BAB III

PEMBAHASAN

3.1 Kegiatan Kerja Peraktek

Dalam kesempatan ini penulis melakukan kerja praktek di PT MELVAR LINTAS NUSA (MELSA) yang berlokasidi Paskal*Hyper Square* Blok C No. 39-41, Jl. HOS Cokroaminoto (Pasirkaliki)No.25-27Bandung. Kerja praktek yang tercantum dalam surat sekertaris jurusanUNIKOM dilaksanakan selama 1 (satu) bulan yaitu mulai tanggal 29 Juli 2012 sampai dengan 29 Agustus 2012, tetapi dilapangan kerja praktek dilaksanakan selama 3 (bulan) yaitu sampai dengan 20September 2012.

Tabel 3.1 jadwal kerja praktek

No	Waktu & Kegiatan	Juli	Agustus				September			
		IV	I	II	III	IV	I	II	III	IV
1	Pengumpulan Data									
2	Analisis									
3	Perancangan									
4	Implementasi									

3.2 Pengembangan *Sistem*

Pada sub bab pengembangan *sistem*, akan dijelaskan tentang tahapan analisis, perancangan, implementasi.

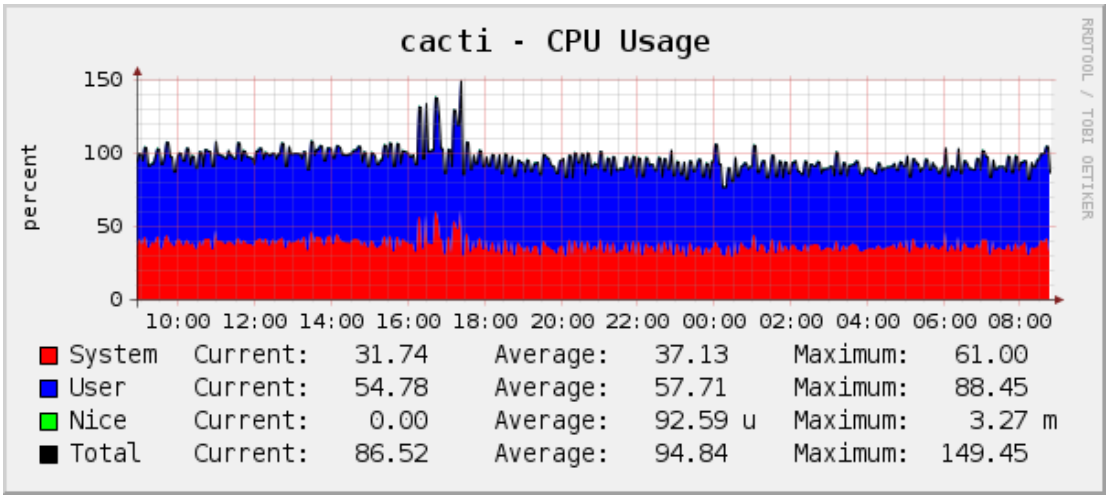
3.2.1 Analisis *Sistem*

Analisis dilakukan untuk memperoleh definisi permasalahan dan gambaran yang tepat tentang meningkatnya permintaan yang berlebihan ke *server*, sehingga terjadinya *overload* di *server*yang berimbas pada kinerja *DNSserver* yang menjadi lambat.

Melakukan analisis atas terjadinya *overload* di *server*bertujuan sebagai dasar perancangan atau perbaikan *sistem*. Dari hasil analisis tersebut dapat diketahui kelemahan atau kekurangan *overload* di *server*dan dapat dirancang atau diperbaiki menjadi sebuah *sistem* atau konfigurasi yang lebih baik dan meliputi analisis kebutuhan fungsional maupun non fungsional.

3.2.1.1 Analisis *Sistem* Yang Berjalan

Saat ini PT MELVAR LINTAS NUSA (MELSA) belum mempunyai suatu *sistem* yang dapat mencegah terjadinya *overload*, sehingga pada saat permintaan yang sangat besar maka kinerja*server*menjadi lambat, oleh karna itu diperlukan sebuah *sistem* yang dapat mencegah terjadinya *overload* di *server*. Berikut adalah grafik terjadinya *DNS overload* yang terdapat pada gambar 3.1 :



Gambar 3.1 DNS Overload

Keterangan :

Dari gambar 3.1bisa di lihat bahwa pemakaian cpu *server* mengalami overload karena grafik dengan warna biru menunjukan di atas 100, hal itu di sebabkan karena jumlah query *DNS* yang sangat tinggi sedangkan sumber daya cpu yang kecil, karena *DNSserver* yang sedang berjalan menggunakan single core, atau prosesor tunggal.

Adapun hasil analisa *sistemDNS* yg ada sekarang adalah sebagai berikut :

1. *DNSserver*yang ada sekarang tidak memiliki *sistem* pembagi beban sehingga sering kali tiap *server* mengalami *overload.*
2. *DNSserver* yang digunakan tidak ada yang berfungsi khusus sebagai *master*, sehingga tiap *DNSserver* tidak saling berhubungan satu dengan yang lainnya.

