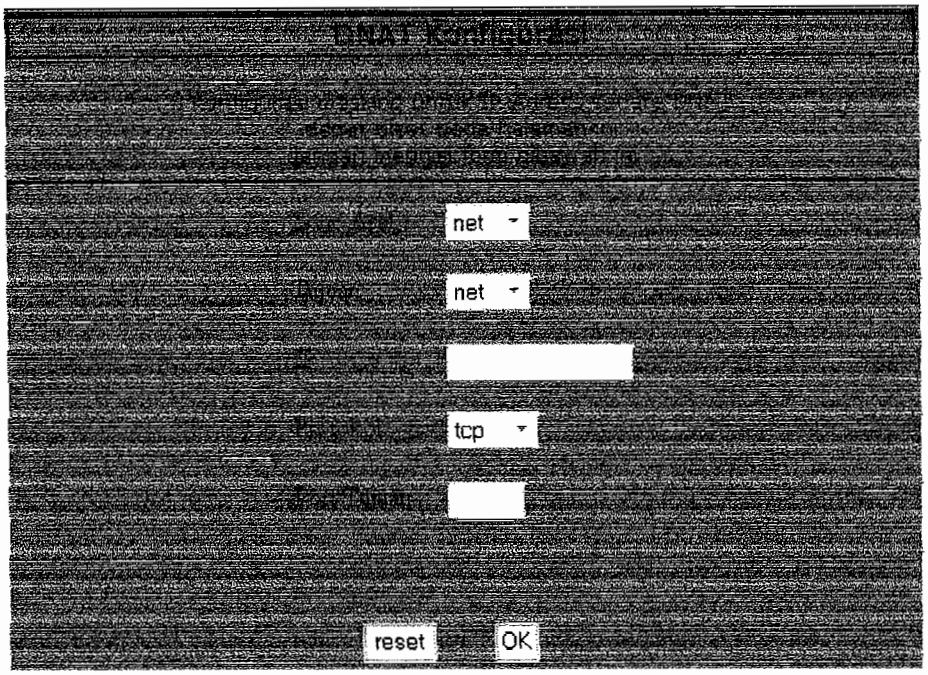
Gambar 4.24 Antarmuka hasil Masqureade konfigurasi

Konfigurasi akan dijalankan setelah dilakukan restart terhadap program.

4.3.13. Implementasi antarmuka DNAT

Menu DNAT adalah menu yang dipergunakan untuk melakukan setting *Destination NAT* yaitu proses diijinkannya host dari luar jaringan mengakses komputer didalam jaringan lokal atau DMZ *(demiliterized zone)*

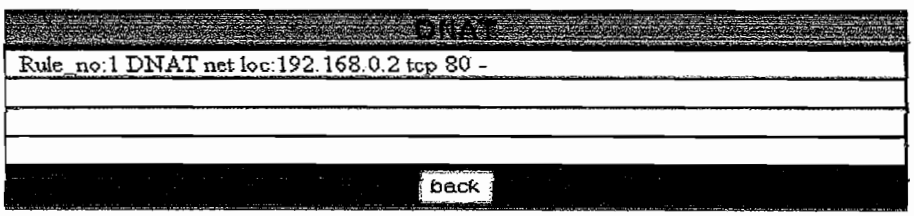
menggunakan alamat IP komputer router. Proses ini dapat di set pada antarmuka menu berikut ini.



Gambar 4.24 Antarmuka DNAT konfigurasi

Asal permintaan akses dapat dipilih dari zone yang sudah ditentukan dengan memilih *jump menu* zone asal. Zone tujuan permintaan akses dapat dipilih pada *jump menu* Tujuan. IP tujuan yang sesuai dengan permintaan akses dapat ditentukan dengan mengisi *text field* IP. Protokol permintaan akses dapat ditentukan pada *jump menu* Protokol. Port tujuan permintaan akses juga dapat ditentukan dengan mengisi *text field* Port Tujuan.

Jika konfigurasi sudah dilakukan dan tombol OK ditekan maka akan muncul tampilan sbb:



Gambar 4.25 Antarmuka hasil DNAT konfigurasi

Konfigurasi akan dijalankan setelah dilakukan restart pada program.

4.4. Implementasi Program

4.4.1. Implementasi Login

Fungsi login digunakan untuk melakukan authentifikasi pengguna program. Hanya orang yang diijinkan saja yang bisa mengakses program ini. Kode program fungsi login adalah sebagai berikut :

```
#!/usr/bin/perl
use CGI;
$q=new CGI;
$id="admin";
$pass="admin";
print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#000066 link=red vlink=#0000CC
alink=#0000CC>";

if(($q->param('pass') eq $pass)&& ($q->param('id') eq $id))
{#--------------------------------
        system("echo '1'>/usr/local/apache2/cgi-
        bin/skywall/html/skywall");
        system("echo '1'>skywall");
       $hai=`cat skywall`;
       #--------------------------------
print "<table width=350 border=0 align=center cellpadding=0
               cellspacing=0 class=login>";
print "<tr>";
print "<td height=150 align=center valign=top></td>";
print "</tr>";
print "</table>";
print "<table width=350 border=0 align=center cellpadding=0
               cellspacing=0 >";
print "<tr>";
print "<td height=100 align=center valign=top class=text>";
   print "<strong><font color=#FF0000 size=4>";
   print "Login Sukses\n</font></strong>";
   print "<font color=#666666 size=4
   align=left>Anda Terkoneksi dengan :</font>";
   print "<font color=#666666 size=4 align=left>Server
         $server\n</font>";
 print "<font color=#666666 size=4 align=left>Dari $IP\n</font>";
 print "<font color=#666666 size=4>Web Browser anda
        $browser\n</font>";
 print "</td>";
 print "</table>";
 }
else
 {
print "<table width=350 border=0 align=center cellpadding=0
cellspacing=0 class=login>";
print "<tr>";
```

```
print "<td height=150 align=center valign=top></td>";
print "</tr>";
        print "</table>";
        print "<table width=350 border=0 align=center
cellpadding=0 cellspacing=0 >";
        print "<tr>";
        print "<td height=100 align=center valign=top
class=text>";
        print "<strong>";
        print "<font color=#FF0000 size=4>";
               print "Login GAGAL!";
               print "<br>";
               print "<br>";
               print "<br>";
               print "User ID Salah";
               print "<br>";
               print "Anda harus melakukan Login Ulang";
       #------------------
       print "</td>";
        print "</table>";
}
```

Kode program diatas akan melakukan pencocokan input dari form login (`$q->param('pass')`) dengan variabel yang sudah didefinisikan yaitu variabel "id" dan "pass". Jika proses login tidak berhasil maka semua menu tidak dapat diakses dan menampilkan peringatan seperti dibawah ini

Peringatan untuk login terlebih dahulu diatas ditampilkan dengan kode program sbb :

```
print "<table width=100% border=0 bordercolor=white>";
print "<table width=75% border=0 align=center valign=center
bordercolor=white>";
    print "<tr>";
    print "<td height=25 align=center valign=center>";
          print "<strong><font face=Verdana size=3 color=red>";
          print "PERINGATAN</font></strong>";
    print "</td>";
    print "</tr>";
    print "<tr>";
    print "<td height=25 align=center valign=center>";
           print "<strong><font face=Verdana size=3 color=red>";
           print "Anda harus melakukan login terlebih dahulu
!!!<br>";
```

```
            print "Anda tidak berhak mengakses sistem
ini</font></strong>";
            print "<hr color=#000066>";
    print "</td>";
    print "</tr>";
    print "<form name=log action=/skywall/tengah.htm>";
    print "<tr>";
    print "<td height=25 align=center valign=center>";
                print "<input type=submit name=login
value=LOGIN>";
            print "</td>";
    print "</tr>";
    print "</form>";
print "</table>";
```

Jika login benar maka file yang bernama 'skywall' akan terbentuk pada directory /usr/local/apache2/cgi-bin/skywall/html/. File tersebut akan diberi nilai 1 untuk fungsi pengecekan akses bagi fungsi lainnya. Proses tersebut dilakukan dengan kode program sbb:

```
#--------------------------------
  system("echo '1'>/usr/local/apache2/cgi-bin/skywall/
          html/skywall");
  system("echo '1'>skywall");
#--------------------------------
```

4.4.2. Implementasi Logout

Proses Logout berfungsi untuk mengamankan program agar ketika program ditutup dan kemudian dibuka kembali diharuskan untuk melakukan login terlebih dahulu. Proses logout dihasilkan dari kode program sbb:

```
system("rm skywall");
system("rm /usr/local/apache2/cgi-bin/skywall/html/skywall");
```

Kode program diatas akan melakukan penghapusan file yang bernama 'skywall' yang berada pada direktory /usr/local/apache2/cgi-bin/skywall/html/skywall.

4.4.3. Implementasi Informasi Sistem

Menu informasi sistem pada halaman utama akan menjalankan modul interface.cgi yang ada pada direktori `/usr/local/apache2/cgi-bin/skywall/` Pada linux. Modul ini berfungsi untuk menampilkan informasi konfigurasi dari komputer router yang digunakan oleh sistem. Proses pengambilan informasi tersebut dilakukan oleh kode program sebagai berikut :

```
$hai=`cat skywall`;

if($hai==1){
system("dmesg > sistem.txt");

$versi=`cat /etc/redhat-release`;
$kernel=`kernelversion`;
$speed=`cat sistem.txt|grep processor`;
$prosesor=`cat sistem.txt|grep CPU:|cut -d: -f2`;
$memori=`cat sistem.txt|grep Memory|cut -d: -f2`;
$hardisk=`cat sistem.txt|grep hda|grep drive`;
$eth0=`cat sistem.txt|grep eth0|grep IRQ|cut -d: -f2`;
$eth1=`cat sistem.txt|grep eth1|grep IRQ|cut -d: -f2`;
$eth2=`cat sistem.txt|grep eth2|grep IRQ|cut -d: -f2`;
```

Perintah `system("dmesg > sistem.txt");` akan memberikan informasi mengenai komputer router dan akan disimpan pada file sistem.txt. Selanjutnya akan diambil dengan perintah `cat` sesuai dengan kebutuhan yang kemudian divariabelkan untuk ditampilkan.

