

---

## BAB II DASAR TEORI

Pada Bab ini yaitu membahas tentang *load balance*, *PCC* (*Per Connection Classifier*) dan *Mikrotik* dimana pada pembahasan ini akan menjelaskan sebuah metode yang digunakan untuk melakukan proses mendistribusikan beban trafik pada dua jalur dari koneksi internet secara seimbang, agar trafik dapat berjalan secara optimal.

### A. LOAD BALANCE

*Load balance* adalah Teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan dengan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi.

Selama ini banyak yang beranggapan salah, bahwa dengan menggunakan *load balance*, maka besar *bandwidth* yang diperoleh menjadi dua kali lipat dari *bandwidth* sebelum menggunakan *load balance*. Hal ini perlu diperjelas dahulu, bahwa *load balance* tidak akan menambah besar *bandwidth* yang diperoleh, tetapi hanya bertugas untuk membagi trafik dari kedua *bandwidth* tersebut agar dapat terpakai secara seimbang. Dalam menggunakan *load balance* tidaklah seperti rumus matematika  $1+1 = 2$ , akan tetapi  $1+1 = 1+1$  [1].

Dalam system *load balance*, proses pembagian bebannya memiliki Teknik dan algoritma tersendiri. Pada perangkat *load balance* yang kompleks biasanya disediakan bermacam-macam algoritma untuk pembagian beban ini. Tujuannya untuk

---

menyesuaikan pembagian beban dengan karakteristik dari *server-server* yang ada dibelakangnya [2].

Dalam *load balance* dimikrotik, hal-hal yang perlu diperhatikan dalam mengatur *load balance* adalah *static route* dan *policy router*, *firewall mangle*, dan *firewall src-nat*. *Static route* dan *policy router* mengatur *uplink flow*, yaitu kebijakan *routing* yang akan dilalui paket yang telah ditandai, sedangkan *Firewall mangle* adalah penandaan paket sebelum masuk *routing*.

Ada beberapa metode *load balance*, yaitu antara lain PCC dan Nth. Setiap metode 9 tersebut memiliki kekurangan dan kelebihan tersendiri, namun lebih dari hal itu, yang paling terpenting dalam menentukan metode *load balance* apa yang akan digunakan adalah harus dimengerti terlebih dahulu karakteristik dari jaringan yang akan diimplementasikan. Berikut ini adalah sedikit pengertian dari metode *load balance* dan disertakan pula kekurangan maupun kelebihan[3].

#### **B. ( *Per Connection Classifier* )**

PCC merupakan metode yang dikembangkan oleh Mikrotik dan mulai diperkenalkan pada *Mikrotik RouterOS* versi 3.24. PCC mengambil bidang yang dipilih dari *header IP*, dan dengan bantuan dari algoritma *hashing* mengubah bidang yang dipilih menjadi 32-bit. Nilai ini kemudian dibagi dengan *denominator* tertentu dan sisanya kemudian dibandingkan dengan *remainder* tertentu, jika sama maka paket akan ditangkap. *Rules* dapat dibuat dengan memilih informasi dari *src-address*, *dst-address*, *src-port*, atau *dst-port* dari bagian *header*

---

IP. *Header* IP memiliki *field* yang berisi beberapa bidang, dua diantaranya adalah alamat IP sumber (*src-address*) paket dan alamat IP tujuan (*dst-address*) dari paket tersebut. Paket TCP dan UDP juga memiliki *header* yang berisi port sumber dan port tujuan [1].

PCC Merupakan metode yang mengelompokkan trafik koneksi melalui atau keluar masuk *router* menjadi beberapa kelompok. Pengelompokan ini bisa dibedakan berdasarkan *src-address*, *dst-address*, *src-port*, dan *dst-port*. *Router* akan menyimpan informasi tentang jalur gateway yang dilewati data di tiap trafik koneksi, sehingga pada paket-paket selanjutnya yang masih berkaitan dengan paket data sebelumnya akan dilewatkan pada jalur *gateway* yang sama juga. Karena metode PCC melewatkan paket data melalui jalur *gateway* yang sama, 10 maka metode tersebut mempunyai kekurangan yaitu dapat terjadi *overload* pada salah satu *gateway*.

