

## IMPLEMENTASI SQUID SEBAGAI REVERSE PROXY UNTUK KEPERLUAN BACKUP SERVER

Cahya Ningsih Fitri

D3 Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya, ningsihcahya11@gmail.com

Ibnu Febry Kurniawan

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya, ibnufebry@unesa.ac.id

### Abstrak

Di era modern ini perkembangan teknologi informasi semakin pesat, hal ini terlihat pada banyaknya fitur-fitur yang terpasang pada laman-laman web. Web-web yang memiliki banyak pengunjung biasanya membutuhkan sumber daya pemrosesan yang tinggi, seringkali mengakibatkan *overload* dan berujung pada *server down*. Salah satu upaya untuk menghadapi hal tersebut adalah dengan melakukan regulasi pembagian beban (*loadbalancing*) dalam suatu *klaster server*. Squid merupakan *software* yang digunakan sebagai *web cache* dan juga *proxy server*. Squid banyak ditemui untuk mempercepat penyimpanan *request*, penyimpanan alamat DNS yang diminta *client* secara terus menerus sehingga tidak perlu diambilkan dari *server* aslinya. Pada studi ini mencoba mensimulasikan squid sebagai *reverse proxy* untuk keperluan *backup server*. Simulasi tersebut menghasilkan apabila terdapat *request* dari *client* langsung diarahkan ke *reverse proxy* terlebih dahulu, *reverse proxy* yang mengatur jika terdapat *request* yang sama maka langsung di balas ke *client* (TCP\_HIT). Jika terdapat *request* yang baru maka dikirim ke *server1* atau *server2* menggunakan round-robin terlihat di *log squid* (TCP\_MISS). Dan apabila menonaktifkan *server1* maka semua *request* diarahkan ke *server 2* sebagai *backup server*.

**Kata Kunci:** Web, Squid, Reverse proxy, Failover

### Abstract

In this fast-growing Information Technology era, available websites have been enriched with many features. Most visited websites usually require high processing resources, and sometimes suffer overload, or worse down state. One way to cope this situation is regulating resource distribution (load balancing) in corresponding server cluster. Squid is one of application used for web-cache, and proxy-server, as well. Squid has been found able to reduce response time, and DNS query by lowering requests to originated servers. This study attempts to implement Squid as reverse proxy to fulfil failover requirement. Experiments have showed that client request will be directed to Squid. If there is same request cached, then Squid will response the request (TCP\_HIT). Conversely, Squid will direct the request to server cluster (server1 or server2) and producing TCP\_MISS log message. Failover has worked well in the experiment when server2 took over server1 during down state.

**Keywords:** Web, Squid, Reverse proxy, Failover

### PENDAHULUAN

Perkembangan dunia internet saat ini sangat pesat sehingga menyebabkan kenaikan dari jumlah pengguna internet. Hal ini terlihat pada banyaknya fitur-fitur yang terpasang pada laman-laman web. Sehingga banyak orang menggunakannya seperti dalam mempromosikan usahanya. Hal tersebut dapat mengakibatkan *traffic* atau lalu lintas data yang semakin padat. Layanan *web server* yang memiliki situs banyak dikunjungi memiliki beban proses yang lebih tinggi dalam melayani permintaan *client*. Sehingga dapat menyebabkan *web server* terancam *down*, dan tidak mampu melayani semua permintaan dari tiap-tiap *client*. Apabila *server down* *client* tidak dapat mengakses *web* tersebut.

Salah satu cara untuk menangani permasalahan agar tidak terancam *down*, dengan menerapkan Squid sebagai *Reverse Proxy* untuk Keperluan *backup server*. Squid merupakan *software* untuk *web cache* dan *proxy server*, squid biasanya digunakan untuk mempercepat *web server* melakukan penyimpanan *request* yang diminta *client*, penyimpanan alamat DNS yang diminta *client*, dan sebagai pengendali *server*. Seperti halnya *client* yang mengakses *web*, *client* telah melakukan pengiriman permintaan HTTP, akan diteruskan ke *reverse proxy* terlebih dahulu sebelum diarahkan ke *web server*, setelah itu di *filter* oleh *cache proxy*, apabila terdapat permintaan permintaan yang sama dengan sebelumnya maka akan langsung dibalas ke *client*, jika permintaan tersebut hal baru maka akan diteruskan ke *web server* asli, dan ketika banyak *client* yang mengakses *webserver1* maka dapat

mengakibatkan *overload*, sehingga otomatis *request* dari *client* di arahkan ke *web server 2* sebagai *server backup*.

Pada studi ini dilakukan penelitian dengan menerapkan *reverse proxy* pada topologi DMZ. Dengan adanya *reverse proxy* dapat meringankan beban kinerja *web server* dan waktu akses menjadi web menjadi lebih cepat. Dengan demikian mencoba mensimulasikan dalam “IMPLEMENTASI *SQUID* SEBAGAI REVERSE PROXY UNTUK KEPERLUAN BACKUP SERVER” yang diharapkan bisa mengatasi apabila *server 1* mengalami kegagalan dapat dialihkan ke *server2* sebagai *server backup*. Manfaat dari uji coba ini adalah dengan adanya *reverse proxy* dapat membantu web dalam meminimalkan terjadinya *overload* pada *web server* dan dapat merancang system untuk mengoptimalkan performa dari *web server*.

