

Pengaturan Akses Internet dengan Pemanfaatan Proxy Server Menggunakan Squid

Nani Mintarsih¹

Witari Aryunani²

nanim@staff.gunadarma.ac.id

witari_aryunani@staff.gunadarma.ac.id

Abstrak

Dalam pemakaian Internet bersama-sama sering kali menimbulkan masalah, seperti tidak adanya pembagian *bandwidth* yang adil pada setiap penggunaanya, tidak adanya filter untuk mengakses situs manapun, dan sampai pemanfaatan Internet yang tidak berdasarkan haknya. Dengan menggunakan Squid pada Proxy Server maka dapat di atur konfigurasi manajemen berdasarkan penggunaan bandwidth, pemakaian waktu serta, penggunaan IP serta alamat internet dan lainnya. Pada aplikasi ini digunakan sistem operasi Linux OpenSUSE yang dirasakan sangat efektif dalam penggunaan pengaturan konfigurasi squid . Hasil yang didapat adalah dapat diaturnya penggunaan bandwidth kebutuhan tertentu sehingga penggunaan layanan internet yang tidak layak dapat di tekan.

Pendahuluan

Seiring terus berkembangnya jaringan Internet di Indonesia, pemanfaatan teknologi Internet pun sudah mulai merambah ke berbagai bidang. Karena Internet kini pandang bukan lagi barang mewah yang penggunaanya didominasi kelas atas. Internet sudah menjadi suatu kebutuhan bagi banyak kalangan. Sekarang tidak asing lagi bagi kita tersedianya sarana umum, sekolah, kampus, dan perkantoran yang menyediakan fasilitas *free* Internet setiap harinya. Hal itu dapat terjadi karena hanya bermodalkan satu *account* yang dikoneksikan ke sebuah jaringan komputer, maka kita sudah dapat menikmati koneksi Internet bersama-sama.

Pemanfaatan satu *account* yang dipakai secara bersamaan banyak terjadi di instansi-instansi pendidikan dan perkantoran. Hal tersebut dilakukan karena pemakaian Internet bersama-sama dapat menghemat anggaran pengeluaran untuk akses Internet. Dalam realita yang ada ternyata pemakaian Internet secara bersama-sama ini sering kali tidak mencapai nilai yang optimal. Contohnya dalam sebuah perusahaan yang memiliki koneksi Internet dalam 24 jam dan perusahaan tersebut sudah menerapkan pemakaian Internet secara bersama, ternyata malah menimbulkan masalah baru. Di mana ada seorang karyawannya sedang asyik mengakses Internet di sela-sela jam kerja, dengan mengabaikan tugas dan kewajibannya sebagai karyawan di perusahaan tersebut. Hal ini tentu secara tidak langsung akan mengganggu kinerja perusahaan. Memutuskan koneksi jaringan komputer pada komputer-komputer yang tidak berkepentingan dengan Internet sangat tidak mungkin karena setiap orang terhubung dengan *server*.

Oleh karena itu diperlukannya suatu manajemen agar Internet bersama dapat dinikmati oleh pengguna dengan adil berdasarkan waktu-waktu yang diperbolehkan mengakses Internet. Setiap pengguna memperoleh hak akses Internet sesuai dengan bidang pekerjaannya atau kebutuhan yang seharusnya diperoleh. Selain itu, orang yang tidak berhak memperoleh hak untuk menggunakan Internet juga dapat dicegah.

Pembahasan

Perancangan Jaringan

Media Transmisi

Dalam membangun sebuah jaringan komputer, memilih media transmisi tidak dapat dianggap remeh. Ada beberapa faktor yang perlu dipertimbangkan dalam memilih media transmisi, yaitu faktor keperluan, biaya, kemudahan pemasangan, perawatan, kehandalan, kecepatan dan jarak.

Secara umum terdapat dua pilihan media transmisi yaitu menggunakan kabel dan tanpa kabel. Media transmisi berupa kabel UTP adalah pilihan yang paling sesuai pada kondisi lokasi seperti ini. Jarak antar komputer pengguna yang tidak terlalu jauh sehingga masih terjangkau oleh kabel UTP. Selain harganya murah dan perawatannya mudah, proses instalasinya pun lebih mudah dibanding kabel-kabel jaringan yang lain.

Pemasangan kabel harus kita usahakan rapi dan sebaik mungkin, mengingat masalah kabel sangat sensitif terhadap kelancaran jaringan. Pemasangan kabel yang asal-asalan dapat menimbulkan masalah di kemudian hari. Misalkan kabel lecet atau terkelupas karena suatu hal, atau pemasangan konektor yang tidak baik.

Jaringan Client-Server

Untuk membangun sebuah *proxy server* dibutuhkan sebuah komputer yang berperan sebagai *server* untuk memberikan fasilitas kepada *workstation*. Karena jaringan Internet kali ini menggunakan *port ethernet* yang sudah ada maka, harus ada satu LAN Card lagi yang terpasang pada komputer *server*. Sehingga nantinya pada komputer *server* akan terdapat dua nomor IP yang terdiri dari satu nomor IP global (dari ISP) dan satu nomor IP lokal.

Proxy server memerlukan penyimpanan data sebagai *cache* yang harus disimpan di dalam komputer *server*. Jaringan *client-server* memiliki akses lebih tinggi karena penyedia fasilitas jaringan dan pengelolaannya dilakukan oleh satu komputer khusus yang tidak terbebani dengan tugas-tugas yang lain. Sistem

keamanan dan administrasi jaringan lebih baik karena tugas tersebut diberikan kepada seorang administrator jaringan.

Pemilihan Topologi

Pemilihan topologi jaringan kita lakukan untuk menentukan kebutuhan kabel bagi instalasi pengkabelan. Berdasarkan kondisi lokasi, digunakan topologi *star*. Karena kemudahan untuk menambah, mengurangi atau mendeteksi kerusakan jaringan yang ada. Dalam topologi *star* setiap komputer *client* terhubung langsung ke komputer *server*, sehingga *bandwidth* komunikasi dalam kabel semakin lebar dan akan meningkatkan kinerja jaringan secara keseluruhan.

Bila dilihat dari segi biaya, khusus untuk kasus ini, membangun jaringan komputer dengan topologi *star* memerlukan biaya lebih terjangkau dibandingkan dengan topologi lain.

Rancangan Kebutuhan

Tahapan yang sangat penting dalam membangun sebuah jaringan adalah identifikasi kebutuhan organisasi atau perusahaan. Layanan yang diberikan pada komputer *client* berupa Internet yang digunakan untuk akses Internet. Semua komputer terhubung sebagai satu jaringan yang berada dalam satu lokasi.

Tabel 1. Hasil identifikasi pengembangan jaringan

Keragaan	Keterangan
Jenis Layanan	Internet
Skalabilitas	Kecil, <i>workstation</i> < 25
Expandable	Ya
Lokasi	Satu lokasi, ruangan
Medium transmisi	Kabel
Besar <i>Bandwidth</i>	Up to 512 KBps
Ketersediaan perangkat keras	1 <i>server</i> dan 1 <i>switch</i>
Perangkat lunak jaringan	<i>Open Source</i> (Linux OpenSUSE 11.1)

SDM	SDM IT
-----	--------

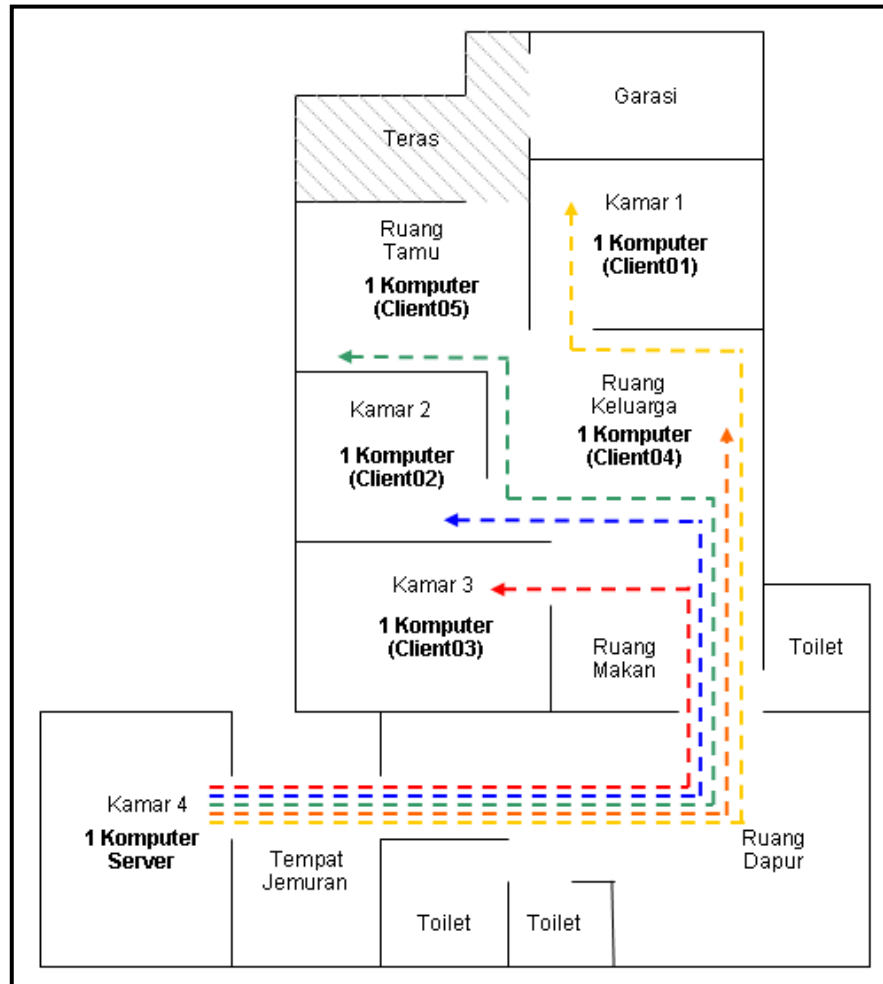
Perangkat keras yang diperlukan untuk membangun jaringan disajikan pada Tabel 2. Perangkat ini merupakan perangkat yang digunakan dalam melakukan implementasi penulisan ini.

