

IMPLEMENTASI MANAJEMEN BANDWIDTH INTERNET BERBASIS KUOTA DAN FILTERING DENGAN OS IPCOP

Mualim

Program Studi Teknik Informatika Fakultas Teknik Universitas Sultan Fatah (UNISFAT)
Jl. Diponegoro 1B Jogoloyo Demak Telpn (0291) 681024

Abstrak : Saat ini perkembangan teknologi informasi semakin pesat, sehingga kebutuhan akan informasi sangat diperlukan oleh masyarakat. Setiap masyarakat belum tentu mempunyai fasilitas internet, salah satu alternatifnya adalah dengan menggunakan jasa warnet. Warnet yang memiliki akses yang cepat dan stabil biasanya menjadi prioritas oleh pengguna. Warnet SMKHK memiliki fasilitas untuk browsing, chatting, printing, scanning, burning dan gaming. Pada dasarnya bandwidth adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari satu tempat ke tempat lain dalam satu waktu tertentu.

Namun, pengelolaan suatu bandwidth yang tidak optimal dan besarnya traffic yang dihasilkan oleh pengguna menyebabkan tidak stabilnya bandwidth yang diterima oleh pengguna. Dalam mengatur pembagian bandwidth sesuai kebutuhan, maka dengan ini dilakukan manajemen bandwidth internet berbasis kuota dan filtering web dengan menggunakan IPCop, sehingga dapat mengoptimalkan penggunaan bandwidth internet yang sangat terbatas

Kata kunci : Warnet Zamzami, *Bandwith*, *Filtering Web*, IPCop

PENDAHULUAN

Saat ini perkembangan teknologi informasi semakin pesat, sehingga kebutuhan akan informasi sangat diperlukan oleh masyarakat. Setiap masyarakat belum tentu mempunyai fasilitas internet, salah satu alternatifnya adalah dengan menggunakan jasa warnet. Warnet yang memiliki akses yang cepat dan stabil biasanya menjadi prioritas oleh pengguna, maka diperlukan suatu mekanisme pengaturan *bandwidth*. Pada dasarnya *bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari satu tempat ke tempat lain dalam satu waktu tertentu. Namun, pengelolaan suatu *bandwidth* yang tidak optimal dan

besarnya *traffic* yang dihasilkan oleh pengguna menyebabkan tidak stabilnya *bandwidth* yang diterima oleh pengguna. Hal yang sering menjadi penyebab tidak stabilnya *bandwidth* yang diterima oleh tiap pengguna adalah adanya permintaan dari pengguna untuk mengunduh file. Selain permasalahan pada bandwidth Internet, warnet juga memiliki masalah dalam pengaksesan internet, dimana pengguna bebas untuk mengakses situs-situs dewasa seperti situs porno dan judi.

TINJAUAN PUSTAKA

Bandwidth

Bandwidth adalah kapasitas atau daya tampung kabel Ethernet agar dapat dilewati *traffic* paket data dalam jumlah

tertentu.

Bandwidth juga bisa berarti jumlah konsumsi paket data per satuan waktu dinyatakan dengan satuan *bytes per second* [bps].

Bandwidth Internet disediakan oleh *Internet Service Provider* (ISP) dengan jumlah tertentu tergantung sewa pelanggan. Dengan *Quality of Service* (QoS) dapat diatur agar *user* tidak menghabiskan *bandwidth* yang disediakan oleh *provider*. Istilah *bandwidth* muncul dari bidang teknik elektro, dimana *bandwidth* mempresentasikan jarak keseluruhan atau jangkauan diantara sinyal tertinggi dan terendah pada kanal (*band*) komunikasi.

Pada dasarnya *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kapasitas, maka umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah perangkat meminta akses ke jaringan dan masa perangkat itu memberi izin untuk melakukan transmisi.

Manajemen Bandwidth

Istilah manajemen *bandwidth* sering dipertukarkan dengan istilah *traffic control*, yang dapat didefinisikan sebagai pengalokasian yang tepat dari suatu *bandwidth* untuk mendukung kebutuhan atau keperluan aplikasi atau suatu layanan jaringan. Manajemen *bandwidth* juga dapat dilakukan melalui beberapa aspek seperti alokasi waktu, pembatasan *qouta*, maupun autentikasi user.

Maksud dari manajemen *bandwidth* adalah bagaimana kita menerapkan permanajemenan atau pengaturan *bandwidth* dengan menggunakan sebuah komputer Linux.