3. *DNSserver*yang ada sekarang belum memiliki *sistem sync* khusus sehingga jika ada *zone* yang *update*, maka harus di *update* satu per satu secara manual.

3.2.1.2 Analisis *Sistem* Yang Akan Dibangun

DNSserver adalah *sistem* yang sangat penting, karena jika *DNSserver* tidak berjalan dengan optimal maka kinerja dari koneksi *internet* pun akan terganggu, hal itu disebabkan karena setiap permintaan *lookup* domain tidak terpenuhi.

Dengan adanya *serverDNScluster*, maka *requestclient* terhadap aplikasi *DNS* maupun sumber daya yang ada pada *server* akan terjamin meskipun pada saat terjadinyakegagalan *sistem(sistem failure)* dan atau sesuatu yang terjadi diluar dugaan. Secara sederhana dapat digambarkan, bila satu *server* di-dalam grup *cluster* mengalami kegagalan yang mengakibatkan ia menjadi *unavailable* dan atau *server* tersebut sedang dalam keadaan di-*maintenance* sehingga harus di"*offline*"-kan sementara waktu, maka semua sumber daya (data dan aplikasi) yang sebelumnya ada pada *server* tersebut akan berpindah tangan kepada *server* lain yang tidak "*offline*". Dengan kata lain keberadaan *server* pertamaakan tergantikan oleh *server* berikutnya yang *available*, dan para pengguna(*client* / *terminal*) tidak akan mengetahui apakah pada saat mereka melakukan akses, kepada *server* sebelumnya atau kepada *server* berikutnya, yang mereka ketahui adalah bahwa *server* dengan alamat sekian tetap *online* dan dapat di-akses.

Untuk *zone file* yang ada di *master* dan mengantisipasi *update* manual *filezone* jika ada *update* subdomain,atau*domain* baru maka, penulis menerapkan *sistemmirroring* di *dnsserver*. *rsync* adalah *fileopensource* program *transfer* untuk *sistemUnix* yang menggunakan "algoritma *rsync*" yang menyediakan metode yang sangat cepat untuk *file* sinkronisasi dan direktori dari satu lokasi ke lokasi lain, penulis juga akan mengotomatisasi tugas ini menggunakan *cronjob* tanpa interaksi pengguna sehingga program bisa berjalan dengan waktu yang telah di tetapkan.

3.2.1.3 Analisis Kebutuhan Non Fungsional

Analisis non fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non fungsional.Spesifikasi kebutuhan non fungsional digunakan untuk menunjang *sistem* yang sudah ada dan dikembangkan untuk mendapatkan *sistem* yang labih baik.Kebutuhan non fungsional meliputi analisis perangkat keras, analisis perangkat lunak dan analisis pengguna.

3.2.1.3.1 Analisis Perangkat Keras (*Hardware*)

Berdasarkan hasil pengamatan diketahui bahwa perangkat keras yang terdapat di Melsa ada pada gambar 3.2 :



Gambar 3.2 Perangkat Keras DNS Server

1. *Intel(R) Pentium(R) 4 CPU 2.40GHz Memori 2 GHz.*
2. *Harddisk*80 GB.
3. *Video Graphic Adapter* (VGA) 64 MB
4. *Network Interfaces Card* (NIC) dengan kecepatan 1000/100/10 Mbps.

Sedangkan perangkat keras yang dibutuhkan dalam pembangunan*DNSCluster* adalah :

1. *Prosessor Intel(R) Xeon(TM) CPU 2.80GHz*
2. *Memori 2 GHz.*
3. *Harddisk 2X80 GB.*
4. *Video Graphic Adapter (VGA) 64 MB*
5. *Network Interfaces Card (NIC) dengan kecepatan 1000/100/10 Mbps.*

Perangkat keras yang di butuhkan sebanyak 2 unit untuk menunjang *DNScluster*, pada gambar 3.3 :



*Gambar 3.3 P*erangkatKeras *ServerDNS*

PerangkatKeras*Router Load Balancing*

Perangkat keras *Router Load Balancing*, penulis telah mempersiapkan *router* dengan spesifikasi sebagai berikut :

1. ***CPU*** ***:*** *AR7161 680/800MHz CPU.*
2. ***Memory*** ***:****32MB DDR SDRAM onboard memory.*
3. ***Boot loader*** ***:*** *RouterBOOT.*
4. ***Data storage*** ***:*** *64MB onboard NAND memory chip.*
5. ***Ethernet*** ***:*** *Five 10/100/1000 gigabit ethernet ports*
6. ***MiniPCI*** ***:*** *none.*

7. ***Extras*** ***:*** *Reset switch, Beeper.*
8. ***Serial port*** ***:*** *no serial port.*
9. ***LEDs*** ***:*** *Power, NAND activity, 5 Ethernet LEDs.*
10. ***Power options*** ***:*** *Power over Ethernet: 9-28V DC*
11. ***Dimensions*** ***:*** *113x89x28mm. Weight without packaging and cables: 130g.*
12. ***Power consumption*** ***:*** *Up to 3W.*
13. ***Operating Sistem*** ***:*** *MikroTik RouterOS v3, Level 4 license.*



Gambar 3.4 PerangkatKeras*Router Load Balancing*

Berdasarkan penelitian dan analisa, spesifikasi perangkat keras yang ada di MELSA sudah mencukupi untuk perancangan *SistemDNSServer* Clustering.