Tabel 2. Perangkat keras yang digunakan

Perangkat keras	Jumlah	Spesifikasi
Komputer <i>Server</i>	1 unit	-AMD Athlon 64 3000+ 1,81Ghz -Memory 1,25 GB -Hard disk 40GB -VGA Card 256MB
Komputer <i>Client</i>	5 unit	Dengan spesifikasi yang beragam
LAN Card	2 unit	-D-Link DFE-528TX PCI Adapter -NVIDIA nForce Networking Controller
Switch	1 unit	-D-Link DES-1008D
UTP dan konektor RJ 45	secukupnya	Kabel UTP category 5e & RJ-45 ori
Tank crimping	1 unit	

Pada komputer *server*, ditambahkan satu buah LAN Card (*external*) yang ditanamkan pada *port* PCI di *motherboard*, hal ini karena LAN Card (*internal*) yang sudah terintegrasi di *motherboard* hanya memiliki satu *port ethernet*. Dan *port ethernet* tersebut sudah digunakan untuk koneksi Internet. Oleh karena itu diperlukan satu *port ethernet*, sehingga dibutuhkan LAN Card tambahan. Yang nantinya *port ethernet* tambahan tersebut akan digunakan untuk membentuk sebuah koneksi antara *server* dengan *client*. Terkecuali jika koneksi Internet yang dipakai menggunakan *port*, selain *port ethernet*, misalnya menggunakan *port* USB. Maka penambahan LAN Card tidak perlu dilakukan.

Design Jaringan

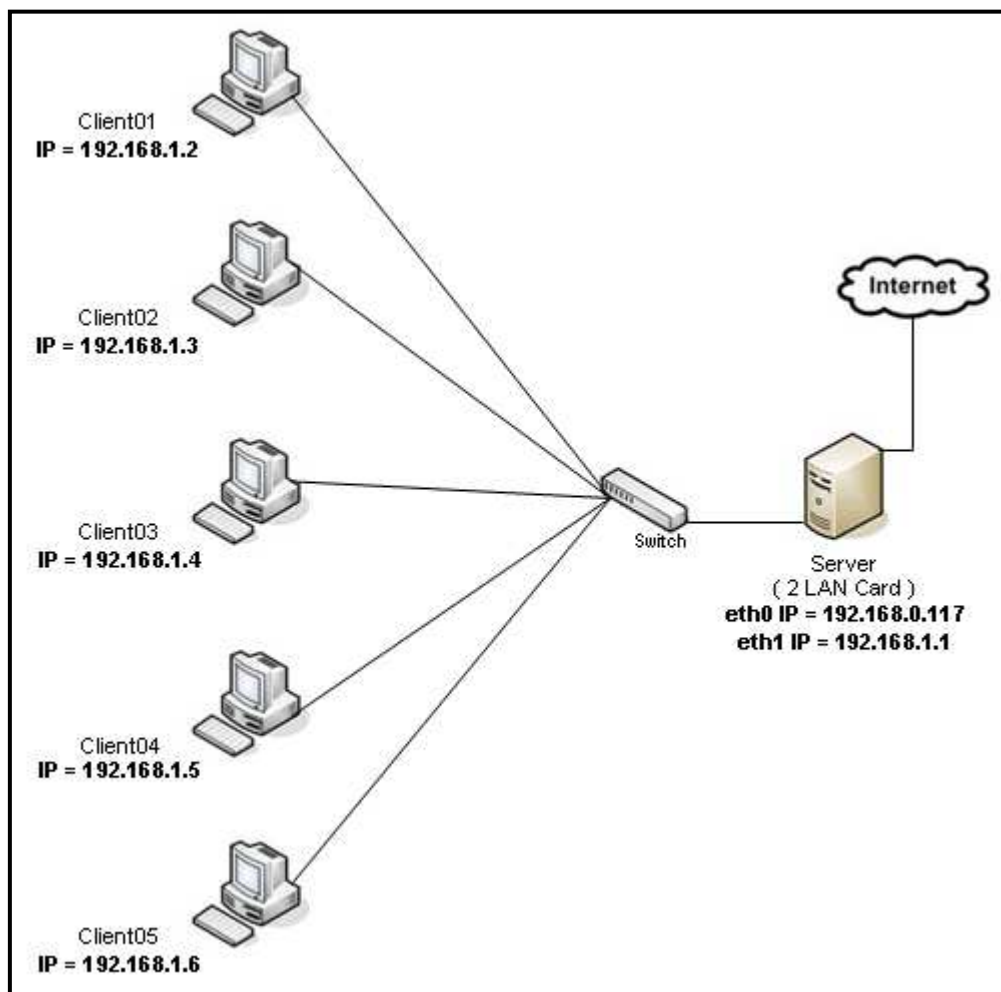


Gambar 1. Alur jaringan komputer *server* dengan *clien*

Design jaringan yang bagus dalam suatu sistem infrastruktur jaringan komputer merupakan suatu pondasi keberhasilan dari sistem komputer yang akan anda bangun. Tidak peduli sebagus apapun sistem komputer yang anda *design* kalau dibangun pada jaringan komputer yang tidak bagus maka sistem komputer anda tidak akan berjalan dengan efisien dikarenakan terhambatnya jaringan komputer yang ada.

Dimisalkan, jalanan di Jakarta ataupun di kota-kota besar yang macet dikarenakan membludaknya jumlah kendaraan bermotor pada jam sibuk, akan bisa butuh waktu jauh lebih lama buat anda untuk sampai ke kantor dibanding jika

jalan lancar di hari libur. Seperti juga jalan raya, suatu jaringan komputer mempunyai keterbatasan kapasitas dalam mentransmisikan data. Jika jumlah piranti di dalam jaringan bertambah, maka kemacetan akan bertambah juga yang pada akibatnya mempengaruhi kinerja dari jaringan. Karenanya, *design* jaringan yang bagus adalah sangat penting sekali untuk mengurangi kemacetan jaringan dan juga menjaga kinerja dari jaringan komputer anda dalam kondisi yang tinggi.



Gambar 2. *Design jaringan*

Setelah menentukan topologi yang akan dibuat, tahapan selanjutnya ialah membangunnya menjadi sebuah jaringan komputer yang nyata. Yang pertama penulis lakukan ialah dengan menggunakan kabel UTP, penulis mengkoneksikan komputer *server* ke *switch*, kemudian dari *switch*, dikoneksikan ke *client* yang

ada. Semua menggunakan kabel UTP sebagai media transmisi dan juga menggunakan RJ-45 sebagai konektornya. Maka terbangunlah jaringan komputer sederhana dengan topologi *star*

Komputer *server* dan *switch* ditempatkan pada ruang *server* dalam hal ini yang dimaksud adalah kamar empat. Hal tersebut bertujuan agar mempermudah kontrol dan kerja administrator dalam mengelola jaringan komputer tersebut, yang bertindak sebagai administrator ada penulis sendiri. Sedangkan komputer *client* ditempatkan pada ruang kamar, ruang tamu dan ruang keluarga.

Nantinya dengan menggunakan koneksi Internet yang berasal dari ISP RTRW Net yang biasa dipakai penulis, koneksi Internet tersebut akan di *sharing* ke *client-client*, dan tentunya akan dimanajemen terlebih dahulu, karena itulah tujuan dari penulisan ilmiah ini.

