

Implementasi Sistem *Load Balancing* Dua ISP Menggunakan Mikrotik dengan Metode Per *Connection Classifier*

Prihatin Oktivasari, Rinaldi Sanjaya
Konsentrasi Teknik Komputer dan Jaringan
Program Studi Teknik Informatika
Jurusan Teknik Informatika dan Komputer
Politeknik Negeri Jakarta
Depok, Jakarta
prihatin.oktivasari@tik.pnj.ac.id, rsanjayar@gmail.com

Diterima: 15 September 2015. Disetujui: 13 Oktober 2015. Dipublikasikan: November 2015

Abstrak - Sistem *Load Balancing* dapat membuat kinerja internet menjadi lebih optimal. Dengan pembagian jalur *traffic* yang lebih merata membuat sistem *Load Balancing* menjadi solusi yang bagus untuk kinerja internet yang lebih stabil. Dalam melakukan konfigurasi, *system Load Balancing* ini dilakukan pada mikrotik. Mikrotik yang digunakan adalah Mikrotik RB2011UiAS-RM. Terdapat beberapa metode yang biasa. Salah satunya adalah Per *Connection Classifier* (PCC). Metode ini pula yang akan digunakan dalam penulisan laporan ini. Selain PCC, metode lain yang ada pada *Load Balancing* adalah ECMP (*Equal Cost Multi Path*), Nth, dan juga *Static Route* dengan *address list*. *Load Balancing* dengan menggunakan metode PCC adalah teknik untuk mendistribusikan beban *traffic* pada 2 jalur koneksi menjadi lebih seimbang agar *traffic* dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. Kelebihan yang dimiliki metode PCC adalah mampu menspesifikasikan *gateway* untuk setiap paket data yang masih berhubungan dengan data yang sebelumnya sudah dilewatkan pada salah satu *gateway*.

Kata Kunci: *Load Balancing*, *bandwidth*, mikrotik RB2011UiAS-RM, *per connection classifier* (PCC), *internet service provider* (ISP).

I. PENDAHULUAN

Jaringan komputer bukan merupakan sesuatu yang baru saat ini. Hampir di setiap instansi pendidikan terdapat jaringan komputer untuk memperlancar arus informasi di dalam instansi tersebut bahkan di perumahan dan perusahaan juga sudah menggunakannya. Internet yang sudah populer saat ini adalah suatu jaringan komputer raksasa yang merupakan jaringan komputer yang terhubung dan dapat saling berinteraksi.

Load Balancing diperlukan karena dapat membuat kinerja internet menjadi lebih cepat dengan memaksimalkan pembagian beban *traffic* yang dilakukan pada mikrotik [1].

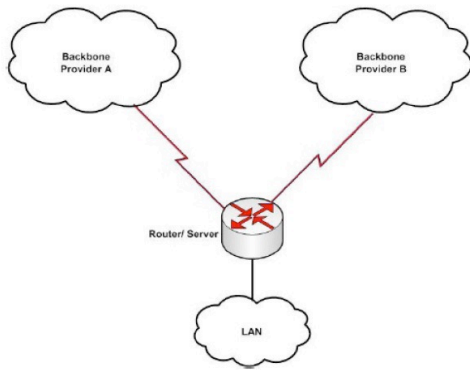
Hal ini dapat terjadi karena adanya perkembangan teknologi jaringan yang sangat pesat, sehingga dalam beberapa tahun saja jumlah pengguna jaringan komputer yang tergabung dalam internet berlipat ganda. Perkembangan pemakaian internet yang meningkat pesat saat ini menyebabkan permintaan akan mutu layanan (*Quality of services/QoS*) yang harus ditingkatkan. Terutama bagi penyedia layanan internet (*Internet Service Provider/ISP*) bahkan di jaringan pribadi dan instansi yang menggunakan internet sebagai kebutuhan utama.

II. TINJAUAN PUSTAKA

A. Pengertian *Load Balancing*

Load Balancing adalah proses pendistribusian beban terhadap sebuah servis yang ada pada sekumpulan *server* atau perangkat jaringan ketika ada permintaan dari pengguna [2]. Ketika banyak permintaan dari pengguna maka *server* tersebut akan terbebani karena harus melakukan proses pelayanan terhadap permintaan pengguna.

Load Balancing dengan 2 ISP ditunjukkan pada Gambar 1.