Menampilkan masing-masing variabel pada tabel dengan kode program sebagai berikut untuk masing-masing variabel:

```
print "<tr>";
  print "<td width=166 height=25 bgcolor=#6699CC><font face=Verdana
         size=-1>  versi</font></td>\n";
  print "<td width=599 valign=top bgcolor=#6699CC>  $versi";
  print "</td>";
print "</tr>";
```

4.4.4. Implementasi IP Konfigurasi

Proses pada menu IP-Konfigurasi ini berfungsi untuk mengubah alamat IP- Pada komputer router. Setelah admin memasukkan ip yang baru maka program akan menjalankan modul ipkonf.cgi pada direktori /usr/local/apache2/cgi-bin/skywall pada linux. Proses perubahan ip dilakukan oleh kode program sebagai berikut pada modul ipkonf.cgi

```
system("chmod 777 /etc/sysconfig/network-scripts/ifcfg-$kartu");
system("sed '/IPADDR/d' /etc/sysconfig/network-scripts/ifcfg-$kartu
        >/tmp/del");
system("cp -f /tmp/del /etc/sysconfig/network-scripts/ifcfg-
        $kartu");
system("echo IPADDR='$ip' >/tmp/ip");
system("sed '/BROADCAST/r /tmp/ip' /etc/sysconfig/network-
        scripts/ifcfg-$kartu >/tmp/ip-ok");
system("cp -f /tmp/ip-ok /etc/sysconfig/network-scripts/ifcfg-
       $kartu");
system("rm /tmp/del");
system("rm /tmp/ip");
system("rm /tmp/ip-ok");
system("chmod 0644 /etc/sysconfig/network-scripts/ifcfg-$kartu");
system("service network restart >/tmp/restart");
$restart=`cat /tmp/restart`;
```

Awalnya file 'ifcfg-$kartu', ($kartu merupakan hasil inputan dari form html pada menu konfigurasi) pada direktori /etc/sysconfig/network-scripts pada linux yang merupakan file konfigurasi ip-address, dikenai perintah chmod 777 yang artinya memperbolehkan file tersebut di edit oleh siapa saja. Kemudian baris yang mengandung kata 'IPADDR' dihapus dan hasilnya dimasukkan ke file /tmp/del. Kemudian file pada yang sudah tidak mengandung kata 'IPADDR' dicopy kembali ke /etc/sysconfig/network-scripts kemudian dituliskan ip-address baru hasil dari pembacaan form inputan menu ip-konfigurasi pada file /tmp/ip. Selanjutnya adalah menyisipkan isi dari

file /tmp/ip dibawah kata 'BROADCAST' pada file `'ifcfg-$kartu'` di direktori `/etc/sysconfig/network-scripts`.

Setelah semua file pada direktori `/tmp` dihapus langkah terakhir adalah melakukan restart terhadap service network pada linux untuk mengaktifkan konfigurasi dengan ip-address baru dengan perintah. `system("service network restart >/tmp/restart")`.

4.4.5. Implementasi Zone Konfigurasi

Proses zone konfigurasi adalah proses pengesetan nama zone (daerah) untuk masing-masing kartu jaringan yang digunakan. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul `zone.cgi` pada direktori `/usr/local/apache2/cgi-bin/skywall/` pada linux. Kode program untuk melakukan pengesetan zone adalah sebagai berikut :

```
$zon=$q->param('zone');
$name=$q->param('nama');
$com=$q->param('comment');
$kartu=$q->param('kartu');
system("echo '$name' '$zon' '$com'>/tmp/zones");
system("sed '/#dibawahini/r /tmp/zones' /etc/shorewall/zones
>/tmp/hasil1");
system("cp -f /tmp/hasil1 /etc/shorewall/zones");
system("rm /tmp/hasil1");
system("rm /tmp/int");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu zone konfigurasi. Variabel $zon untuk inputan zona, $nama untuk nama yang akan diberikan untuk zona tersebut. Dan variabel $com untuk keterangan tentang zona tersebut. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah:

```
System("echo '$name' '$zon''$com'>/tmp/zones");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori /etc/shorewall/zones dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori /tmp/hasil1 dengan perintah

```
System("sed'/#dibawahini/r/tmp/zones'/etc/shorewall/zones>/tmp/hasi
l1");.
```

Langkah terakhir adalah mengcopy file direktori /tmp/hasil1 ke file direktori setting zone shorewall yaitu pada /etc/shorewall/zones dengan perintah.

```
system("cp -f /tmp/hasil1 /etc/shorewall/zones");
```

setelah program di restart file direktori pada /etc/shorewall/zones akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.6. Implementasi Interface Konfigurasi

Proses interface konfigurasi adalah proses pengesetan / pendefinisian zona (daerah) untuk masing-masing kartu jaringan yang digunakan. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul int.cgi pada direktori /usr/local/apache2/cgi-bin/skywall/ pada linux. Kode program untuk melakukan pengesetan interface adalah sebagai berikut :

```
$zone=$q->param('zone');
$ket=$q->param('ket');
$kartu=$q->param('kartu');
system("echo '$zone' '$kartu' 'detect' '$ket'>/tmp/int");
system("sed '/#dibawahini/r /tmp/int' /etc/shorewall/interfaces
>/tmp/hasil2");
system("cp -f /tmp/hasil2 /etc/shorewall/interfaces");
system("rm /tmp/hasil2");
system("rm /tmp/int");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu zone konfigurasi. Variabel $zone untuk inputan zona,

$ket untuk keterangan yang akan diberikan untuk zona tersebut. Dan variabel $kartu untuk pilihan kartu jaringan yang akan digunakan. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
System("echo '$zone' '$kartu' 'detect' '$ket'>/tmp/int");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori `/etc/shorewall/interfaces` dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori `/tmp/hasil2` dengan perintah

```
System("sed '/#dibawahini/r /tmp/int' /etc/shorewall/interfaces
>/tmp/hasil2");
```

Langkah terakhir adalah mengcopy file direktori `/tmp/hasil2` ke file direktori setting interfaces shorewall yaitu pada `/etc/ shorewall /interfaces` dengan perintah.

```
System("cp -f /tmp/hasil2 /etc/shorewall/interfaces");
```

setelah program di restart file direktori pada `/etc/shorewall /interfaces` akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.7. Implementasi Host Allow

Proses pada menu *Host Allow* adalah proses mengijinkan sebuah *host* untuk dapat mengakses program. Pada program ini maksimal 3 *host* yang diijinkan mengakses program. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul hostallow.cgi pada direktori `/usr/local/apache2 /cgi-bin/skywall/` pada linux. Kode program untuk melakukan pengesetan *Host Allow* adalah sebagai berikut :

```
$ip1=$q->param('ip1');
$ip2=$q->param('ip2');
$ip3=$q->param('ip3');
system("sed '/#HA/d' /etc/shorewall/rules >/tmp/hostbaru");
system("echo 'ACCEPT' 'loc:''$ip1' '$ip2' '$ip3' 'fw' 'TCP' '80'
'#HA'>/tmp/host");
system("sed '/#dibawahini/r /tmp/host' /tmp/hostbaru >/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/host");
```

Langkah kedua adalah memberikan variabel untuk masing-masing inputan untuk host yang diijinkan dari form menu *Host allow*. Variabel $ip1,$ip2 dan $ip3 adalah variabel untuk inputan masing-masing host yang diijinkan. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
system("echo'ACCEPT''loc:''$ip1''$ip2''$ip3''fw''TCP''80'
'#HA'>/tmp/host");
```

Namun sebelum itu dilakukan penghapusan setting untuk *Host allow* sebelumnya dengan perintah sbb:

```
system("sed '/#HA/d' /etc/shorewall/rules >/tmp/hostbaru");
```

Perintah diatas akn melakukan penghapusan baris yang mengandung karakter "#HA". Selanjutnya isi dari hasil penulisan yang ada pada *temporary* disisipkan kedalam file direktori `/etc/shorewall/rules` dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori `/tmp/hasil` dengan perintah

```
system("sed '/#dibawahini/r /tmp/host' /tmp/hostbaru >/tmp/hasil");
```

Langkah terakhir adalah mengcopy file direktori `/tmp/hasil` ke file direktori setting *rules* untuk *host allow* shorewall yaitu pada `/etc/ shorewall /rules` dengan perintah.

```
System("cp -f /tmp/hasil /etc/shorewall/rules");
```

setelah program di restart, isi file direktori `/etc/shorewall/rules` akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.8. Implementasi Restart Sistem

Restart sistem dilakukan dengan melakukan restart pada service Shorewall. Jika ada perubahan yang terjadi pada file konfigurasi Shorewall maka isi dari file yang terbarulah yang akan dieksekusi oleh Shorewall. Perintah untuk melakukan restart pada Shorewall pada program ini adalah sebagai berikut:

```
system("sudo-u root shorewall restart >rest.txt");
```

Perintah kode program diatas akan melakukan proses restart Shorewall sebagai *root*. Hasil tampilan restart akan disimpan pada file rest.txt untuk ditampilkan.

4.4.9. Implementasi View Konfigurasi (Hapus, Aktifkan dan Nonaktifkan)

Pada menu view konfigurasi terdapat fasilitas hapus, aktifkan dan nonaktifkan konfigurasi. Berikut ini kode program untuk fungsi hapus, aktifkan dan nonaktifkan konfigurasi.

d. Hapus kofigurasi (*rules* konfigurasi)

Proses ini akan mengeksekusi file "`rule`" yang ada pada direktori `/usr/local/apache2/cgi-bin/skywall/hapus`. Berikut ini kode program yang dieksekusi untuk melakukan fungsi hapus.

```
#!/bin/bash
let no=`cat /usr/local/apache2/cgi-bin/skywall/hapus/norule`+327
echo $no > /tmp/brshapus
let del=`cat /tmp/brshapus`
```

```
awk NR~/$del/ /etc/shorewall/rules >/tmp/brsawk2
awk '{print $1"."$2"."$3"."$4"."$5}' /tmp/brsawk2 >/tmp/brsawk3
del2=`cat /tmp/brsawk3`
sed '/'"$del2"'/d' /etc/shorewall/rules >/tmp/rulebaru
cp -f /tmp/rulebaru /etc/shorewall/rules
rm /tmp/brshapus
#rm /tmp/brsawk2
rm /tmp/brsawk3
rm /tmp/rulebaru
```

Langkah pertama adalah membuat variabel yang isinya dihasilkan dari hasil penjumlahan antara isi file direktori `/usr/local/apache2/cgi-bin/skywall/hapus/norule` yang merupakan angka baris konfigurasi yang akan dihapus ditambah dengan 327 yang merupakan jumlah baris sampai dengan penanda kata "#dibawahini". Hasilnya dimasukkan ke dalam file direktori /tmp/brshapus. File direktori tersebut diambil isinya dan divariabelkan dengan nama variabel "del". Proses ini dilakukan kode program :

```
let no=`cat /usr/local/apache2/cgi-bin/skywall/hapus/norule`+327
echo $no > /tmp/brshapus
let del=`cat /tmp/brshapus`
```

Selanjutnya baris yang akan dihapus tersebut diambil dengan fungsi awk untuk ditulis ulang dan ditampilkan. Hasil tulis ulang disimpan pada file direktori /tmp/brsawk3 dan untuk ditampilkan ada pada file direktori /tmp/brsawk2. kode programnya :

```
awk NR~/$del/ /etc/shorewall/rules >/tmp/brsawk2
awk '{print $1"."$2"."$3"."$4"."$5}' /tmp/brsawk2 >/tmp/brsawk3
```

Selanjutnya isi dari /tmp/brsawk3 yang merupakan hasil tulis ulang baris yang akan dihapus divariabelkan dengan nama "del2". Selanjutnya baris tersebut dihapus dari file direktori /etc/shorewall/rules dengan perintah:

```
sed '/'"$del2"'/d' /etc/shorewall/rules >/tmp/rulebaru
```

Hasil dari proses ini disimpan pada file direktori /tmp/rulebaru yang selanjutnya akan dicopy kembali ke file direktori /etc /shorewall/rules dengan perintah :

```
cp -f /tmp/rulebaru /etc/shorewall/rules
```

Konfigurasi rules yang baru akan dijalankan setelah program di restart.