Implementasi

Konfigurasi Network

Protokol yang paling populer yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*). Jaringan dibentuk menggunakan protokol TCP/IP. *Network* TCP/IP dibangun menggunakan prinsip *packet switching*. Data yang dikirim ke *host* lain dibagi-bagi menjadi paket-paket dengan ukuran *byte* tertentu. Tiap-tiap paket berisi *header* yang salah satu bagiannya adalah alamat tujuan dari paket tersebut. Alamat tujuan dikenal dengan nama IP *address*. Setiap *host* pada jaringan TCP/IP dikenal melalui IP *address*-nya. Setiap *host* mempunyai IP *address* yang unik dan alamat IP tidak tergantung pada antarmuka jaringan (NIC) yang digunakan atau sistem operasi yang digunakan. Sesuai dengan medium transmisi yang digunakan yaitu kabel UTP, maka sebagai antarmuka jaringan akan menggunakan NIC berjenis *ethernet*. Harus dipastikan bahwa NIC yang digunakan sudah ter-*instal* dan dapat dikenali oleh sistem Linux. Komputer *server* memerlukan dua buah *ethernet* yang akan digunakan untuk berhubungan dengan antena grid dan *switch*. Untuk *ethernet* yang terhubung ke Internet diberikan alamat IP 192.168.0.117, alamat IP tersebut merupakan IP statis ketetapan dari

ISP RTRW Net yang penulis gunakan. Dan *ethernet* yang terhubung dengan *switch* diberikan alamat IP 192.168.1.1.

Mendeteksi NIC

Pada umumnya Linux yang sudah ada sekarang termasuk OpenSUSE sudah mampu mendeteksi keberadaan *ethernet card* dan sekaligus menginstallkan *driver*-nya jika terdapat dalam paket CD/DVD *installer*. *Ethernet card* nantinya akan dikenal sebagai eth0, eth1, eth2 dan seterusnya. Untuk mengetahui *ethernet* sudah aktif atau belum maka kita dapat menggunakan perintah berikut pada terminal.

```
masteguh:~ # ifconfig
```

Maka perintah di atas akan menampilkan informasi sebagai berikut.

```
eth0      Link encap:Ethernet  HWaddr 00:14:85:15:E5:AE
          inet addr:192.168.0.117  Bcast:192.168.0.255
          Mask:255.255.255.0
          inet6 addr: fe80::214:85ff:fe15:e5ae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:219 errors:0 dropped:0 overruns:0 frame:0
          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17807 (17.3 Kb)  TX bytes:9253 (9.0 Kb)
          Interrupt:21

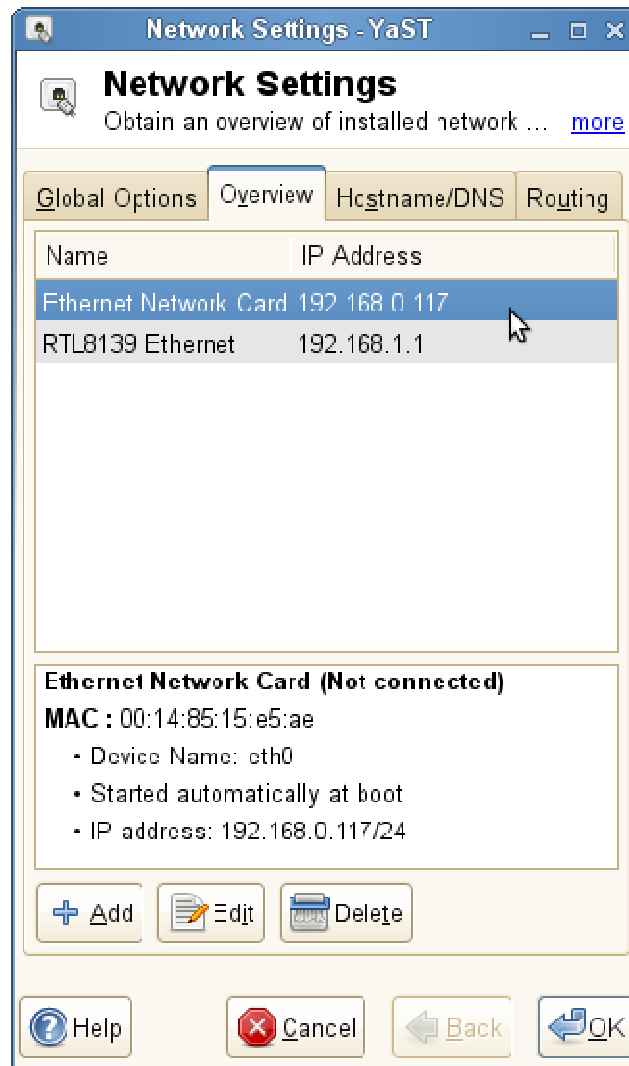
eth1      Link encap:Ethernet  HWaddr 00:24:01:62:26:21
          inet addr:192.168.1.1  Bcast:192.168.1.255
          Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:19 Base address:0x8000
```

Pesan yang tampil menunjukan bahwa terdapat dua buah NIC yang dikenal oleh sistem operasi Linux OpenSUSE. Yaitu eth0 yang memiliki *mac address* 00:14:85:15:E5:AE dan eth1 yang memiliki *mac address* 00:24:01:62:26:21.

Konfigurasi IP Address

Pemberian alamat IP *address* pada komputer *server* dilakukan secara manual (statically) dan dilakukan dengan *mode* GUI. Berikut ini ialah

tahap-tahapnya, YaST >> *Network Settings* >> pilih tab *Overview*. Masukkan IP *address* yang ingin digunakan dengan memilih *mode static* klik tombol *Edit* untuk dapat mengkonfigurasi.



Gambar 3. *Setting IP address* pada Linux OpenSUSE

Kedua *ethernet* diberikan alamat IP berdasarkan tabel berikut:

Tabel .3 Alamat IP *server*

<i>Ethernet</i>	eth0	eth1
IP <i>address</i>	192.168.0.117	192.168.1.1
netmask	255.255.255.0	255.255.255.0
gateway	192.168.0.1	

Kedua *ethernet card* di atas harus diberikan alamat jaringan yang berbeda, jika keduanya memiliki alamat jaringan yang sama maka tidak bisa terhubung. Untuk memastikan apakah *ethernet card* sudah mempunyai alamat IP atau belum dapat diketahui dengan menggunakan perintah berikut.

```
masteguh:~ # ifconfig eth0
masteguh:~ # ifconfig eth1
```

Konfigurasi *Proxy*

Proxy server memiliki banyak kemampuan dalam mengelola jaringan di antaranya dapat memanajemen *bandwidth*, menghemat *bandwidth*, meningkatkan keamanan dan mempercepat *web surfing*. *Proxy server* adalah *server* yang berfungsi untuk menyimpan sementara *file html* dari *server* lain, sehingga mempercepat akses Internet untuk alamat-alamat yang sudah pernah diakses sebelumnya.

Instalasi Squid

Squid merupakan *software proxy server* yang bersifat *open source* yang sangat terkenal. Prinsip kerja Squid yaitu menyimpan data suatu site kedalam *cache* sehingga untuk menampilkan site yang sama tinggal mengambil dari *cache*-nya.

Ketika menginstal Linux OpenSUSE, secara *file* biasanya Squid belum terinstal. Untuk menginstal Squid dapat dilakukan dengan dua mode yaitu mode GUI dan mode *command*. Jika melalui mode GUI tahapannya ialah, YaST → *Managemen Software* → *searching* kata Squid, kemudian pilih dan klik tombol *Install*. Dan berikut ini merupakan perintah untuk *mode comman*.

```
masteguh:~ # zypper install Squid
```

Pada saat meng-*instal*, baik dengan *mode* GUI maupun *mode command*. Komputer tersebut harus sudah terhubung dengan koneksi Internet. Sehingga paket yang di-*instal* diambil langsung dari penyedia repositori OpenSUSE. Penulis menggunakan Squid versi 3.0 STABLE10, versi paling akhir di saat penulis membuat penulisan ilmiah ini.

Manajemen Bandiwdth dengan Memodifikasi Konfigurasi Sqiud

Konfigurasi Squid dilakukan dengan cara mengedit *file* yang terdapat pada direktori `/etc/squid/squid.conf` dengan menggunakan editor gedit atau menggunakan editor teks yang lainnya yang terdapat pada Linux. Sebelum melakukan perubahan sebaiknya *file* ini di-*backup* terlebih dahulu untukantisipasi jika terjadi kerusakan. Berikut perintah di Linux dalam mem-*backup* suatu *file*. Jadi cara kerjanya ialah membuat duplikat tetapi dengan namanya berbeda dari *file* aslinya.

```
masteguh:~ # cp /etc/squid/squid.conf /etc/squid/squid.conf.copy
```

Setelah berhasil di-*backup*, maka kita dapat segera memulai memodifikasi konfigurasi dengan menggunakan perintah berikut.

```
masteguh:~ # gedit /etc/squid/squid.conf
```

Maka akan tampak tampilan seperti di bawah ini.

```
#      WELCOME TO SQUID 3.0.STABLE10
#      -----
#      This is the default Squid configuration file. You may wish
#      to look at the Squid home page (http://www.Squid-cache.org/)
#      for the FAQ and other documentation.
#
#      The default Squid config file shows what the defaults for
#      various options happen to be. If you don't need to change the
#      default, you shouldn't uncomment the line. Doing so may cause
#      run-time problems. In some cases "none" refers to no default
#      setting at all, while in other cases it refers to a valid
#      option - the comments for that keyword indicate if this is the
#      ^^^
```

Tampilan di atas merupakan hanya potongan bagian paling atas dari *file* konfigurasi Squid (*squid.conf*). Karena panjangnya *listing* dan mengingat tidak semua konfigurasi di modifikasi, maka akan sangat tidak efektif apabila penulis tampilkan semua *listing* konfigurasinya di pembahasan ini. Kemudian pada halaman selanjutnya, berisi *listing* yang terdiri dari tag-tag konfigurasi. Tag-tag konfigurasi tersebutlah yang nantinya akan penulis modifikasi sehingga akan tercapai tujuan dalam manajemen *bandwidth*. Untuk kita ketahui bersama, hanya *listing* yang tidak diawali tanda pagar (#) saja yang akan aktif. Jadi, apabila ada tag yang diawali dengan tanda pagar maka tag tersebut tidak aktif.

