

BAB IV

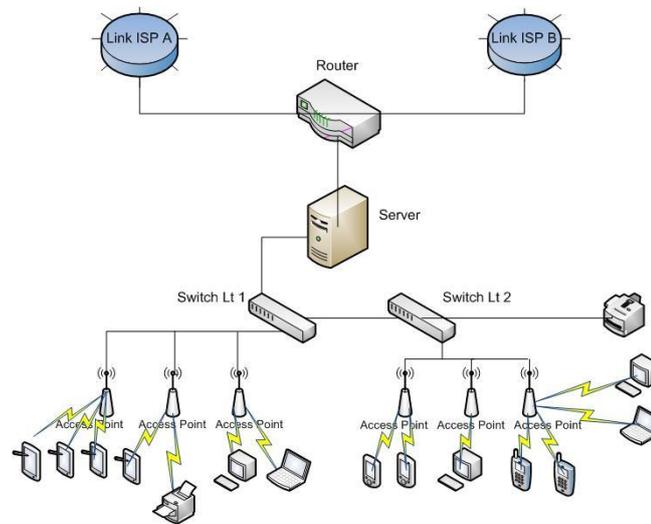
RANCANGAN JARINGAN USULAN

4.1 Jaringan Usulan

Load Balancing adalah teknik untuk membagi beban *traffic* pada dua atau lebih jalur koneksi secara seimbang, oleh karena itu penulis mengusulkan untuk menerapkan teknologi *load balancing* agar masalah *bandwidth* yang selama ini menjadi kendala bisa teratasi. Teknologi yang saat ini diterapkan hanyalah *failover* yaitu apabila *link* utama terganggu maka akan otomatis memfungsikan *link* kedua, teknologi tersebut tidak penulis hilangkan tetapi ditambah dengan *load balancing* dan *manajemen bandwidth*. Penerapan *load balancing* nya sendiri menggunakan metode ECMP (*Equal Cost Multi Path*) karena saat ini PT.Astra International,Tbk – Auto2000 Cempaka Putih menggunakan dua ISP dengan *bandwidth* yang sama sehingga dengan metode ECMP tiap proses akan melewati kedua *gateway* dengan beban yang sama pada masing-masing *gateway*.

4.1.1 Topologi Jaringan

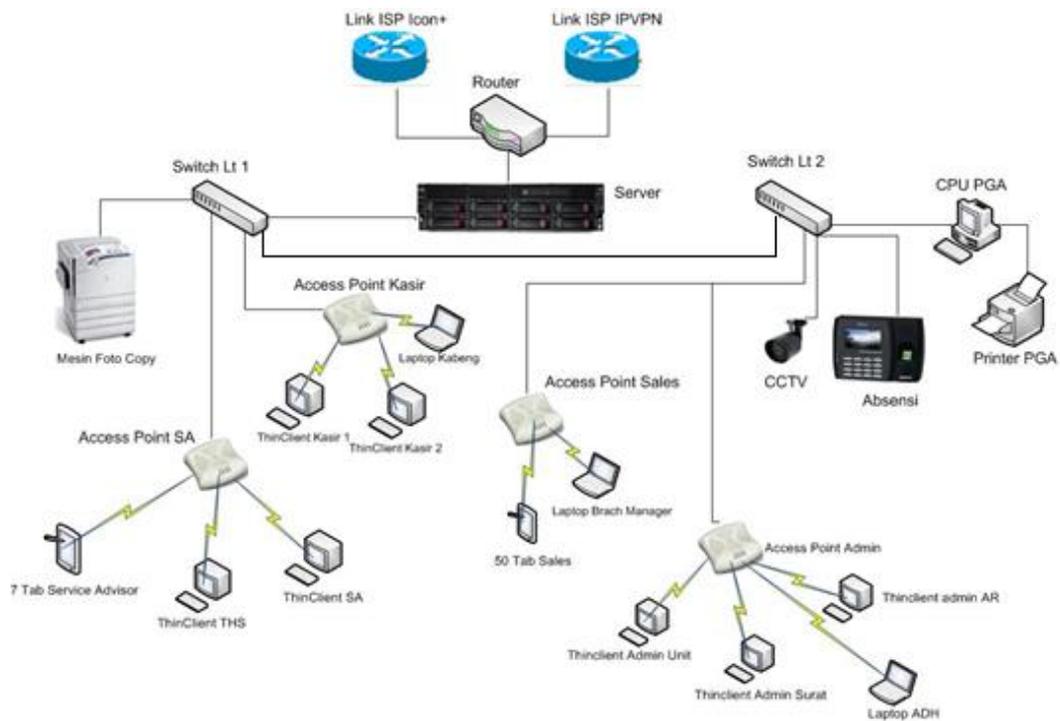
Penulis tidak merubah topologi yang saat ini berjalan, hanya menambahkan teknologi *load balancing* dengan metode ECMP agar pemakaian dua ISP menjadi maksimal. Berikut topologi usulan dalam gambar :