Proses ini juga terjadi untuk file konfigurasi yang lain (Zone kofigurasi, interface konfigurasi, masqureade konfigurasi, dan one-to-one konfigurasi) hanya saja file yang dieksekusi pada direktori /usr/local/apache2/cgi-bin/skywall/hapus berbeda sesuai dengan nama konfigurasi masing-masing. Berikut ini daftar file yang dieksekusi pada direktori /usr/local/apache2/cgi-bin /skywall/hapus untuk melakukan proses hapus berdasarkan konfigurasi tertentu :

- Zone konfigurasi : file zone
- Interface konfigurasi : file int
- Masqureade konfigurasi : file masq
- One to one konfigurasi : file nat

e. Aktifkan konfigurasi (*rules* konfigurasi)

Proses ini adalah proses mengaktifkan sebuah konfigurasi dari kondisi tidak aktif. Proses ini akan mengeksekusi file "aktif" pada direktori /usr/local/apache2/cgi-bin/skywall/hapus. Langkah pertama adalah membaca baris yang akan diaktifkan dengan kode program dibawah ini dan disimpan pada file direktori /tmp/brsact2.

```
let no=`cat /usr/local/apache2/cgi-bin/skywall/hapus/noact`+327
echo $no > /tmp/brsact
let act=`cat /tmp/brsact`
awk NR~/$act/ /etc/shorewall/rules >/tmp/brsact2
```

Selanjutnya adalah menghilangkan karakter '#' yang berarti *remark* pada file konfigurasi Shorewall dengan kode program dibawahini :

```
sed -n 's/#/''/p' /tmp/brsact2 >/tmp/actbaru
```

Untuk menyisipkan kembali baris baru yang sudah disimpan pada file direktori `/tmp/actbaru` ke dalam file konfigurasi asalnya perlu diketahu baris sebelum baris yang diubah dan disimpan dalam variabel "atasnya". Caranya adalah dengan kode program berikut

```
let atasnya=`cat /tmp/brsact`-1
awk NR~/$atasnya/ /etc/shorewall/rules >/tmp/atasnya
atasnya=`cat /tmp/atasnya`
```

Kemudian hapus baris yang lama dengan perintah

```
awk '{print $1"."$2"."$3"."$4"."$5}' /tmp/brsact2 >/tmp/brsact3
del2=`cat /tmp/brsact3`
sed '/'"$del2"'/d' /etc/shorewall/rules>/tmp/rulebaru
```

Setelah baris yang lama terhapus maka sisipkan baris baru dibawah baris yang sudah diketahui yang disimpan pada variabel "atasnya"dan simpan ke dalam file *temporary* untuk selanjutnya dicopy kan ke file konfigurasi Shorewall untuk mendapatkan file konfigurasi yang baru. Kode programnya adalah :

```
Sed '/'"$atasnya"'/r /tmp/actbaru' /tmp/rulebaru >/tmp/rulebaru1
Cp -f /tmp/rulebaru1 /etc/shorewall/rules
```

Proses pengaktifan ini juga terjadi untuk file konfigurasi yang lain (Zone kofigurasi, interface konfigurasi, masqureade konfigurasi, dan one-to-one konfigurasi) hanya saja file yang dieksekusi pada direktori `/usr/local/apache2/cgi-bin/skywall/hapus` berbeda sesuai dengan nama konfigurasi masing-masing. Berikut ini daftar file yang dieksekusi pada direktori `/usr/local/apache2/cgi-bin /skywall/hapus` untuk melakukan proses pengaktifan baris konfigurasi berdasarkan konfigurasi tertentu :

- Zone konfigurasi : file aktifzone
- Interface konfigurasi : file aktifint
- Masqureade konfigurasi : file aktifmasq
- One to one konfigurasi : file aktifnat

f. Nonaktifkan konfigurasi (*rules* konfigurasi)

Proses ini adalah proses me-nonaktifkan sebuah baris konfigurasi dari kondisi aktif. Proses ini akan mengeksekusi file "rem" pada direktori `/usr/local/apache2/cgi-bin/skywall/hapus`. Langkah pertama adalah membaca baris yang akan di non-aktifkan dan disimpan pada fille direktori `/tmp/brsrem2` dengan kode program dibawah ini :

```
let no=`cat /usr/local/apache2/cgi-bin/skywall/hapus/norem`+327
echo $no > /tmp/brsrem
let rem=`cat /tmp/brsrem`
awk NR~/$rem/ /etc/shorewall/rules >/tmp/brsrem2
```

Setelah baris yang akan dinonaktifkan disimpan, maka baris tersebut akan ditulis ulang ditambah dengan karakter '#' yang berarti *remark* dan tidak akan dieksekusi oleh program. Hasil penulisan ulang tersebut disimpan pada file direktori /tmp Kode programnya adalah sebagai berikut

```
awk '{print "#"$1" "$2" "$3" "$4" "$5}' /tmp/brsrem2
>/tmp/rembaru
```

Sama seperti proses hapus, pada proses ini perlu juga diketahui baris diatas baris yang diproses dan disimpan ke dalam sebuah variabel "atasnya" dengan kode program sbb:

```
let atasnya=`cat /tmp/brsrem`-1
awk NR~/$atasnya/ /etc/shorewall/rules >/tmp/atasnya
atasnya=`cat /tmp/atasnya`
```

Kemudian sisipkan baris baru yang sudah memiliki karakter "#" dengan kode program :

```
sed '/'"$atasnya"'/r /tmp/rembaru' /tmp/rulebaru >/tmp/rulebaru1
```

Tetapi sebelum dilakukan penyisipan harus dilakukan penghapusan terhadap baris yang lama terlebih dahulu dengan kode program:

```
awk '{print $1"."$2"."$3"."$4"."$5}' /tmp/brsrem2 >/tmp/brsrem3
del2=`cat /tmp/brsrem3`
sed '/'"$del2"'/d' /etc/shorewall/rules>/tmp/rulebaru
```

Langkah terakhir adalah mengcopy konfigurasi baru yang disimpan di file direktori /tmp/rulebaru1 ke dalam file konfigurasi Shorewall di /etc/shorewall/rules dengan perintah kode program :

```
cp -f /tmp/rulebaru1 /etc/shorewall/rules
```

Setting konfigurasi yang baru akan dieksekusi setelah dilakukan restart pada program.

Proses pengaktifan ini juga terjadi untuk file konfigurasi yang lain (Zone kofigurasi, interface konfigurasi, masqureade konfigurasi, dan one-to-one konfigurasi) hanya saja file yang dieksekusi pada direktori `/usr/local/apache2/cgi-bin/skywall/hapus` berbeda sesuai dengan nama konfigurasi masing-masing. Berikut ini daftar file yang dieksekusi pada direktori `/usr/local/apache2/cgi-bin /skywall/hapus` untuk melakukan proses hapus berdasarkan konfigurasi tertentu :

- Zone konfigurasi : file aktifzone
- Interface konfigurasi : file aktifint
- Masqureade konfigurasi : file aktifmasq
- One to one konfigurasi : file aktifnat

Proses penon-aktifan ini juga terjadi untuk file konfigurasi yang lain (Zone kofigurasi, interface konfigurasi, masqureade konfigurasi, dan one-to-one konfigurasi) hanya saja file yang dieksekusi pada direktori

/usr/local/apache2/cgi-bin/skywall/hapus berbeda sesuai dengan nama konfigurasi masing-masing. Berikut ini daftar file yang dieksekusi pada direktori /usr/local/apache2/cgi-bin /skywall/hapus untuk melakukan proses penon-aktifan baris konfigurasi berdasarkan konfigurasi tertentu :

- Zone konfigurasi : file remzone
- Interface konfigurasi : file remint
- Masqureade konfigurasi : file remmasq
- One to one konfigurasi : file remnat

4.4.10. Implementasi Policy Setting

Policy setting adalah proses setting keadaan *default* program. Keadaan ini akan dieksekusi ketika *rule* telah didefinisikan pada keadaan *default (policy)* tidak ada. Fungsi pengesetan *policy* pada program ini akan berjalan dengan mengeksekusi modul policy.cgi pada direktori /usr/local /apache2/cgi-bin/skywall/default. Proses setting *policy* pada program ini dilakukan dengan kode program berikut ini:

```
$a1=$q->param('1');
$a2=$q->param('2');
$a3=$q->param('3');
$a4=$q->param('4');
$a5=$q->param('5');
$a6=$q->param('6');
$a7=$q->param('7');
$a8=$q->param('8');
$a9=$q->param('9');
```

Setelah masing masing inputan untuk 9 keadaan pada *policy* setting divariabelkan dengan kode program diatas, maka selanjutnya isi dari variabel tersebut dituliskan ke dalam file def1 di direktori /usr/local /apache2 /cgi-bin/skywall/default dengan kode program dibawah ini.

```
system("echo 'loc' 'fw' '$a1' > def1");
```

```
system("echo 'loc' 'fw' '$a2' >> def1");
system("echo 'loc' 'fw' '$a3' >> def1");
system("echo 'loc' 'fw' '$a4' >> def1");
system("echo 'loc' 'fw' '$a5' >> def1");
system("echo 'loc' 'fw' '$a6' >> def1");
system("echo 'loc' 'fw' '$a7' >> def1");
system("echo 'loc' 'fw' '$a8' >> def1");
system("echo 'loc' 'fw' '$a9' >> def1");
```

Setelah file `def1` terisi dengan susunan konfigurasi *policy* yang baru maka yang dilakukan adalah mengcopy file setting *policy* yang masih kosong pada `/usr/local/apache2/cgi-bin/skywall/default/policy` ke direktori `/etc/shorewall` agar file tersebut kembali kosong. Tentunya dengan mengubah *file permission* direktori `/etc/shorewall` dengan perintah `chmod 755` terlebih dahulu .Setelah file yang masih kosong tercopy maka selanjutnya adalah menyisipklan file def1 yang sudah berisi setting *policy* yang baru ke file direktori setting *policy* Shorewall yaitu pada `/etc/shorewall/policy`. Hasil penyisipan tersebut dimasukkan kedalam file temporary dan langkah terakhir adalah menyalinnya ke direktori `/etc/shorewall`.

```
System ("chmod 755 /etc/shorewall/");
system ("cp -f /usr/local/apache2/cgi-bin/skywall/default/policy
         /etc/shorewall");
system ("chmod 755 /etc/shorewall/policy");
system ("sed '/#dibawahini/r /usr/local/apache2/cgi-
        bin/skywall/default/def1' /etc/shorewall/policy
        >/tmp/policy");
System ("cp -f /tmp/policy /etc/shorewall");
```

File konfigurasi policy yang baru akan dieksekusi setelah program direstart.