Umumnya komputer Linux dapat digunakan sebagai *gateway / router* sehingga memungkinkan untuk mengatur *traffic* data atau memanajemenkan *bandwidth* dari *traffic* data yang melewati komputer linux tersebut untuk memberikan jaminan kualitas akses layanan *Internet* bagi komputer dalam jaringan lokal.

IPCop

IPCop adalah suatu distribusi Linux yang menyediakan fitur *simple-to-manage firewall* appliance berbasis perangkat keras PC. IPCOP juga

merupakan suatu *stateful firewall* dibuat berdasarkan pada Linux netfilter framework.

Distro ini awalnya dikembangkan oleh tim yang mengembangkan Smoothwall Linux firewall, perkembangan selanjutnya, proyek IPCop dikembangkan dengan bebas, dan saat ini sudah terpisah sepenuhnya. IPCop sangat simple, dan memiliki fitur *user-managed* untuk mekanisme update security-nya. Bahkan cenderung mudah dipahami untuk yang para pemula dan handal untuk yang sudah berpengalaman.

Fitur IPCop

Fitur yang terdapat pada IPCop, yaitu:

- 1) Aman, stabil dan merupakan Linux berbasis *firewall* yang sangat mudah dikonfigurasi.
- 2) Mudah untuk melakukan administrasi lewat *web* akses.
- 3) IPCop dapat menggunakan *DHCP IP address* dari ISP yang kita gunakan.
- 4) Dapat berfungsi sebagai *DHCP server* untuk memudahkan konfigurasi *internal network*.
- 5) Memiliki kemampuan sebagai *caching DNS proxy*, untuk membantu menambah kecepatan *query domain*

name.

- 6) Memiliki *web caching proxy*, untuk menambah
- 7) kecepatan akses *web*.
- 8) Sebagai *intrusion detection system* untuk mendeteksi serangan ke *internal network* kita.
- 9) Kemampuan untuk memisahkan *network*, konfigurasi *GREEN* untuk *internal network* yang aman, *network* terlindungi dari internet, konfigurasi *BLUE* untuk *network* dengan *wireless LAN* dan *DMZ* or *ORANGE* untuk *network* yang diperbolehkan diakses oleh public seperti *webserver/ mailserver*.
- 10) Fasilitas VPN yang digunakan untuk koneksi ke *internal network* dari *eksternal network* melalui internet secara aman karena telah ditingkatkan untuk *support x509 certificate*.
- 11) Memiliki *traffic shaping* untuk mengatur prioritas service seperti *web browsing*, FTP, telnet dan lain-lain sesuai keinginan.
- 12) Di bangun dengan *Pro police* untuk mencegah serangan pada semua aplikasi.
- 13) Memiliki pilihan konfigurasi kernel yang mengizinkan kita memilih

sesuai dengan keadaan yang kita inginkan.

ACL (Access Control List)

Berikut ini adalah konfigurasi yang ada di squid, selain sebagai *cache server*, squid yang memang bertindak sebagai *parent* untuk meminta objek dari kliennya dapat juga di konfigurasi untuk pengaturan hak akses lebih lanjut, untuk pertama kali yang dibicarakan adalah ACL (*Access Control List*), ACL sendiri terdiri dari beberapa tipe antara lain (Ashari, 2010):

- 1) Src : IP Address asal yang digunakan klien
- 2) Dst: IP Address tujuan yang diminta klien
- 3) Myip: IP Address local dimana klien terhubung
- 4) Srcdomain : Nama domain asal klien
- 5) Dstdomain: Nama domain tujuan klien
- 6) Srcdom_regex: Pencarian pola secara string dari nama domain asal klien
- 7) Dstdom_regex: Pencarian pola secara string dari nama domain tujuan klien
- 8) Time: Waktu dinyatakan dalam hari

dan jam

- 9) Proto: Protokol transfer (http, ftp, gopher)
- 10) Method: Metode permintaan http (get, post, connect)

Berikutnya adalah control list yang akan digunakan untuk mengatur control dari ACL, control list tersebut antara lain :

- 1) http_access : memperbolehkan akses http
- 2) icp_access : memperbolehkan peer untuk
- 3) mengirimkan icp untuk menquery object
- 4) miss_access : memperbolehkan klien meminta object
- 5) yang belum ada (*miss*) didalam cache.
- 6) No Cache : object yang diminta klien tidak perlu disimpan ke hardisk.
- 7) always_direct: permintaan yang di tangani langsung ke *server*.
- 8) never direct : permintaan yang di tangani secara tidak langsung ke *server*.