Hak Akses Dengan IP tertentu

Pada subbab ini kita akan membahas bagaimana konfigurasi di Squid untuk memberikan hak akses pada IP tertentu. Penulis akan membuat konfigurasi di Squid hanya komputer client01, komputer client02, komputer client03, komputer client04, dan komputer client05 yang bisa menikmati akses Internet. Sedangkan jika ada komputer lain yang terhubung ke jaringan komputer, maka tidak diizinkan untuk mendapatkan hak akses Internet.

- Komputer client01, nomor IP *address* komputernya adalah 192.168.1.2
- Komputer client02, nomor IP *address* komputernya adalah 192.168.1.3
- Komputer client03, nomor IP *address* komputernya adalah 192.168.1.4
- Komputer client04, nomor IP *address* komputernya adalah 192.168.1.5
- Komputer client05, nomor IP *address* komputernya adalah 192.168.1.6

Agar dapat diimplementasikan, perlu dilakukan perubahan pada konfigurasi Squid yang terletak di direktori */etc/squid* di komputer *server*. Perhatikan potongan *listing* konfigurasi di bawah ini. Penambahan yang dilakukan penulis hanya yang dicetak tebal.

```

...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255

...
...
# TAG: http_access
...
...
http_access allow client01
http_access allow client02
http_access allow client03
http_access allow client04
http_access allow client05
http_access deny all
...
...
...

```

Dengan *setting* konfigurasi seperti di atas, maka hanya komputer client01, komputer client02, komputer client03, komputer client04, dan komputer client05 saja yang dapat memiliki akses Internet. Hal ini dimungkinkan karena tag *http_access* hanya memperbolehkan (*allow*) *IP address* ke pengguna di atas. Sedangkan jika terdapat *IP address* lain, maka akan ditolak (*deny*) dan tidak akan bisa memiliki hak akses Internet.

Namun pada suatu saat karena alasan tertentu, dua komputer dari kelima komputer *client* yang ada tidak diizinkan untuk menggunakan akses Internet. Memutuskan kedua koneksi jaringan kedua komputer tersebut merupakan bukan solusi yang tepat, karena tentu saja akan menyulitkan interaksi data atau penyimpanan *file-file* ke *server*, antara *client* dengan *server*. Komputer *clint* yang hak aksesnya dimatikan yaitu client02 dan client05. Agar memenuhi ketentuan tersebut, perubahan konfigurasi Squid dapat dilakukan seperti di bawah ini.

```

...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255

...
...
# TAG: http_access
...
...

http_access allow client01
http_access deny client02
http_access allow client03
http_access allow client04
http_access deny client05
http_access deny all
...
...

```

Dengan begitu, hak akses Internet hanya dapat digunakan oleh komputer client01, komputer client03, dan komputer client04 saja. Hal ini dimungkinkan karena *http_access* hanya memperbolehkan (*allow*) komputer-komputer tersebut. Sedangkan komputer komputer client01, dan komputer client05 akan ditolak (*deny*).

Hak Akses Internet pada Jam Tertentu

Subbab ini akan membahas bagaimana membatasi hak akses Internet pada jam-jam tertentu saja. Kali ini penulis ingin membatasi hak akses pada masing-masing *client*. Akses Internet komputer client01 hanya bisa pada pukul 06.00 sampai 09.00, akses Internet komputer client02 hanya bisa pada pukul 09.00 sampai 12.00, akses Internet komputer client03 hanya bisa pada pukul 12.00 sampai 15.00, akses Internet komputer client04 hanya bisa pada pukul 15.00 sampai 18.00, akses Internet komputer client05 hanya bisa pada pukul 18.00 sampai 21.00. Untuk itu diperlukan tambahan *Access Control List* (ACL) yang membatasi pemakaian pada jam tertentu saja.

```

...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl jam-client01 time 06:00-09:00
acl jam-client02 time 09:00-12:00
acl jam-client03 time 12:00-15:00
acl jam-client04 time 15:00-18:00
acl jam-client05 time 18:00-21:00
...
...
# TAG: http_access
...
...
http_access allow jam-client01 client01
http_access allow jam-client02 client02
http_access allow jam-client03 client03
http_access allow jam-client04 client04
http_access allow jam-client05 client05
http_access deny all
...
...

```

Dengan memanfaatkan perintah tambahan *time* pada ACL, maka koneksi Internet hanya dapat digunakan pada komputer *client* dengan waktu yang sudah ditentukan. Hal itu dimungkinkan karena *http_access* hanya memperbolehkan (*allow*) komputer *client01* pada saat yang diatur dalam *jam-client01*, begitu juga dengan komputer *client* lainnya.

Hak Akses Internet pada Hari Tertentu

Hak akses Internet juga dapat diatur penggunaannya berdasarkan hari tertentu. Hal ini akan penulis coba implementasikan pada jaringan komputer di rumah penulis. Apabila komputer *client01* hanya dapat digunakan Senin dan Selasa, komputer *client02* hanya dapat digunakan Selasa dan Rabu, komputer *client03* hanya dapat digunakan Rabu dan Kamis, komputer *client04* hanya dapat digunakan Kamis dan Jumat, dan komputer *client05* hanya dapat digunakan Jumat dan Sabtu. Maka konfigurasi Squid dapat diatur seperti di bawah ini.


```

...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl hari-client01 time MT
acl hari-client02 time TW
acl hari-client03 time WH
acl hari-client04 time HF
acl hari-client05 time FS
...
...
# TAG: http_access
...
...
http_access allow hari-client01 client01
http_access allow hari-client02 client02
http_access allow hari-client03 client03
http_access allow hari-client04 client04
http_access allow hari-client05 client05
http_access deny all
...
...

```

Kombinasi huruf MT pada ACL *hari-client01*, merupakan singkatan hari untuk penulisan *Access Control*. Kita dapat melihat arti dari masing-masing singkatan pada tabel di bawah ini.

Tabel 4. Singkatan nama hari

Singkatan	Arti	Bahasa Indonesia
S	<i>Sunday</i>	Minggu
M	<i>Monday</i>	Senin
T	<i>Tuesday</i>	Selasa
W	<i>Wednesday</i>	Rabu
H	<i>Thursday</i>	Kamis
F	<i>Friday</i>	Jum'at
A	<i>Saturday</i>	Sabtu

Jika konfigurasi sudah diubah seperti contoh di atas, maka hak akses Internet komputer *client01* hanya dapat digunakan pada hari Senin, dan Selasa (MT), begitu juga dengan ruangan lainnya. Akses Internetnya hanya dapat dilakukan pada hari-hari yang sudah ditentukan pada konfigurasi Squid. Hal ini dimungkinkan karena *http_access* hanya memperbolehkan (*allow*) komputer

client01 pada saat yang telah diatur dalam *hari-client01*. Di luar waktu tersebut, akses Internet komputer di komputer client01 akan ditolak (*deny*).

Hak Akses Internet pada Hari dan Jam Tertentu

Ternyata pada Squid, memungkinkan diadakannya kombinasi *access control* yang akan menghasilkan pembatasan untuk hari dan jam sekaligus. Nantinya hak akses Internet dapat dibatasi dengan lebih spesifik. Pada konfigurasi sebelumnya, yang mana komputer client01 akses Internetnya hanya bisa pada hari Senin, dan Selasa kali ini akan dibatasi juga penggunaannya pada jam tertentu saja, yaitu pukul 06.00 sampai dengan pukul 09.00.

