



**IMPLEMENTASI *PROXY SERVER* DAN *LOAD BALANCING*
MENGUNAKAN METODE *PER CONNECTION*
CLASSIFIER (PCC) BERBASIS MIKROTIK
(Studi kasus : Shmily.net)**

¹Fiki Justisia Bhayangkara, ²Imam Riadi (0510088001)

¹Program Studi Teknik Informatika

²Program Studi Sistem Informasi

Universitas Ahmad Dahlan

Prof. Dr. Soepomo, S.H., Janturan, Umbulharjo, Yogyakarta 55164

¹Email: fikijbhayangkara@gmail.com

²E-mail: imam.riadi@is.uad.ac.id

ABSTRAK

Jaringan Internet sudah merupakan suatu kebutuhan untuk setiap Negara berkembang, semua itu terkendala ketika bermasalah pada suatu jaringan Internet seperti link disconnect maupun down hingga terputusnya jaringan Internet. Permasalahan tersebut akan menjadi suatu hal yang sangat menyusahakan bagi pengguna Internet ketika pengguna sedang dalam keadaan penting atau pada jam-jam kerja yang sangat tergantung pada koneksi Internet dan kemudian koneksi terputus.

Pengembangan proxy server dan load balancing dilakukan tahapan penelitian yang terdiri dari analisa sistem dilanjutkan dengan perancangan sistem yang meliputi perancangan topologi, perancangan server, dan perancangan sistem kemudian pengujian penelitian digunakan uji validasi oleh user Shmily.net.

Berdasarkan hasil pengujian proxy dan load balancing memperlihatkan bahwa kedua koneksi yang berasal dari dua ISP Speedy dapat berjalan secara bersamaan dan akan otomatis berganti ketika salah satu koneksi terputus dan berjalan normal secara bersamaan ketika kedua koneksi tidak bermasalah. Berdasarkan hasil penelitian dapat disimpulkan bahwa sistem load balancing dan proxy server merupakan solusi untuk kebutuhan Internet secara real time.

Kata Kunci : *Jaringan, Ubuntu, Proxy, Mikrotik, load balancing*

1. PENDAHULUAN

Perkembangan dan kemajuan teknologi informasi sangatlah cepat, kebutuhan akan aksesibilitas data yang cepat dan stabil telah mendorong penggunaan teknologi informasi disegala bidang, baik politik, ekonomi, kesehatan, hiburan dan terutama di bidang pendidikan, dimana teknologi informasi menjadi pendukung utama untuk meningkatkan mutu dan performa suatu bidang. Kemampuan komunikasi yang tidak mengenal batasan ruang dan waktu melalui teknologi *networking*, merupakan syarat utama disetiap bidang untuk dapat terus meningkatkan efisiensi kinerja. Penggunaan teknologi komputer dan Internet menjadi pilihan terbaik untuk dapat

memaksimalkan mutu dan kualitas sebuah sistem, yang hasilnya akan menjadi tolak ukur suatu keberhasilan.

Shmily.net sebagai penyedia layanan Internet berbasis *wireless* yang memperhatikan kebutuhan pelanggan akan pentingnya kecepatan dan kestabilan komunikasi data yang dilakukan melalui Internet dimana pada area tersebut tersedia koneksi Internet berbasis *wireless* yang dapat diakses melalui *notebook*, laptop, maupun perangkat lainnya yang mendukung teknologi *wireless*. Dengan adanya jaringan hotspot di Shmily.net maka setiap orang bisa menikmati akses Internet dimanapun dan kapanpun selama di area hotspot Shmily.net tanpa harus menggunakan kabel. Layanan inilah yang diharapkan akan mempercepat akses informasi bagi yang sudah terdaftar sebagai pelanggan Shmily.net, akan tetapi jaringan Shmily.net belum sepenuhnya memenuhi standar kebutuhan pelanggan. Masalah yang sering terjadi di Shmily.net adalah *trouble* pada koneksi dan tidak cukupnya koneksi untuk memenuhi semua kebutuhan pelanggan. Oleh karena itu, Shmily.net perlu peningkatan dan kestabilan koneksi dengan menggunakan *load balancing* dan *proxy server*. Diharapkan dengan menggunakan *load balancing* dan *proxy server* nantinya akan mencukupi kebutuhan koneksi dan kestabilan koneksi tersebut dan penelitian ini dilakukan di Shmily.net yang berlokasi di jalan Glagahsari, Yogyakarta.