4.4.11. Implementasi Rule Setting

Proses *rule setting* adalah proses setting *rule* (aturan) yang akan dijalankan untuk melakukan proses *firewalling* pada sistem dan juga pada program ini. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul `rules.cgi` pada direktori `/usr/local/apache2/cgi-`

bin/skywall/ pada linux. Kode program untuk melakukan pengesetan *rules* (aturan) firewalling adalah sebagai berikut :

```
$aksi=$q->param('action');
$asal=$q->param('source');
$tuj=$q->param('dest');
$ips=$q->param('ips');
$ipd=$q->param('ip');
$prot=$q->param('proto');
$port=$q->param('destport');

system("echo '$aksi' '$asal''$ips' '$tuj''$ipd' '$prot'
'$port'>/tmp/rules");
system("sed '/#dibawahini/r /tmp/rules' /etc/shorewall/rules
>/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/dnat");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu *rules setting*. Variabel $aksi untuk inputan *action* terhadap suatu operasi apakah setuju atau ditolak, variabel $asal untuk inputan zone asal, variabel $tuj untuk inputan zone tujuan, variabel $ips untuk inputan alamat IP asal, variabel $ipd untuk inputan alamat IP tujuan, variabel $prot untuk protokol tujuan dan variabel $port untuk port tujuan. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
system("echo'$aksi''$asal''$ips''$tuj''$ipd''$prot'
'$port'>/tmp/rules");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori /etc/shorewall/rules dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori /tmp/hasil dengan perintah

```
system("sed'/#dibawahini/r/tmp/rules'/etc/shorewall/rules
>/tmp/hasil");
```

Langkah terakhir adalah mengcopy file direktori /tmp/hasil ke file direktori setting *rules* pada Shorewall yaitu ada pada /etc /shorewall/rules dengan perintah:

```
System("cp -f /tmp/hasil /etc/shorewall/rules");
```

setelah program di restart, isi file direktori /etc/shorewall/rules akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.12. Implementasi One to One

Proses One to One adalah proses *Network Address Translation (NAT)* untuk satu alamat ip lokal dengan satu alamat ip-global. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul onotoone.cgi pada direktori /usr/local/apache2/cgi-bin/skywall/ pada linux. Kode program untuk melakukan NAT secara *one to one* adalah sebagai berikut :

```
$iplok=$q->param('iplokal');
$ipglo=$q->param('ipglobal');

system("echo '$ipglo' eth1 '$iplok' no>/tmp/nat");
system("sed '/#dibawahini/r /tmp/nat' /etc/shorewall/nat
>/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/nat");
system("rm /tmp/hasil");
system("rm /tmp/nat");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu *one to one* konfigurasi. Variabel $iplok untuk inputan alamat ip-lokal, variabel $ipglo untuk inputan alamat ip-address yang akan digunakan untuk alamat global. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
system("echo '$ipglo' eth1 '$iplok' no>/tmp/nat");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori /etc/shorewall/nat dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori /tmp/hasil dengan perintah

```
system("sed'/#dibawahini/r/tmp/nat'/etc/shorewall/nat
>/tmp/hasil");
```

Langkah terakhir adalah mengcopy file direktori /tmp/hasil ke file direktori setting NAT secara *ono to one* pada Shorewall yaitu ada pada /etc/shorewall/nat dengan perintah:

```
System("cp -f /tmp/hasil /etc/shorewall/nat");
```

setelah program di restart, file direktori pada /etc/shorewall/nat akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.13. Implementasi Masqureade

Proses masqureade adalah proses *Network Address Translation (NAT)* untuk sekelompok segment jaringan dengan satu alamat ip-global. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul masq.cgi pada direktori /usr/local/apache2/cgi-bin/skywall/ pada linux. Kode program untuk melakukan NAT secara *Masqureade* adalah sebagai berikut :

```
$int=$q->param('internal');
$ext=$q->param('external');

system("echo '$ext' '$int'>/tmp/masq");
system("sed '/#dibawahini/r /tmp/masq' /etc/shorewall/masq
>/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/masq");
system("rm /tmp/hasil");
system("rm /tmp/masq");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu *masqureade* konfigurasi. Variabel $int untuk inputan

kartu jaringan internal, variabel $ext untuk inputan kartu jaringan external yang alamat ip-addressnya akan digunakan untuk alamat global bagi kartu jaringan internal. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
system("echo '$ext' '$int'>/tmp/masq");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori `/etc/shorewall/masq` dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori `/tmp/hasil` dengan perintah

```
system("sed'/#dibawahini/r/tmp/masq'/etc/shorewall/masq
>/tmp/hasil");
```

Langkah terakhir adalah mengcopy file direktori `/tmp/hasil` ke file direktori setting masqureade Shorewall yaitu pada `/etc/shorewall /masq` dengan perintah.

```
System("cp -f /tmp/hasil /etc/shorewall/masq");
```

setelah program di restart file direktori pada `/etc/shorewall/masq` akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

4.4.14. Implementasi DNAT

Proses DNAT *(Destination NAT)* adalah proses mengijinkan host dari luar untuk mengakses komputer server yang berada dibelakang router. Fungsi ini akan dijalankan dengan melakukan eksekusi pada modul `dnat.cgi` pada direktori `/usr/local/apache2/cgi-bin/skywall/` pada linux. Kode program untuk melakukan DNAT adalah sebagai berikut :

```
$asl=$q->param('asal');
$tuj=$q->param('tujuan');
$ipnya=$q->param('ip');
$prot=$q->param('protokol');
```

```
$port=$q->param('porttuj');
$tujasl=$q->param('tujasli');

system("echo DNAT '$asl' '$tuj'':$ipnya' '$prot' '$port' -
'$tujasl'>/tmp/dnat");
system("sed '/#dibawahini/r /tmp/dnat' /etc/shorewall/rules
>/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/dnat");
```

Langkah pertama adalah memberikan variabel untuk masing-masing inputan dari form menu DNAT konfigurasi. Variabel $asl untuk inputan zona asal, $tuj untuk zona tujuan, $ipnya untuk alamat ip tujuan, $prot untuk protokol tujuan, $port untuk port tujuan, $tujasl untuk tujuan aslinya. Kode program akan menuliskan hasil inputan dari form melalui variabel tersebut ke dalam direktori *temporary* dengan perintah :

```
system("echo DNAT '$asl' '$tuj'':$ipnya' '$prot' '$port' -
'$tujasl'>/tmp/dnat");
```

Selanjutnya isi dari hasil penulisan tersebut disisipkan kedalam file direktori `/etc/shorewall/rules` dibawah kata '#dibawahini' dan hasilnya dimasukkan ke dalam file direktori `/tmp/hasil` dengan perintah

```
System("sed'/#dibawahini/r/tmp/dnat'/etc/shorewall/rules
>/tmp/hasil");
```

Langkah terakhir adalah mengcopy file direktori `/tmp/hasil` ke file direktori setting DNAT Shorewall yaitu pada `/etc/ shorewall /interfaces` dengan perintah.