Akses Internet komputer client02 hanya dapat digunakan pada hari Selasa dan Rabu dengan waktu penggunaan hanya pada pukul 09.00 sampai dengan 12.00, akses Internet komputer client03 hanya dapat digunakan pada hari Rabu dan Kamis dengan waktu penggunaan hanya pada pukul 12.00 sampai dengan 15.00, akses Internet komputer client04 hanya dapat digunakan pada hari Kamis dan Jumat dengan waktu penggunaan hanya pada pukul 15.00 sampai dengan 18.00, akses Internet komputer client05 hanya dapat digunakan pada hari Jumat dan Sabtu dengan waktu penggunaan hanya pada pukul 18.00 sampai dengan 21.00. Maka konfigurasi Squid dapat diatur seperti di bawah ini.

```
...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl waktu-client01 time MT 06:00-09:00
acl waktu-client02 time TW 09:00-12:00
acl waktu-client03 time WH 12:00-15:00
acl waktu-client04 time HF 15:00-18:00
acl waktu-client05 time FS 18:00-21:00

...
...
# TAG: http_access
...
...
http_access allow waktu-client01 client01
http_access allow waktu-client02 client02
http_access allow waktu-client03 client03
http_access allow waktu-client04 client04
http_access allow waktu-client05 client05
http_access deny all
...
...
```

Hasil dari konfigurasi Squid di atas ialah hak akses Internet hanya dapat digunakan komputer client01 pada hari Senin, dan Selasa (MT) dengan jam akses pada pukul 06.00-09.00, sama halnya dengan komputer *client* lainnya, hak akses hanya akan didapatkan pada hari dan jam yang sudah ditentukan pada konfigurasi di atas.

Hal ini dimungkinkan karena *http_access* hanya memperbolehkan (*allow*) penggunaan komputer client01 pada saat yang telah diatur dalam *waktu-client01*. Di luar waktu tersebut, akses Internet pada komputer client01, komputer client02, komputer client03, komputer client04, dan komputer client05 tersebut akan ditolak (*deny*).

Hak Akses dengan Pembatasan Alamat Internet Tertentu

Salah satu cara untuk mengelola hak akses Internet secara optimal adalah dengan membatasi akses ke alamat Internet tertentu. Beberapa situs yang dianggap porno, menampilkan kekerasan, atau berisi material yang tidak layak akan diblokir. Alamat-alamat tersebut akan ditolak untuk ditampilkan meskipun pengguna memintanya. Konfigurasi untuk menangani penutupan akses alamat-alamat yang diblokir adalah sebagai berikut.

```
...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl blockdomain dstdomain -i "/etc/squid/domain-terlarang.txt
...
...
# TAG: http_access
...
...
http_access deny blockdomain
http_access allow client01
http_access allow client02
http_access allow client03
http_access allow client04
http_access allow client05
http_access deny all
...
...
```

Pada konfigurasi kali ini, kita diharuskan membuat *file domain-terlarang.txt* yang disimpan di direktori */etc/squid*. *File* ini berisi alamat-alamat Internet (*domain*) yang tidak dapat diakses karena alasan-alasan tertentu. *File* tersebut dapat menggunakan *text editor* vi, gedit, atau lainnya. Berikut ini contoh isi *file domain-terlarang.txt*.

```
.17tahun.com
.playboy.com
.nude.com
.hardcore.com
.xxx.cm
.muffia.com
```

Kita dapat menambahkan daftar tersebut sesuai dengan kondisi lingkungan kita. Semakin banyak alamat Internet yang kita masukkan, semakin banyak pula pengurangan kuota pemakaian untuk alamat-alamat yang tidak perlu. Jika ada permintaan menuju alamat tertentu yang termasuk dalam *domain-terlarang.txt* maka *browser* akan menampilkan pesan penolakan

Hak Akses dengan Pembatasan Kata Tertentu

Squid juga dapat menolak permintaan dari *client* apabila situs yang dikunjungi mengandung kata-kata yang tidak diinginkan. Cara yang digunakan hampir mirip dengan cara menyaring situs berdasarkan *domain*. Berikut ini konfigurasinya.

```
...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl blockkata url_regex-i "/etc/squid/kata-terlarang.txt
...
...
# TAG: http_access
...
...
http_access deny blockkata
http_access allow client01
http_access allow client02
http_access allow client03
http_access allow client04
http_access allow client05
http_access deny all
...
...
```

Sama seperti subbab sebelumnya, kali ini kita diharuskan membuat sebuah *file kata-terlarang.txt* yang berisi kata-kata yang tidak diperbolehkan untuk diakses. *File* tersebut harus diletakkan di direktori */etc/squid*. Berikut ini contoh dari isi *file kata-terlarang.txt*.

```
adult
sex
xxx
homo
porn
porno
lesbian
bokep
```

Hak Akses dengan Pembatasan IP Address Tertentu

Pada subbab sebelumnya penulis sudah melakukan pembatasan hak akses berdasarkan alamat situs atau *domain* tertentu. Ternyata hal tersebut tidak cukup efektif, karena apabila pengguna mengetahui alamat IP *address* dari situs yang sudah dilarang. Maka pengguna tersebut dapat dengan mudahnya mengakses situs tersebut dengan cara mengetikkan alamat IP *address* situs tertentu pada *web browser*. Guna mengantisipasinya diperlukan penyaringan berdasarkan IP *address*. Buka *file /etc/squid/squid.conf* dengan menambahkan baris di bawah ini.

```
...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
acl blockip dst -i "/etc/squid/ip-terlarang.txt
...
...
# TAG: http_access
...
...
http_access deny blockip
http_access allow client01
http_access allow client02
http_access allow client03
http_access allow client04
http_access allow client05
http_access deny all
...
...
```

Sama seperti hak akses berdasarkan alamat dan kata, kita juga diharuskan membuat sebuah *file ip-terlarang.txt* yang berisi *IP address* yang tidak diperbolehkan untuk diakses. *File* tersebut harus diletakkan di direktori */etc/squid*. Berikut ini contoh dari isi *file ip-terlarang.txt*.

```
115.124.65.234
209.247.228.203
116.145.45.45
206.251.29.35
202.145.6.214
```

Konfigurasi *Transparent Proxy*

Fungsi dari *transparent proxy* adalah untuk menjamin semua pengguna yang mengakses Internet selalu melewati *Squid proxy server*. Semua pengguna secara tidak langsung dipaksa untuk melewati *proxy server* setiap mengakses Internet. Cara kerja dari *transparent proxy* yaitu membelokkan pengiriman paket data baik dari luar maupun dari dalam jaringan yang melalui *port 80* untuk diarahkan lebih dahulu ke *proxy server*. Tambahkan perintah *transparent* pada konfigurasi *Squid*. Dengan cara ini, maka *client* akan membuat *proxy automatic* dan kita tidak perlu untuk men-*setting* *IP proxy* di *web browser* pada komputer-komputer *client*.

```
...
...
# ACCESS CONTROLS
...
...
acl client01 src 192.168.1.2/255.255.255.255
acl client02 src 192.168.1.3/255.255.255.255
acl client03 src 192.168.1.4/255.255.255.255
acl client04 src 192.168.1.5/255.255.255.255
acl client05 src 192.168.1.6/255.255.255.255
...
...
# TAG: http_access
...
...
http_access allow client01
http_access allow client02
http_access allow client03
http_access allow client04
http_access allow client05
http_access deny all
...
...
```

Pada listing konfigurasi di atas, selain menambahkan kata *transparent*, penulis juga menambahkan perintah *cache_mgr* dan *visible_hostname*, sebagai identitas dan contact *administrator newtwok* yang bisa dihubungi jika ada pesan *error* yang tampil dikarenakan pembatasan hak akses.

Hak Akses dengan Pembatasan Bandwidth

Ide pembatasan *bandwidth* pada Squid sebetulnya berasal dari Universitas Western Australia yang ingin membatasi biaya dan kepadatan lalu lintas jaringan ke Internet untuk mahasiswa, tetapi tidak mempengaruhi akses Internet bagi staf dan akses lokal di dalam universitas. Jika pada Squid versi 1.0 kemampuan tersebut disebut dengan DELAY_HACK maka mulai versi 2 kemampuan ini disebut dengan delay pools.

Pada jaringan yang penulis buat, penulis akan mengimplementasikan pembatasan *bandwidth* dalam penggunaan akses Internet di komputer *client* Namun sebelum implementasi, kita harus mempelajari pengaturan konfigurasi minimal pada delay pools.

```
...  
...  
...  
acl <nama_acl> scr <nomor_ip>  
delay_pools <jumlah_pengaturan>  
delay_class <no_urut_pengaturan> <pilihan_pool_1-3>  
delay_access <no_urut_pengaturan> allow <nama_acl>  
delay_parameters <no_urut_pengaturan> <max_bandwidth>  
...  
...
```

Diatas merupakan format dasar dalam melakukan konfigurasi di delay pools, penulis akan mengimplementasikannya ke jaringan komputer yang ada. *Bandwidth* dari ISP yang diterima penulis adalah 512 KBps. Penulis ingin membatasi *bandwidth* untuk setiap *client* maksimum sebesar 64 Kilobyte per detik. Sebelumnya, penulis harus mendefinisikan ACL yang akan menerima pengaturan tersebut, yaitu IP *address* komputer dengan awalan 192.168.1.

```
...  
...  
acl user64kbps src 192.168.1.0/255.255.255.255  
...  
...
```

Selanjutnya membuat pengaturan dengan satu class saja dengan pool tiga.

```
...  
...  
delay_pools 1  
delay_class 1 3  
...  
...
```

Kemudian mengatur ACL yang menggunakan pengaturan class 1 dan tolak semua pengguna di luar ACL (termasuk dalam *acl all*).

```
...  
...  
delay_access 1 allow user64kbps  
delay_access 1 deny all  
...  
...
```