Gambar IV.1 Topologi Jaringan Usulan

4.1.2 Skema Jaringan

Tidak banyak yang berubah dari skema jaringan usulan dikarenakan infrastruktur yang sudah ada cukup bagus. Berikut skema jaringan usulan penulis :



Gambar IV.2 Skema Jaringan

Dari skema diatas maka langkah awal yang diperlukan dalam implementasi *load balancing* ECMP yang sesuai dengan rancangan topologi yang penulis buat ditahap desain adalah konfigurasi *interface*, konfigurasi NAT, konfigurasi mangle, konfigurasi *routing*, manajemen jaringan. Semua hal tersebut akan penulis bahas di sub-bab berikutnya.

4.1.3 Keamanan Jaringan

Selain keamanan jaringan yang sudah pernah penulis bahas di bab 3 yaitu menggunakan antivirus symantec *enduser protection*. Penulis akan membahas tentang di dalam router mikrotik sendiri terdapat *firewall* yang bisa dimanfaatkan, adapun fungsi dari *firewall* adalah sebagai berikut :

1. NAT (*Network Address Translation*)

Salah satu fungsi *firewall* pada mikrotik yang bertugas melakukan perubahan IP Address pengirim dari sebuah paket data.

2. Mangle

Fungsinya adalah menandai paket data dan koneksi tertentu yang dapat diterapkan pada fitur *firewall* yang ada di mikrotik.

3. Packet Filtering

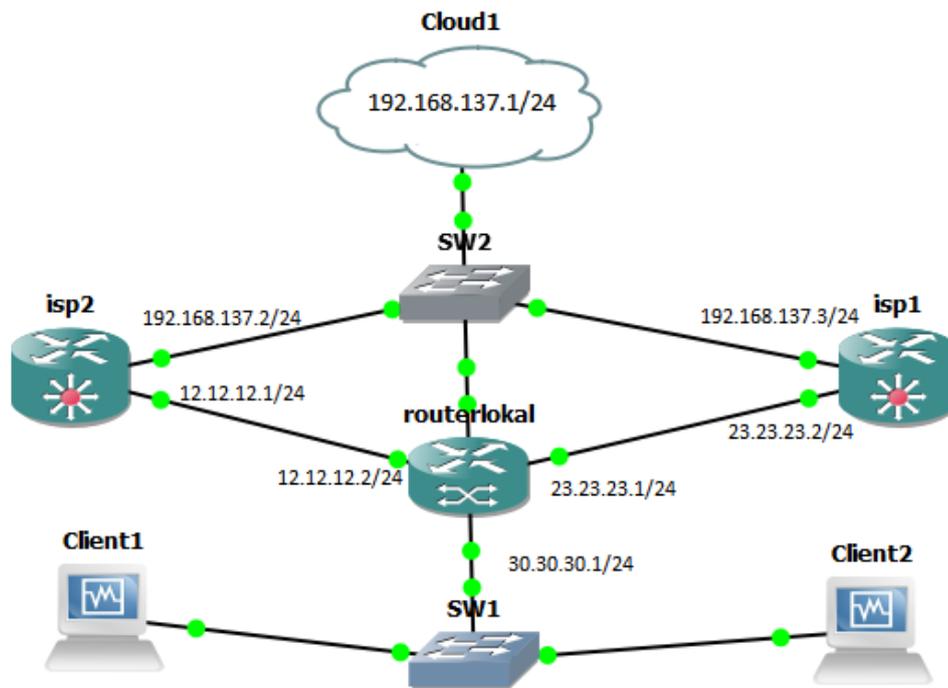
Berfungsi memblokir paket data jaringan sesuai dengan peraturan yang sudah di tentukan.