Delay Pools

Delay pools digunakan untuk menspesifikasi berapa jumlah *pool* yang digunakan untuk membatasi jumlah

bandwidth dari ACL.

Opsi ini akan dirangkaikan bersama opsi *delay_class* dan *delay_parameters* yang akan di bahas di bawah ini.

Delay Class

Delay class menspesifikasikan kelompok dari masing – masing pool yang telah didefinisikan pada opsi *delay* pools. Ada tiga *class* yang didukung Squid, antara lain :

- 1) Class 1: Semua akses dibatasi dengan single bucket, artinya hanya bisa mendefinisikan *overall bandwidth* untuk suatu ACL saja, tidak bisa mendefinisikan *bandwidth* dengan lebih mendetail .
- 2) Class 2: Semua akses dibatasi dengan *single aggregate* dengan dua parameter *bandwidth*. Parameter pertama mendefinisikan berapa *bandwidth* maksimal yang didapatkan ACL, parameter kedua mendefinisikan berapa *bandwidth overall* untuk ACL yang spesifik yang ada pada *network* tersebut .
- 3) Class 3: Kelompok yang definisi *bandwidth*-nya paling mendetail. Parameter pertama mendefinisikan berapa *bandwidth* maksimal yang didapatkan ACL, parameter kedua

mendefinisikan berapa *bandwidth* normal yang didapatkan ACL secara umum, dan parameter yang ketiga adalah mendefinisikan *bandwidth* yang didapatkan ACL jika mengakses ACL - ACL tertentu yang spesifik, misalnya *file* Squid[4]

Delay Parameters

Delay parameters menspesifikasikan rumus *bandwidth* yang akan didapatkan oleh ACL yang akan memasuki *delay_pool*. Misalnya ada *entry* berikut ini pada *delay_parameters*:

delay_parameters 1 -1/-1 2100/4000

Angka 1 berarti rumus ini berlaku untuk pool 1. Angka -1/-1 berarti *bandwidth* maksimal yang diberikan Squid adalah tidak terbatas untuk pool ini. Angka 2100/4000 berarti *bandwidth* yang didapatkan oleh ACL setelah masuk ke pool ini. Angka ini berada dalam kelipatan 8 b, sehingga untuk mendapatkan nilai *bandwidth* yang sebenarnya harus dikalikan delapan. Angka 2100 adalah *bandwidth* yang didapatkan ACL pada masa – masa normal. Jika dikalikan 8, maka *bandwidth* normal yang akan didapatkan ACL sekitar 18 Kbps. Angka 4000

adalah *bandwidth* maksimal yang didapatkan ACL pada masa-masa jalur sedang kosong. Jika dikalikan 8, maka *bandwidth* yang di dapat sekitar 32 Kbps

Delay acces

Delay access mendefinisikan ACL apa yang akan dimasukkan ke pool tertentu untuk mendapatkan perlambatan *bandwidth*. Bentuk umumnya adalah seperti ini:

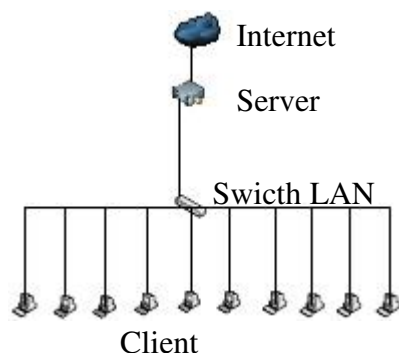
delay_access 1 allow labprog

Opsi di atas berarti kita memasukkan ACL labprog ke dalam pool 1

ANALISIS

Arsitektur sistem

Perancangan dari arsitektur sistem yang akan dibangun digambarkan sebagai berikut :



Gambar 1 : Arsitektur Sistem

Gambar di atas adalah rancangan desain dari sistem yang akan di buat. Server menggunakan IPCop. Untuk

menghubungkan Server ke internet menggunakan akses internet dengan kecepatan up to 1 Mbps. PC Client akan dihubungkan ke Server menggunakan switch.

Tahap pertama di dalam perancangan sistem ini adalah mengkonfiguraskan PC yang akan bertugas sebagai server.

Server ini akan melakukan tugasnya sebagai pembatasan kecepatan *download* ketika *file* yang di *download* melebihi 5 MB maka kecepatan *download* akan turun menjadi 10 KB/s. Ukuran ini diambil sesuai dengan kebutuhan *user* yang dapat dilihat pada hasil kuesioner. Server ini juga melakukan tugas menyaring situs dan judi, sehingga bila client ingin membuka situs porno dan judi maka akan terblokir.