```
System("cp -f /tmp/hasil /etc/shorewall/rules");
```

setelah program di restart, isi file direktori `/etc/shorewall/rules` akan dijalankan dengan settingan yang baru sesuai dengan inputan yang dilakukan.

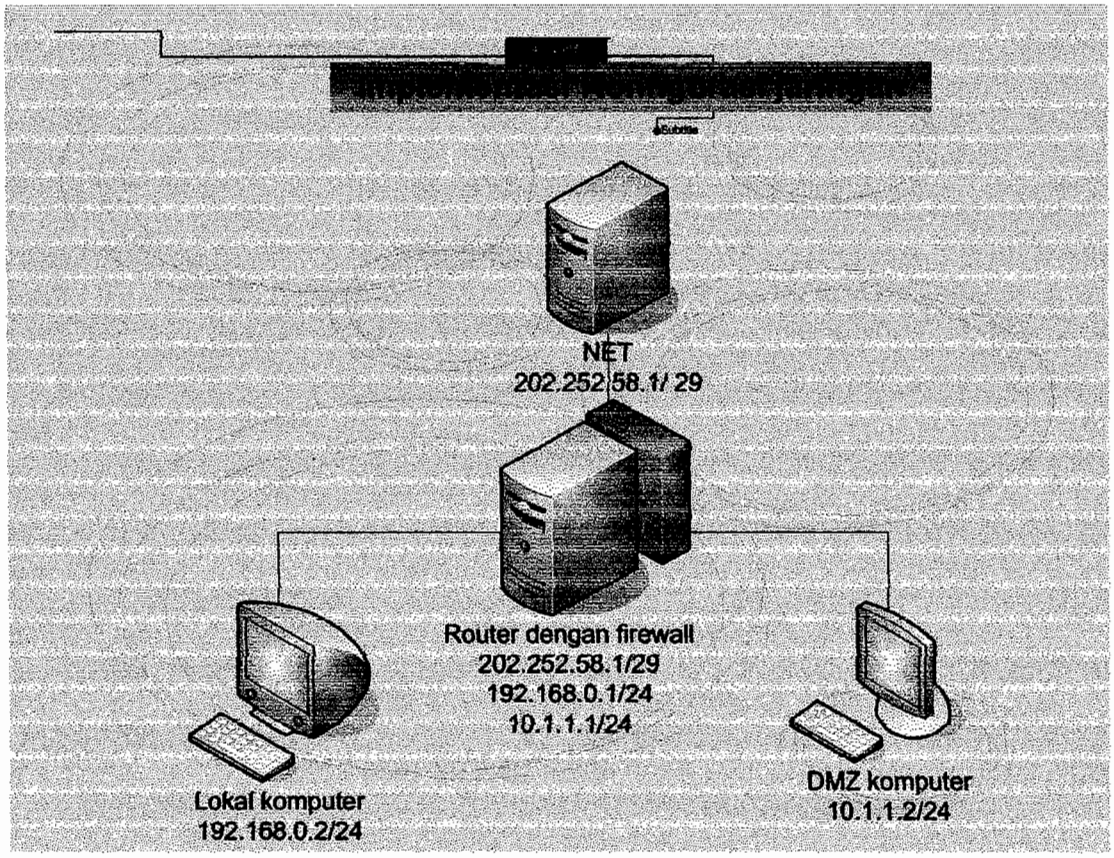
BAB V

ANALISA HASIL IMPLEMENTASI

5.1. Hasil ujicoba program

Pada bagian ini akan dilaporkan uji coba program ke dalam jaringan untuk penggunaan servis-servis tertentu.

Pada uji coba program digunakan 4 buah komputer yang diimplementasikan sebagai komputer router dengan firewall, komputer lokal, komputer Dmz (server). Berikut ini adalah gambar dari konfigurasi jaringan yang dipakai pada implementasi program yang sudah dibuat :



Gambar 5.1 konfigurasi jaringan

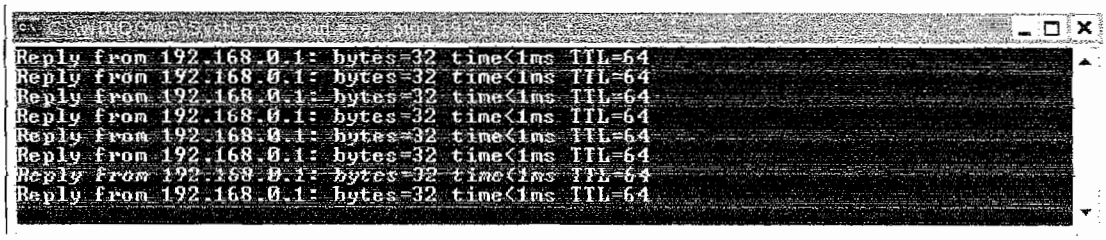
Komputer router yan digunakan untuk firewall menggunakan 3 kartu jaringan yang masing-masing berbeda *network id*. Masing masing jartu jaringan dihubungkan dengan komputer yang mewakili tiap-tiap zone (daerah). Alamat IP dan kartu jaringan tersebut adalah sebagai berikut :

1. Kartu jaringan yang ke 1 (eth 0) : 202.252.58.1/29
2. Kartu jaringan yang ke 2 (eth 1) : 192.168.0.1 /24
3. Kartu jaringan yang ke 3 (eth 2) : 10.1.1.1 /24

Masing masing kartu jaringan tersebut terhubung dengan 1 buah komputer yang satu *network id* dengan alamat IP yang dimilikinya dan sekaligus mewakili 3 zone yang digunakan pada sistem ini.

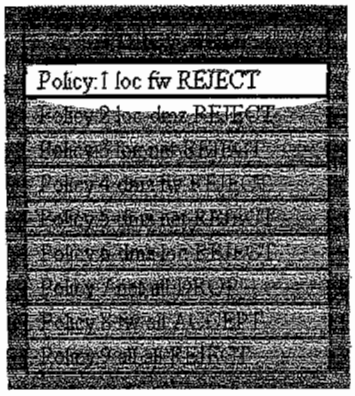
5.1.1. Ping icmp

Sebelum firewall diaktifkan dapat dilakukan ping melalui protokol ICMP ke alamat 192.168.0.1 (alamat firewall zone lokal) oleh komputer 192.168.0.2.



Gambar 5.2 Ping reply ke firewall

Tetapi setelah firewall diaktifkan dan pada policy di set dari zone lokal ke firewall tidak diijinkan melakukan operasi apapun seperti tampak pada gambar dibawah ini,

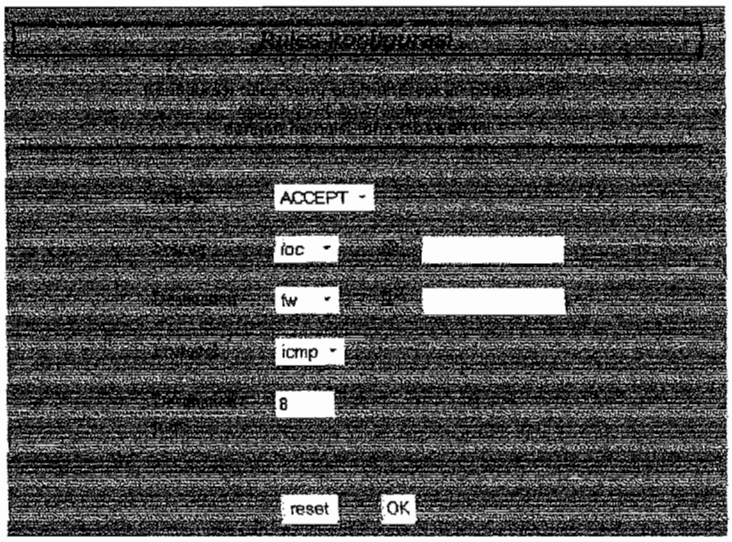


Gambar 5.3 Policy untuk lokal ke firewall

maka tidak dapat dilakukan ping icmp ke komputer firewall dan akan tampak sebagai berikut.

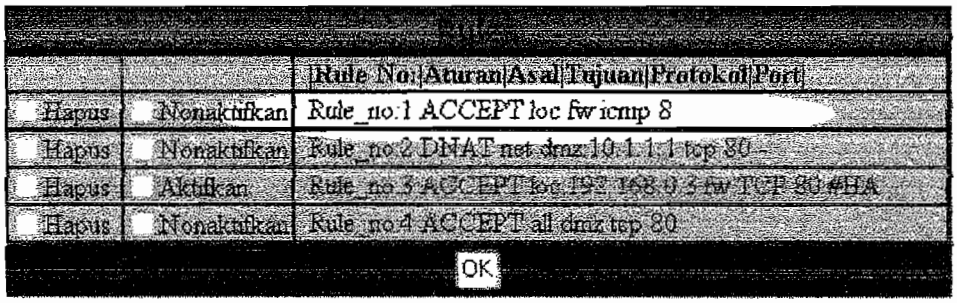
Gambar 5.4 Ping gagal ke firewall

Jika pada menu rule setting ditambah aturan untuk mengijinkan ping dari komputer lokal ke firewall yaitu dengan setting seperti tampak pada gambar dibawah ini.



Gambar 5.5 konfigurasi rules ping icmp

maka ping ke alamat 192.168.0.1 akan dapat dilakukan kembai karena aturan sudah ditambahkan pada pabel rules seperti tampak pada menu view konfigurasi atau gambar dibawah ini.

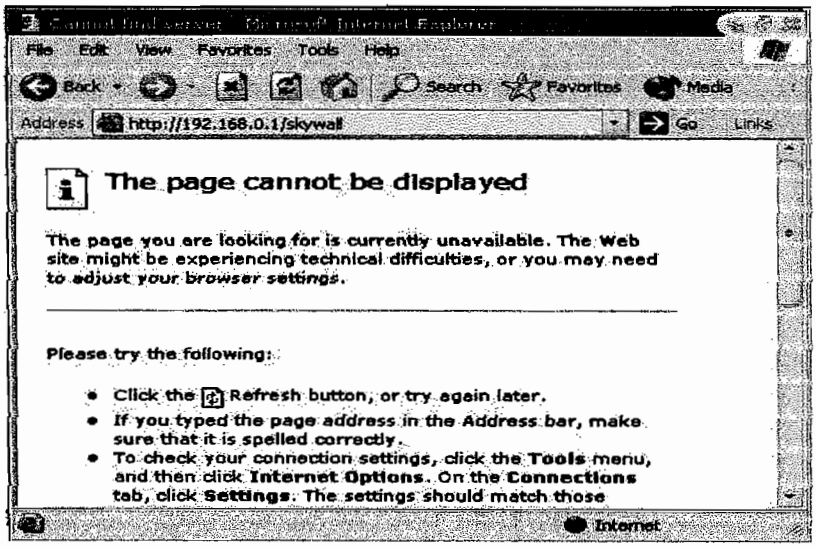


Gambar 5.6 Tabel rules ping icmp

5.1.2. Host allow (Akses Program ke Firewall)

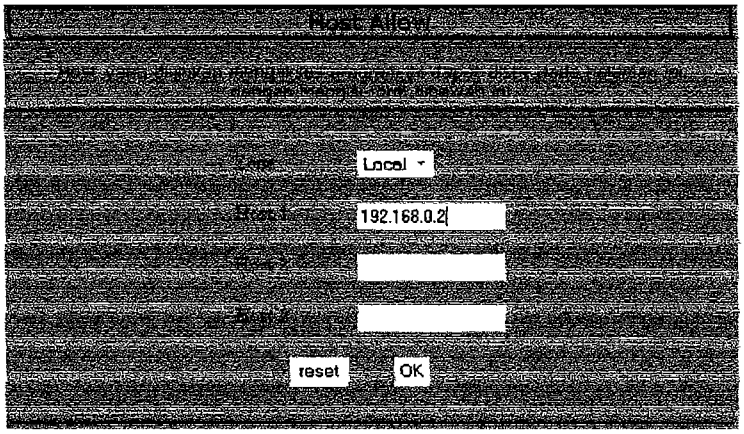
Pada program ini disediakan menu host allow yang digunakan untuk mengijinkan host mana yang diijinkan mengakses program ini.

Host pada lokal dengan IP 192.168.0.2 tidak dapat melakukan akses program ini, pada web browser ditampilkan peringatan sebagai berikut:



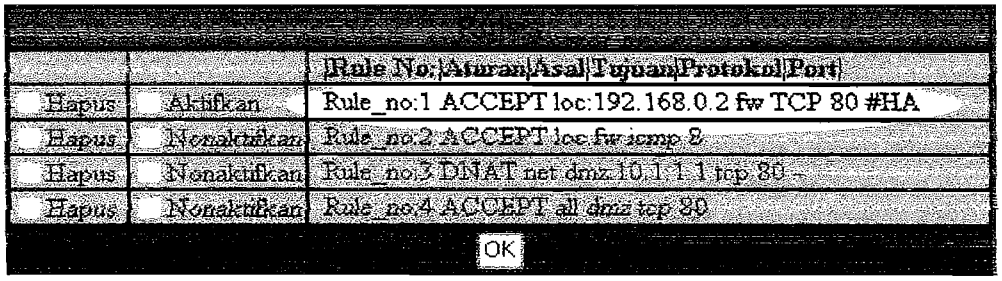
Gambar 5.7 Host yang tidak diijinkan

Selanjutnya dilakukan setting untuk mengijinkan host dengan alamat IP 192.168.0.2 untuk dapat mengakse program ini. Konfigurasi dilakukan dengan mengisi form dibawah ini pada menu Host Allow.



Gambar 5.8 Setting Host yang diijinkan

Setelah tombol Ok ditekan maka konfigurasi akan masuk pada tabel *rules* untuk mengijinkan *host* tersebut mengakses program. Konfigurasi tersebut pada menu View Rules ditampilkan sebagai berikut :



Gambar 5.9 Host yang tidak diijinkan

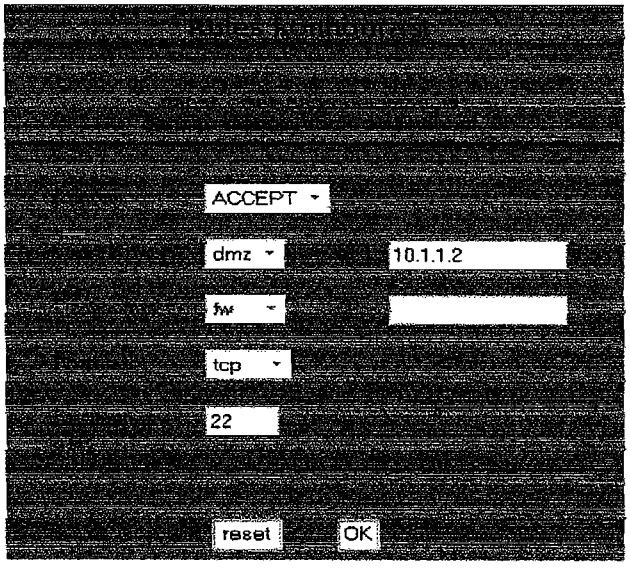
Setelah program di restart host dengan IP 192.168.0.2 pada jaringan lokal dapat mengakses program ini dan menggunakannya. Sedangkan komputer dengan alamat IP lain tidak dapat meggunakan program ini.

5.1.3. SSH

Sebelum ditambahkan *rules*, komputer pada daerah dmz tidak dapat dapat melakukan ssh ke komputer firewall karena policy yang aktif tidak mengijinkan itu. Hal ini tampak pada gambar berikut ini :

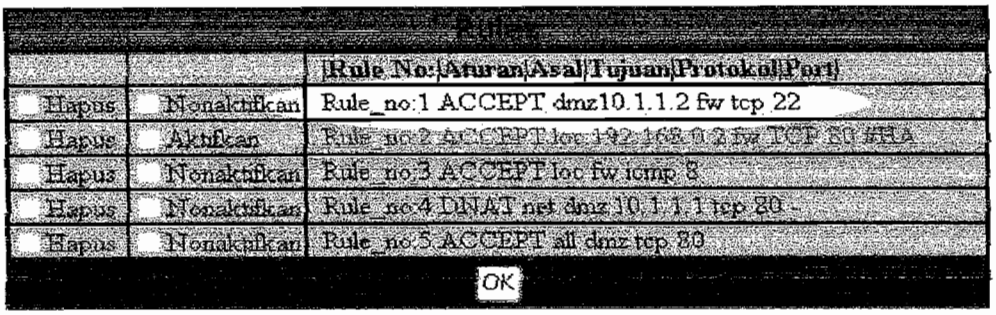
Gambar 5.10 SSH tidak diijinkan

Untuk dapat melakukan ssh dari komputer pada wilayah dmz dengan alamat IP 10.1.1.2 perlu ditambah konfigurasi menggunakan menu Rules Setting seperti tampak gambar berikut :



Gambar 5.11 Setting rules SSH

Konfigurasi tersebut masuk ke dalam tabel *rules*. Seperti tampak pada gambar dibawah ini :



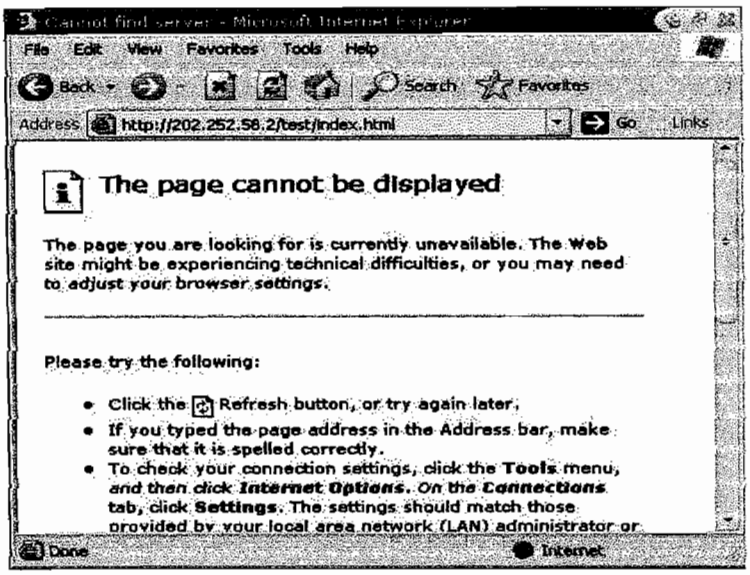
Gambar 5.12 Setting rules SSH

Setelah program di restart maka servis SSH dari wilayah DMZ kw komputer firewall dapat dilakukan dan tampak koneksi yang diijinkan pada gambar berikut :

Gambar 5.13 SSH diijinkan

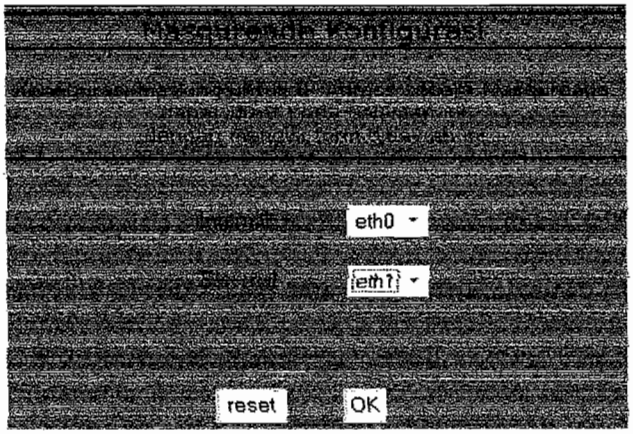
5.1.4. Masqureade

Sebelum ditambahkan aturan untuk mengijinkan komputer pada jaringan lokal dapat mengakses web pada jaringan luar (Net), maka komputer 192.168.0.1 tdak dapat melakukan browsing ke jaringan luar seperti tampak pada gambar berikut ini.



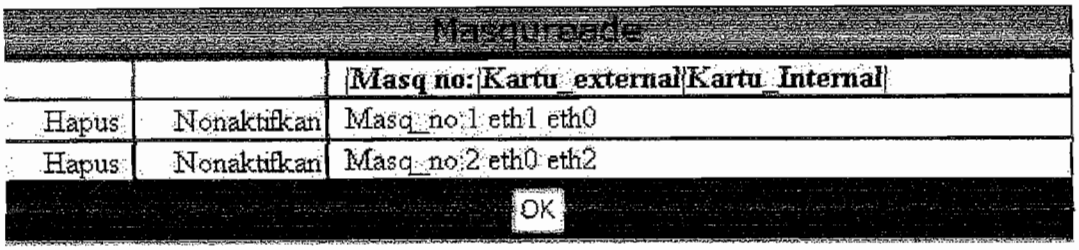
Gambar 5.14 Masqureade tidak dilakukan

Untuk mengijinkan komputer lokal dapat mengakses jaringan luar (Net) maka dilakukan setting pada menu Masqureade.



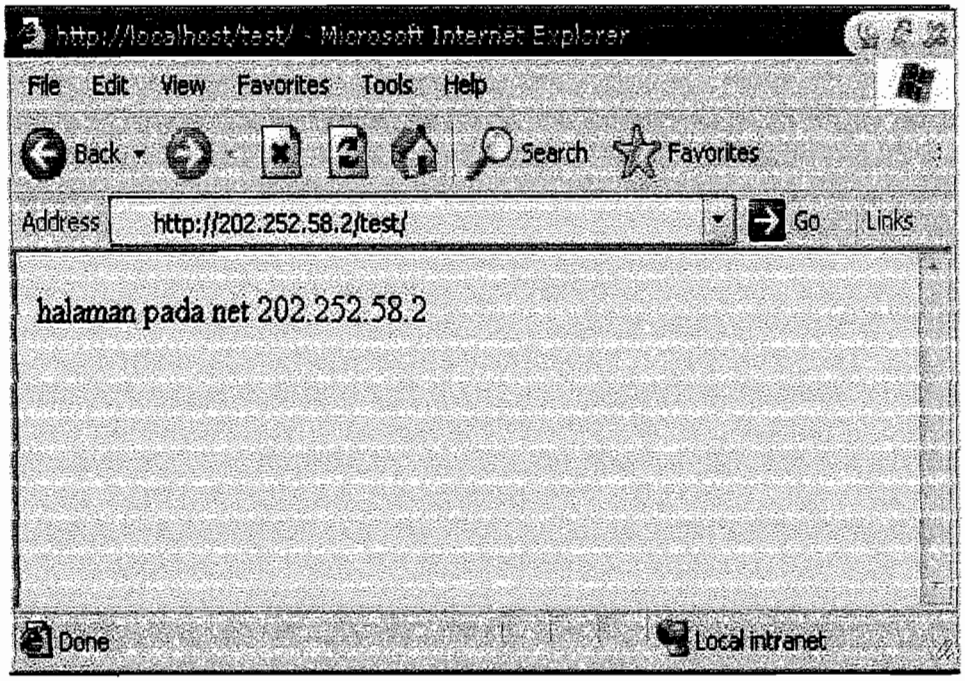
Gambar 5.15 Setting Masqureade

Setelah konfigurasi masqureade ditambahkan maka konfigurasi akan masuk pada tabel masqureade seperti tampak pada gambar berikut ini.



Gambar 5.16 Masqureade tabel

Konfigurasi pada tabel masqureade tersebut dijalankan dan komputer lokal dapat mengakses halaman web yang terdapat pada jaringan luar (browsing). Seperti terlihat pada gambar berikut :



Gambar 5.17 Masqureade dilakukan

5.2. Kelebihan dan Kekurangan Program

Program yang dihasilkan ini mempunyai kelebihan dan kekurangan. Kelebihan dan kekurangan program aplikasi *web base firewall* yang dibuat adalah diantaranya adalah :

5.2.1. Kelebihan

Kelebihan yang dimiliki program adalah :

1. Program dapat dijalankan pada komputer dengan spesifikasi yang minimal. Sehingga dengan program ini dapat diberdayakan komputer dengan spesifikasi yang minimal untuk digunakan sebagai komputer router dengan firewall
2. Program menyediakan fasilitas untuk melakukan perubahan alamat IP yang digunakan router dan sekaligus melakukan restart untuk *service network*-nya sehingga membantu seorang admin dalam melakukan konfigurasi jaringan.
3. Masing masing setting ditampilkan dengan tabel yang dikelompokkan sesuai dengan jenis setting yang dilakukan. Sehingga memudahkan untuk melihat setting yang sudah dilakukan
4. Pada tabel yang ditampilkan tersedia *checkbox* untuk menghapus, mengaktifkan atau me-nonaktifkan baris konfigurasi yang ditampilkan sehingga memudahkan pengguna untuk melakukan managemen baris konfigurasi.