4. *Proxy*

Fungsi nya adalah untuk menyediakan layanan dan meneruskan setiap permintaan kepada server di *internet*. Dengan *proxy* maka setiap identitas IP komputer menjadi tersembunyi karena yang dikenali server yang di *request* adalah IP *proxy* kita.

4.1.4 Rancangan Aplikasi

Pada sub-bab ini kita bahas lebih detail mengenai perancangan load balancing dengan metode ECMP, berikut gambaran sederhana untuk konfigurasi load balancing disertai failover.



Gambar IV.3 Rancangan Load Balancing

Berikut adalah tahapan untuk konfigurasi Load balancing dan juga failover :

1. Konfigurasi Router ISP1

Konfigurasi ini dilakukan melalui virtualbox yang sudah di setting sebagai ISP1, setelah masuk kedalam mikrotikOS setting router :

```
[admin@Mikrotik]> system identity set name=ISP1
```

```
[admin@ISP1]> ip address add address=192.168.137.2/24 interface=ether1
```

```
[admin@ISP1]> ip address add address=12.12.12.1/24 interface=ether2
```

```
[admin@ISP1]> ip dns set servers=8.8.8.8
```

```
[admin@ISP1]> ip dns set allow-remote-request=no
[admin@ISP1]> ip dns set allow-remote-request=yes
[admin@ISP1]> ip route add gateway=192.168.137.1
```

2. Konfigurasi Router ISP2

```
[admin@Mikrotik]> system identity set name=ISP2
[admin@ISP1]> ip address add address=192.168.137.3/24 interface=ether1
[admin@ISP1]> ip address add address=23.23.23.2/24 interface=ether2
[admin@ISP1]> ip dns set servers=8.8.8.8
[admin@ISP1]> ip dns set allow-remote-request=no
[admin@ISP1]> ip dns set allow-remote-request=yes
[admin@ISP1]> ip route add gateway=192.168.137.1
```

3. Konfigurasi Router

```
[admin@Lokal]> ip address add address=12.12.12.2/24 interface=ether1
[admin@Lokal]> ip address add address=23.23.23.1/24 interface=ether2
[admin@Lokal]> ip address add address=192.168.137.4/24 interface=ether3
[admin@Lokal]> ip address add address=30.30.30.1/24 interface=ether4
```

4. Konfigurasi PC Client1

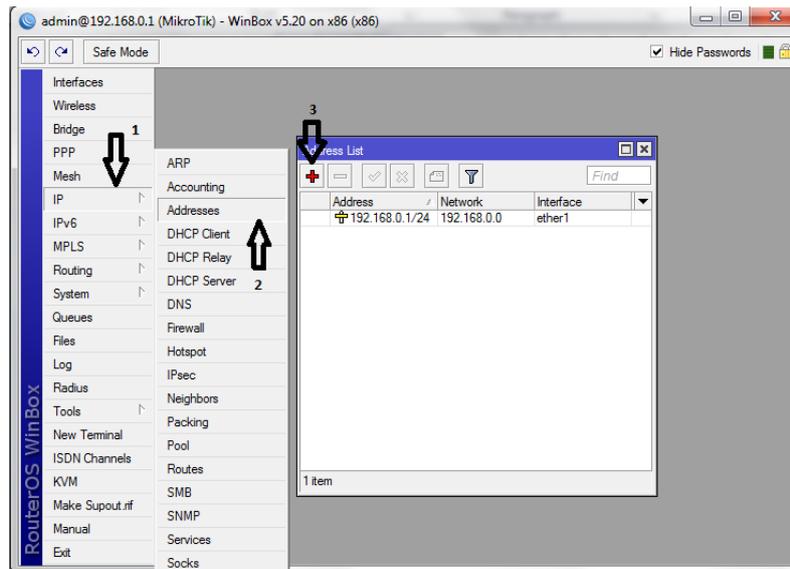
Buat IP untuk Client 1 menjadi IP 30.30.30.2 dengan gateway 30.30.30.1