Berdasarkan latar belakang masalah tersebut, maka akan diteliti implementasi *load balancing* dan *proxy server* sebelum dan sesudah menggunakan, untuk mendapatkan gambaran dan implementasi serta penerapan sistem jaringan.

2. KAJIAN PUSTAKA

Penelitian mengacu pada skripsi yang disusun Purwangga Bagja [1], Teknik Informatika Universitas Komputer Indonesia Bandung yang berjudul “Konfigurasi *load balancing* pada sistem jaringan komputer di Balai Besar, Arief Rakhman Aziz [2], Teknik Informatika Universitas Ahmad Dahlan Yogyakarta, Membangun Sistem VRRP Menggunakan Sistem Operasi *FreeBSD*, dan Mochamad Irfan [3], Teknik Informatika Politeknik Bandung, Penerapan *bandwidth management* menggunakan metode PCC (*Per Connection Classifier*) di PT. Neuronworks. Dalam skripsi Sistem yang akan dibuat pada penelitian kali ini yang akan melakukan implementasi *load balancing* dan *proxy server* untuk kecepatan dan kestabilan koneksi Internet.

2.1. Jaringan

Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan antara satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program, penggunaan bersama perangkat keras seperti printer, *harddisk*, dan sebagainya. Selain itu jaringan komputer bisa diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada diberbagai lokasi yang terdiri dari lebih satu komputer yang saling berhubungan.[4]

2.2. Jenis Jaringan

Jaringan komputer dapat dikelompokkan berdasar luas area yang dapat dijangkau atau dilayani. Secara umum jaringan komputer terbagi menjadi tiga jenis, yaitu *Local area network* (LAN), *Metropolitan area network* (MAN), dan *Wide area network* (WAN). [4]

2.3. Tujuan Jaringan Komputer

Jaringan komputer menjadi penting bagi manusia dan organisasinya karena jaringan komputer mempunyai tujuan yang menguntungkan. Tujuan jaringan komputer adalah untuk *resource sharing* atau berbagi sumber, *high reliability* atau kehandalan tinggi, menghemat biaya, *Scalability* atau skalabilitas, medium komunikasi, akses informasi luas, komunikasi dua arah, hiburan interaktif. [4]

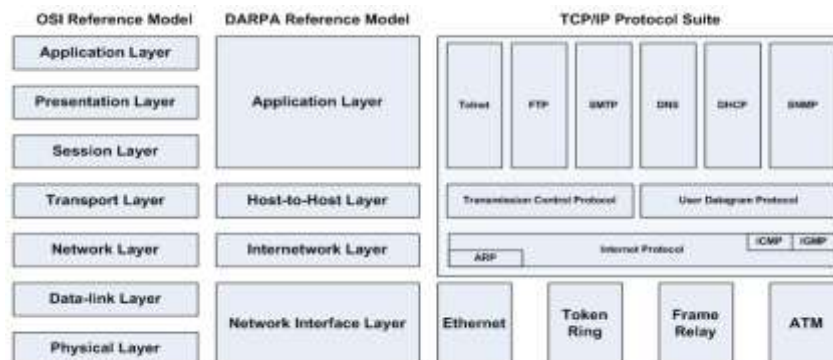
2.4. Topologi Jaringan

Topologi merupakan bagaimana cara dan bentuk secara fisik untuk menghubungkan komputer yang satu dengan yang lainnya. Bentuk topologi terdapat banyak alternatif, akan tetapi terdapat tiga bagian yang umum dipakai seperti topologi bus, topologi star dan topologi ring. [4]

2.5. TCP/IP

TCP/IP dikembangkan mengacu pada model *Open Sistem Interconnection* (OSI), di mana, *layer* yang terdapat pada TCP tidak persis sama dengan *layer* yang terdapat pada model OSI. Terdapat empat *layer* pada TCP/IP, yaitu: *network interface*, *network*, *transport* dan *application*. Tiga *layer* pertama pada TCP/IP menyediakan *physical standards*, *network interface*, *Internet working*, dan fungsi *transport*, yang mengacu pada empat *layer* pertama pada model OSI. [4]