5.2.2. Kekurangan

Kekurangan dari program yang telah dibuat adalah :

1. Tidak menggunakan metode *encrypt* data sehingga data yang dikirimkan masih belum terenkripsi.

2. Tidak adanya fungsi session pada CGI mengakibatkan harus dilakukan logout untuk mengamankan program, agar ketika ada pengguna lain yang ingin menggunakan program harus melakukan login kembali.
3. Managemen sebuah file menggunakan *Shell programming* dilakukan dengan baris-per baris sehingga tampilan antarmuka yang dihasilkan kurang maksimal.

5.3. Analisis Metode berbasis Web

Pada bagian ini akan dibahas mengenai efektivitas dan efisiensi penggunaan metode berbasis web yang digunakan dalam membangun atau mengembangkan suatu perangkat-lunak untuk aplikasi *firewalling.*

Metode yang digunakan dalam pembuatan aplikasi *firewalling* ini adalah metode berbasis web. Dengan metode ini penulis dapat membangun sebuah aplikasi *firewalling* yang memenuhi kebutuhan pengguna akan akses kontrol jarak jauh.

Dengan penerapan metode ini dalam pembuatan program, maka dimungkinkan diwujudkan sebuah aplikasi yang berguna untuk melakukan setting *firewalling* secara jarak jauh dalam satu lingkungan jaringan. Hal ini mengingat karena metode berbasis web mempunyai sifat operasi *client-server* dimana sebuah *client* dapat mengakses program atau data yang terletak pada *server*. Sifat inilah yang digunakan penulis untuk melakukan pemenuhan kebutuhan akan setting *firewalling* dari jarak jauh.

5.4. Analisis Bahasa Pemprograman

Bagian ini akan menganalisa bahasa pemprograman yang digunakan untuk mengembangkan aplikasi *web base firewall.*

5.4.1. CGI

CGI dapat digunakan untuk mengembangkan aplikasi yang dibuat oleh penulis karena CGI mempunyai kemampuan sebagai berikut :

1. CGI dapat menjadi pengantara web server dengan program aplikasi.
2. CGI dapat merealisasikan antarmuka yang interaktif dengan *user* (pengguna).
3. CGI dapat menjadi perantara *(gateway)* dengan aplikasi yang dikembangkan pada linux. Aplikasi ini dapat berupa shell programming ataupun berupa file konfigurasi yang terdapat pada direktori linux.
4. Meskipun dapat menjadi gateway untuk program atau aplikasi yang ada pada linux, CGI mempunyai kelemahan dalam fungsi *login session* sehingga menyulitkan penulis untuk mengatur aplikasi *login session* pada program.

Dalam pembuatan program ini penulis menggunakan file konfigurasi Shorewall untuk memudahkan pengaturan file konfugurasi *firewalling* berbasis *IP-Tables.* Karena Shorewall merupakan file konfigurasi yang dikembangkan pada linux, maka penggunaan CGI dapat memudahkan dalam pembuatan program.

5.4.2. Shell Programming

Shell Programming dapat digunakan untuk mengembangkan aplikasi program yang dibuat karena mempunyai kemampuan sebagai berikut :

1. Dalam melakukan pengaturan atau manajemen terhadap file konfigurasi Shorewall *Shell programming* sangat membantu dengan kemampuan *redirectionnya.*
2. Dalam melakukan parsing parameter *Shell script* Mempuyai kemampuan seperti bahasa pemprograman lainnya, bahkan mempunyai kemampuan yang lebih.
3. Shell programming kurang cocok digunakan untuk membuat aplikasi program yang ditekankan pada tampilan dari suatu karena dasar operasinya dilakukan dengan baris-perbaris

Pada pembuatan program ini penulis menggunakan sebuah file konfigurasi Shorewall. *Shell programming* sangat membantu dalam melakukan manajemen file konfigurasi ini terutama dengan kemampuan *redirectionnya.*

BAB VI

PENUTUP

Pada bagian akhir penulisan skripsi ini akan dicantumkan beberapa kesimpulan dan saran mengenai hal-hal yang terkait dengan pembuatan aplikasi *web base firewall* sebagai sarana bantu setting firewall berbasis ip-tables pada linux Redhat 9.0 menggunakan file konfigurasi Shorewall.

6.1. Kesimpulan

Dalam pembuatan aplikasi *web base firewall* berbasis ip-tables menggunakan file konfigurasi Shorewall dapat disimpulkan :

1. Program seting *firewall* yang dikembangkan dengan berbasis web dapat membantu seorang administrator dalam melakukan setting firewalling pada sistem secara jarak jauh.
2. Metode pengembangan berbasis web dapat membantu penulis dalam mengembangkan aplikasi setting *firewalling* jarak jauh.
3. CGI *(Common Gateway Interface)* adalah bahasa untuk merealisasikan teknologi berbasis web dan dapat digunakan dalam mengembangkan aplikasi yang membutuhkan manajemen file konfigurasi pada linux. Namun CGI mempunyai kekurangan dalam hal *login session.*
4. *Shell programming* adalah bahasa pemprograman yang cukup efektif dalam melakukan manajemen file konfigurasi yang dapat digunakan untuk melakukan pengembangan aplikasi berbasis file konfigurasi.

6.2. Saran

Saran yang diberikan penulis untuk mengembangkan aplikasi lebih lanjut adalah sebagai berikut :

1. Dapat dikembangkan aplikasi *web base firewall* yang menggunakan teknologi data *encryption* sehingga data yang dikirim akan lebih aman.
2. Dapat dikembangkan aplikasi *web base firewall* yang menggunakan pengaturan *login session* sehingga proses *login* dapat diatur lebih aman.
3. Pengembangan aplikasi web base Firewall akan lebih mudah jika menggunakan file konfigurasi karena akan memudahkan dalam manajemen file setting yang digunakan.

DAFTAR PUSTAKA

Susanto, Budi (2001). *UNIX dan Pemprograman Script.* Yogyakarta, J&J Learning

Gunadi, Hariman (2002). *Visual Modeling Menggunakan UML dan Rational Rose.* Bandung. Informatika

Frans (2002). *Membuat Web Dinamis dan Interaktif dengan CGI.* Yogyakarta, ANDI

Wahana (2003). *Panduan Lengkap Pengembangan Jaringan Linux.* Yogyakarta, ANDI

Wahana (2003). *Mari Mengenal Linux.* Yogyakarta, ANDI

Grennan, Mark (2000). *Firewall and Proxy Server HOWTO* http://www.grennan.com/Firewall-HOWTO.html. Diakses 12 Mei 2004

Stein, L. (2004) *CGI Tutorial.* http://www.stein.cshl.org/WWW/software/CGI/cgi_docs. html. Diakses 12 Mei 2004

Harlan's, David (1996). *Special Edition Using Perl for Web Programming* http://www.leaf.stpn.soft.net/. Diakses 25 Mei 2004

Eastep, Thomas M (2004). *Shorewall Documentation* http://www.shorewall.net/documentation.pdf. Diakses 14 Oktober 2004

Brockmeier, joe (2001) *IP-Tables* http://www.sns.ias.edu/~jns/security/iptables/iptables_manpage.html #lbAZ. Diakses 5 Mei 2004

N.N http://www.iana.org/assignments/portnumbers.html. Diakses 2 Juli 2004

Mudji, Basuki (2004) *Network Address Translation (NAT)* http://www.ilmukomputer.com/. Diakses 31 Mei 2004

Muammar, Ahmad (2001). *Firewall.* http://www.ilmukomputer.com/. Diakses 31 Mei 2004

LISTING PROGRAM

DNAT