Gambar 1. Load Balancing dengan 2 ISP

Solusi yang cukup bermanfaat adalah dengan membagi beban yang datang ke beberapa *server*, Jadi tidak berpusat ke salah satu *server* jaringan saja. Teknologi itulah yang disebut Teknologi *Load Balancing*. Dengan teknologi *Load Balancing* maka dapat diperoleh keuntungan seperti menjamin *reabilitas* servis, *availabilitas* dan skalabilitas suatu jaringan [3]. Keuntungan menggunakan *Load Balancing* adalah jika salah satu *line speedy* bermasalah maka koneksi internet tidak akan terputus karena masih ada *line speedy* yang satu. Tapi jika menggunakan model *konvensional* bila salah satu *line speedy* mengalami gangguan maka semua komputer yang terhubung dengannya akan mengalami gangguan koneksi.

B. Metode Load Balancing

Metode *Load Balancing* sendiri ada 4 cara yang umum dipergunakan. Diantaranya *Static Route* dengan *Address List*, *Equal Cost Multi Path (ECMP)*, *Nth*, dan *Per Connection Classifier (PCC)*. Masing-masing metode mempunyai kelebihan, kekurangan dan karakteristiknya masing-masing [4].

Load Balancing dengan menggunakan konsep *PCC* ini adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi supaya seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. Kelebihan yang dimiliki metode *PCC* adalah mampu menspesifikasikan *gateway* untuk tiap paket data yang masih berhubungan dengan data yang sebelumnya sudah dilewatkan pada salah satu *gateway* [5]. Sementara kekurangan dari metode ini lebih memungkinkan beresiko terjadi *overload* pada salah satu *gateway* yang disebabkan oleh pengaksesan situs yang sama.

C. Synching

Winbox adalah sebuah utility yang digunakan untuk melakukan *remote* ke *server* Mikrotik kita dalam mode GUI. Jika untuk mengkonfigurasi Mikrotik dalam text mode melalui PC itu sendiri,

maka untuk mode GUI yang menggunakan Winbox ini kita mengkonfigurasi Mikrotik melalui komputer client. Gambar 2 menunjukkan logo Winbox.



Gambar 2. Logo Winbox

D. OwnCloud

Mikrotik adalah sebuah sistem operasi termasuk di dalamnya perangkat lunak yang dipasang pada suatu komputer sehingga komputer tersebut dapat berperan sebagai jantung *network*, pengendali atau pengatur lalu-lintas data antar jaringan, komputer jenis ini dikenal dengan nama *router*. Jadi intinya Mikrotik adalah salah satu sistem operasi khusus untuk *router*. Mikrotik dikenal sebagai salah satu *Router OS* yang handal dan memiliki banyak sekali fitur untuk mendukung kelancaran *network*. Gambar 3 menunjukkan tampilan Mikrotik RB2011UiAS-RM.



Gambar 3. Mikrotik RB2011UiAS-RM

E. Bandwidth

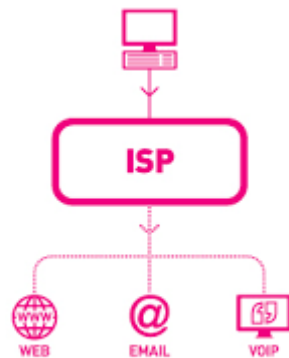
Bandwidth adalah nilai hitung atau perhitungan konsumsi *transfer* data telekomunikasi yang dihitung dalam satuan bit per detik atau yang biasa disingkat *bps* yang terjadi antara komputer *server* dan komputer *client* dalam waktu tertentu dalam sebuah jaringan komputer.

Bandwidth sendiri akan dialokasikan ke komputer dalam jaringan dan akan mempengaruhi kecepatan transfer data pada jaringan komputer tersebut sehingga semakin besar *Bandwidth* pada jaringan komputer maka semakin cepat pula kecepatan transfer data yang dapat dilakukan oleh *client* maupun *server*. Pada sebuah jaringan komputer *bandwidth* terbagi menjadi 2 yaitu *bandwidth* digital dan *bandwidth* analog.

F. Internet Service Provider (ISP)

ISP (Internet Service Provider). Untuk bisa bergabung dengan internet, kita harus mempunyai akses dengan cara berlangganan ke penyedia jasa internet (*Internet Service Provider*). *ISP* adalah perusahaan yang menawarkan jasa pelayanan kepada kita untuk berhubungan dengan internet. Untuk

mengakses internet, kita cukup menghubungi ISP melalui modem dan komputer lalu ISP akan mengurus detail-detail yang diperlukan untuk berhubungan dengan internet, termasuk biaya koneksi tersebut. Jadi, misalnya kita sedang mengakses *homepage* luar negeri, maka ISP-lah yang menganggung biaya hubungan ke luar negeri. Kita cukup membayar pulsa lokal yang digunakan untuk menghubungi ISP tadi. Alur kerja ISP ditunjukkan pada Gambar 4. Saat ini telkom menyediakan fasilitas *TelkomNet@Instan* yang bersifat semi, yaitu akses internet melalui telkom tanpa harus berlangganan ke ISP tertentu, biaya pemakaian dihitung berdasarkan menit pemakaian.



Gambar 4 Alur Kerja ISP

III. PERANCANGAN PROGRAM APLIKASI

A. Deskripsi Program Aplikasi

Load Balance 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 Balance tidak akan menambah besar *bandwidth* yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang.

Dengan penulisan ini, penulis akan membuktikan bahwa dalam penggunaan *Load Balancing* tidak seperti rumus matematika $512 + 256 = 768$, akan tetapi $512 + 256 = 512 + 256$, atau $512 + 256 = 256 + 256 + 256$.

Pada penulisan ini *Router* Mikrotik yang digunakan adalah Mikrotik RB2011UiAS-RM dengan kondisi sebagai berikut :

1. *Ether1* dan *Ether2* terhubung pada *ISP* yang berbeda dengan besar *bandwidth* yang berbeda. *ISP1* sebesar 512kbps dan *ISP2* sebesar 256kbps.
2. Mikrotik *Router OS* menggunakan versi 4.5 karena fitur *PCC* mulai dikenal pada versi 3.24.

B. Cara Kerja Program Aplikasi

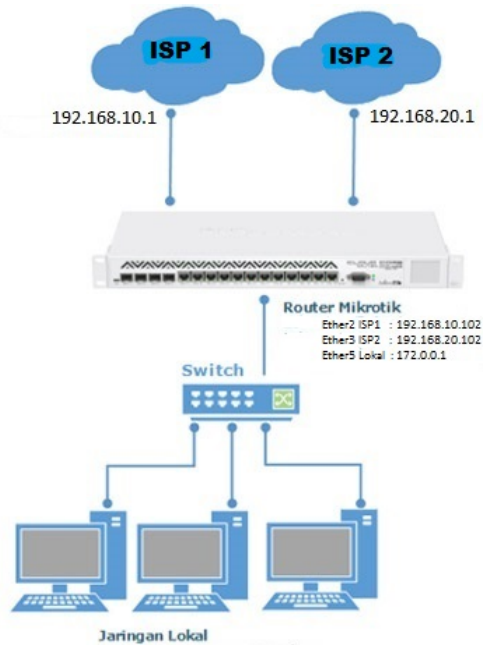
Yang perlu dipahami dalam melakukan konfigurasi untuk *PCC* kuncinya adalah "per-connection-classifier=src-address-and-port:" selain itu juga dari sisi pembagian *bandwidth* dimana harus kita bagi secara merata, semisal dari *ISP 1* kita mendapatkan *bandwidth* 768 kbps sedangkan pada *ISP 2* kita mendapatkan *bandwidth* 512kbps, maka agar merata pembagiannya untuk *ISP 1* $256\text{kbps} + 256\text{kbps} + 256\text{kbps}$ dan *ISP 2* $256\text{kbps} + 256\text{kbps}$ sehingga dari pembagian tersebut terdapat 5 koneksi dimana masing-masing *bandwidth*-nya 256kbps, sehingga untuk pembagiannya kurang lebih seperti berikut : 5/0, 5/1, 5/2, 5/3, 5/4. Langkah selanjutnya adalah kita membuat penandaan untuk semua trafik menggunakan *mangle*, tapi sebelum kita membuat penandaan, kita perlu membuat 2 buah *rule* NAT, jika kita sebelumnya sudah memiliki koneksi ke satu provider, kita hanya perlu menambahkan *rule* NAT dengan *action masquerade* dimana untuk *out-interface*-nya di arahkan ke *interface* ethernet yang mengarah ke *provider* yang kedua, sebagai contoh seperti berikut :

```
/ip firewall nat add action=masquerade chain=srcnat out-interface=ether1 (mengarah ke ISP1)
/ip firewall nat add action=masquerade chain=srcnat out-interface=ether2 (mengarah ke ISP2)
```

C. Rancangan Program Aplikasi

Load Balancing sendiri 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 *agonize* pada salah satu jalur koneksi. Selama ini kita beranggapan bahwa jika kita memiliki 2 koneksi dan menggabungkan antara kedua dengan *bulk acclimation* maka *bandwidth* kita menjadi dua kali lebih besar. Hal ini perlu kita perjelas dahulu, bahwa *Load Balance* tidak akan menambah besar *bandwidth* yang kita peroleh, tetapi hanya bertugas untuk membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang dan maksimal, dan untuk metode *PCC* adalah salah satu alternatif jika kita ingin menerapkan *Load Balance* dimana Mikrotik akan mengingat kembali koneksi sebelumnya untuk digunakan.

Topologi sistem pada *Load Balancing* ditunjukkan oleh Gambar 5.



Gambar 5. Topologi Sistem Load Balancing

D. Realisasi Program Aplikasi

Pada bagian ini akan membahas langkah – langkah yang akan ditempuh dalam upaya membangun sistem Load Balancing. Software yang digunakan untuk melakukan konfigurasi disini adalah Winbox. Setelah menginstall Winbox, maka tahap selanjutnya adalah melakukan konfigurasi Load Balancing menggunakan metode Per Connection Classifier.

IV. PENGUJIAN

A. Deskripsi Pengujian

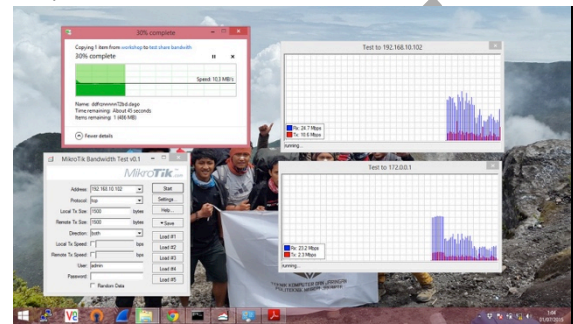
Pengujian dilakukan dengan menggunakan 4 komputer serta software Mikrotik Bandwidth Test untuk mengetahui grafik bandwidth saat dilakukan pengunduhan. Komputer pertama dan kedua dijadikan sebagai ISP1 dan ISP2. Komputer 3 sebagai server yang digunakan untuk mengkonfigurasi Mikrotik, dan komputer 4 sebagai client yang akan melakukan pengujian sistem Load Balancing dengan metode Per Connection Classifier.

B. Prosedur Pengujian

Dalam melakukan proses pengujian menggunakan 4 komputer dan juga sebuah Mikrotik. Pertama setiap komputer dihubungkan dengan menggunakan Mikrotik. Lalu komputer client mengunduh sebuah file dari komputer 1, dan secara bersamaan client juga mengunduh file dari komputer 2. Pada saat melakukan pengunduhan akan terlihat bandwidth yang terbagi jika dilakukan pengunduhan secara bersamaan.

1. Pengujian Sistem Load Balancing Dengan Metode Per Connection Classifier (PCC).

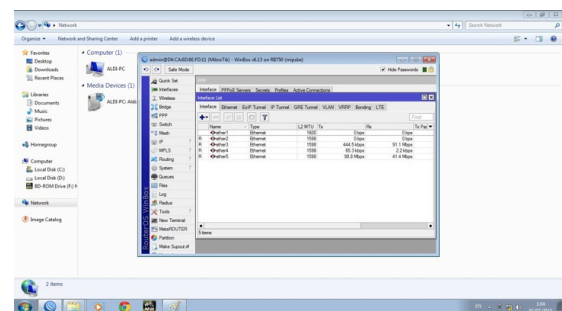
Uji coba dilakukan dengan menjalankan software Mikrotik Bandwidth Test untuk mengetahui grafik bandwidth yang terdapat pada ISP1 dan ISP2. Setelah itu melakukan pengunduhan yang dilakukan komputer client kepada komputer 1. Selanjutnya pengunduhan kepada komputer 2. Jika sudah dilakukan pengunduhan secara bergantian, maka lakukan pengunduhan secara bersamaan. Lalu disitu akan terlihat bandwidth yang terbagi jika dilakukan pengunduhan secara bersamaan. Tampilan grafik bandwidth seperti pada Gambar 6.



Gambar 6. Tampilan Grafik Bandwidth yang Diterima Client Ketika Melakukan Pengunduhan dari Komputer 1 dengan IP 192.168.10.102

2. Pengujian pada sisi Mikrotik

Uji coba yang dilakukan kurang lebih sama seperti uji coba sebelumnya. Namun pada uji coba kali ini dapat terlihat ISP mana yang sedang bekerja ketika client melakukan pengunduhan. Tampilan Interfaces List Ether3 sebagai ISP2 bekerja ketika dilakukan pengunduhan dari komputer 2 seperti ditunjukkan oleh Gambar 7.



Gambar 7. Tampilan Interfaces List Ether3

3. Pengujian Ketika Salah Satu Koneksi Down

Disini dapat dilihat ketika salah satu koneksi mati maka akan secara otomatis akan dialihkan kepada koneksi yang masih aktif. Ether3 sebagai ISP2 mengalami down maka seluruh beban akan dialihkan kepada Ether2 sebagai ISP1.

4. Data Hasil Pengujian

Berikut adalah hasil setelah melakukan pengujian sistem Load Balancing menggunakan metode Per Connection Classifier. Ketika dilakukan

pengunduhan secara bersamaan maka *bandwidth* akan terbagi rata. Tampilannya seperti pada Gambar 8.

Name	Type	L2 MTU	Tx	Rx
ether1	Ethernet	1600	0 bps	0 bps
ether2	Ethernet	1598	392.2 kbps	39.4 Mbps
ether3	Ethernet	1598	403.6 kbps	52.9 Mbps
ether4	Ethernet	1598	75.9 kbps	3.3 kbps
ether5	Ethernet	1598	100.3 Mbps	40.5 Mbps

Gambar 8. Hasil Pengujian *Bandwidth* Terbagi Rata

C. Analisis Data / Evaluasi

Dari hasil pengujian, sangat terlihat jelas pembagian *bandwidth* yang merata jika pengunduhan dilakukan lebih dari 1. Apabila hanya melakukan satu pengunduhan, maka *bandwidth* yang diterima akan lebih maksimal. Namun perlu diperhatikan disini, sistem *Load Balancing* dua ISP ini tidak menambahkan besar *bandwidth* menjadi 2 kali lipat dari sebelum menggunakan *Load Balancing*. Akan tetapi *Load Balancing* hanya bertugas membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang.

V. KESIMPULAN DAN SARAN

Berdasarkan uraian diatas dapat disimpulkan :

1. Penerapan sistem *Load Balancing* dengan metode PCC ini dapat membagi beban *traffic* secara seimbang sehingga dapat meningkatkan kecepatan internet.
2. Jika salah satu link jaringan internet terputus, maka seluruh beban akan dialihkan secara otomatis ke jaringan internet yang masih aktif.
3. Hubungan *client server* terjalin utuh karena selalu pada jalur yang sama, ini dikarenakan pada *rule* PCC akan selalu mengingat IP *address* sumber tujuan.

REFERENSI

- [1] Chun-Cheng Lin, Hui-Hsin Chin, Der-Jiunn Deng, "Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System", IEEE Systems Journal , Vol. 8, Issue: 1, 2014, Pages: 225 – 234.
- [2] Lukitasari, Desy & Oklilas, Ahmad Fali. 2010. *Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server*. Universitas Sriwijaya : Jurnal Generic Fakultas Ilmu Komputer.
- [3] Sirajuddin, Affandi, Ahmad, & Setijadi, Eko. 2012. *Rancang Bangun Server Learning Management System Menggunakan Load Balancer dan Reverse Proxy*. Surabaya : Jurnal Teknik POMITS Vol. 1, No. 1.
- [4] Shancang Li; Shanshan Zhao; Xinheng Wang; Kewang Zhang; Ling Li, "Adaptive and Secure Load-Balancing Routing Protocol for Service-Oriented Wireless Sensor Networks". IEEE Systems Journal, Vol. 8, Issue: 3, 2014, Pages: 858 – 867.
- [5] Hengky Susanto; Byung-Guk Kim, "Per-Connection Return Routability Test in Mobile IPv6.", International Conference on Network-Based Information Systems, 2009, Pages: 142 – 146.