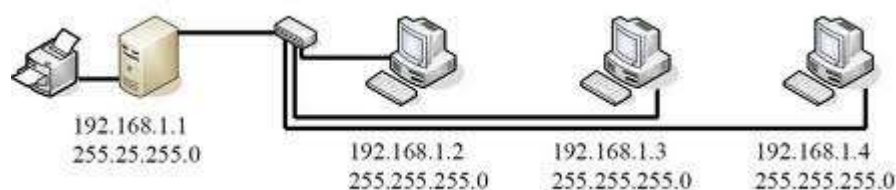
Tiga *layer* teratas dari model OSI direpresentasikan di model TCP/IP sebagai satu *layer*, yaitu *application layer*. Arsitektur protokol TCP/IP seperti tampak pada Gambar 1. [4]



Gambar 1. Arsitektur TCP/IP [4]

2.6. IP Address

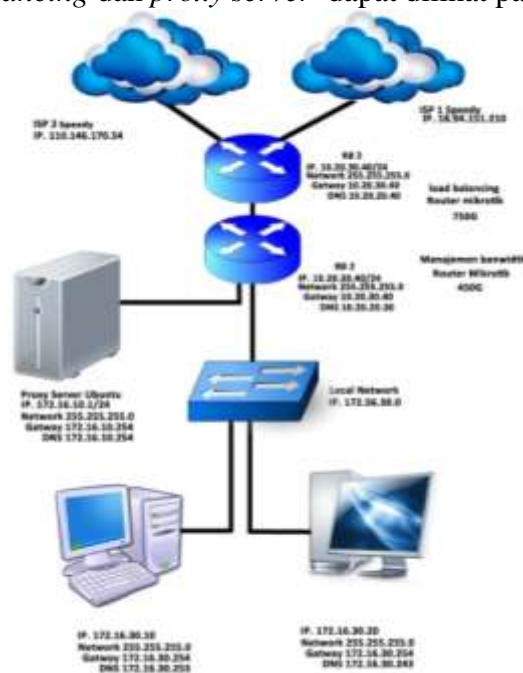
IP address (Internet protokol *address*) adalah metode pengalamatan pada jaringan komputer dengan memberikan sederetan angka pada komputer (*host*), *router* atau peralatan jaringan lainnya. *IP Address* sebenarnya bukan diberikan kepada komputer (*host*) atau *router*, melainkan pada *interface* jaringan dari *host* atau *router* tersebut. Contoh penerapan *IP Address* pada jaringan komputer dapat dilihat pada Gambar 2.[4]



Gambar 2. Konsep IP address [4]

2.7. Load balancing

Load balancing pada mikrotik adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. *Load balancing* tidak akan menambah besar *bandwidth* yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang. Dalam penggunaan *load balancing* menggunakan 2 koneksi sebesar 2 Mbps hasilnya tidak menjadi 4 Mbps. Melainkan tetap 2Mbps + 2 Mbps. Dalam penggabungan koneksi *load balancing* semua koneksi akan didistribusikan secara seimbang. [4] Gambar Topologi *load balancing* dan *proxy server* dapat dilihat pada Gambar 3.



Gambar 3. Topologi *load balancing* dan *proxy server*

2.8. Mikrotik Router

MikroTik *routerOS*TM adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk IP *network* dan jaringan *wireless*, cocok digunakan oleh ISP dan *provider* hotspot. Mikrotik *routerOS* sebuah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk IP *network* dan jaringan *wireless*. Fitur-fitur tersebut diantaranya : *Firewall & Nat, Routing, Hotspot, Point to Point Tunneling Protocol, DNS server, DHCP server, Hotspot*, dan masih banyak lagi fitur lainnya. Mikrotik dapat digunakan dalam 2 tipe, yaitu dalam bentuk perangkat keras dan perangkat lunak. Dalam bentuk perangkat

keras, mikrotik biasanya sudah diinstalasi pada suatu *board* tertentu, sedangkan dalam bentuk perangkat lunak, mikrotik merupakan satu distro Linux yang memang dikhususkan untuk fungsi *router*. MikroTik RouterOS™, merupakan sistem operasi Linux *base* yang diperuntukkan sebagai *network router*. Didesain untuk memberikan kemudahan bagi penggunaanya. Administrasinya bisa dilakukan melalui *Windows Application* (*Winbox*). Selain itu instalasi dapat dilakukan pada standar komputer PC (*Personal Computer*). PC yang akan dijadikan *router* mikrotik pun tidak memerlukan *resource* yang cukup besar untuk penggunaan standar, misalnya hanya sebagai *gateway*. Untuk keperluan beban yang besar (*network* yang kompleks, *routing* yang rumit) disarankan untuk mempertimbangkan pemilihan resource PC yang memadai. [4]