Data	Jumlah	Keterangan
<i>Bandwidth</i>	1 Mbps = 1024 Kbps = 128 KBps	<i>Bandwidth</i> yang terdapat di warnet
<i>Bandwidth</i> optimal yang didapat <i>user</i>	64 KBps	Di dapat dengan rumus : $BW_{Optimal\ Client} = (BW_{total\ jaringan}) / 2$ $128\ ps / 2 = 64\ KBps$

User Requirement

Spesifikasi Kebutuhan dan Perangkat

Adapun kebutuhan perangkat keras (*hardware*) yang dibutuhkan dalam pengerjaan proyek akhir ini antara lain:

1. Spesifikasi *Server*

1 buah PC dengan spesifikasi minimal:

- a) Processor Intel Pentium 4
- b) Memory 128 MB
- c) Hard disk 40 GB

2. Spesifikasi *User*

1. PC/laptop dengan spesifikasi minimal:

- a) Procesor intel Pentium Core2 Duo
- b) Memory 1 GB
- c) Hard disk 80 GB
- d) 1 Fast Ethernet card 10/100 MBps

2. Perangkat tambahan

- a) Switch 16 port
- b) Kabel UTP yang sudah terpasang konektor

Spesifikasi Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak (*software*) yang dibutuhkan dalam pengerjaan proyek akhir ini adalah:

1. Server

- a) IPCOP OS versi 1.4.20

2. Operator

- a) GUI ipcop mode webbased

3. User

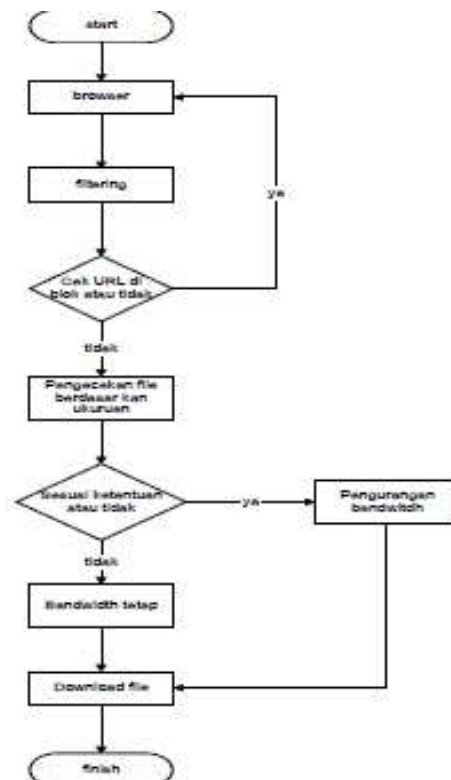
- a) Windows XP 2003

- b) Web browser (Firefox Mozilla)

- c) Web browser (opera)

- d) Web browser (Internet Explorer)

Perancangan Sistem yang Akan Berjalan



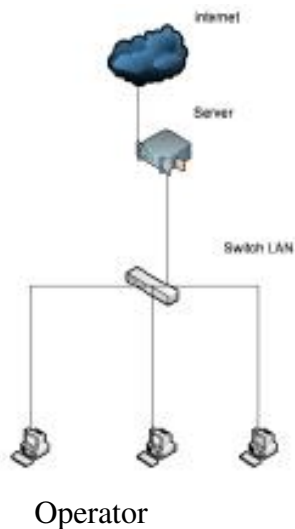
Gambar 2 Sistem yang Akan Berjalan

Pertama *user* masuk ke *browser* kemudian isi URL. URL akan masuk kedalam *filtering web*, apakah URL yang dimasukan dapat diakses atau diblok. Jika diblok, maka akan kembali ke proses awal *browser* dan URL yang dimasukan tidak bisa dibuka. Jika tidak diblok, maka URL yang dimasukan bisa

dibuka dan jika *user* ingin melakukan *download* dari URL tersebut maka akan masuk dalam pengecekan *file*. Apakah besaran *file download* sudah sesuai atau belum. Jika tidak sesuai, maka kecepatan *bandwidth* tetap. Tetapi jika besaran *file* sesuai dengan yang ditentukan maka kecepatan *bandwidth* akan menurun. Kemudian *user* dapat melakukan *download file*.