5. Konfigurasi PC Client2

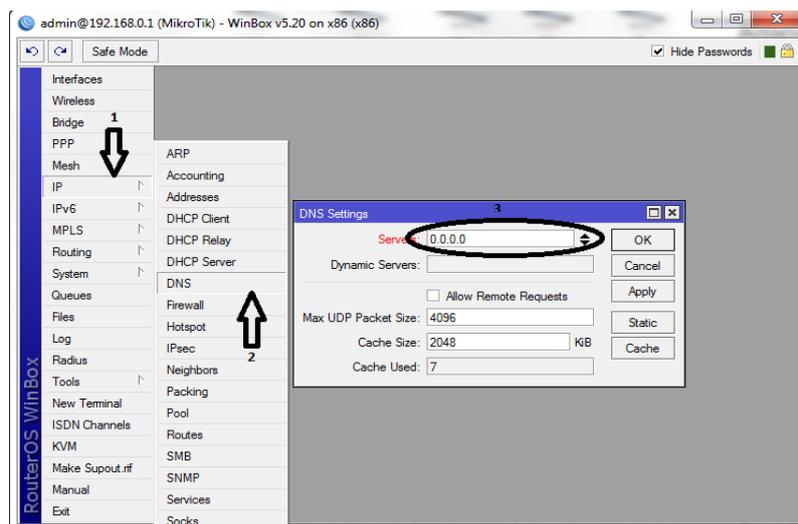
Buat IP untuk Client 1 menjadi IP 30.30.30.3 dengan gateway 30.30.30.1

6. Konfigurasi Interface

Pertama-tama kita harus memberikan IP Address dan DNS pada interface router mikrotik. Berikut tahap pemberian IP Address an DNS :



Gambar IV.4 Setting IP Address



Gambar IV.5 Setting DNS

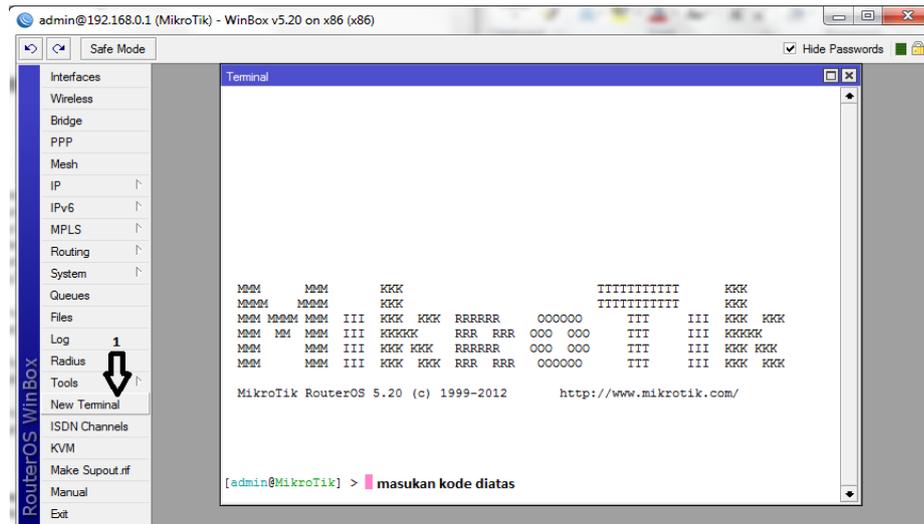
7. Konfigurasi NAT (*Network Address Translation*)