2.9. *Per Connection Clasifier (PCC)*

Metode PCC (*Per Connection Clasifier*) merupakan salah satu metode yang dapat digunakan pada *load balancing*, dengan PCC dapat digunakan untuk mengelompokkan trafik koneksi yang melalui *router* menjadi beberapa kelompok, sehingga *router* akan mengetahui jalur *gateway* yang dilewati diawal trafik koneksi dan pada paket-paket selanjutnya yang masih berkaitan dengan koneksi awalnya akan dilewatkan pada jalur *gateway* yang sama juga.[5]

2.10. *Proxy Server*

Proxy server (peladen *proxy*) adalah sebuah komputer *server* atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan *request* terhadap *content* dari Internet atau Intranet.

Proxy Server bertindak sebagai *gateway* terhadap dunia ini Internet untuk setiap komputer clien. *Proxy server* tidak terlihat oleh komputer *client*: seorang pengguna yang berinteraksi dengan Internet melalui sebuah *proxy server* tidak akan mengetahui bahwa sebuah *proxy server* sedang menangani *request* yang dilakukannya. *Web server* yang menerima request dari *proxy server* akan menginterpretasikan *request-request* tersebut seolah-olah *request* itu datang secara langsung dari komputer *client*, bukan dari *proxy server*. [5]

2.11. *Ubuntu*

Ubuntu suatu sistem operasi bebas dan *open source* yang menggunakan Debian sebagai fondasinya dan dirilis secara berkala (setiap enam bulan), fokus utama sistem operasi Ubuntu adalah kemudahan penggunaan dan pada setiap rilis Ubuntu akan memberikan perbaikan keamanan selama 18 bulan. Ubuntu menyertakan lingkungan desktop Gnome/KDE/XFCE terbaru di setiap rilis dan juga menyertakan beragam pilihan perangkat lunak untuk *server* dan desktop yang semuanya dikemas ke dalam satu CD.[6]

2.12. *Keuntungan Menggunakan load balancing dan proxy server*

Ketika salah satu jaringan *down* maka akan otomatis ditopang oleh koneksi yang lain atau ketika jaringan diakses oleh banyak pengguna, maka *load balancing* maka dengan adanya *load balancing* bisa dialihkan ke *server* lain. Secara garis besar, keuntungan dari penerapan *load balancing* adalah sbb:

2.12.1 Menjamin Reliabilitas sistem artinya tingkat kepercayaan terhadap sebuah sistem untuk dapat terus melayani pengguna dengan sebaik-baiknya. Reliabilitas yang terjamin artinya tingkat kepercayaan yang

selalu terjaga agar para penggunanya dapat menggunakan servis tersebut dan melakukan pekerjaannya dengan lancar. Hal ini amat penting bagi situs-situs komersial.

2.12.2 Skalabilitas dan *available* satu buah *server* yang digunakan untuk melayani berribu-ribu pengguna, tentunya tidak mungkin dapat menghasilkan pelayanan yang baik. Meskipun telah menggunakan sebuah *server* dengan teknologi tercanggih sekalipun, tetap saja bisa kewalahan melayani penggunanya. Selain itu, satu buah *server* artinya satu buah titik masalah. Jika tiba-tiba *server* tersebut mati, masalah pasti akan terjadi terhadap situs atau servis di dalamnya. Namun dengan menggunakan sistem *load balancing*, *server* yang bekerja mendukung sebuah situs atau servis dapat lebih dari satu buah. Artinya jika ternyata satu buah *server* kewalahan melayani pengguna. Selain itu juga titik masalah menjadi terpecah. Jika ada sebuah *server* bermasalah, maka masih ada dukungan dari yang lain. Situs atau servis yang Anda jalankan belum tentu bermasalah ketika sebuah *server* mengalami masalah.