Terakhir membuat pengaturan *bandwidth*, untuk keseluruhan jaringan (512KBps), group dari semua host (maksimum *bandwidth* yang ada), dan setiap individu (64 KBps dari total *bandwidth* 512 KBps).

```
...  
...  
#delay_parameters 1 <aggregate> <network> <individual>  
delay_parameters 1 64000/64000 -1/-1 8000/64000  
...  
...
```

Maka akan didapatkan konfigurasi delay pools untuk mengatur *bandwidth* setiap *user* maksimum 64 KBps seperti di bawah ini.

```
...  
...  
acl user64kbps src 192.168.1.0/255.255.255.255  
...  
delay_pools 1  
delay_class 1 3  
  
delay_access 1 allow user64kbps  
delay_access 1 deny all  
  
#delay_parameters 1 <aggregate> <network> <individual>  
delay_parameters 1 64000/64000 -1/-1 8000/64000  
...  
...
```


Pengujian dan Hasil

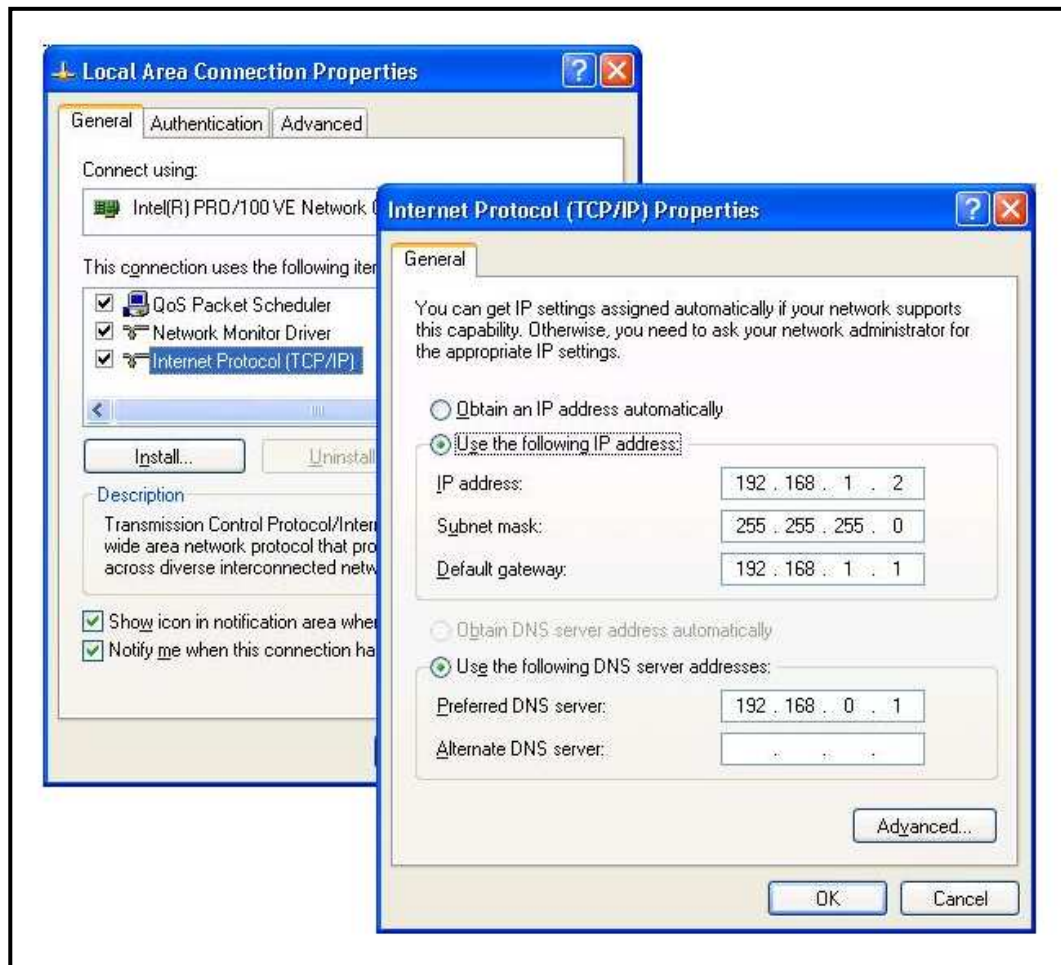
Sebelum dilakukannya pengujian, ada tahap yang harus dilakukan terlebih dahulu, yaitu memastikan koneksi harus berjalan dengan baik, baik koneksi jaringan maupun koneksi Internet. Pengujian ini dilakukan pada setiap komputer *client*.

Pengujian Koneksi Jaringan

Pada subbab ini kita akan mencoba memastikan koneksi antara komputer *server* ke *client* dan *client* ke *server* dapat terhubung dengan baik. Pengujian dapat dilakukan dari komputer *server* maupun *client*. Sebelum membahas lebih jauh, kita harus memastikan pemasangan kabel sudah benar, baik pemasangan pada komputer, maupun pemasangan pada *switch*. Jika pemasangan terpasang dengan baik, maka pada LAN *Card* biasanya terdapat lampu indikator yang akan menyala, begitu juga pada *switch*, akan ada lampu indikator yang menyala, menandakan pemasangan kabel sudah dipasang dengan benar.

Setelah pemasangan kabel dilakukan dengan baik, maka kita akan meng-*setting* IP *address* pada komputer *client*. Jika pada subbab sebelumnya penulis sudah menjelaskan *setting* IP *address* pada komputer *server* yang menggunakan sistem operasi Linux OpenSUSE, maka sekarang penulis akan menjelaskan *setting* IP *address* pada komputer *client* yang menggunakan sistem operasi Windows XP.

Tahap-tahapnya ialah Start → Control Panel → Network and Internet Connections → Network Connections → pilih Local Area Connection → klik kanan → Properties → pilih Internet Protocol (TCP/IP) → klik button Properties → isi IP *address* yang sesuai.

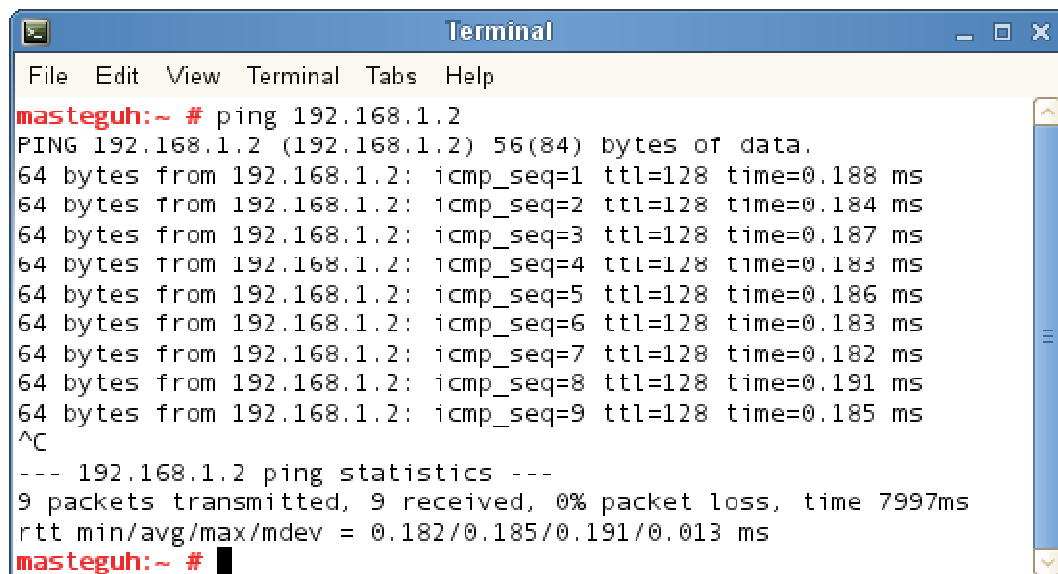


Gambar 4. Tampilan *setting* IP address pada salah satu komputer *client*

Setelah melakukan *setting* pada semua *client*, maka untuk lebih meyakinkan koneksi jaringan sudah terkoneksi dengan baik, kita dapat menggunakan perintah ping pada terminal jika menggunakan Linux dan pada DOS jika menggunakan Windows. Baik pada Linux maupun Windows formatnya sama, ketik perintah, ping<spasi>nomor IP komputer yang dituju. Sebagai contoh berikut ini kita akan melakukan ping dari komputer *server* yang memiliki IP komputer 192.168.1.1 ke komputer *client* yang memiliki IP komputer 192.168.1.2..

```
masteguh:~ # ping 192.168.1.2
```