PCC bekerja dengan cara mengambil beberapa *field* dari IP *header* dan TCP atau UDP header, kemudian dengan bantuan algoritma hashing akan menghasilkan sebuah *output*. *Output* tersebut didapat dengan cara melakukan penjumlahan dari beberapa *field* IP *header*, kemudian dibagi oleh penyebut yang telah ditentukan, dan sisanya jika dibandingkan dengan remainder tertentu, jika sama, maka paket akan dicapture. Kita dapat memilih *source-address*, *destination-address*, *source-port*, dan *destination* port dalam operasi ini. *Source-address* dan *destination-address* dapat diambil dari IP paket *header* dan *source-port* dan *destination-port* diambil dari TCP atau UDP

---

paket *header*. Salah satu metode hash yang dapat digunakan adalah Modulo [2].

Modulo merupakan sebuah operasi bilangan yang menghasilkan sisa pembagian dari suatu bilangan terhadap bilangan lainnya. Misalkan dua bilangan  $a$  dan  $b$ ,  $a$  modulo  $b$  (disingkat  $a \bmod b$ ) adalah bilangan bulat sisa pembagian  $a$  oleh  $b$ . misalnya, “1 mod 3”, “4 mod 3”, dan “7 mod 3” memiliki hasil 1, karena ketiga bilangan tersebut memiliki sisa 1 jika dibagi oleh 3, sedangkan “9 mod 3” sama dengan 0. Penerapan operasi modulus dalam teori bilangan tergolong aritmatika modulo. Fungsi *hashing* dipakai karena mempunyai salah satu sifat yang *deterministik*. Maksudnya adalah jika kita memasukkan *input* yang bertuliskan “hello” dan menghasilkan *output* “1”, dan pernyataan itu bersifat mutlak, sehingga jika kita menginputkan “hello” kedua kalinya akan menghasilkan *output* “1”. Misalkan *source-address* dari *router client* (*router load balance*) adalah 11 192.168.1.1 kemudian *destination-address* dari *router A* (256 Kbps) 192.168.3.2 dan *destination-address* dari *router B* (512 Kbps) 192.168.2.2, jika pada saat ada file video yang lewat pada *router client*, maka *router client* akan mengecek ping *gateway* pada *router A* dan *router B* (dicari yang terendah). Setelah diketahui bahwa nilai ping *gateway* yang terendah ada pada *router B* (karena *bandwidth*nya lebih besar dari *router A*), maka file video tersebut di-dekapsulasi kemudian pada semua *header packet*-nya di *mangle* pada *router B*. Jika dengan algoritma *hashing*, nilai *source-address* + nilai *destination-address* kemudian di modulus jumlah jalur yang ada. Jadi

---

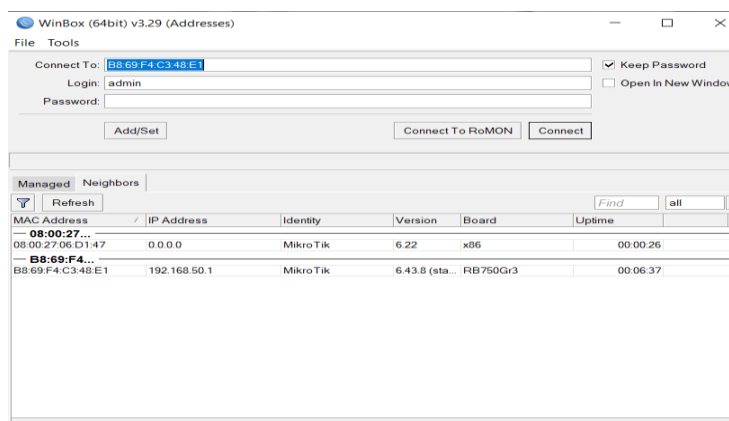
$192+168+1+1+192+168+2+2 \text{ mod } 2 = 1$ . Nilai 1 untuk *router B*, nilai 0 untuk *router A* [3].

### C. MIKROTIK

“MikroTik Router OS adalah system 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 untuk digunakan oleh ISP dan provider hotspot”. MikroTik Router OS, merupakan sistem operasi Linux base yang diperuntukkan sebagai sistem network router. Administrasinya bisa dilakukan melalui *WinBox (Windows Application)* [1].



**Gambar 2.1 Mikrotik**



**Gambar 2.2 WinBox (Windows Application).**