## KAJIAN PUSTAKA

### Web server

*Web server* merupakan software yang digunakan untuk menerima *request* HTTP atau HTTPS dari suatu *client*. Dan ditugaskan untuk membuat berupa halaman-halaman web yang berextensi dokumen HTML diantaranya yaitu *apache* dan *microsoft information service(IIS)*. *Web server* juga menjadi software yang menjadi *backbone* dari *world wide web* (Sugeng, 2010).

### Server Proxy

*Server proxy* adalah *server* yang berada antara *client* dan aplikasi. Manfaat *proxy server* diantaranya meningkatkan kinerja jaringan dan untuk mengatur situs mana yang diperbolehkan untuk dikunjungi dan yang situs yang dilarang untuk mengaksesnya. *Proxy server* mempunyai fungsi utama diantaranya *connection Sharing* (jaringan lokal yang dapat terhubung ke internet jika telah melewati suatu *gateway*), *filtering* (untuk menjaga jaringan), *caching* digunakan untuk tempat penyimpanan file-file yang pernah diakses sebelumnya. Sedangkan jenis-jenis *proxy* berdasarkan cara kerjanya diantaranya *forward proxy* (*proxy* yang berada antara *client* dan internet yang digunakan untuk *caching* halaman web yang sudah pernah dikunjungi), *Open proxy* (*forward proxy server* yang diakses oleh user di internet untuk semua keperluan. *Reverse proxy* berada diantara *web server* dan *client* yang berfungsi sebagai *web acceleration* dan sebagai *front-end* untuk mengcontrol dan melindungi akses menuju ke *web server*. *Reverse proxy* bekerja pada port 80. *Reverse proxy* bertugas yang mengendalikan *request* diarahkan ke *server1* atau *server 2*. Konsep *reverse proxy* agar *client* hanya mengetahui IP dari *reverse proxy* tidak dapat mengetahui IP dari *web server*, sehingga *web server* aman terhindar dari pencurian data (Wagito, 2005).

Tabel 1. Perbedaan reverse proxy dengan proxy biasa

No	Reverse Proxy	Proxy Biasa
1.	<i>Reverse Proxy</i> menyetting jaringannya langsung diservernya	<i>Proxy biasa</i> menyetting jaringannya di <i>clientnya</i>
2.	<i>Reverse proxy</i> berjalan di port 80 untuk melayani <i>request</i> Http.	<i>Proxy</i> yang berjalan di port 8080
3.	Sebuah <i>reverse proxy</i> menyembunyikan identitas <i>server</i> .	<i>Proxy biasa</i> Menyembunyikan identitas klien
4	<i>Server</i> yang menyediakan layanan perantara antara <i>client</i> dengan <i>web server</i>	<i>server</i> yang menyediakan layanan perantara antara <i>client host</i> dengan <i>server</i> lain

Pada tabel 1 dijelaskan tentang perbedaan dari *Reverse proxy* dengan *proxy* biasa salah satunya perbedaan dalam penyettingan jaringannya terdapat di *server* nya dan sebaliknya jika *proxy* biasa penyettingannya terdapat di sisi *clientnya*.

### Apache

Menurut Sugeng (2010) Apache adalah *web server* yang paling banyak diminati. Apache didesain pertama kali untuk sistem operasi UNIX. Apache versi berikutnya mengeluarkan programnya yang dapat dijalankan dengan *microsoft windows NT*. Kelebihan menggunakan *web server apache* yaitu termasuk *freeware*.

### GNS (Graphical Network Simulator 3)

GNS3 adalah aplikasi untuk melakukan simulasi jaringan komputer. Di GNS3 biasanya melakukan simulasi untuk jaringan yang sudah kompleks dengan menggunakan sistem operasi asli dari perangkat jaringan seperti *cisco* membuat kondisi lebih *real* dalam melakukan konfigurasi (Dewannanta, 2013).

### Virtualbox

Virtualbox adalah suatu perangkat lunak *opensource* untuk membuat virtualisasi dan simulasi. Dari sistem operasi komputer yang sedang berjalan tanpa ada gangguan dari aktifitas sistem operasi yang sebenarnya (Pranantyo, 2012).

### Squid

*Squid* adalah software sebagai *proxy server* dan *web cache*. *Squid* banyak ditemui untuk mempercepat penyimpanan *request*, penyimpanan alamat DNS yang diminta *client* secara terus menerus permintaan yang terus menerus, sehingga tidak perlu diambilkan dari *server* aslinya dan *squid* memiliki kemampuan dalam menangani beban yang terlalu besar. *Squid* juga mempunyai kemampuan memperkecil penggunaan *bandwith* dan

mempercepat waktu respon halaman web karena sudah pernah dikunjungi sebelumnya (Rafiudin, 2008).