Keuntungan penggunaan *proxy*, yaitu dapat menghemat biaya *bandwidth*. Mempercepat koneksi karena *file* web yang direquest (selanjutnya disebut *object*) disimpan di dalam *cache* sehingga tidak perlu keluar menuju Internet. Mengatur kecepatan *bandwidth* untuk subnet yang berbeda-beda, dapat melakukan pembatasan untuk *file* tertentu, dapat melakukan pembatasan akses kepada situs-situs tertentu (misalnya situs porno), dapat melakukan pembatasan *download* untuk *file* tertentu (misalnya file-file mp3, wav, dsb), dapat melakukan pembatasan waktu-waktu yang diperbolehkan untuk *download*.

3. METODE PENELITIAN

Subjek penelitian yang akan dibahas pada tugas akhir ini adalah: “Implementasi *Load Balancing* dengan metode *PCC* dan *Proxy Server* (Studi Kasus: *Shmily.net*)”. Sistem yang akan dibuat diharapkan dapat meningkatkan kualitas koneksi yang digunakan oleh pelanggan *Shmily.net* sehingga dapat menjadi standar kualitas dan mutu pada *Shmily.net*.

3.1. Metode Pengumpulan Data

Metode Pengumpulan data pada penelitian ini menggunakan metode wawancara kepada pemilik *Shmily.net*, metode literatur dan metode eksperimen. Wawancara berkaitan dengan kualitas koneksi jaringan *Shmily.net*. Metode literatur dengan cara studi pustaka dengan membaca dan membandingkan buku-buku referensi tentang *load balancing*, metode *PCC*, *proxy server*, *squid* dan artikel yang terkait. Metode eksperimen dilakukan dengan cara melakukan penelitian secara langsung terhadap objek yang diteliti yaitu *Shmily.net* dengan melakukan percobaan *load balancing* dan *proxy server*.

3.2. Alat Penelitian

Untuk melakukan penelitian ini memerlukan beberapa peralatan yang terbagi dua bagian yaitu: perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras meliputi seperangkat komputer. Perangkat lunak yang dibutuhkan meliputi sistem informasi dan perangkat lunak aplikasi lainnya.

3.3. Analisis Data

Data yang berhasil dikumpulkan adalah data *user* atau pelanggan di Shmily.net. Dan penelitian ini terfokus kecepatan koneksi dengan *proxy* dan *load balancing*.

3.4. Perancangan

Dalam proses perancangan sistem ini dilakukan uji coba. Adapun beberapa uji coba pengerjaan untuk mendukung pengerjaan proyek ini yaitu pengujian pertama menguji coba koneksi dari sisi *user*, kedua menguji koneksi dengan *load balancing* kemudian menguji dengan penggabungan *proxy server* dan *load balancing*.

3.5. Uji Kelayakan

Uji kelayakan dilakukan untuk mengetahui suatu kelayakan dari sistem yang diimplementasikan, apakah sistem tersebut sudah sesuai dengan tujuan yang ingin dicapai dalam penelitian ini. Pengujian ini dilakukan dengan mengamati dan mencari kesalahan terhadap jalannya *proxy* dan *load balancing*. Jika sesuai dengan yang diinginkan, maka proses implementasi jaringan menggunakan *proxy* dan *load balancing* dinyatakan berhasil.

4. HASIL DAN PEMBAHASAN

4.1. Tahap Persiapan

Pada tahap persiapan ini merupakan tahap untuk melakukan percobaan koneksi terhadap Shmily.net, adapun persiapan yang telah dilakukan yaitu telah dilakukan penelitian dan pengamatan terhadap jaringan Shmily.net. Persiapan awal yang dilakukan adalah menguji koneksi dengan menggunakan 10 *user* Shmily.net dan melihat hasil yang diperoleh yaitu hasil presentase kualitas koneksi.

4.2. Instalasi dan Konfigurasi *proxy server*.

Melakukan perubahan terhadap *file repository apt* untuk diarahkan ke *mirror local* Indonesia dengan alamat <http://kambing.ui.ac.id/ubuntu/lucid> seperti, pada Tabel 1 dibawah ini.