Perancangan Implementasi

Setelah melakukan observasi dan menentukan *user requirement* di dapatkan hasil untuk perancangan implementasi sebagai berikut:



Gambar 3 Perancangan Implementasi

Topologi yang akan di gunakan dalam pengujian menggunakan topologi star, yang akan terdiri atas satu PC/laptop

sebagai Server, 1 buah switch, 1 buah pc/laptop sebagai operator dan 2 buah PC/laptop sebagai *client*.

1. Skenario Pengujian

Pengujian sistem dalam proyek akhir ini dilakukan dengan menggunakan 3 buah PC/Laptop yang telah terhubung dengan Server, dengan Server yang telah terhubung dengan internet.

Gambar 3-5 menggambarkan tentang skenario dari pengujian sistem yang akan dilakukan pada proyek akhir ini, dimana server yang akan di hubungkan ke 3 buah PC/laptop melalui sebuah switch, setelah ketiga PC/laptop itu terhubung, maka akan di lakukan pengujian dengan cara mengakses situs porno dan judi.

Bila situs yang kita akses diblok maka pengujian *filtering URL* berhasil. Selanjutnya untuk pengujian pembatasan *download* dilakukan dengan cara *download file* dengan ukuran di atas 8 MB. Bila kecepatan turun ketika *file* mencapai ukuran 8 MB, maka pengujian pembatasan *download* berhasil.

Implementasi dan Pengujian

Pada bab ini akan dijelaskan langkah-langkah konfigurasi pada IPCop, yang

selanjutnya akan dilanjutkan dengan pengujian Sistem

Konfigurasi untuk menyaring URL pada IPCop

Pada tahap awal konfigurasi masuk pada *web based*, kemudian pilih menu *service-advance proxy*. Akan muncul tampilan seperti gambar dibawah, lalu tandai pada bagian *enabled on GREEN* untuk mengaktifkan *proxynya*. Selanjutnya beri tanda lagi pada *URL filter*, lalu *Save* dan *Restart*.



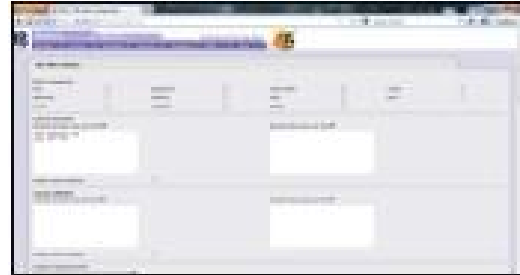
Gambar 4 Konfigurasi Proxy 2



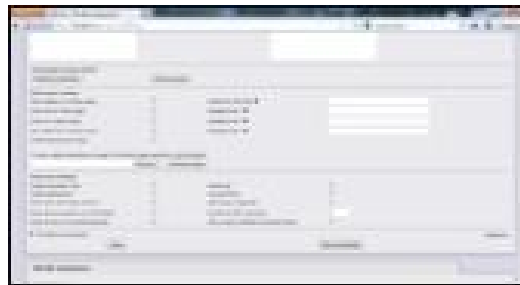
Gambar 5 Konfigurasi Proxy 1

Pada tahap berikut masuk menu *Service-URL Filter*. Lalmasukan URL yang akan disaring. Pada contoh gambar

dibawah URL yang akan disaring adalah www.redtube.com, www.mansion88.com Setelah selesai kemudian *Save* dan *Restart*.



Gambar 6 Konfigurasi URL Filter 1



Gambar 7 Konfigurasi URL Filter 2

Konfigurasi Manajemen Bandwidth Membatasi Download

Untuk mengkonfigurasi *shaping* manajemen *bandwidth* pindah ke *console*, IPCop, lalu ketikan *common nano/etc/squid/squid.conf*. Kemudian tambahkan *script* berikut:

```
# delay Pool
acl local src 192.168.0/24
delay_pools 1
delay_class 1 1
delay_parameters
```

```
1 10000/8000000
```

```
delay_access 1 allow local
```

```
delay_access 1 deny ALL
```

Keterangan script di atas, 10000 bps adalah maksimum *download* speed untuk *file* yang termasuk dalam rule (acl), sedangkan 16000 bps adalah maksimum ukuran file yang di-*download*

Pengujian

Pengujian URL Filter

Setelah konfigurasi di *Save and Restart*. Kita lakukan pengujian dengan memasukan URL yang akan di *filter* pada *web browser*. Situs judi yang diuji untuk *filtering* yaitu www.m888.com



Gambar 8 Filtering www.m888.com

Situs porno yang diuji untuk filtering yaitu www.redtube.com



Gambar 9 Filtering www.redtube.com

Pengujian Download

Setelah menambahkan *script* pada *console* IPCop. Lalu lakukan *restart* dengan perintah berikut `/usr/local/bin/restartsquid` pada *console* IPCop. Kemudian lakukan *download* untuk melakukan pengujian.