Setting NAT penting karena agar komputer server dan client dapat menggunakan internet. NAT akan mengubah alamat sumber paket yang memiliki IP private supaya dapat dikenali oleh jaringan internet dengan cara mentranslasikan menjadi IP Address publik. Berikut tahapan setting NAT :


```

add chain=output connection-mark=ispmain_conn action=mark-routing new-
routingmark=ke_isp-main 5
add chain=output connection-mark=ispbackup1_conn action=mark-routing newrouting-
mark=ke_isp-backup1
add chain=output connection-mark=ispbackup2_conn action=mark-routing newrouting-
mark=ke_isp-backup2
/ip route
add dst-address=0.0.0.0/0 gateway=192.168.1.88 routing-mark=ke_ispmain
add dst-address=0.0.0.0/0 gateway=120.189.44.53 routing-mark=ke_ispbackup1
add dst-address=0.0.0.0/0 gateway=202.67.41.51 routing-mark=ke_ispbackup2

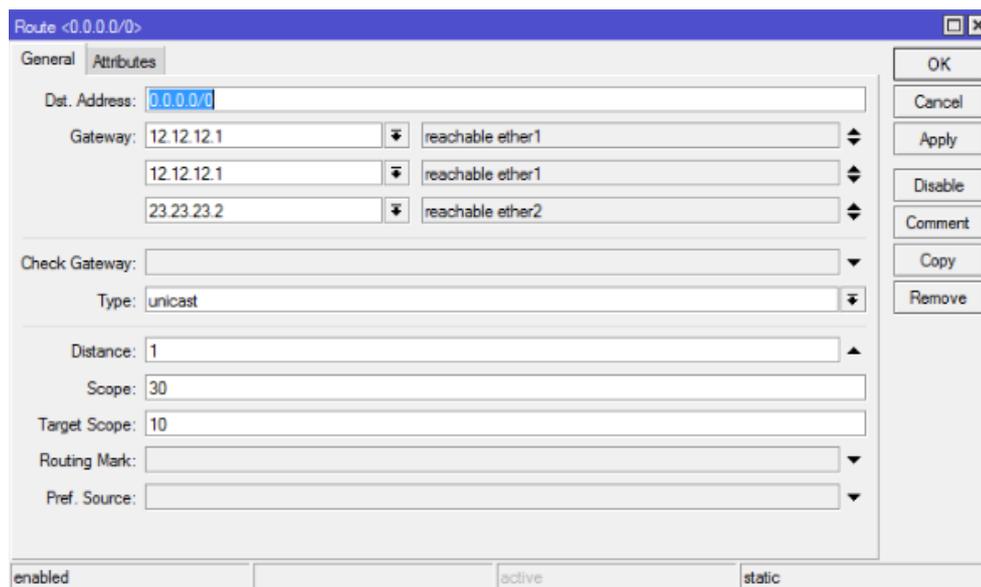
```



Gambar IV.8 Konfigurasi Route

9. Konfigurasi Load Balance ECMP

Terakhir adalah setting load balance pada router seperti gambar berikut :



Gambar IV.9 Konfigurasi Load Balance

4.1.5 Manajemen Jaringan

Saat ini jaringan auto2000 menggunakan manajemen *failover* dengan dua ISP yaitu jika satu ISP putus maka akan langsung digantikan oleh ISP *backup*, jadi *bandwidth* yang tersedia tetap sama dengan ISP pertama oleh karena itu tidak akan ada perubahan dari sisi *performance client*. Penulis ingin merancang *load balancing* dengan manajemen *bandwidth*. Garis besar nya adalah perusahaan tetap menggunakan dua ISP yang tersedia, tetapi penggunaanya dimaksimalkan dengan metode *load balancing* sehingga *bandwidth* yang tersedia menjadi dua kali lipat dari sebelumnya karena kedua *bandwidth* ISP terpakai. Untuk teknologi *failover* nya sendiri tidak di hilangkan tetapi digabungkan dengan teknologi *load balancing*.

4.2 Pengujian Jaringan

4.2.1 Pengujian Jaringan Awal

Untuk mengetahui besaran *speed download* dan *speed upload* serta besaran nilai ping ISP yang digunakan. Pengujian ini akan dilakukan sebanyak empat kali pada server yang sama. Menguji jaringan awal dengan melakukan tes ping menggunakan CMD (*Command Promt*) dengan perintah tes ping 8.8.8.8 –t untuk menguji dan mengetahui apakah terdapat jeda pada ISP *primary* di jam sibuk pengujian jaringan bisa dilihat pada gambar berikut :



Gambar IV.10 Hasil Speedtest Jaringan Awal

Data –data hasil pengujian *speedtest* akan dibuat dalam bentuk table seperti table berikut :