Tabel 1: Listing kode konfigurasi *squid*

1	<code>mv /etc/apt/sources.list /etc/apt/sources.list</code>
2	<code>nano /etc/apt/sources.list</code>
3	<code>deb http://kambing.ui.ac.id/ubuntu/ lucid-backports main restricted universe multiverse</code> <code>deb-src http://kambing.ui.ac.id/ubuntu/ lucid-backports main restricted universe multiverse</code> <code>deb http://kambing.ui.ac.id/ubuntu/ lucid-proposed main restricted universe multiverse</code> <code>deb-src http://kambing.ui.ac.id/ubuntu/ lucid-proposed main restricted universe multiverse</code> <code>deb http://kambing.ui.ac.id/ubuntu/ lucid-security main restricted universe multiverse</code> <code>deb-src http://kambing.ui.ac.id/ubuntu/ lucid-security main restricted universe multiverse</code> <code>deb http://kambing.ui.ac.id/ubuntu/ lucid-updates main restricted universe multiverse</code> <code>deb-src http://kambing.ui.ac.id/ubuntu/ lucid-updates main restricted</code>

	universe multiverse deb http://kambing.ui.ac.id/ubuntu/ lucid main restricted universe multiverse deb-src http://kambing.ui.ac.id/ubuntu/ lucid main restricted universe multiverse
4	ctrl + x

4.3 Konfigurasi Load Balancing

Router mikrotik dikonfigurasi sebagai *load balancing* dimana router ini sebagai penggabung kedua koneksi dari Speedy. Langkah-langkah yang dilakukan untuk menerapkan teknik diatas adalah sebagai berikut:

- 4.3.1** Mengkonfigurasi jalur koneksi dengan nama Speedy 01 yang berasal dari koneksi Speedy 2 Mbps dengan menggunakan modem Aztec dengan mengkonfigurasi *routing mark* dengan nama *line 1* atau jalur satu dengan jalur koneksi yang akan masuk pada mode *load balancing*.
- 4.3.2** Konfigurasi *load balancing* pada tahap sebelumnya dengan melakukan beberapa konfigurasi pada menu *mangle* agar *load balancing* bisa berjalan dengan baik dan stabil. Konfigurasi *load balancing* dengan metode PCC menggunakan terminal yang sudah ada pada router seperti pada Gambar 4.



```

Mikrotik RouterOS 5.17 (rc) 1999-2012      http://www.mikrotik.com/

[fish@LS 750] > ip firewall mangle add chain=input in-interface=Modem3 action=mark
-connection new-connection=mark-Modem3_conn
input does not match any value of interface
[fish@LS 750] > add chain=input in-interface=Modem3 action=mark-connection new-connec
to=mark-Modem3_conn
add command succeeded (line 1 column 3)
[fish@LS 750] > ip firewall mangle add chain=output connection=mark-Modem3_conn act
ion=mark-routing new-routing=mark-to-Modem3_conn
expected end of command (line 1 column 37)
[fish@LS 750] > add chain=output connection=mark-Modem3_conn action=mark-routing ne
w-routing=mark-to-Modem3_conn

```

Gambar 4. Konfigurasi IP firewall mangle.

Seperti yang terlihat pada kode *script 1* adalah jalur konfigurasi IP firewall mangle.

Kode script 1. Konfigurasi IP firewall mangle.

```

addchain=output connection mark-
mark=Modem3_conn action=mark-routing new-
routing-mark=to Modem3 conn

```

- 4.3.3** Membuat *Diffie Helman* yaitu digunakan untuk *assign request* atau persetujuan koneksi antara *client* dengan *server*. *Diffie Helman* adalah protokol kriptografik yang menunjukkan kedua pihak yang bertukar informasi, walaupun mereka tidak mengenal satu sama lainnya, dapat secara bersama menciptakan sebuah kunci rahasia bersama melalui sebuah jalur komunikasi yang tak aman sekalipun.
- 4.3.4** Konfigurasi ini melakukan perintah menambahkan *output* pada modem 3 dengan memasukan kode *add chain = output* koneksi pada modem 3.

- 4.3.5 Menambahkan pendefinisian *packet* yang akan masuk ke dalam tubuh *router* melalui *interface* konfigurasi *passthrough* pada modem 3 dan modem 4.
- 4.3.6 konfigurasi IP *firewall nat* dengan *script* untuk memasukan perintah pada *router* seperti pada Gambar 5.



Gambar 5. Konfigurasi trafik modem 3 dan modem 4 pada terminal.