```
#!/usr/bin/perl

use CGI;
$q=new CGI;

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";

$asl=$q->param('asal');
$tuj=$q->param('tujuan');
$ipnya=$q->param('ip');
$prot=$q->param('protokol');
$port=$q->param('porttuj');
$tujasl=$q->param('tujasli');

system("echo DNAT '$asl' '$tuj":$ipnya' '$prot' '$port'
- '$tujasl'>/tmp/dnat");
system("sed '/#dibawahini/r /tmp/dnat'
/etc/shorewall/rules >/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/dnat");

system("./hitung");
print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>";
   print "<tr>";
   print "<td height=25 colspan=2 align=center
valign=top ><font face=Verdana size=+1>";
   print "DNAT</font></td>";
   print "</tr>";
print "</table>";

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>\n";
$baris=`cat /tmp/jmlbaris`;
for ($i=1;$i<=$baris-1;$i++)
{
$as=`cat ruleku |egrep 'Rule_no:$i DNAT'`;
  print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  $as";
   print "</td>";
  print "</tr>";
}
print "</table>";

print "<table border=0 bgcolor=#000066
width=100%>\n";
print"<form method=post action=/cgi-
bin/skywall/html/dnat.cgi>";
            print "<tr><td align=center><font
color#000000>\n";
            print "<input type=submit name=back
value=back>";
            print "</tr></td></table>";
print "</form>";
print "</table>";

print "</body>";
print $q->end_html();
```

HOST ALLOW

```
#!/usr/bin/perl

use CGI;
$q=new CGI;

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";

$ip1=$q->param('ip1');
$ip2=$q->param('ip2');
            #system ("cat $ip2>ip2a");
            #system ("awk '{print ",",$1}'
ip2a>ip2new");
            #$ip2new=`cat ip2new`;
            #system ("rm ip2a");
$ip3=$q->param('ip3');
            #system ("cat $ip2>ip3a");
            #system ("awk '{print ",",$1}'
ip3a>ip3new");
            #$ip3new=`cat ip3new`;
            #system ("rm ip3a");

$iplama=$q->param('iplama');

#system("echo 'DROP' 'all' 'all' 'TCP'
'80'>/tmp/host");
system("sed '/#HA/d' /etc/shorewall/rules
>/tmp/hostbaru");
system("echo 'ACCEPT' 'loc:"$ip1' '$ip2' '$ip3' 'fw'
'TCP' '80' '#HA'>/tmp/host");
system("sed '/#dibawahini/r /tmp/host' /tmp/hostbaru
>/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/host");

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>";
   print "<tr>";
   print "<td height=25 colspan=2 align=center
valign=top ><font face=Verdana size=+1>";
   print "Host Allow</font></td>";
   print "</tr>";
print "</table>";