### Jmeter

Apache jmeter dikembangkan oleh *Apache software foundation (ASF)*. Digunakan untuk sebagai alat uji beban untuk menganalisa dan mengukur kinerja berbagai layanan pada aplikasi web. Jmeter juga digunakan untuk koneksi database *JDBC*, *FTP*, *LDAP*, layanan web, *JMS*, *HTTP*, koneksi *TCP generik*. Hal ini dilakukan untuk beberapa pengujian fungsional. Jmeter umumnya dilakukan untuk uji coba performa penggunaan web server dan untuk memudahkan untuk melihat hasil laporan pengujian berupa grafik, tabel sehingga ditampilkan lebih menarik.

### Httpperf

Httpperf adalah alat untuk mengukur kinerja server web. Sistem web yang diuji terdiri dari server web, sejumlah client, dan jaringan yang menghubungkan client ke server. Httpperf menyediakan berbagai pilihan seperti --server (berupa nama dan alamat IP dari suatu situs web), --rate menentukan rata-rata koneksi dalam jumlah permintaan HTTP per detik. Yang menunjukkan jumlah client bersamaan yang mengakses server dan --num-conns menentukan banyak koneksi HTTP yang dibuat selama pengujian, semakin tinggi jumlah koneksi semakin lama uji coba yang dilakukan (Sung-Jae Jung, 2011).

## METODE

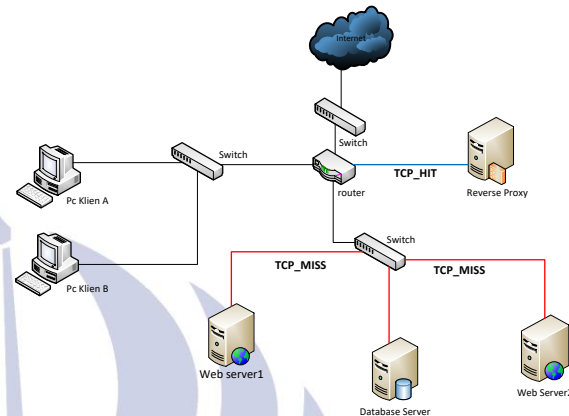
### Analisa Sistem

Analisa sistem yang akan dirancang adalah implementasi Squid sebagai reverse proxy untuk keperluan backup server. Analisa sistem hal yang terpenting dalam merancang sistem, merupakan konsep awal dari suatu sistem yang akan dibangun untuk mendapatkan gambaran awal dari sistem yang dibuat dan langkah yang dilakukan pada konfigurasi tersebut. Reverse proxy digunakan untuk melakukan konfigurasi web server ke satu dengan web server2 dapat terhubung, sehingga apabila web server 1 terlalu banyak menerima request dari client, hal tersebut dapat mengakibatkan web server1 overload, sehingga request dialihkan ke server kedua sebagai server backup seperti terjadinya failover pada web server. Apabila salah satu web server mengalami kegagalan maka otomatis dialihkan ke web server lain. Dengan memanfaatkan reverse proxy maka kinerja web server tersebut akan meningkat dan terhindar dari overload.

Dalam membuat rancangan jaringan pertama yang dilakukan instalasi virtualbox, GNS3, dan OS pada tiap komputer di virtualbox, setelah terinstal OS dilanjutkan dalam penginstalan masing-masing aplikasi

pada komputer berdasarkan fungsi setiap server, setelah itu baru melakukan konfigurasi pada aplikasi-aplikasi tersebut. Setelah semua aplikasi berjalan, terakhir melakukan pengujian kinerja web server dengan menggunakan Jmeter dan httpperf

### Desain Topologi



Gambar 1. Desain Topologi

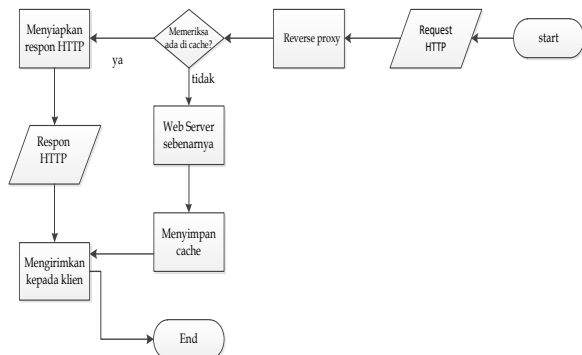
Pada Gambar 1 terdapat 4 buah server dan 2 client, server tersebut diantaranya web server1, web server2, database server, squid dan 2 client windows xp. Dan pada gambar tersebut menunjukkan alur jalannya sistem reverse proxy. Dilihat dari topologi diatas termasuk sebagai redirect. Pada reverse proxy ini mempunyai satu interface yang bertugas mengarahkan ke jaringan yang dibawahnya yaitu ke web server1 sebagai server pertama dan server kedua sebagai server backup. Apabila server pertama mengalami kegagalan.

Ketika client A mengunjungi suatu web, maka Client A telah melakukan request, request tersebut tidak diarahkan ke web sebenarnya tetapi diarahkan ke reverse proxy terlebih dahulu untuk di cek. Reverse proxy ini sebagai pengendali request nantinya diarahkan ke server1 atau server2. Apabila request tersebut telah ada di reverse proxy maka akan langsung dibalas ke client yang biasa dikenal TCP\_HIT dalam log squid yang artinya salinan valid dari objek yang direquest telah ada dalam cache, jika request tersebut tidak terdapat di reverse proxy maka akan diteruskan ke web server 1 biasa disebut TCP\_MISS (Objek yang di request sudah tidak ada dalam cache). Selanjutnya balasan request dari server satu diarahkan kembali ke reverse proxy untuk disimpan sebelum dibalas ke client. Ketika banyak request yang diterima di server 1 hal tersebut dapat mengakibatkan overload, maka request tersebut akan dialihkan ke server kedua sebagai server backup.