Download file yang ukurannya diatas 8 MB pada Browser



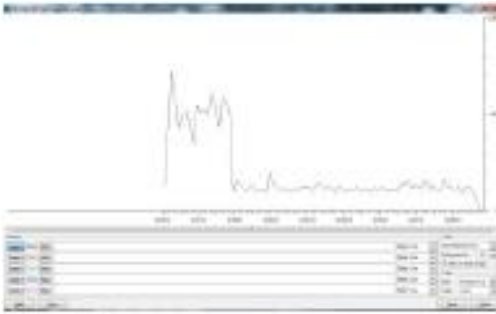
Gambar 10 Download file 1

Lakukan *capture* pada *interface* jaringan.



Gambar 11 Tampilan *capture* pada Wireshark

untuk mengetahui grafik dari data rate buka **IO graph** diwireshark pada **Statistics > IO graphs**.



Gambar 12 IO Graphs

Analisis paket data pada Wireshark



Gambar 13 Tampilan *capture* dan IO graphs

Dari penjelasan di atas, kecepatan *transfer* data akan mulai turun pada saat paket data mencapai ukuran 8 MB. Hal ini ditunjukkan dengan menurunnya kecepatan pada IO graph ketika besar paket pada tampilan *capture* wireshark sebesar 8 MB.

KESIMPULAN

Kesimpulan yang dapat ditarik dari proyek akhir ini adalah:

- a) Dengan IPCop sebagai manajemen *bandwidth* setiap *client* tidak berebut *bandwidth* dan

mendapatkan *bandwidth* yang sama sebesar 10 KB/s ketika ukuran *file* mencapai 8 MB.

- b) Dengan *add-on* web GUI IPCop seluruh situs porno dan judi yang ada pada web GUI itu dapat diblokir sehingga *client* tidak seenaknya membuka situs porno dan judi..

Saran

Beberapa poin yang dapat dijadikan sebagai saran diantaranya adalah:

- a) Diharapkan pada pengembangan baru di *Add-on Advance Proxy* server pada web based IPCop, dapat ditambahkan fungsi *shaping* untuk mengatasi agar admin bisa langsung mengkonfigurasi tanpa perlu menambahkan *script console* IPCop .
- b) Untuk penyusunan selanjutnya yang ingin menggunakan Operating Sistem IPCop, lebih baik gunakan juga fitur-fitur lain yang masih tersedia antara lain *QoS* untuk *bandwidth management*, E-mail Virus scanner, OpenVPN dan fitur lainnya yang masih bias dikembangkan.

- c) Untuk menghindari terjadinya gangguan dari luar dapat ditambahkan pengamanan pada sistem

DAFTAR PUSTAKA

- Ashari, A. (2010). *Linux System Administrator*. Informatika.
- Eichstaedt, Tom. (2012, january 2). *IPCop addon binaries*. Retrieved january 11, 2012, from wintermute website: <http://www.ipadd.de/index.html>
- Maryanto, D. (2001). *Optimasi Akses Internet Dengan SQUID*. PT Elex Media Komputindo.
- Oktavianus. (n.d.). *Parameter Delay Pools Di Dalam Squid Cache*. Retrieved Mei 25, 2011, from Forum Mikrotik: <http://www.forummikrotik.com>
- Roberts, S. (2009). *Information System: Now and Tomorrow*. Chicago: Adventure Press.
- Rokoko, J. (2005). *Pseudo-2D Hidden Markov Model*. New York: McGraw Hill.
- Sanjay, R. (2005). *Trik mengelola kuota Internet bersama dengan squid*. Jakarta: PT Elex Media

Komputindo.

- Santoso, B. (n.d.). *Manajemen Bandwidth Internet dan Intranet*. Retrieved May 15, 2011, from Budi Santoso: <http://www.budisantoso.com>
- Setio Dewo, E. (n.d.). *Bandwidth dan Throughput*. Retrieved July 1, 2011, from Bandwidth dan Throughput: <http://arieffirmanto.blog.upi.edu>
- The IPCop team. (2001). Retrieved december 3, 2011, from IPCop-The Bad Packets Stop Here: <http://www.ipcop.org/index.php>
- Router IPCOP murah (2010) . Mualim,