Seperti yang terlihat pada kode *script* 2 dibawah adalah jalur konfigurasi IP *firewall nat*.

Kode *script* 3. Konfigurasi IP *firewall nat*.

```
addchain=srcnat out-interface=MODEM3  
action=masquerade
```

- 4.3.7 Hasil *load balancing* pada *interface list* yang berjalan dengan baik terlihat pada Gambar 6.

Name	Type	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...	Tx Drops	Rx Drops	Tx Errors	Rx Errors
X Modem01	Ethernet	1524	0 bps	0 bps	0	0	0	0	0	0
X Modem02	Ethernet	1524	0 bps	0 bps	0	0	0	0	0	0
R Modem03 To Public	Ethernet	1524	84.9 kbps	218.4 kbps	42	260	0	0	0	0
R Modem04 Tp-Link	Ethernet	1524	53.7 kbps	30.5 kbps	32	22	0	0	0	0
R Modem05 Aztec	Ethernet	1524	177.0 kbps	15.0 kbps	225	16	0	0	0	0
R Speedy 02 Tp-Link	PPPoE Client		47.1 kbps	25.9 kbps	32	22	0	0	0	0
R Speedy01 Aztec	PPPoE Client		130.2 kbps	11.7 kbps	225	16	0	0	0	0

Gambar 6. Hasil *load balancing* pada *interface list*

4.4 Percobaan 1 Load Balancing

Percobaan *load balancing* adalah pengujian sistem *load balancing* tanpa menggunakan *proxy server* agar dapat diketahui apakah *load balancing* berfungsi dengan baik seperti terlihat pada Gambar 7 adalah hasil *load balancing* yang dilihat melalui *tab download*.



avast_free_antivirus_setup.exe 250 KB/s - 8.2 MB of 112 MB, 7 mins left
http://files.avast.com/iavs5x/avast_free_antivirus_setup.exe
Pause Cancel

Gambar 7. *Download dengan load balancing*

4.5 Percobaan 2 *Proxy server*

Percobaan *proxy server* adalah percobaan dengan menguji fungsi *proxy server* sebelum menggunakan *load balancing* agar dapat diketahui apakah *proxy server* berfungsi dengan baik. Seperti terlihat pada Gambar 8 adalah hasil *proxy server* yang dilihat melalui *putty*.

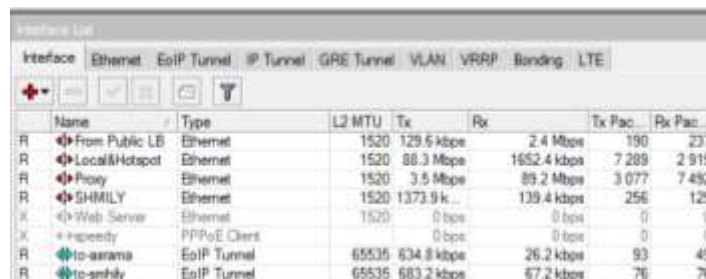


Gambar 8. *Cache proxy*

Gambar 8 merupakan hasil *cache* pada *server proxy* yang memberikan report bahwa *proxy* berjalan dengan baik.

4.6 Percobaan ketiga penggabungan *proxy* dan *load balancing*

Percobaan penggabungan *load balancing* dan *proxy server* seperti terlihat pada Gambar 9.



Name	Type	L2 MTU	Tx	Rx	Tx Pac.	Rx Pac.
From Public LB	Ethernet	1520	129.6 kbps		2.4 Mbps	190 237
LocalHotspot	Ethernet	1520	88.3 Mbps	1652.4 kbps	7.289	2.915
Proxy	Ethernet	1520	3.5 Mbps	89.2 Mbps	3.077	7.492
SHMILY	Ethernet	1520	1373.9 k...	139.4 kbps	256	125
Web Server	Ethernet	1520	0 bps	0 bps	0	0
ispemily	PPPoE Client		0 bps	0 bps	0	0
to-asrama	EoIP Tunnel	65536	634.8 kbps	26.2 kbps	93	49
to-smily	EoIP Tunnel	65536	683.2 kbps	67.2 kbps	76	76

Gambar 9. Tampilan *interface* dengan menjalankan fungsi *load balancing* dan *proxy*

4.7 Hasil perbandingan

Dibawah ini adalah hasil test sebelum dan sesudah menggunakan *proxy* dan *load balancing* seperti terlihat pada Tabel 4.