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>\n";
print "<tr>";
   #print "<td width=75 valign=top
bgcolor=#6699CC>";
   #print "<input name=Hapus$i type=checkbox
value=cek$i><font size=3>Hapus </font></td>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  Host Allow diset pada
Local : $ip1,$ip2,$ip3";
   print "</td>";
  print "</tr>";
print "</table>";

print "<table border=0 bgcolor=#000066
width=100%>\n";
```

```
print"<form method=post action=/cgi-
bin/skywall/html/hostallow.cgi>";
        print "<tr><td align=center><font
color#000000>\n";
        print "<input type=submit name=back
value=back>";
       print "</tr></td></table>";
print "</form>";
print "</table>";

print "</body>";
print $q->end_html();


IP Konf
#!/usr/bin/perl

use CGI;
$q=new CGI;

$kartu=$q->param('menu_eth');
$mask=$q->param('netmask');
$ip=$q->param('ip');
$broad=$q->param('broadcast');

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<table width=100% border=0 cellspacing=0
align=center cellpadding=0 bordercolor=#000066>";
   print "<tr>";
      print "<td height=25 colspan=2 align=center
valign=top> <font face=Verdana size=-1>
      Anda sudah melakukan perubahan pada $kartu
dari :</font>";
      print "<hr color=#000066>";
      print "</td>";
      print "</tr>";
print "</table>";
################################tabel eth1
print "<p>";
print "<p>";
print "<p>";
$ip1=`cat /etc/sysconfig/network-scripts/ifcfg-$kartu
|grep IPADDR`;
$mask1=`cat /etc/sysconfig/network-scripts/ifcfg-
$kartu |grep NETMASK`;
$broad1=`cat /etc/sysconfig/network-scripts/ifcfg-
$kartu |grep BROADCAST`;
print "<table width=50% border=1 align=center
cellspacing=0 cellpadding=0
bordercolor=#000066>";
print "<tr>";
print "<td height=25 colspan=2 align=center
valign=top><font face=Verdana size=+1>
         Konfigurasi Lama $kartu</font></td>";
print "</tr>";
print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  $ip1";
   print "</td>";
  print "</tr>";
print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  $mask1";
   print "</td>";
print "</tr>";
#print "<tr>";
#   print "<td width=166 height=25
bgcolor=#6699CC><font face=Verdana size=-
1>  Kernel</font></td>\n";
#    print "<td width=599 valign=top
bgcolor=#6699CC>  $broad1";
#    print "</td>";
#print "</tr>";
print "</table>";
system("sudo -u root chmod 777
/etc/sysconfig/network-scripts/ifcfg-$kartu");
system("sed '/IPADDR/d' /etc/sysconfig/network-
scripts/ifcfg-$kartu >/tmp/del");
system("cp -f /tmp/del /etc/sysconfig/network-
scripts/ifcfg-$kartu");
system("echo IPADDR='$ip' >/tmp/ip");
system("sed '/BROADCAST/r /tmp/ip'
/etc/sysconfig/network-scripts/ifcfg-$kartu >/tmp/ip-
ok");
system("cp -f /tmp/ip-ok /etc/sysconfig/network-
scripts/ifcfg-$kartu");
system("rm /tmp/del");
system("rm /tmp/ip");
system("rm /tmp/ip-ok");
system("sudo -u root chmod 0644
/etc/sysconfig/network-scripts/ifcfg-$kartu");
#system("sudo -u root ifconfig $kartu down >
/tmp/restart");
#system("sudo -u root ifconfig $kartu up >>
/tmp/restart");
system("sudo -u root service network restart
>/tmp/restart");
$restart=`cat /tmp/restart`;

#-------------------------------------------------
--------
print "<p>";
print "<p>";
print "<p>";
$ip1=`cat /etc/sysconfig/network-scripts/ifcfg-$kartu
|grep IPADDR`;
$mask1=`cat /etc/sysconfig/network-scripts/ifcfg-
$kartu |grep NETMASK`;
$broad1=`cat /etc/sysconfig/network-scripts/ifcfg-
$kartu |grep BROADCAST`;
print "<table width=50% border=1 align=center
cellspacing=0 cellpadding=0
bordercolor=#000066>";
print "<tr>";
print "<td height=25 colspan=2 align=center
valign=top><font face=Verdana size=+1>
         Konfigurasi $kartu Baru</font></td>";
print "</tr>";
print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  $ip1";
   print "</td>";
  print "</tr>";
print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>  $mask1";
   print "</td>";
print "</tr>";
print "</table>";
#####################tabel eth1
print "<p>";
print "<p>";
print "<p>";
print "<p>";
```

```
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";
print "<table width=75% border=1 align=center
cellspacing=0 cellpadding=0
bordercolor=#000066>";
print "<tr>";
print "<td height=25 colspan=2 align=center
valign=top><font face=Verdana size=+1>
Restart Network Service</font></td>";
print "</tr>";
print "<tr>";
   print "<td width=599 valign=top
bgcolor=#6699CC>$restart";
   print "</td>";
print "</tr>";
print "</table>";
print "<p>";
print "<p>";
print "<p>";
print "<table border=0 bgcolor=#000066
width=100%>\n";
print"<form method=post action=/cgi-
bin/skywall/html/ipkonf.cgi>";
        print "<tr><td align=center><font
color#000000>\n";
        print "<input type=submit name=back
value=back>";
      print "</tr></td></table>";
print "</form>";
print "</table>";
print "</body>";
print $q->end_html();
```

MASQ

```
#!/usr/bin/perl

use CGI;
$q=new CGI;

#$sub=$q->param('subnet');
$int=$q->param('internal');
$ext=$q->param('external');

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";

system("echo '$ext' '$int'>/tmp/masq");
system("sed '/#dibawahini/r /tmp/masq'
/etc/shorewall/masq >/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/masq");
system("rm /tmp/hasil");
system("rm /tmp/masq");

##############################
system("./hitmasq");

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>";
   print "<tr>";
   print "<td height=25 colspan=2 align=center
valign=top ><font face=Verdana size=+1>";
   print "Masqureade</font></td>";
   print "</tr>";
print "</table>";

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>\n";
$baris3=`cat /tmp/jmlmasq`;
for ($i=1;$i<=$baris3-1;$i++)
{
$as3=`cat masqku |egrep 'Masq_no:$i '`;
  print "<tr>";
    print "<td width=599 valign=top
bgcolor=#6699CC>  $as3";
    print "</td>";
  print "</tr>";
}
print "</table>";

print "<table border=0 bgcolor=#000066
width=100%>\n";
print"<form method=post action=/cgi-
bin/skywall/html/masq.cgi>";
        print "<tr><td align=center><font
color#000000>\n";
        print "<input type=submit name=back
value=back>";
        print "</tr></td></table>";
print "</form>";
print "</table>";

print "</body>";
print $q->end_html();
```

RULES

```
#!/usr/bin/perl
use CGI;
$q=new CGI;

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";

$aksi=$q->param('action');
$asal=$q->param('source');
$tuj=$q->param('dest');
$ips=$q->param('ips');
$ipd=$q->param('ip');
$prot=$q->param('proto');
$port=$q->param('destport');

system("echo '$aksi' '$asal''$ips' '$tuj''$ipd' '$prot'
'$port'>/tmp/rules");
system("sed '/#dibawahini/r /tmp/rules'
/etc/shorewall/rules >/tmp/hasil");
system("cp -f /tmp/hasil /etc/shorewall/rules");
system("rm /tmp/hasil");
system("rm /tmp/dnat");

##########################################
######
system("./hitung");
print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>";
   print "<tr>";
   print "<td height=25 colspan=2 align=center
valign=top ><font face=Verdana size=+1>";
   print "Rules</font></td>";
   print "</tr>";
print "</table>";

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>\n";
$baris=`cat /tmp/jmlbaris`;
for ($i=1;$i<=$baris-1;$i++)
```

```
{

$as=`cat ruleku |egrep 'Rule_no:$i '`;
  print "<tr>";
    #print "<td width=75 valign=top
bgcolor=#6699CC>";
    #print "<input name=delete$i type=checkbox
value=cek$i><font size=3>Delete </font></td>";
    print "<td width=599 valign=top
bgcolor=#6699CC>  $as";
    print "</td>";
  print "</tr>";
}
print "</table>";
########################################
########
print "<table border=0 bgcolor=#000066
width=100%>\n";
print"<form method=post action=/cgi-
bin/skywall/html/rules.cgi>";
            print "<tr><td align=center><font
color#000000>\n";
            print "<input type=submit name=back
value=back>";
            print "</tr></td></table>";
print "</form>";
print "</table>";

print "</body>";
print $q->end_html();
```

ZONES

```
#!/usr/bin/perl

use CGI;
$q=new CGI;

print $q->header();
print $q->start_html(-title=>'SKYWALL');
print "<body bgcolor=#006699 link=red
vlink=#0000CC alink=#0000CC>";

$zon=$q->param('zone');
$name=$q->param('nama');
$com=$q->param('comment');
$kartu=$q->param('kartu');


system("echo '$name' '$zon' '$com'>/tmp/zones");
#system("echo '$name'          '$kartu'
'detect'>/tmp/int");
system("sed '/#dibawahini/r /tmp/zones'
/etc/shorewall/zones >/tmp/hasil1");
#system("sed '/#dibawahini/r /tmp/int'
/etc/shorewall/interfaces >/tmp/hasil2");
system("cp -f /tmp/hasil1 /etc/shorewall/zones");
#system("cp -f /tmp/hasil2 /etc/shorewall/interfaces");
system("rm /tmp/hasil1");
#system("rm /tmp/hasil2");
system("rm /tmp/int");

########################################
###
system("./hitzone");

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>";
   print "<tr>";
   print "<td height=25 colspan=2 align=center
valign=top ><font face=Verdana size=+1>";
   print "Zone</font></td>";
   print "</tr>";
print "</table>";

print "<table width=100% border=1 cellspacing=0
cellpadding=0 bordercolor=#000066>\n";
$baris=`cat /tmp/jmlzone`;
for ($i=1;$i<=$baris-1;$i++)
{
$as3=`cat zoneku |egrep 'Zone:$i '`;
  print "<tr>";
    print "<td width=599 valign=top
bgcolor=#6699CC>  $as3";
    print "</td>";
  print "</tr>";
}
print "</table>";
########################
print "<table border=0 bgcolor=#000066
width=100%>\n";
print"<form method=post action=/cgi-
bin/skywall/html/zone.cgi>";
            print "<tr><td align=center><font
color#000000>\n";
            print "<input type=submit name=back
value=back>";
            print "</tr></td></table>";
print "</form>";
print "</table>";

print "</body>";
print $q->end_html();
```

next on CD...