3.2.1.3.2 Analisis Perangkat Lunak (*Software*)

Dalam analisis kebutuhan perangkat lunak, dibutuhkan beberapa *software* yang mendukung untuk digunakan dalam perancangan *sistemDNS* clustering dan dapat dilihat pada Tabel 3.2

Tabel 3.2Perangkat lunak DNSServer Melsa

Uraian	**Perangkat Lunak *DNSServer* Melsa**
Sistem Operasi	*FreeBSD*
Aplikasi *DNS*	*Bind*

Tabel 3.3Analisis perangkat lunak (software)

Uraian	**Perangkat Lunak Yang dibutuhkan**
Sistem Operasi	*Linux Ubuntu 12*
Aplikasi DNS	*BIND*
Tools	*Rsync*
Bahasa Pemrograman	*Bash*
Load Balancing	*RouterOS 5.14*

Berdasarkan penelitian dan analisa, spesifikasi perangkat lunak yang ada di MELSA sudah mencukupi untuk perancangan *SistemDNSServerClustering*

3.2.1.3.3 Analisis Pengguna

Berikut adalah analisis pengguna keterkaitan dalam proses maintenance, pembuatan zone *DNS* baru, dan troubleshooting dapat dilihat pada Tabel 3.4.

Tabel 3.4 Analisis pengguna Sistem yang sedang berjalan

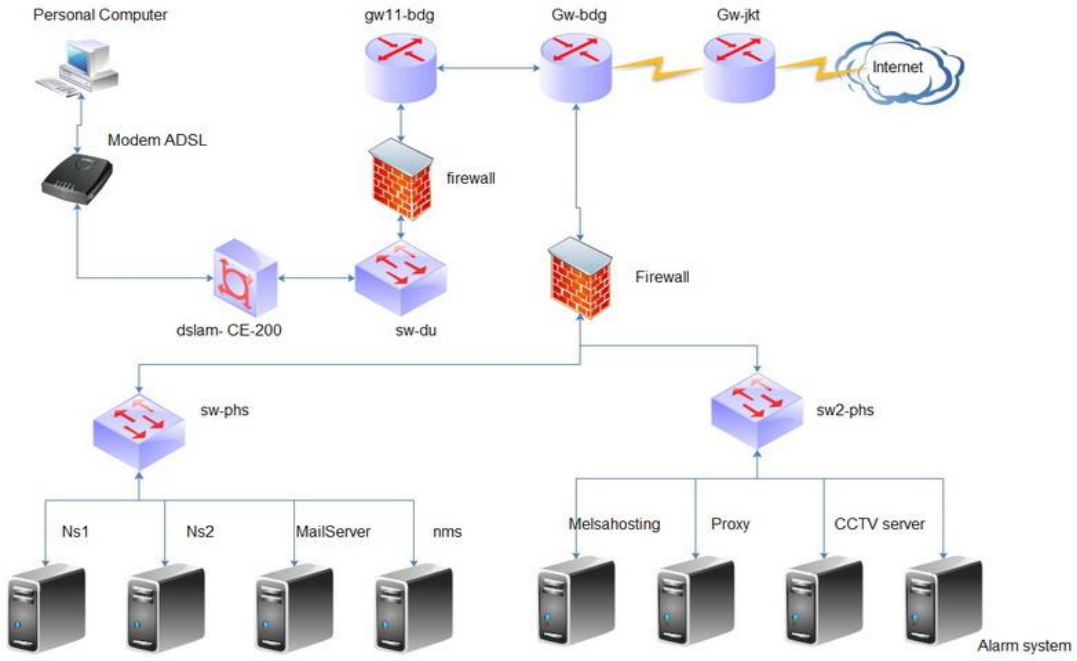
Pengguna	Tanggung Jawab
SysAdmin	Maintenace dan Troubleshooting *sistem*
Helpdesk	Memonitor *sistem* yang sedang berjalan
Network Engineer	Memonitor trafik yang sedang berjalan
Technical Support	Input Zone baru dan subdomain

3.2.2 Pengembangan *SistemDNS Clustering*

Pada sub bab pengembangan perangkat lunak, akan dijelaskan tentang tahapan analisis, perancangan, implementasi serta pengujian dari *serverdnsclustering* yang akan dibangun.

3.2.2.1 Topologi *DNSserver*

Berikut topologi jaringan MELSA yang menghubungkan antara *serverDNS* dan *internet* di gambar 3.5 :



Gambar 3.5Topologi DNSserver

Dari gambar 3.5 bisa dilihat bahwa *sistemDNS* yang sedang berjalan saat ini, yaitu, *Ns1* dan *Ns2* tidak saling melakukan *Sinkronisasi*, artinya jika di lakukan *update* di *server* ns1 maka harus di lakukan juga *update* di ns2, di karenakan tidak adanya pembagi beban maka salah satu dari *server* tersebut sering mengalami kelebihan beban (*overload*)

3.2.2.2 Metode *Load balancing* Untuk Pembagian Beban

Load balancing 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* digunakan pada saat

sebuah *server*telah memiliki jumlah *user* yang telah melebihi maksimal kapasitasnya. *Load balancing* juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, *link* jaringan, *CPU, hard drive,* atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.