## Perancangan Sistem

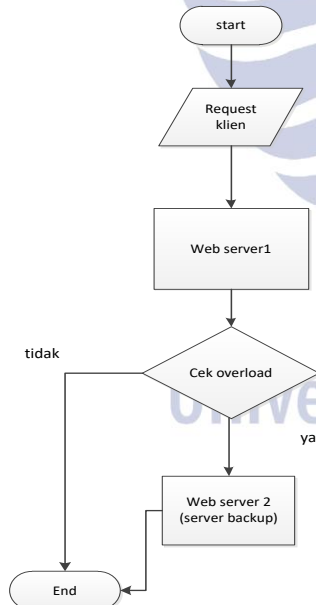
### 1. Perancangan alur kerja reverse proxy



Gambar 2. Perancangan alur kerja reverse proxy

Pada gambar 2 menjelaskan tentang sistem ini dirancang untuk menggambarkan bagaimana *reverse proxy* ini berjalan. Dengan mengirimkan *request HTTP* ke *reverse proxy* nantinya akan dicek terlebih dahulu oleh *cache proxy*, apabila *request http* dari *client* tersebut ada maka akan langsung dibalas ke *client*, apabila *request* tersebut tidak ada dicache maka akan diambilkan dari *web server* sebenarnya, setelah itu dikembalikan lagi ke *reverse proxy* untuk disimpan di *cache* dan barulah *request* tersebut dibalas ke *client*.

### 2. Perancangan alur kerja backup server



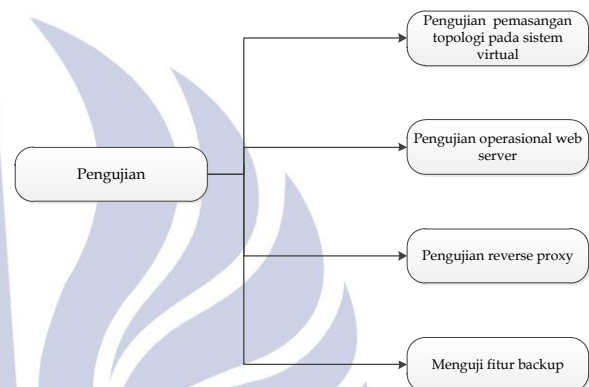
Gambar 3. Perancangan alur kerja backup server

Pada Gambar 3 menjelaskan tentang Sistem ini dirancang untuk menggambarkan bagaimana *backup server* ini berjalan. Dengan adanya *server backup* data dari sebuah system dapat diduplikasi dan dicadangkan sehingga data cadangan dapat digunakan

kembali pada saat *server* mengalami kegagalan. Dengan mengirimkan *request HTTP* dari *client* ke *server*. Jika dalam hal ini *request* tersebut terlalu banyak yang datang, sehingga membuat beban *server* bertambah sehingga jika dibiarkan terlalu lama dapat mengakibatkan *overload*, Apabila *web server* pertama mengalami *overload* maka otomatis akan diarahkan ke *web server* kedua sebagai *server backup*.

## Skenario Pengujian

Dalam studi ini topologi dan sistem yang berhasil dipasang akan diuji validitas dan oprasional konfigurasinya. Berikut adalah skenario pengujian yang dilakukan:



Gambar 4. Skenario pengujian

1. Pada Gambar 4 dijelaskan Pengujian pemasangan *topologi* pada sistem *virtual*, dilakukan untuk mendeteksi *topologi* sudah berjalan dengan lancar waktu *play topologi* di GNS 3, jika dalam pemasangan tersebut terdapat kesalahan seperti tidak dapat *play vm nya*, maka perlu melakukan penyetingan kembali.
2. Pengujian operasional *web server* dilakukan dengan mencoba mengakses *web server*, dan mencoba mengakses fitur yang ada di *web server* tersebut. Uji coba penambahan data di *server 1* maka otomatis dapat terlihat juga penambahan data di *server* kedua.
3. Pengujian *reverse proxy* dilakukan dengan mencoba mengirim *request HTTP* ke *web server*, apabila dalam *request* tersebut terdapat *request* yang sudah pernah dilakukan, maka tidak perlu mengambil dari *server* asli cukup mengambil di *reverse proxy*.
4. Menguji fitur *backup* dilakukan dengan mencoba mengirim banyak *request* ke *web server 1*, sehingga *web server 1* terjadi *overload*, maka *request* dari *client* otomatis akan dialihkan ke *webserver* kedua sebagai *server backup*. Dapat juga dengan mematikan *server 1* maka otomatis *request* diarahkan ke *server* kedua.

## HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mencegah *web server* supaya tidak *overload*, disebabkan banyaknya *client* yang mengakses web, dan untuk menanggulangi masalah tersebut maka dibuatlah penelitian yang berjudul implementasi *squid* sebagai *reverse proxy* untuk keperluan *backup server*. Berikut adalah hasil setting yang dilakukan:

1. Setting *Web server1* dan *Web server2*
  - a) Konfigurasi IP address
  - b) Konfigurasi host dan hostname pada *web server1*
  - c) Membuat web dan meletakkan di *var/www*
  - d) Pengaturan hak akses untuk web
  - e) Pengaturan Name dari localhost menjadi domain
  - f) Konfigurasi *nameserver*
2. Setting database server
  - a) Konfigurasi IP address
  - b) Konfigurasi hosts dan hostname
  - c) Konfigurasi bind-address di *my.cnf*
  - d) Konfigurasi penambahan data admin, data bengkel, produk\_baru pada database hnynd
  - e) Pengaturan hak akses mysql
  - f) Konfigurasi hostname *server 1* dan *server2* di *phpmyadmin*
  - g) Konfigurasi *nameserver*
3. Setting *squid server*
  - a) Konfigurasi IP address
  - b) Konfigurasi hosts dan hostname di *squid server*
  - c) Konfigurasi config *squid* sebagai reverse proxy untuk keperluan backup server
  - d) Konfigurasi *nameserver*
4. Setting router mikrotik
  - a) Konfigurasi IP address tiap-tiap ether
  - b) Konfigurasi DNS server dan DNS Static
  - c) Konfigurasi DHCP Server
  - d) Konfigurasi routing statik
  - e) Konfigurasi filter ruler
  - f) Konfigurasi *dstnat* untuk web agar terforward di laptop host
5. Konfigurasi *client A* dan B
  - a) Konfigurasi IP address
  - b) Penambahan DNS sttic di hosts
6. Berikut ini adalah config *squid* untuk *reverse proxy* sebagai backup server:
 

```
http_port 80 accel defaultsite =www.cahya.net
vhost
forwarded_for on
refresh_pattern ^ftp:1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern . 0 20% 4320
```

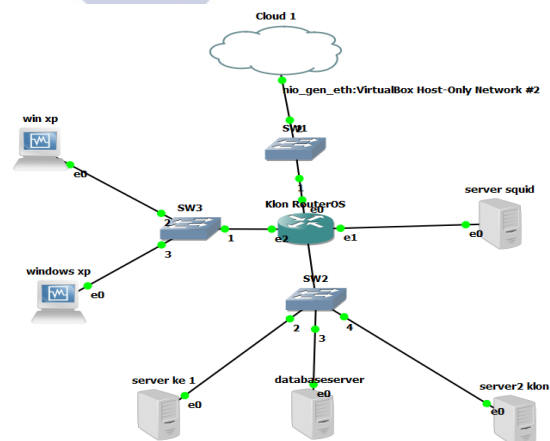
```
#server1
cache_peer 192.168.5.7 parent 80 0 no-query
originserver round-robin name= server1.cahya.net
```

```
#server2
cache_peer 192.168.5.6 parent 80 0 no-query
originserver round-robin name= server2.cahya.net
#server1
acl sites_server1.cahya.net dstdomain
www.cahya.net cahya.net
acl_our_sites dstdomain www.cahya.net
cache_peer_access server1.cahya.net allow
sites_server1.cahya.net
#server2
acl sites_server2.cahya.net dstdomain
www.cahya.net cahya.net
acl_our_sites2 dstdomain www.cahya.net
cache_peer_access server2.cahya.net allow
sites_server2.cahya.net
http_port 192.168.15.3:80 accel defaultsite=
www.cahya.net
http_access allow our_sites2
http_access allow our_sites
```

## Pengujian dan Pembahasan

Pengujian yang dilakukan yaitu sesuai dengan skenario peengujian yang sudah dibuat pada tahap sebelumnya

1. Pengujian yang pertama dilakukan adalah pengujian pemasangan topologi pada sistem *virtual* dilakukan untuk mendeteksi topologi sudah berjalan dengan lancar, jika dalam pemasangan tersebut terdapat kesalahan seperti tidak dapat *play vm* nya, maka perlu melakukan penyettingan kembali. Berikut gambar pemasangan topologi yang telah sukses.



Gambar 5. Pemasangan topologi yang berjalan lancar

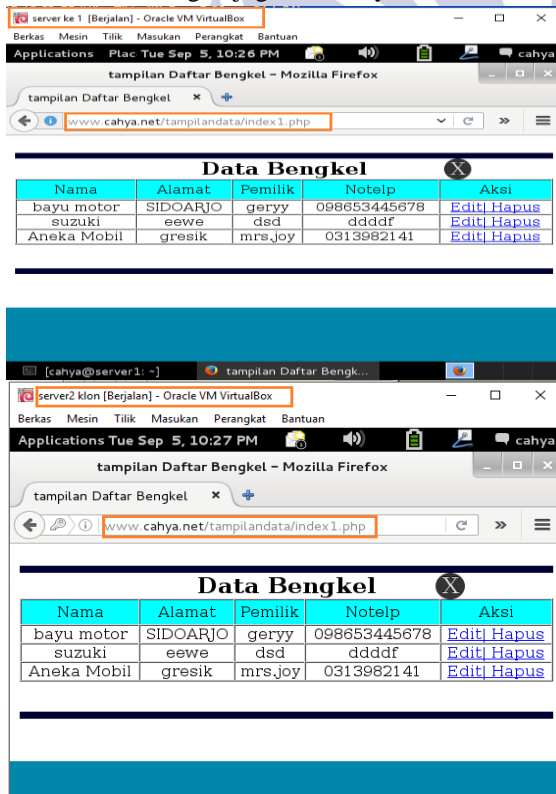
Pada Gambar 5. Menjelaskan tentang topologi yang digunakan saat simulasi:

- a. 192.168.10.8 merupakan IP address client A

- b. 192.168.10.7 merupakan IP *address client* B
- c. 192.168.5.7 merupakan IP statik yang disetting pada *web server1*
- d. 192.168.5.6 merupakan IP statik yang disetting di *web server 2*
- e. 192.168.5.2 merupakan IP statik yang disetting *database server*
- f. 192.168.15.3 merupakan IP statik yang disetting di *squid*

## 2. Pengujian oprasional *web server1* dan *web server2*.

Uji coba oprasional *web server* dimaksud adalah mencoba mengakses web, akses fitur apa saja yang ada di *web server* diantaranya penambahan data bengkel dan produk terbaru. Jika di *web server 1* terdapat penambahan data bengkel, maka perubahan tersebut dapat dikenali di *server 2* begitu juga sebaliknya.

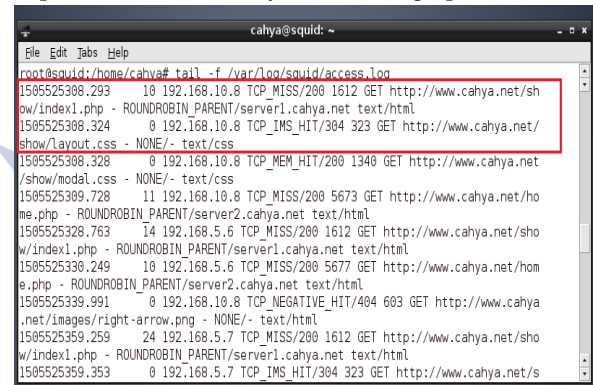


Gambar 6. Tampilan penambahan data bengkel *web server 1* dan *server 2*

Pada Gambar 6 dijelaskan jika *client* melakukan penambahan data pada *server1*, maka perubahan tersebut dapat terlihat juga pada tampilan *server2*

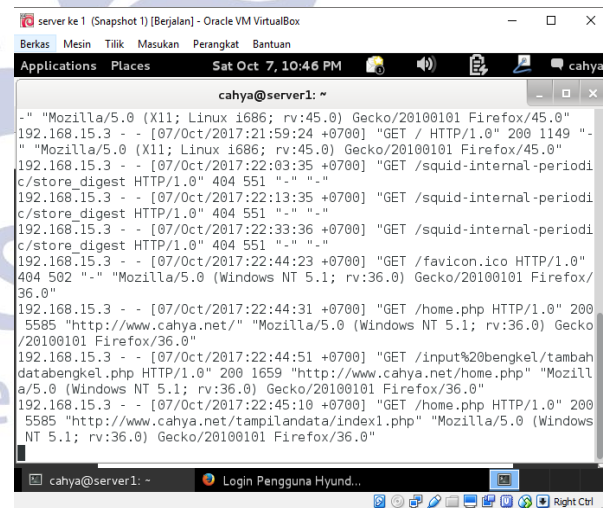
## 3. Pengujian reverse proxy

Apabila *client* mengakses web nantinya *request* bukan langsung diarahkan ke web aslinya melainkan ke *reverse proxy* terlebih dahulu. Di *reverse proxy* jika sudah terdapat *request* maka akan langsung dibales ke *client*. Misalnya *client A* windows XP dengan IP 192.168.10.8 mengakses web [www.cahya.net](http://www.cahya.net) maka *request* diarahkan ke *reverse proxy*. Jadi seolah-olah *client* langsung mengambil *request* dari *server* aslinya. Berikut log *squid*



Gambar 7. Tampilan log *squid*

Pada gambar 7 dapat dijelaskan tampilan log saat melakukan uji coba reverse proxy, dapat diketahui IP 192.168.10.8 telah melakukan *request* ke [www.cahya.net](http://www.cahya.net) namun diarahkan ke reverse proxy terlebih dahulu.



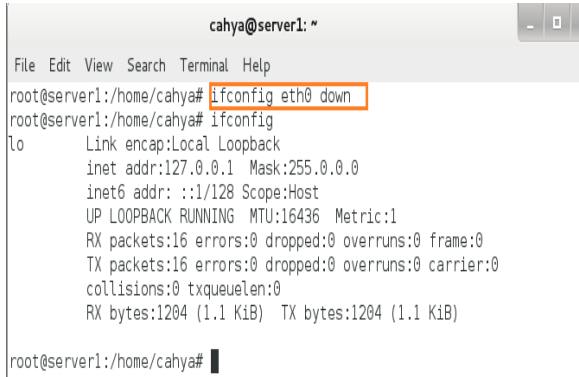
Gambar 8. Tampilan log apache2 di *server 1*

Pada gambar 8 dapat dijelaskan hasil tampilan log apache sewaktu klien A dengan IP 192.168.10.8 mengakses [www.cahya.net](http://www.cahya.net), maka hasil log nya menunjukkan IP yang mengakses adalah IP reverse proxy.