Table IV.1
Hasil Pengujian Speed Download dan Upload Awal

No	Ping	Speed Download	Speed Upload
1	14 ms	0.80 Mbps	1.06 Mbps
2	12 ms	0.79 Mbps	1.04 Mbps
3	13 ms	0.67 Mbps	0.98 Mbps
4	16 ms	0.44 Mbps	0.99 Mbps

```

C:\Windows\system32\cmd.exe - ping 8.8.8.8 -t
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=21ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=27ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=23ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=25ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=21ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=18ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=2281ms TTL=55
Reply from 8.8.8.8: bytes=32 time=19ms TTL=55
Reply from 8.8.8.8: bytes=32 time=21ms TTL=55
Reply from 8.8.8.8: bytes=32 time=21ms TTL=55

```

Gambar IV.11 Hasil Tes Ping Pengujian Awal

Dapat dilihat pada gambar tes *bandwidth* diatas saat jam sibuk *traffic* beban cabang sangat kecil dikarenakan hanya mengandalkan satu ISP sedang kan ISP yang lain hanya sebagai *backup*. Sehingga menimbulkan masalah seperti jaringan lambat atau terkadang terputus.

4.2.2 Pengujian Jaringan Akhir

Sebelum masuk kepengujian besaran *bandwidth*, kita uji dahulu teknologi failover yaitu dengan mematikan sala satu ISP dan kita lihat hasil PING dan Trace IP 8.8.8.8 dari salah satu client PC.

```

PC2> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  30.30.30.1   2.883 ms  1.477 ms  2.032 ms
 2  23.23.23.2   9.879 ms  5.293 ms  6.579 ms
 3  192.168.137.1 9.962 ms  5.756 ms  6.580 ms
 4  * * *
 5  192.168.1.1   40.489 ms 30.676 ms 25.921 ms
 6  10.56.0.1     56.496 ms 75.131 ms 50.287 ms
 7  172.16.88.101 55.814 ms 80.736 ms 137.858 ms
 8  172.16.88.134 79.587 ms 83.854 ms 190.520 ms

```

Gambar IV.12 PING dan Trace Failover 1

Dapat dilihat dari hasil PIN dan Trace di gambar atas bahwa jalur nya berubah dari 12.12.12.1 menjadi 23.23.23.2 (ISP2). Bagaimana jika kita nyalakan kembali ISP1, maka hasilnya sebagai berikut :

```

PC2> trace 8.8.8.8
trace to 8.8.8.8, 8 hops max, press Ctrl+C to stop
 1  30.30.30.1   2.008 ms  2.100 ms  3.315 ms
 2  12.12.12.1  12.111 ms 4.610 ms  5.989 ms
 3  192.168.137.1 11.341 ms 4.090 ms  5.104 ms
 4  * * *
 5  192.168.1.1   30.452 ms 18.499 ms 11.332 ms
 6  10.56.0.1     14.601 ms 10.028 ms 13.795 ms
 7  172.16.88.101 32.585 ms 24.679 ms 34.503 ms
 8  172.16.88.134 97.832 ms 25.457 ms 27.549 ms

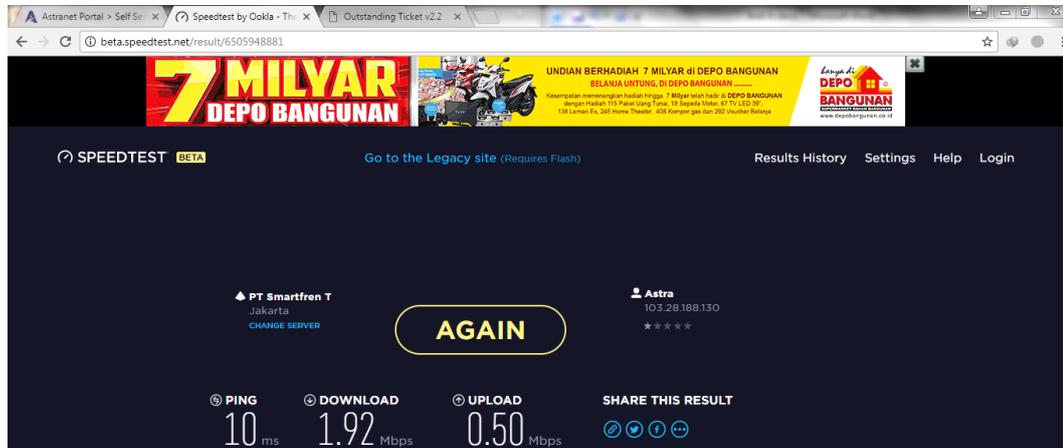
```

Gambar IV.13 PING dan Trace Failover 2

Jalurnya kembali berubah lagi ke 12.12.12.1 (ISP1) karena distancenya 1, dan pada perbandingan ECMP 1:1.