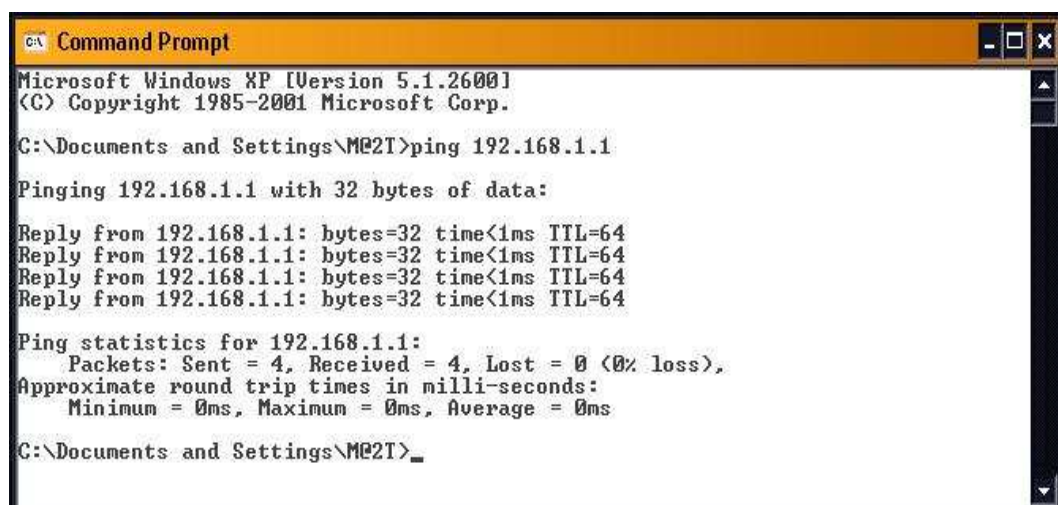
Jika koneksi memang terhubung dengan baik, maka perintah tersebut akan menampilkan pesan seperti tampilan berikut ini.



```
Terminal
File Edit View Terminal Tabs Help
masteguh:~ # ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=128 time=0.188 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=128 time=0.184 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=128 time=0.187 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=128 time=0.183 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=128 time=0.186 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=128 time=0.183 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=128 time=0.182 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=128 time=0.191 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=128 time=0.185 ms
^C
--- 192.168.1.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 7997ms
rtt min/avg/max/mdev = 0.182/0.185/0.191/0.013 ms
masteguh:~ #
```

Gambar 5. Hasil dari perintah ping di Terminal pada Linux

Kemudian kita akan melakukan lagi hal yang sama, akan tetapi kali ini tes dilakukan dari *client* yang memiliki IP *address* komputer 192.168.1.2 ke komputer *server* yang memiliki IP *address* komputer 192.168.1.1. Untuk kita ketahui terlebih dahulu, *client* ini menggunakan sistem operasi Windows, maka perintah tersebut akan dilakukan pada aplikasi DOS. Berikut ini hasilnya.



```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\MQ2T>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\MQ2T>
```

Gambar 6. Hasil dari perintah ping di DOS pada Windows

Maksud dari isi pesan yang kita peroleh ialah bahwa komputer yang menjadi tempat menjalankan perintah ping mendapatkan respon atau balasan tentang keberadaan IP komputer yang dituju. Sehingga dapat disimpulkan adanya sambungan atau koneksi antara komputer yang memberi perintah dengan komputer yang dituju. Akan tetapi walaupun adanya koneksi tidak menjamin koneksi tersebut terhubung dengan baik. Ada kalanya suatu hubungan bisa dikatakan kurang baik apabila *time* dari *reply* komputer yang dituju memiliki nilai yang besar bahkan bisa mencapai status RTO (*request time out*).

Pada pengujian kali ini, *time* yang diperoleh bisa dibilang baik, jika pada Terminal di Linux, akan terlihat dengan lebih spesifik dari segi penilaian *time*-nya. Pada baris pertama di gambar 16 disana *time*-nya bernilai 0.188 ms. Sedangkan pada DOS di Windows, tidak seperti di Terminal pada Windows, penilaian *time*-nya tidak spesifik, disana *time*-nya diberi nilai <1ms, lihat gambar 7.

Sharing Koneksi Internet

Jika sebelumnya penulis sudah memastikan koneksi jaringan dapat berjalan lancar sebagaimana mestinya, maka kali ini penulis akan mencoba *sharing* koneksi Internet yang sudah ada di komputer *server* ke komputer-komputer *client*. Karena walaupun koneksi jaringan sudah terpasang dengan baik, koneksi Internet tidak ada terbagi langsung ke *client*. Untuk mencapai tujuan tersebut ada beberapa perintah yang harus diketikan di Terminal komputer *server*. Berikut ini perintah yang dimaksud.

```
masteguh:~ # echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
masteguh:~ # iptables -t nat -A PREROUTING -s 192.168.1.0/24 -o  
eth1 -j MASQUERADE
```

```
masteguh:~ # iptables -A PREROUTING -t nat -p tcp --dport 80 -j  
REDIRECT -to-port 3128
```

Perlu dilakukan pengaturan pada *iptables*-nya dengan menggunakan *table nat*. Secara konsep, *iptables* diatur agar semua pengguna yang mengakses *web* ke alamat mana saja akan diarahkan untuk masuk ke Squid yang telah dibuat.

Dengan memasukan perintah di atas maka koneksi Internet yang terdapat pada komputer *server* akan tersalurkan ke komputer *client*. Akan tetapi masih belum termanajemen, karena Squid belum diaktifkan.

Pengujian Manajemen Bandwidth Internet dengan Squid

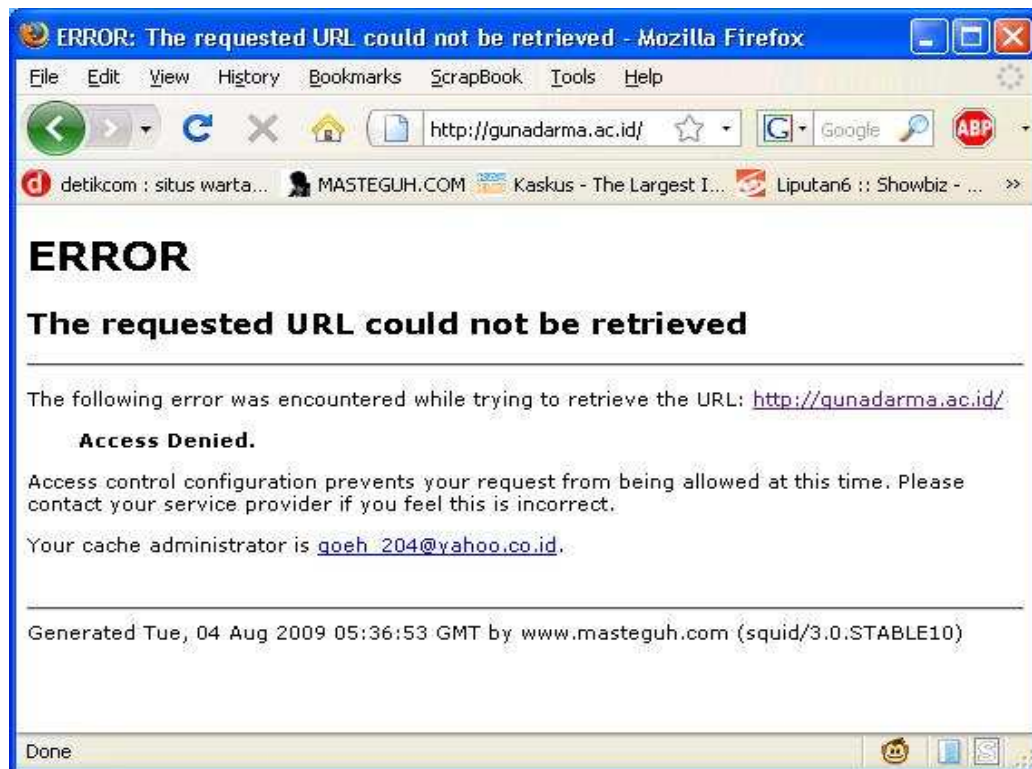
Setelah melakukan *sharing* koneksi jaringan dan koneksi Internet, kini saatnya kita melakukan manajemen *bandwidth* yang merupakan tujuan dari penulisan ini. Sekarang penulis akan mengaktifkan Squid yang sudah di-*instal* dan dikonfigurasi sedemikian rupa. Sebelum diaktifkan, Squid harus terlebih dahulu di *compile*, ini bertujuan untuk memeriksa apakah ada sintaks yang salah pada *file* konfigurasi Squid tersebut. Jika terdapat kesalahan pada penulisan sintaks, maka pada proses *compile* akan diberi peringatan tentang kesalahan sintaks tersebut. Berikut ini perintah untuk meng-*compile* yang harus dimasukkan di Terminal.

```
masteguh:~ # squid -z
```