Tabel 4. Hasil Perbandingan

Sebelum menggunakan <i>proxy</i> dan <i>load balancing</i>	Setelah menggunakan <i>proxy</i> dan <i>load balancing</i>
Pada percobaan sebelum menggunakan <i>proxy</i> dan <i>load balancing</i> yang dilakukan oleh 10 user Shmily.net kecepatan koneksi sering terputus dikarenakan ISP yg bermasalah dan belum cukup memenuhi kebutuhan user.	Pada percobaan setelah menggunakan <i>proxy</i> dan <i>load balancing</i> yang dilakukan oleh 10 user Shmily.net koneksi Internet menjadi stabil dan kecepatan sudah mencukupi kebutuhan user.

4.8 Uji Kelayakan

Pengujian ini dilakukan dengan mengamati dan mencari kesalahan terhadap jalannya *load balancing* dan *proxy server*. Pengujian ini dilakukan oleh 10 *user* Shmily.net sebagai penguji kelayakan. Hasil Pengujian yang dilakukan 10 *user* Shmily.net sebelum menggunakan *proxy* dan *load balancing* mendapatkan hasil sebesar 75% merasa koneksi belum memenuhi kebutuhan *user* dan hasil pengujian sesudah menggunakan *proxy* dan *load balancing* mendapatkan hasil 77,5% merasa cukup.

5. PENUTUP

5.1. Kesimpulan

Berdasarkan penulisan yang telah dilakukan selama mengimplementasikan *load balancing* dan *proxy*, maka dapat diambil kesimpulan antara lain:

1. Teknik *load balancing* dan *proxy* pada jaringan Shmily.net diimplementasikan dengan metode PCC berbasis mikrotik dengan pengujian melalui 10 responden dengan hasil 77,5% *user* sudah mencukupi kebutuhan koneksi Internet dan artinya penerapan *load balancing* dan *proxy* sudah maksimal dan berjalan sesuai yang diharapkan.
2. Berdasarkan hasil pemanfaatan *load balancing* dan *proxy* pada Shmily.net telah didapatkan hasil bahwa kualitas koneksi Internet, Kecepatan *download* dan *browsing* di Shmily.net lebih cepat setelah menggunakan *load balancing* dan *proxy*.

5.2. Saran

Penulis menyadari bahwa masih banyak terdapat kekurangan-kekurangan dalam mengimplementasikan *load balancing* dan *proxy* pada Shmily.net ini, oleh karena itu demi meningkatkan kualitas pelayanan keamanan jaringan pada Shmily.net, maka penulis menyarankan:

1. Bagi pihak Shmily.net atau administrator Shmily.net untuk lebih meningkatkan kualitas koneksi dengan selalu menerapkan teknologi baru untuk performa koneksi yang lebih baik.
2. Bagi *user* Shmily.net diharapkan dengan penerapan *load balancing* dan *proxy* dapat meningkatkan kenyamanan dan performa dalam melakukan aktifitas dalam dunia maya.
3. Demi meningkatkan nilai kualitas *load balancing* dan *proxy* yang lebih baik pada jaringan Shmily.net, diharapkan penelitian dan implementasi berikutnya mengenai penggabungan metode *load balancing* PCC dan NTH kemudian dalam *proxy* dapat ditingkatkan dalam sisi keamanan dengan menggunakan Os *FreeBSD*.



DAFTAR PUSTAKA

- [1] Bagja P., 2010, Konfigurasi *load balancing* pada sistem jaringan komputer di Balai Besar , Teknik Informatika Universitas Komputer Indonesia Bandung.
- [2] Rakhman A., 2011, Membangun Sistem VRRP Menggunakan Sistem Operasi FreeBSD, Teknik Informatika Universitas Ahmad Dahlan Yogyakarta.
- [3] Irfan M., 2010, Penerapan *Bandwidth* Management menggunakan metode PCC (Per Connection Classifier) di PT. Neuronworks, Teknik Informatika Politeknik Bandung.
- [4] Manual book, 2010, Mikrotik *Router OS*.
- [5] Towidjojo R., 2013, Konsep dan Implementasi Routing dengan Router MikroTik – 200% Connected.
- [6] Hudson P., 2011, Ebook Ubuntu Unlieshed.