Metode yang digunakan penulis adalah *PCC (Per Connection Clasifier)*, *PCC*mengambil*fields* yang dipilih dari*header IP*, dan denganbantuan sebuahalgoritma*hashing*mengubah*field* yang dipilihkenilai 32-bit. Nilai inikemudiandibagiolehpenyebuttertentu dansisanyakemudiandibandingkan dengan*Remainder*, jika samamaka paketakanditangkap. Disini penulis menambahkan *dst-port* 53 sebagai *portdefault* dari *DNSserver* itu sendiri.

3.2.2.3 Tools *Sinkronisasi Untuk File Zone*

Rsync adalah adalah sebuah program kecil yang berfungsi hampir sama dengan *rcp*, namun memiliki fitur tambahan yang lebih banyak dan menggunakan protokol rsync remote *update* untuk mempercepat transfer file bila file tujuan sudah ada.

Membuat sebuah skrip sesi *FTP* atau berbagai bentuk skrip *transferfile* lainnya, *rsync* hanya menyalinkan perbedaan *file-file* saja, mengkompresinya dan mengirimkannya menggunakan *ssh* jika anda ingin keamanan.

Berikut ini adalah beberapa buah fitur tambahan *rsync* :

1. Mendukung penyalinan *link*, *device*, *owner*, grup, dan permisi.
2. *Option exclude* dan *exclude-from* serupa dengan *GNUtar*.

3. Sebuah *mode CVS exclude* untuk mengabaikan *file-file* yang sama yang akan diabaikan oleh *CVS.*
4. Dapat menggunakan sembarang *remote shell* transparan, termasuk *rsh* atau *ssh.*
5. Tidak membutuhkan *root privilege.*
6. Dukungan bagi *server rsync anonymous* atau *authenticated.*

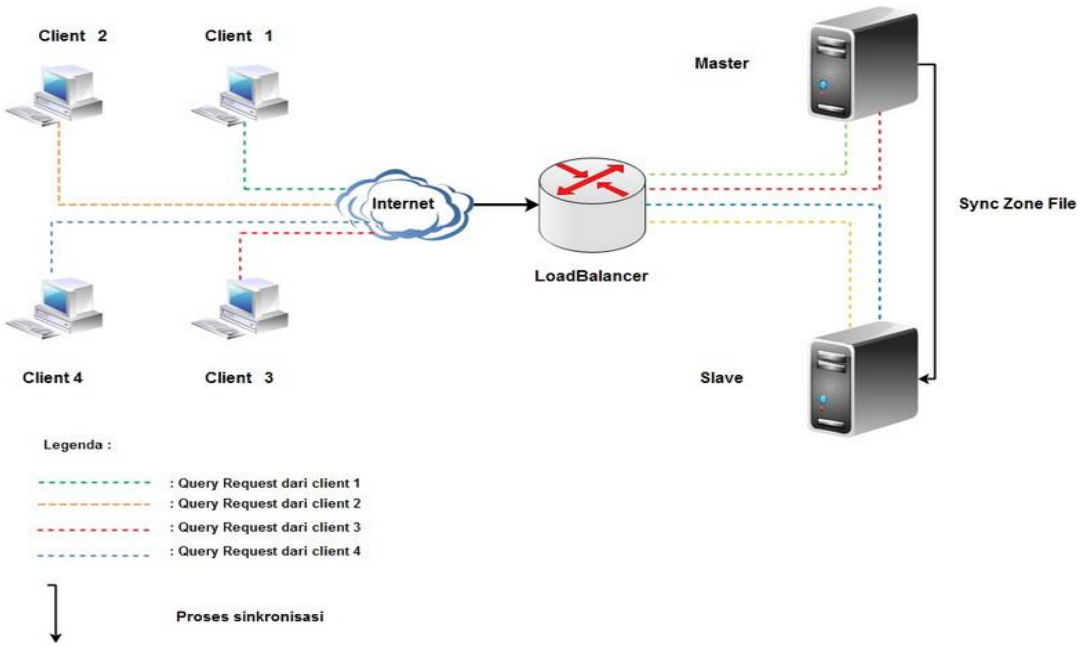
3.3 Perancangan *SistemDNS Cluster*

Perancangan sistem dns cluster merupakan tahap untuk menentukan jumlah server yang akan digunakan dan metode apa yang akan di terapkan.

3.3.1 Perancangan Jaringan *DNS Cluster*

Dalam perancangan jaringan *DNS cluster*, penulis menggunakan dua buah *server*, dimana satu unit *server* difungsikan sebagai *Master* dan satu unit *server* lainnya sebagai *client*. *Routerloadbancing* bertugas mendistribusikan permintaan klien dan mengoptimalkan kinerja *DNSserver* agar tidak terjadinya kelebihan beban, dalam perancangan, loadbalancing di letakan di antara klien dan *server*.

Loadbalancing menggunakan *metode* yang disebut PCC (*PerConnectionClassifier)*, yaitu nilai dari jumlah *ip* dan dengan bantuan sebuah algoritma *hashing* mengubah *field* yang dipilih ke nilai 32 *bit*, nilai ini kemudian dibagi oleh penyebut tertentu dan sisanya kemudian di bandingkan dengan sisa, jika sama maka paket di proses. Seperti yang terlihat pada gambar 3.6 :



Gambar 3.6 Perancangan Jaringan *ServerDNS*

Dapat di lihat dari gambar 3.6, traffic request dari client-client menuju *DNSserver*, akan di bagi rata oleh loadbalancing ke setiap *DNSserver*. Sehingga bisa mencegah overload pada satu *server*, dan bisa juga jika salah satu *serverofflinerequest* dari *client* masih bisa terpenuhi. Kemudian dengan adanya *toolsrsync* maka proses *mirroringfilezone* dan *configbind* lainnya bisa diduplikasi dengan cepat tanpa harus melakukan secara *manual*, karena *toolsrysnc* terpicu jika ada perubahan data modifikasi pada file *configbind*.

Di gambar 3.5 juga di tunjukan kalo *server master* dengan *slave* saling berhubungan melaluli media kabel *peer-to-peer*, hal ini hal ini diperuntukkan agar proses *mirror* bisa lebih cepat dan optimal, karena dengan *peer-to-peer*, tingkat *delay* yang di dapat sangat kecil itu artinya sangat bagus untuk proses *transfer file*.

3.3.2 Perancangan Alamat IP Address

Dalam perancangan alamat *ip address*, penulis hanya mengikuti pengalokasian alamat *ip* yang telah di tentukan oleh pihak perusahaan baik di sisi *client* maupun di sisi *server*, berikut adalah alamat *ip address* untuk *router* dan *server*yang dijelasakan ditabel3.5 :

*Tabel 3.5*Perancangan Alamat IP Address

Hardware	IP	Gateway
Mikrotik RB750G	Eth1 : 10.10.10.254/24 Eth2 : 10.10.20.254/24 Eth3 : 202.138.224.3/25	202.138.224.1
DNSServer 1	Eth0 : 10.10.10.1/24	10.10.10.254
DNSServer 2	Eth0 : 10.10.20.1/24	10.10.10.254

3.4 Implementasi

Pokok bahasan implementasi akan menjelaskan berbagai hal yang memiliki pengaruh dalam mewujudkan terbentuknya *sistemdnsclustering*, seperti konfigurasi *bind*, *mikrotik* dan pembuatan *script* pendukung*toolsrysnc*.

3.4.1 Konfigurasi *Bind*

BIND adalah *DNSServer* yang umum digunakan dimana mana dan biasanya berjalan dibawah *platform unix/linux*. Untuk mengatur agar *BIND* memanfaatkan *DNS*Melsa sebagai *DNS*Utama, maka semua permintaan *NS*dapat

di arahkan oleh *BIND (forward request)* langsung ke *DNS*Melsa. Instalasi sebagai berikut :

Konfigurasi file-file yang harus diperhatikan untuk mengkonfigurasikan BIND, antara lain:

1. */etc/named.conf* (berisi keterangan letak dan jenis database yangdibutuhkan oleh BIND).
2. */home/admin/melsahostingdb/melsahosting.com*(berisi file reverse zone dari melsahosting.com)
3. */home/admin/melsahostingdb/melsahosting*(berisi file zone dari melsahosting.com)
4. */etc/resolv.conf*: (berisi alamat domain atau alamat IP dari name*server*).

File konfigurasi di ambil dari *server* yang sudah berjalan agar proses sync bisa optimal dengan tools rsync. Data konfigurasi terlampir

3.4.2 Konfigurasi *Rsync*

Rsync adalah *tool* untuk *transfer* dan sinkronisasi file atau *tree* (struktur direktori dan file) secara satu arah, baik *transfer* lokal (di *sistem* yang sama) maupun remote (jaringan/internet). *Rsync* merupakan *tools* kecil yang *powerful* karena bisa bertindak sebagai *server* sekaligus sebagai klien.

Rsync bertujuan agar file *Zone* yang sudah *update* di *DNSserver Primary* bisa secara otomatis di *update* juga di *server secondary* sehingga semua *DNS* record di kedua *server* sama, berikut script bash yang di buat khusus untuk *rsync*

1. **Buat file dengan nama */home/admin/bin/sync.sh***

```
#!/bin/bash

IPSYNC=ns4.melsa.net.id;

LOGFILE=/var/log/rsync.log;

/usr/bin/rsync  -auz --progress --log-file=$LOGFILE
admin@$IPSYNC:/home/admin/gibodb/gibo /home/admin/gibodb/gibo;

NEWSIZE=`tail -1 $LOGFILE|awk '{print $7}'`;

OLDSIZE=`cat /temp/oldsize`;

if [ "$NEWSIZE" != "$OLDSIZE" ]; then

        echo $NEWSIZE > /temp/oldsize;

        sudo /usr/sbin/rndc reload

fi;

/usr/bin/rsync  -auz --progress --log-file=$LOGFILE
admin@$IPSYNC:/home/admin/namedb/Zone /home/admin/namedb/Zone;

NEWSIZE1=`tail -1 $LOGFILE|awk '{print $7}'`;

OLDSIZE1=`cat /temp/oldsize.1`;

if [ "$NEWSIZE1" != "$OLDSIZE1" ]; then

        echo $NEWSIZE1 > /temp/oldsize.1;

        sudo /etc/init.d/bind9 reload

fi;
```

2. **Agar file tersebut dapat di eksekusi maka dilakukan perubahan *mode***

```
root@nms:/home/siddiq# chmod 777 /home/admin/bin/sync.sh
```

Agar software tersebut *update* secara berkala maka wajib di tambahkan di *crontab*

```
*/15    *       *       *       *       admin   /home/admin/bin/sync.sh
```

3.4.3 Konfigurasi Mikrotik

Loadbalance 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. Untuk melakukan konfigurasi dasar mikrotik mulai dari konfigurasi *IP address, gateway*.

1.Konfigurasi IP Address

IP ⇨ *Address* ⇨ *add* ⇨ (input ip Address)

2. Konfigurasi Dns yang akan di gunakan

IP ⇨ *DNS* ⇨ *Set* ⇨ (input ip dns)

3. Konfigurasi Gateway

IP ⇨ *Route* ⇨ *add* ⇨ *gateway*(⇨ input ip dns)

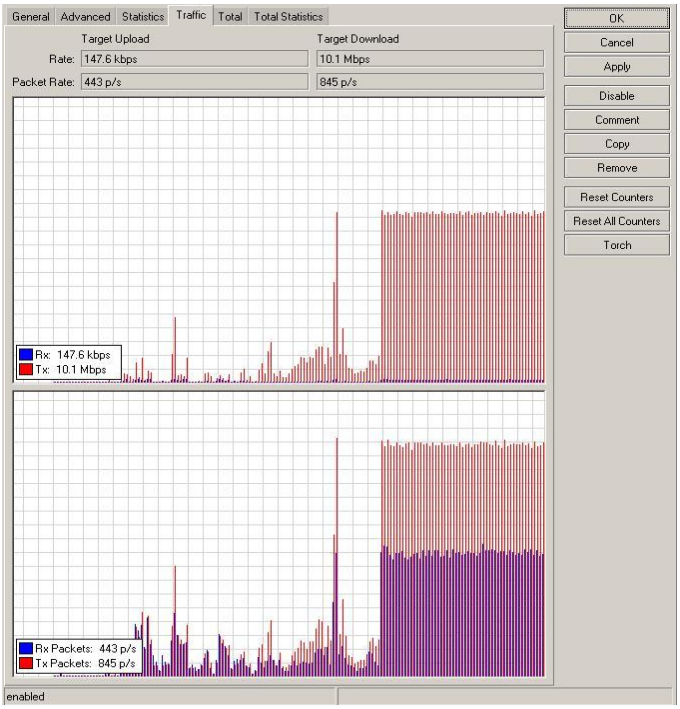
4. Konfigurasi Mangle

IP ⇨ *Firwall* ⇨ *mangle* ⇨ *add* ⇨ (input mangle)

Atau bisa juga dengan menggunakan *command*yang terlampir

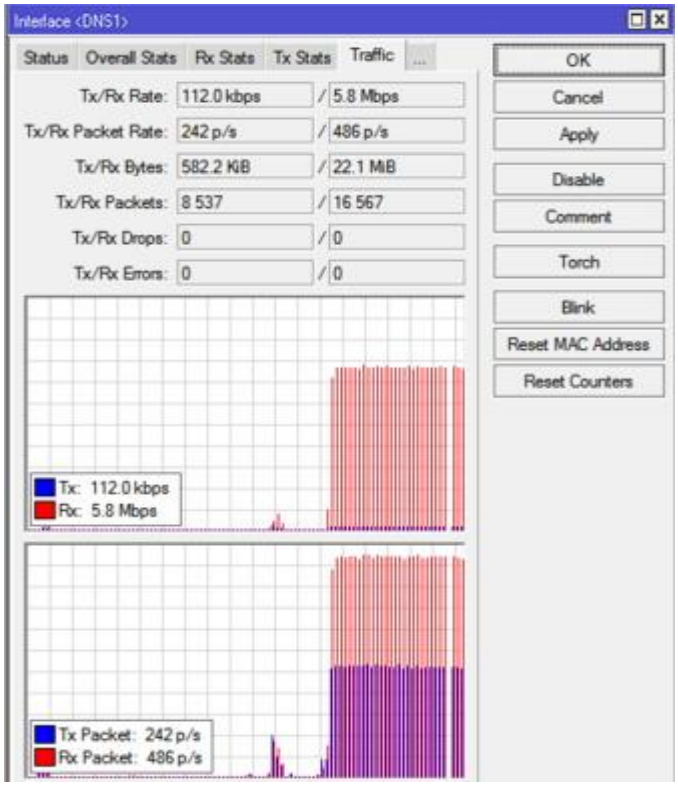
3.5 Pengujian

Pengujian dilakukan dengan cara mem *forward* semua paket di *port* 53 (service*DNS*) dari client ke IP *DNS* cluster yang sedang di uji coba, maka dapat di lihat di Gambar 3.5:



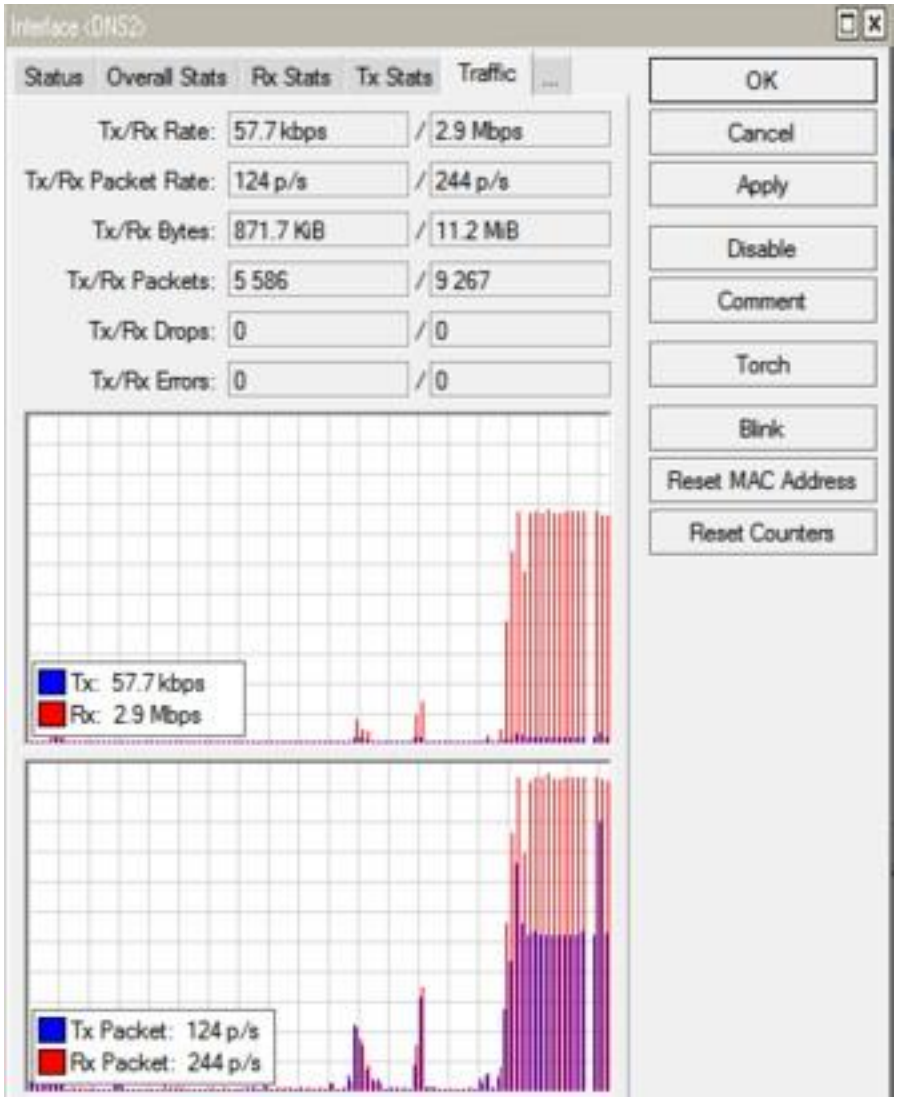
Gambar 3.7 Traffik request ke interface DNS server

Dari gambar 3.5 bisa di lihat traffik yang masuk ke *interface dns server* sebesar *10mbps*, maka dengan otomatis *serverloadbalancing* membagi kepada dua *interface* tujuan *dnsserver*, dapat di lihat di gambar 3.6



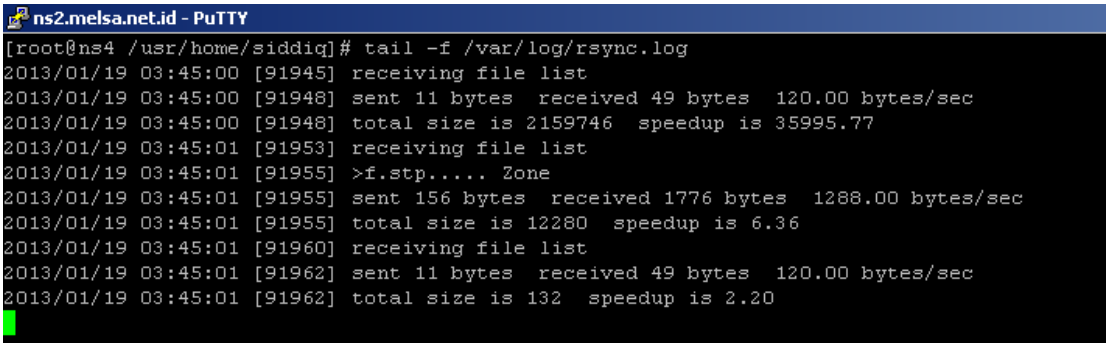
*Gambar 3.8*Uji Coba Interface*DNS* 1 Mikrotik

Jumlah *requestDNS* menuju *IPloadbalancing* dari *client* yaitu 10 mbps, dan bisa dilihat di gambar 3.8 uji coba interface *DNS* 1, traffik yang masuk ke *interface* tersebut sebesar *5.8Mbps*, hal tersebut di sebabkan oleh proses *loadbalancing*dari *routermikrotik*, dan sisa dari traffik, di alihkan ke *DNS*2 di *interfaceDNS*2 seperti pada *Gambar 3.7*



*Gambar 3.9*Uji Coba Interface *DNS* 2 Mikrotik

Untuk memastikan *sistemsync* berjalan dengan seharus nya maka penulis menyediakan *log* untuk setiap aktifitas dari *rsync*, setelah di lakukan *updatezone* pada *serverns1 (master)* maka yang terjadi di *ns2 (slave)* terjadi proses *transfer* atau dengan kata lain bahwa *filezone* di *serverns2* mengalami perubahan, bisa di lihat di gambar 3.10



Gambar 3.10.log proses rsync

Pengujian lain dilakukan secara objektif dimana diuji secara langsung ke setiap pengguna. Dari pengujian yang dilakukan menggunakan metode wawancara dengan menggunakan teknik kualitatif mengenai Performance *DNSserver* setelah menggunakan *sistem*cluster pada Tabel 3.6.

Tabel 3.6 Narasumber wawancara

Nama Narasumber	Jabatan	Waktu	Tempat
Edy Purwanto	Sysadmin	Selasa, 17 Desember 2012 Pukul 13.00	Melvar Lintas Nusa

Berikut daftar pertanyaan yang diajukan kepada narasumber:

1. Menurut Anda dengan adanya *sistemDNS*cluster apakah query *DNS* jadi lebih cepat ?
2. Menurut Anda dengan adanya *sistemDNS*cluster beban cpu *DNS* jadi lebih kecil atau sama saja ?

3. Perbandingan dari sebelum adanya *sistemDNS*cluster apakah user yang complain bertambah banyak atau jadi menurun ?
4. Menurut Anda apakah *sistemDNS*cluster ini membantu beban pekerjaan helpdesk, admin dan staff lain ?

Dari hasil wawancara tersebut, menghasilkan jawaban dari narasumber. Rincian hasil jawaban yang diperoleh dari hasil wawancara kepada narasumber dapat dilihat pada tabel 3.7.

Tabel 3.7 Jawaban Narasumber (Admin)

No	Pertanyaan dan Jawaban
1	Menurut Anda dengan adanya system dns cluster apakah query dns jadi lebih cepat ?
	Narasumber :
	iya. Setelah adanya sistem dns tersebut permintaan query dns menjadi lebih cepat dan stabil
2	Menurut Anda dengan adanya system dns cluster beban cpu dns jadi lebih kecil atau sama saja ?
	Narasumber :
	dengan adanya sistem tersebut beban cpu yang awal nya sering overload di 100% menjadi menurun, dan sekarang lebih sering di bawah 30persen
3	Perbandingan dari sebelum adanya system dns cluster apakah user yang complain bertambah banyak atau jadi menurun ?
	Narasumber :
	dengan adanya sistem ini, jumlah user yang mengeluh lambat dan melaporkan dns offline menjadi sedikit bahkan tidak ada sama sekali
4	Menurut Anda apakah system dns cluster ini membantu beban pekerjaan helpdesk, admin dan staff lain ?
	Narasumber :
	ya dengan adanya sistem ini beban perkerjaan untuk staff helpdesk dan admin menjadi lebih ringan, karena staff sedikit menerima keluhan dari pelanggan atau pengguna dns

Maka dari hasil wawancara dengan narasumber sysadmin Melvar Lintas Nusa , menyatakan bahwa pembangunan *sistemDNS* sluster ini sudah sesuai dengan kebutuhan melsa, yang mana dari segi fungsi sangat membantu kinerja dan meringankan beban perkerjaan helpdesk dan meringankan beban.

3.6 Monitoring

Dalam aktivitas *monitoring* penulis melakukan pengamatan setelah proses pembuatan mesin *Linux* dan semua paket paket program yang telah di sebutkan diatas selesai . Adapun kegiatan-kegiatan selama monitoring meliputi:

1. *Monitoring log* /var/log/local0 dan /var/log/security.log untuk mengetahui jumlah *query DNS* yang masuk ke *server* dan termasuk *error* jika *sistem* mengalami kerusakan.
2. *Monitoring log cron* untuk mengetahui jika ada *errorsistem* otomatisasi.
3. Mengamati performa *server* setelah *sistem* yang dibuat sudah digunakan dilingkungan kerja PT MELVAR LINTAS NUSA khususnya efek terhadap pelanggan.