Bagaimana jika kita tes beban melalui internet, dengan pengujian ini akan di ketahui besaran *speed download* dan *speed upload* beserta besaran nilai ping. Pengujian akan dilakukan sebanyak lima kali uji coba pada server yang sama pada saat pengujian awal sebelumnya. Setelah melakukan pengujian *speedtest* penulis pun menguji jaringan akhir dengan melakukan test ping pada CMD dengan perintah test ping 8.8.8.8 -t untuk menguji performa *fail over* pada *load balancing* apakah masih terdapat jeda ketika salah satu ISP di matikan. Tak hanya melakukan pengujian penulis pun mengamati *parameter interface* pada mikrotik

untuk melihat pemerataan *traffic gateway* pada setiap ISP. Hasil pengujian jaringan bisa dilihat pada gambar berikut :



Gambar IV.14 Hasil Speedtest Jaringan Akhir

Data-data hasil pengujian *speedtest* tersebut akan dibuat dibuat table seperti di bawah ini :

Table IV.2
Hasil Pengujian Speed Download dan Upload Akhir

No	Ping	Speed Download	Speed Upload
1	10 ms	1.92 Mbps	0.50 Mbps
2	12 ms	2.07 Mbps	0.75 Mbps
3	13 ms	1.47 Mbps	0.78 Mbps
4	16 ms	2.0 33 Mbps	0.69 Mbps

```

C:\Windows\system32\cmd.exe - ping 8.8.8.8 -t
Reply from 8.8.8.8: bytes=32 time=31ms TTL=56
Reply from 8.8.8.8: bytes=32 time=493ms TTL=56
Reply from 8.8.8.8: bytes=32 time=24ms TTL=56
Reply from 8.8.8.8: bytes=32 time=84ms TTL=56
Request timed out.
Request timed out.
Reply from 8.8.8.8: bytes=32 time=22ms TTL=56
Reply from 8.8.8.8: bytes=32 time=1141ms TTL=56
Request timed out.
Reply from 8.8.8.8: bytes=32 time=19ms TTL=56
Reply from 8.8.8.8: bytes=32 time=18ms TTL=56
Request timed out.
Reply from 8.8.8.8: bytes=32 time=43ms TTL=56
Reply from 8.8.8.8: bytes=32 time=158ms TTL=56
Reply from 8.8.8.8: bytes=32 time=29ms TTL=56
Reply from 8.8.8.8: bytes=32 time=100ms TTL=56
Reply from 8.8.8.8: bytes=32 time=23ms TTL=56
Reply from 8.8.8.8: bytes=32 time=31ms TTL=56
Reply from 8.8.8.8: bytes=32 time=18ms TTL=56
Reply from 8.8.8.8: bytes=32 time=21ms TTL=56
Reply from 8.8.8.8: bytes=32 time=92ms TTL=56
Reply from 8.8.8.8: bytes=32 time=20ms TTL=56
Reply from 8.8.8.8: bytes=32 time=19ms TTL=56
  
```

Gambar IV.15 Hasil Test PING Pengujian Jaringan Akhir 1

Berdasarkan hasil hasil pengujian *speedtest* jaringan awal dan jaringan akhir dapat dilihat bahwa terdapat perbedaan kualitas jaringan internet.