#### 4. Pengujian fitur backup

Apabila suatu *web server* mengalami banyak *request* dari *client* maka beban *web server* akan bertambah besar jika terlalu lama dijalankan dapat mengakibatkan *server* dapat *overload* bahkan *down*. Jika *server 1* mengalami *overload* dapat dialihkan ke *server* ke dua sebagai *server backup* atau dengan menonaktifkan *eth0 web server 1* maka otomatis diarahkan ke *server 2* sebagai *server backup*



```

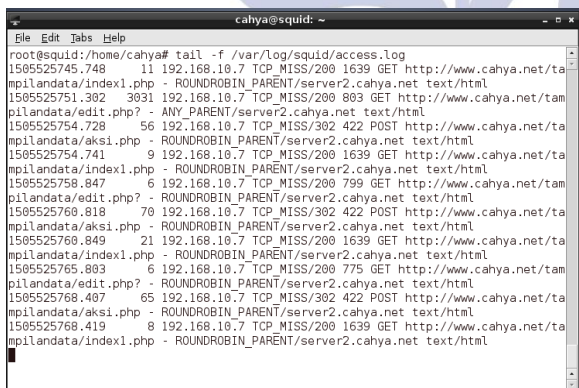
cahya@server1: ~
File Edit View Search Terminal Help
root@server1:/home/cahya# ifconfig eth0 down
root@server1:/home/cahya# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1204 (1.1 KiB)  TX bytes:1204 (1.1 KiB)

root@server1:/home/cahya#

```

Gambar 9. Perintah non aktifkan *eth0* di *web server1*

Pada Gambar 9 dijelaskan perintah untuk menonaktifkan interface di *eth0*. *Ifconfig eth0 down* maka otomatis *server1* tidak bisa digunakan maka jika ada *request* masuk otomatis diarahkan ke *server2*



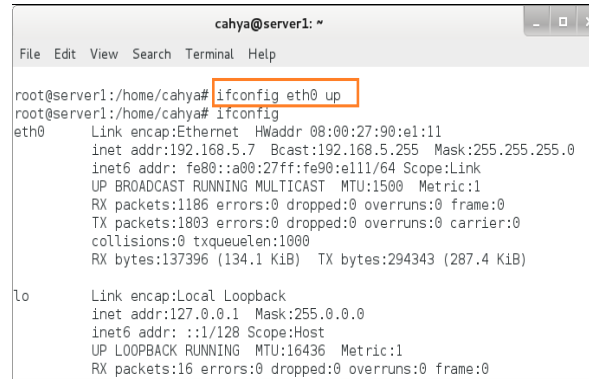
```

cahya@squid: ~
File Edit Tabs Help
root@squid:/home/cahya# tail -f /var/log/squid/access.log
1505525745.748 11 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
mpiladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525751.302 3031 192.168.10.7 TCP_MISS/200 803 GET http://www.cahya.net/tam
piladata/edit.php? - ANY_PARENT/server2.cahya.net text/html
1505525754.728 56 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
piladata/aksi.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525754.741 9 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
mpiladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525758.847 6 192.168.10.7 TCP_MISS/200 799 GET http://www.cahya.net/tam
piladata/edit.php? - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525760.818 70 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
mpiladata/aksi.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525760.849 21 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
mpiladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525765.803 6 192.168.10.7 TCP_MISS/200 775 GET http://www.cahya.net/tam
piladata/edit.php? - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525768.407 65 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
mpiladata/aksi.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525768.419 8 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
mpiladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html

```

Gambar 10. Log *squid* saat terjadi backup *server*

Pada Gambar 10 merupakan hasil tampilan log saat *server1 eth0* di nonaktifkan. Jika ingin dikembalikan lagi di aktifkan lagi *server1* settingan *eth0* dengan *ifconfig eth0 up* seperti gambar berikut:



```

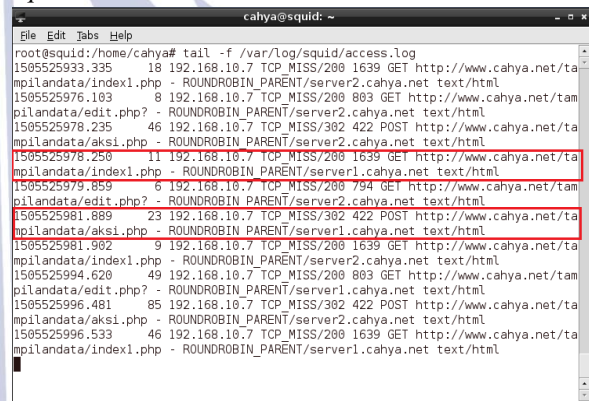
cahya@server1: ~
File Edit View Search Terminal Help
root@server1:/home/cahya# ifconfig eth0 up
root@server1:/home/cahya# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:90:e1:11
          inet addr:192.168.5.7  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe90:e111/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1803 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:137396 (134.1 KiB)  TX bytes:294343 (287.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0