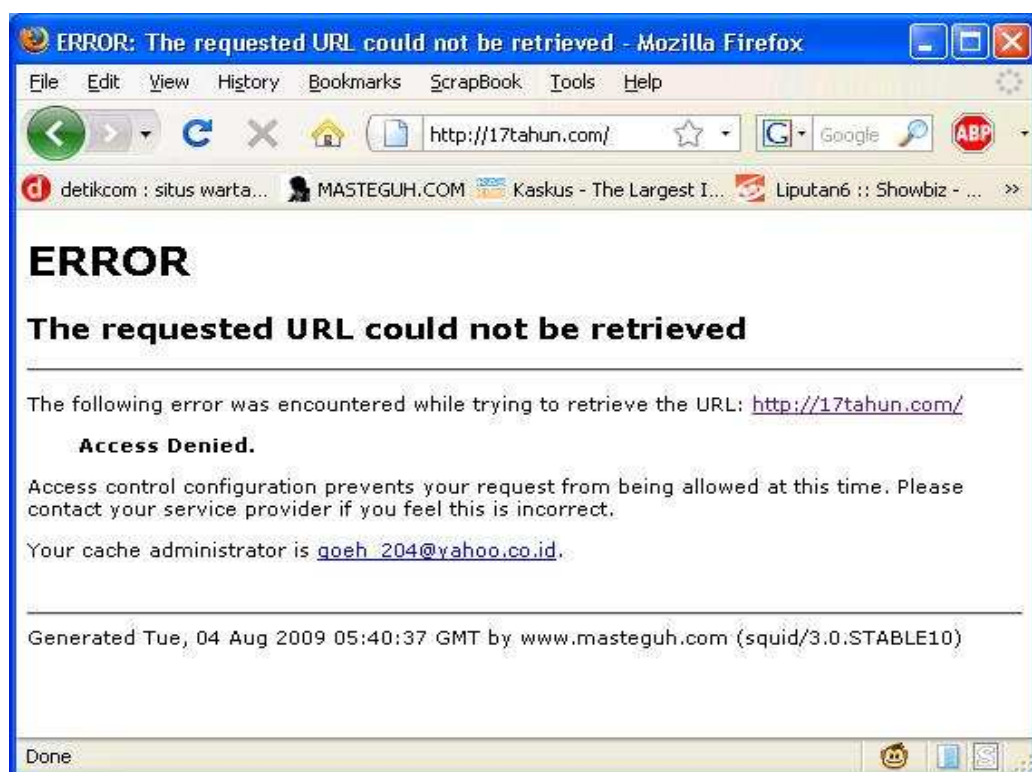
Jika proses *compile* berhasil maka saatnya mengaktifkan Squid. Berikut ini perintah untuk mengaktifkannya.

```
masteguh:~ # service squid start
```

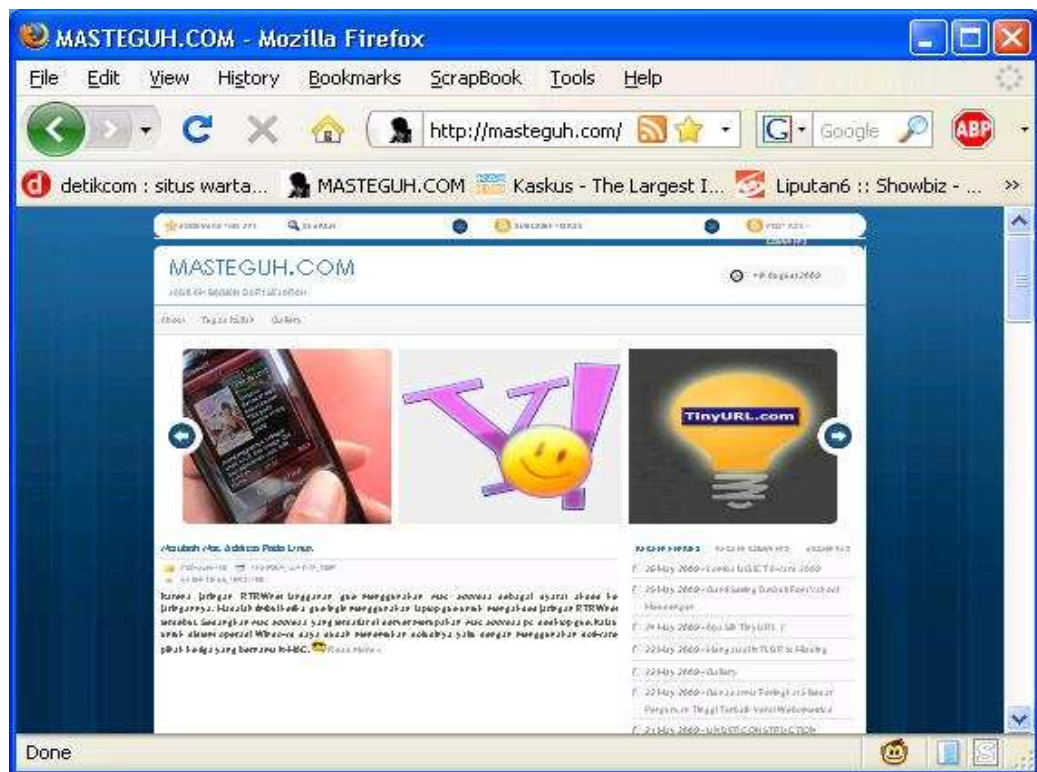
Dengan begitu Squid sudah aktif dan akan melakukan pemblokiran pada *client-client* sesuai dengan konfigurasi yang sudah dibuat. Selanjutnya melakukan pengujian, uji coba dilakukan dengan mengakses Internet pada jam-jam yang dilarang untuk mengakses Internet dan uji coba juga dilakukan dengan cara mengakses alamat URL yang termasuk dalam daftar situs terlarang. Jika tidak dapat mengakses pada jam-jam yang memang dilarang maka konfigurasi yang dilakukan berhasil. Begitu juga dengan mengakses situs terlarang, jika tidak dapat membukanya maka pemblokiran juga berhasil. Berikut ini tampilan yang tampak ketika Squid tidak mengizinkan untuk mengakses Internet.



Gambar 7. Pesan *Error*, tidak diizinkan mengakses pada jam-jam tertentu



Gambar 8. Pesan *Error*, tidak diizinkan mengakses situs yang dilarang



Gambar 9. Akses diizinkan

Jika ingin mengubah konfigurasi pada Squid, maka perintah yang digunakan ialah.

```
masteguh:~ # gedit /etc/squid/squid.conf
```

Setelah melakukan perubahan konfigurasi pada *squid.conf*. Maka agar perubahan konfigurasi dapat aktif, harus menggunakan perintah berikut ini.

```
masteguh:~ # service squid reload
```

Sedangkan untuk menghentikannya proses Squid dapat menggunakan perintah berikut.

```
masteguh:~ # service squid stop
```

Kesimpulan

Jika sebelumnya pemakaian Internet bersama-sama sering kali menimbulkan masalah, seperti tidak adanya pembagian *bandwidth* yang adil pada setiap penggunaanya, tidak adanya filter untuk mengakses situs manapun, dan sampai pemanfaatan Internet yang tidak berdasarkan haknya. Maka dengan memanfaatkan Squid pada server proxy dapat di atur konfigurasinya berdasarkan hak pemakaian sehingga dapat memberikan layanan sesuai kebutuhan. Adapaun secara umum manajemen penggunaan server proxy ini, antara lain :

1. Mempercepat tampilan halaman web pada saat browsing.
2. Membatasi hanya user-user tertentu yang dapat browsing.
3. Mencegah user untuk browsing ke situs-situd tertentu.
4. Pada fungsi ini squid berfungsi sebagai perantara pada saat web transaction, dimana squid menerima request dari client, memproses request tersebut lalu melanjutkan / memforward request tersebut ke server yang di tuju. Request tersebut bisa saja di tolak, dimodifikasi terlebih dahulu sebelum diteruskan serta di log oleh squid.
5. Sebagai cache server squid menyimpan contents web yang pernah dikunjungi sehingga dapat dipergunakan kembali. Jadi bila ada request yang sama maka akan dilayani oleh squid sehingga tidak perlu lagi menghubungi server yang dituju.

Dengan melakukan konfigurasi ini banyak keuntungan yang didapat dengan cara melakukan konfigurasi pada sebuah server linux. Penggunaan squid pada server linux dapat diterapkan diberbagai tempat seperti untuk jaringan perkantoran yang membutuhkan kontrol yang baik atas pemanfaatan penggunaan jaringan internet sehingga dengan kapasitas bandwidth yang tersedia dapat dimanfaatkan dengan sebaik- baiknya.

Daftar Pustaka

- [1] Farah Rizqie, *Pemanfaatan Proxy Server Pada Squid Untuk Mempercepat Akses Internet*, Gunadarma, 2004
- [2] Rahmat Rafiudin, *Squid Koneksi Anti Mogok*, Andi, Yogyakarta, 2008.
- [3] Rakhmat Farunuddin, *Membangun Firewall dengan IPTables di Linux*, PT Elex Media Komputindo, Jakarta, 2005.
- [4] Ridwan Sanjaya, *Trik Mengelola Kouta Internet Bersama dengan Squid*, PT. Elex Media Komputindo, Jakarta, 2005.
- [5] URL: <http://gratianet.wordpress.com/2008/12/07/sharing-koneksi-internet-opensuse-dan-winxp.html>, 25 Juli 2009.
- [5] URL: <http://myamin.net/index.php/linux/proxy-server-transparent-di-opensuse-110-dengan-squid-30.html>, 25 Juli 2009.