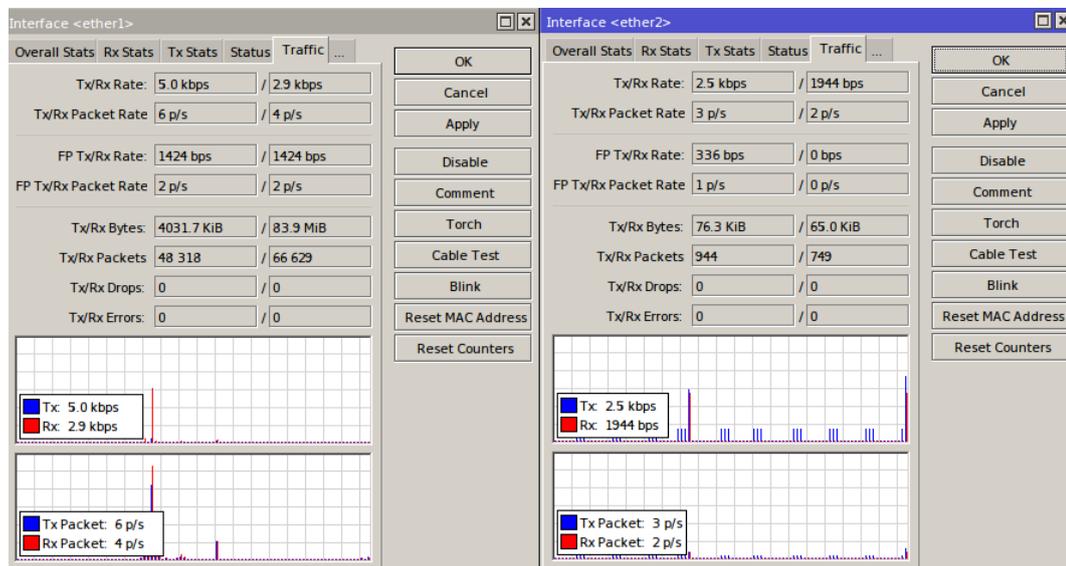
```

C:\Windows\system32\cmd.exe - ping 8.8.8.8 -t
Reply from 8.8.8.8: bytes=32 time=158ms TTL=56
Reply from 8.8.8.8: bytes=32 time=29ms TTL=56
Reply from 8.8.8.8: bytes=32 time=100ms TTL=56
Reply from 8.8.8.8: bytes=32 time=23ms TTL=56
Reply from 8.8.8.8: bytes=32 time=31ms TTL=56
Reply from 8.8.8.8: bytes=32 time=18ms TTL=56
Reply from 8.8.8.8: bytes=32 time=21ms TTL=56
Reply from 8.8.8.8: bytes=32 time=92ms TTL=56
Reply from 8.8.8.8: bytes=32 time=20ms TTL=56
Reply from 8.8.8.8: bytes=32 time=19ms TTL=56
Reply from 8.8.8.8: bytes=32 time=18ms TTL=56
Reply from 8.8.8.8: bytes=32 time=25ms TTL=56
Reply from 8.8.8.8: bytes=32 time=21ms TTL=56
Reply from 8.8.8.8: bytes=32 time=17ms TTL=56
Reply from 8.8.8.8: bytes=32 time=18ms TTL=56
Reply from 8.8.8.8: bytes=32 time=44ms TTL=56
Reply from 8.8.8.8: bytes=32 time=22ms TTL=56
Reply from 8.8.8.8: bytes=32 time=985ms TTL=56
Reply from 8.8.8.8: bytes=32 time=855ms TTL=56
Reply from 8.8.8.8: bytes=32 time=901ms TTL=56
Reply from 8.8.8.8: bytes=32 time=1165ms TTL=56
Reply from 8.8.8.8: bytes=32 time=1382ms TTL=56

```

Gambar IV.16 Hasil Tes PING Pengujian Jaringan Akhir 2

Test ping di atas sudah tidak terdapat *request timed out* atau jeda pada saat salah satu ISP dimatikan. Hasil test ping jaringan akhir ini membuktikan bahwa ketika salah satu ISP dapat membackup dengan baik ketika salah satu ISP dimatikan.



Gambar IV.17 Grafik Parameter Gateway Pada Setiap ISP

Berdasarkan pada grafik *parameter gateway* di atas besaran *traffic* pada setiap ISP kurang lebih hampir sama. Ini membuktikan bahwa *load balancing* telah berhasil

menyeimbangkan trafik pada kedua ISP sehingga tidak ada penumpukan beban traffic pada salah satu ISP.

Jadi kombinasi load balancing dan failover ini sangat bermanfaat apabila kita memiliki lebih dari satu ISP, kita bias membuat jalur cadangan dan mengoptimalkan jalur yang sudah ada.