```

Gambar 11. Perintah aktifkan *eth0* pada *server1*

Pada gambar 11 merupakan perintah untuk mengaktifkan *eth0* dengan *ifconfig eth0 up*, maka hasil log *squid* kembali normal *server1 server1* dan *server2*



```

cahya@squid: ~
File Edit Tabs Help
root@squid:/home/cahya# tail -f /var/log/squid/access.log
1505525933.335 18 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
piladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525976.103 8 192.168.10.7 TCP_MISS/200 803 GET http://www.cahya.net/tam
piladata/edit.php? - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525978.235 46 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
piladata/aksi.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525978.258 11 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
piladata/index1.php - ROUNDROBIN_PARENT/server1.cahya.net text/html
1505525979.859 6 192.168.10.7 TCP_MISS/200 794 GET http://www.cahya.net/tam
piladata/edit.php? - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525981.889 23 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
piladata/aksi.php - ROUNDROBIN_PARENT/server1.cahya.net text/html
1505525981.902 9 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
piladata/index1.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525994.620 49 192.168.10.7 TCP_MISS/200 803 GET http://www.cahya.net/tam
piladata/edit.php? - ROUNDROBIN_PARENT/server1.cahya.net text/html
1505525996.481 85 192.168.10.7 TCP_MISS/302 422 POST http://www.cahya.net/tam
piladata/aksi.php - ROUNDROBIN_PARENT/server2.cahya.net text/html
1505525996.533 46 192.168.10.7 TCP_MISS/200 1639 GET http://www.cahya.net/tam
piladata/index1.php - ROUNDROBIN_PARENT/server1.cahya.net text/html

```

Gambar 12. log access *squid* kembali normal

Pada gambar 12 dijelaskan hasil tampilan log *squid* saat *eth0* telah diaktifkan kembali.

## PENUTUP

### Simpulan

Hasil implementasi yang dilakukan jika terdapat *request* dari klien langsung diarahkan ke reverse proxy terlebih dahulu, reverse proxy yang mengatur jika terdapat *request* yang sama maka langsung di balas ke klien (TCP\_HIT), jika *request* baru akan dikirim ke *server1* atau *server2* menggunakan round-robin terlihat di log *squid* (TCP\_MISS), jika mematikan *server1* maka semua *request* diarahkan ke *server 2* sebagai backup *server*.

### Saran

Saran untuk pengembangan dan penelitian selanjutnya adalah perlu dilakukan implementasi secara real, penambahan protokol web yaitu https, dapat di tambahkan sinkronisasi database dan juga lebih diperhatikan tentang pembobotan *squid* menggunakan sibling

## DAFTAR PUSTAKA

- Apache JMeter. (2017, May 03). Diambil kembali dari Apache: <http://jmeter.apache.org/>
- Asrofi, M. (2011). *Rancang bangun aplikasi implementasi otentikasi squid dalam mode transparant proxy menggunakan fitur URL Rewrite*. Surabaya: Universitas Negeri Surabaya.
- Bullock, T. (2007). Httpperf web workload generator quickstart guide [internet]. Tersedia pada <http://www.comlores.com/httpperf/httpperf-quickstart-guide.pdf>.
- Dewannanta, D. (2013, May 1). *Mengenal Software Simulator Jaringan Komputer GNS3*. Diambil kembali dari Ilmu Komputer: <http://ilmukomputer.org/2013/01/29/gns3/>
- Krisna, A. (2011). *Analisis Pemanfaatan Reverse Proxy untuk meningkatkan efisiensi pelayanan web server*. Yogyakarta: UNIVERSITAS ISLAM INDONESIA YOGYAKARTA.
- MADCOMS. (2016). *Manajemen sistem jaringan komputer dengan mikrotik routerOS*. Yogyakarta: C.V ANDI OFFSET.
- Penulis, T. (2006). *Panduan Penulisan dan Penilaian Tugas Akhir*. Surabaya: Universitas Negeri Surabaya.
- Pranantyo, A. (2012). *Penggunaan Media Pembelajaran Virtual Box Sebagai Upaya Untuk Meningkatkan Kemampuan Siswa Dalam Melakukan Instalasi Sistem Operasi Di SMK Negeri 2 Pengasih*. Yogyakarta: UNIVERSITAS NEGERI YOGYAKARTA.
- Rafiudin, R. (2006). *IP Routing dan Firewall dalam Linux*. Yogyakarta: ANDI OFFSET.
- Rafiudin, R. (2008). *Squid Koneksi Anti Mogok*. Yogyakarta: C.V ANDI OFFSET.
- Sugeng, W. (2010). *Jaringan komputer dengan TCP/IP*. Bandung: Modula.
- Sung-Jae Jung, Y.-M. B. (2011). Web Performance Analysis of Open Source Server Virtualization Techniques . *International Journal of Multimedia and Ubiquitous Engineering* , 1-8.
- Wagito. (2005). *Jaringan komputer teori dan implementasi berbasis linux*. Yogyakarta: