

Analisis Litensi Metode PCC, NTH dan ECMP untuk *Load Balance* dan *Failover*

Nerissa Hansa Wijaya ^{#1}, Billy Susanto Panca ^{*2}

[#] Program Studi SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Prof. Drg. Surya Sumantri 65 Bandung

¹nerissa_hansa@yahoo.co.id

²billy.sp@it.maranatha.edu

Abstract — *Failover and Load Balance are needed to handle disaster on the server. Failover has a function as a backup system when the main system is interrupted. Whereas load balance functions as a server balancer when a request from a client causes overload on one of the servers when the traffic is not balanced. In this study analyze the comparative advantages and disadvantages of the three load balancing and failover methods, namely Equal Cost Multi Path (ECMP), Nth, and Per Connection Classifier (PCC). These three methods will be implemented in a virtual network topology using two ISPs using Winbox software and MikroTik RB-941-2nD-TC. Testing is done using Webstress Tools, Internet Download Manager (IDM), Ping Tools. The results showed the PCC method because classifying connection traffic in and out of the router into several groups. This grouping can be distinguished by src-address, dst-address, src-port and dst-port. Mikrotik will remember the gateway path that has been passed at the beginning of the connection traffic. So that the next data packet that is still related will pass on the same gateway path as the previous data packet that was sent.*

Keywords - PCC, ECMP, NTH, failover, load balancing

I. PENDAHULUAN

Perkembangan teknologi di era global ini menuntut layanan yang tersedia mampu memiliki kemampuan untuk beradaptasi dengan suatu masalah, salah satunya karena adanya kapasitas yang melebihi batas sehingga memungkinkan terputusnya jaringan yang membuat *client* tidak dapat mengakses jaringan.

Agar tidak terjadi terputusnya jaringan dapat menggunakan teknik *load balancing* dan *failover*. Jika menggunakan teknik *load balancing* tanpa menggunakan *failover* dapat mengatasi kegagalan yang terjadi pada sisi server tetapi terdapat kegagalan yang mengakibatkan layanan dapat terhenti. Selain itu dapat terjadi kerusakan pada server, yang berdampak hilangnya data yang bersifat kritikal. Maka dari itu untuk meminimalisir kerusakan tersebut harus menggunakan sistem jaringan *load balancing* dan *failover* secara bersamaan. Cara ini memungkinkan sistem dapat terus berjalan walaupun sistem utama mengalami kegagalan, selama sistem cadangan tetap tersedia.

Mikrotik adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer sebuah router *network* yang handal, mencakup berbagai fitur yang dibuat untuk IP *network* dan jaringan *wireless*, baik untuk digunakan oleh ISP dan *provider hotspot*.

Penelitian ini dipilih karena Mikrotik merupakan sistem operasi yang memiliki tiga metode yaitu PCC, ECMP dan NTH yang dapat menjalankan teknik *failover* dan *load balancing*. Maka dari itu penulis tertarik untuk mengimplementasikan dan mengevaluasi *load balancing* pada server agar dapat meningkatkan kinerja sistem, serta mampu mengantisipasi kegagalan sistem melalui teknik *failover*.

Pembahasan ini bertujuan melakukan implementasi terhadap teknik failover dan load balancing dengan menggunakan metode PCC, ECMP, dan NTH; Dapat menggunakan teknik *failover* dan *load balancing* dengan metode PCC, ECMP dan NTH jika salah satu jalur koneksi terjadi masalah seperti terputus atau mati; Dapat melakukan pengujian teknik *failover* dan *load balancing* dengan metode PCC,

ECMP, dan NTH; dan memperoleh hasil berupa waktu efisien yang didapat dari hasil perbandingan metode PCC, ECMP dan NTH.

II. KAJIAN TEORI

Di dalam penelitian ini, terdapat beberapa teori yang dijadikan landasan dalam penelitian ini. Berikut ini adalah teori-teori yang digunakan:

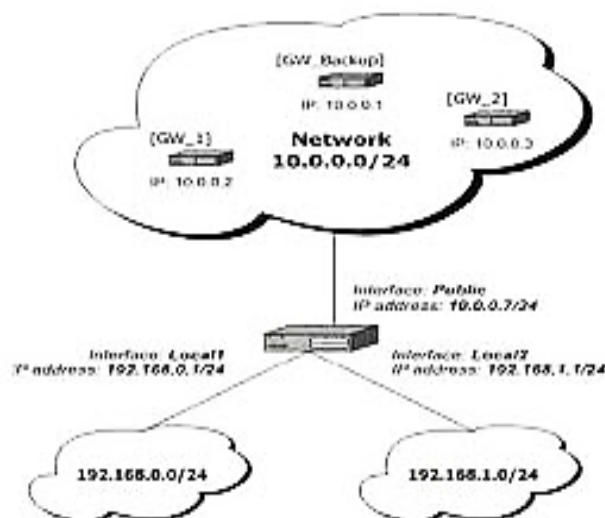
A. Load Balancing

Teknik *load balancing* dapat membagi beban kerja ke beberapa server secara optimal yang menghasilkan waktu tanggap yang baik dan meningkatkan efisiensi penggunaan sumber daya yang ada serta berdampak pada kepuasan pengguna dan meningkatkan performa secara keseluruhan. *Load balancing* dapat di bedakan menjadi dua bagian, yakni:[6]

- Per-Destination Load Balancing
Per-Destination load balancing berarti router mendistribusikan paket berdasarkan paket berdasarkan alamat tujuan. Apabila ada dua buah jalur untuk mencapai ke suatu jaringan maka paket untuk tujuan pertama kan melewati jalur pertama dan paket untuk tujuan kedua akan melewati jalur kedua. Ini memungkinkan pengguna jalur yang tidak seimbang atau tergantung dengan tujuan.
- Per-Packet Load Balancing
Per-Packet Load Balancing berarti router mengirimkan paket pertama melalui jalur pertama dan paket berikutnya melalui jalur kedua untuk tujuan yang sama. Per-paket load balancing menjamin pembagian yang seimbang pada semua link.

B. Failover

Failover, yang merupakan merupakan kemampuan sebuah sistem untuk berpindah secara manual atau otomatis jika salah satu sistem mengalami kegagalan sehingga sistem lain menjadi cadangan bagi sistem yang mengalami kegagalan [4]. Contoh *failover* ditunjukkan pada Gambar 1.



Gambar 1. Failover

Untuk mempermudah dan memperjelas maksud dari *failover* dapat dilihat contoh gambar 2.1. Pada gambar tersebut dapat dilihat sebuah *local area network* menggunakan lebih dari satu jalur jaringan *isp*. Jaringan lokal dengan ip 192.168.0.1/24 menggunakan *gateway* 1, sedangkan ip 192.168.0.0/24 menggunakan *gateway* 2. Jika *gateway* 1 mengalami *disconnect* (putus) maka *gateway backup* akan menggantikan *gateway* 1. Begitu pula sebaliknya. [4]

Dengan begitu dapat disimpulkan bahwa tujuan dari *failover* pada studi *multihomed* kali ini adalah untuk menggantikan atau sistem *backup* koneksi *isp* yang terputus dengan koneksi *isp* yang lainnya.[4]

C. PCC

Teknik PCC (*Per Connection Classifier*) merupakan *proprietary* dari Mikrotik yang belum lama ini diluncurkan. Metode ini menggunakan tipe *both address*, yaitu mengambil IP tujuan dan IP penerima dan akan melakukan proses *hashing* dan selanjutnya dilakukan proses komputasi untuk menentukan apakah suatu paket dapat dilewatkan menggunakan jalur *load balancing* pertama atau jalur *load balancing* kedua [5].

Secara khusus PCC akan mengambil *field* tertentu dari IP *header* dan dengan algoritma *hashing* akan mengubah *field* yang diambil menjadi bernilai 32-bit. Nilai ini kemudian akan dibagi dengan suatu *denominator* yang bernilai spesifik dan hasilnya akan dibandingkan dengan suatu nilai *remainder*, jika cocok maka paket tersebut akan diteruskan. *Field* yang digunakan pada metode ini antara lain *src-address*, *dst-address*, *src-port*, *dst-port* di mana *field* tersebut dapat dikombinasikan ataupun digunakan secara tunggal seperti, *both-addresses\both-ports\dst-address-and-port\src-address\src-port\both-address-and-ports\dst-address\dst-port\src-address-and-port* [3].

Fungsi *hashing* dipakai karena mempunyai salah satu sifat yang *deterministic*. 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 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* (dari yang terendah). Setelah diketahui bahwa nilai *ping gateway* yang terendah ada pada *router B* (karena *bandwith*-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*. [11]

D. NTH

Teknik NTH dikenal merupakan teknik yang menggunakan metode *round-robin* dalam pembagian bebannya. Paket yang ada akan dikelompokkan menjadi beberapa *group* secara berurutan paket pertama masuk dalam *group* pertama, paket kedua masuk dalam *group* kedua, dst. Setelah itu setiap *group* akan dikeluarkan melalui *exit interface* yang tersedia secara berurutan, *group* pertama akan keluar melalui *interface* pertama, *group* kedua akan keluar melalui *interface* kedua, dst [8].

NTH adalah sebuah *integer* (bilangan ke-N). Pada dasarnya koneksi yang masuk ke proses di *router* akan menjadi satu arus yang sama, walaupun mereka datang dari *interface* yang berbeda. Maka pada saat menerapkan metode NTH, tentunya akan memberikan batasan ke *router* untuk memproses koneksi dari sumber tertentu saja. Ketika *router* telah membuat semacam antrian baru untuk batasan yang kita berikan diatas, baru proses NTH dimulai. [15]

Pada NTH terdapat beberapa *variable* yang harus dimengerti, yaitu:

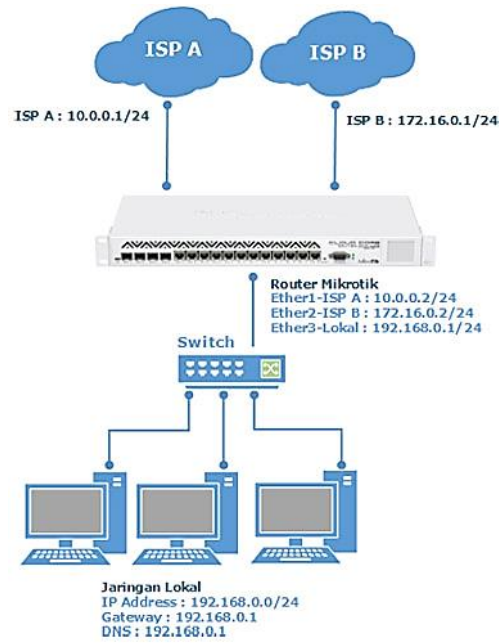
1. *Every*: Angka *every* ialah jumlah kelompok yang ingin dihasilkan. Jadi, apabila *administrator* ingin membagi alur koneksi yang ada menjadi 2 kelompok yang nantinya akan di-*load balance* ke 2 koneksi yang ada, maka angka *every* = 2.
2. *Packet*: Angka *packet* adalah jumlah koneksi yang akan ditandai atau di-*mangle*. Jika ingin membuat 2 buah kelompok, tentunya harus membuat 2 *mangle rules*. Pada *rules* tersebut, angka untuk *every* haruslah sama, namun untuk angka *packet* harus berubah. Untuk 2 kelompok, berarti angka *packet* untuk 2 *rules* tersebut adalah 1 dan 2. Salah satu kekurangan metode NTH ini adalah kemungkinan dapat terjadi terputusnya koneksi yang disebabkan perpindahan *gateway* karena *load balance*.

Misalkan *source-address* dari *router client (router load balance)* adalah 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 *file video* yang lewat ada *router client*, maka *router client* akan men-*dekapsulasi file video*. Kemudian *header packet* yang kedua di-*mangle* pada *route A (192.168.3.2)* dan *header packet* yang kedua di-*mangle* pada *router B (192.168.2.2)*. Proses tersebut berlangsung hingga semua *packet* yang melalui *router client* habis.

E. ECMP

Algoritma (*ECMP*) *Equal-Cost Multi-Path* dikenal sebagai teknik *load balancing* yang paling sederhana dan lebih cocok digunakan pada jaringan dengan tingkat kompleksitas yang tidak terlalu tinggi. Pembagian beban dalam metode ini menggunakan pengaturan perbandingan langsung antara beban yang diterima oleh setiap *gateway*. Tujuan penerapan algoritma ini yaitu untuk dimungkinkannya penggunaan lebih dari 1 jalur *gateway* digunakan sebagai *exit interface* dan mendapatkan pembagian beban yang merata antar tiap *gateway* dalam membagi *traffic* yang ada. Dalam penggunaannya algoritma ini juga mendukung penggunaan *routing protocol* baik *static* maupun *dynamic* [7].

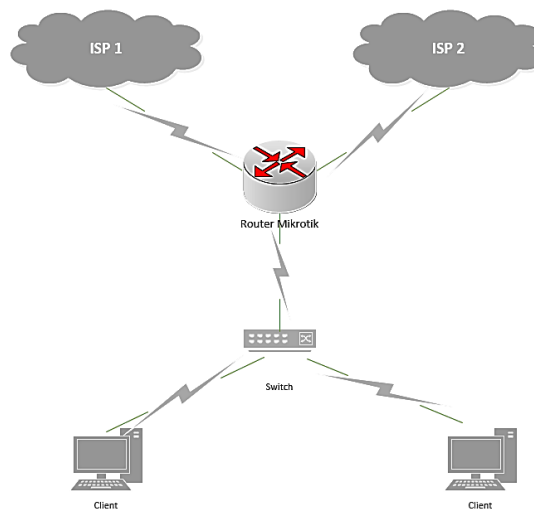
Metode *ECMP* memiliki kelemahan dalam penerapannya yaitu tidak dapat memprediksi *link ISP* yang akan digunakan oleh *router gateway* apabila salah satu *gateway ISP* utama ketika digunakan mengalami *down link*, karena *router gateway* akan selalu menentukan secara acak. Contoh cara kerja *ECMP* dapat dilihat pada gambar 2.2.[16]



Gambar 2. Cara Kerja ECMP

III. ANALISIS DAN RANCANGAN SISTEM

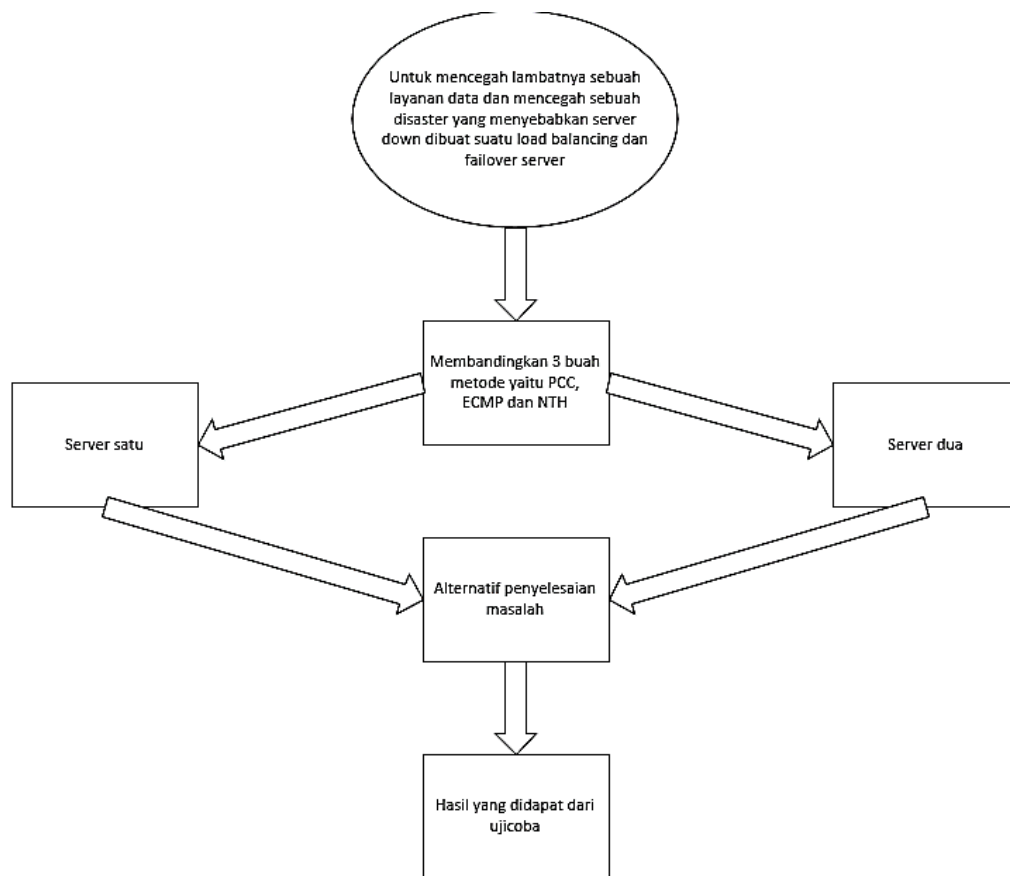
A. Topologi



Gambar 3. Topologi Jaringan

Dalam Gambar 3 memiliki komponen sebagai berikut dua buah server, satu buah switch, satu buah router, dua buah komputer. Kegunaan switch Mikrotik dalam topologi tersebut adalah melakukan request yang diminta oleh client yaitu komputer untuk tersambung ke server

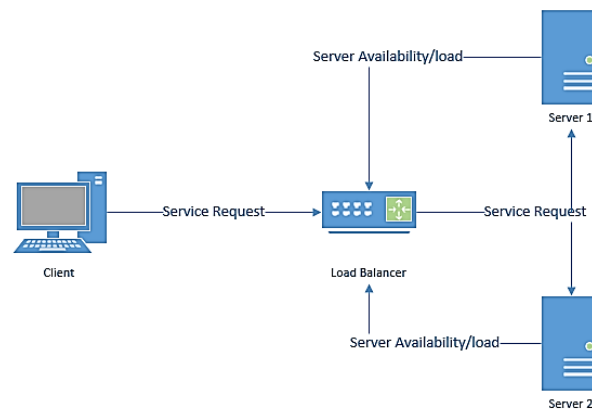
B. Rancangan Komunikasi Client Server



Gambar 4. Rancangan Komunikasi Client Server

Gambar 4 merupakan rancangan *client server* dimana sebuah kasus yaitu untuk mencegah lambatnya sebuah layanan data dan mencegah sebuah *disaster* yang menyebabkan server *down* dibuat suatu solusi yaitu dengan menggunakan *load balancing* dan *failover* dengan membandingkan tiga buah metode yakni PCC, NTH, dan ECMP melalui dua buah server. Alternatif penyelesaian masalah yaitu efektif atau tidaknya metode yang telah digunakan dalam penanganan *disaster* hasil yang didapat dari hasil perbandingan antara tiga buah metode adalah sebuah waktu yang paling cepat yang didapatkan dari hasil ujicoba.

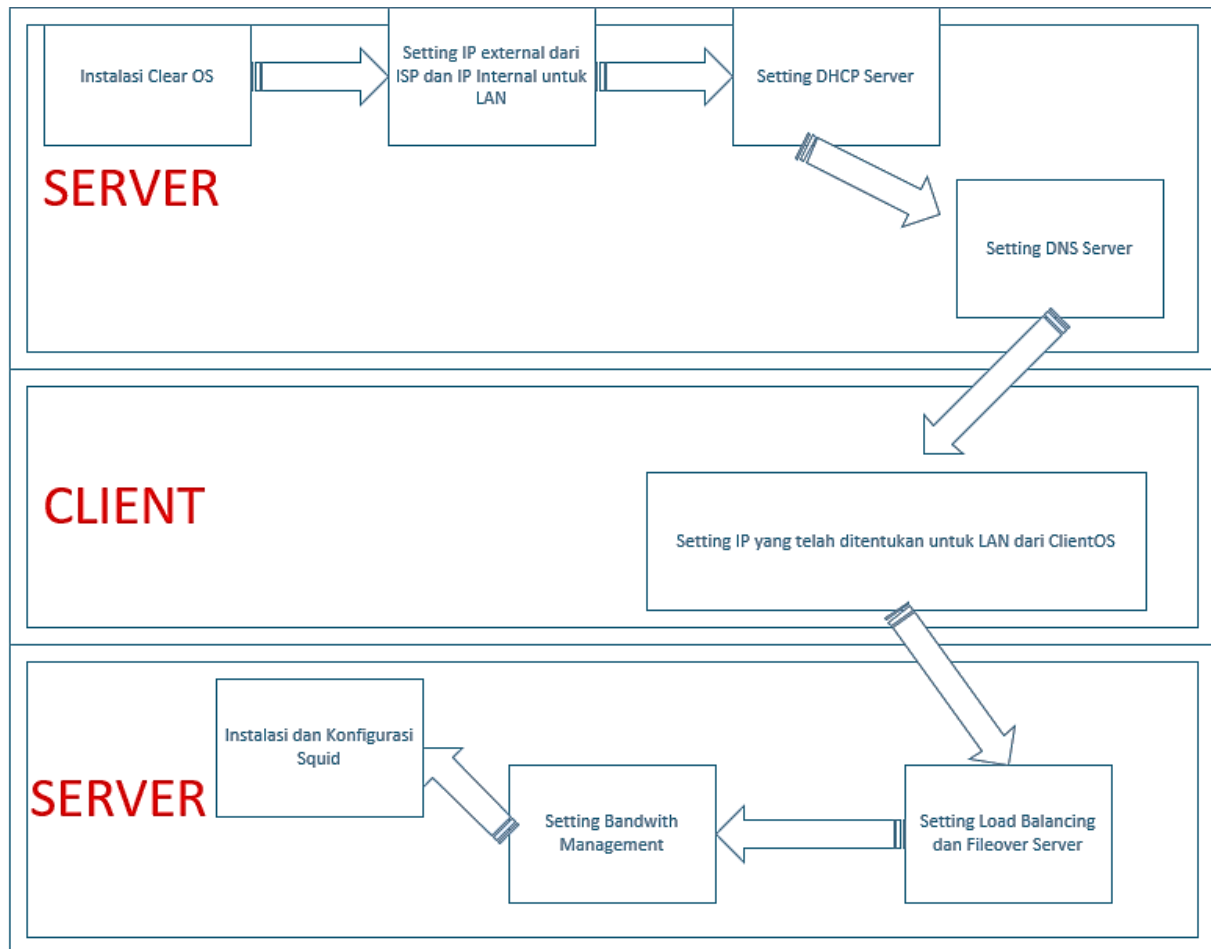
C. Environment Server



Gambar 5. Environment Server

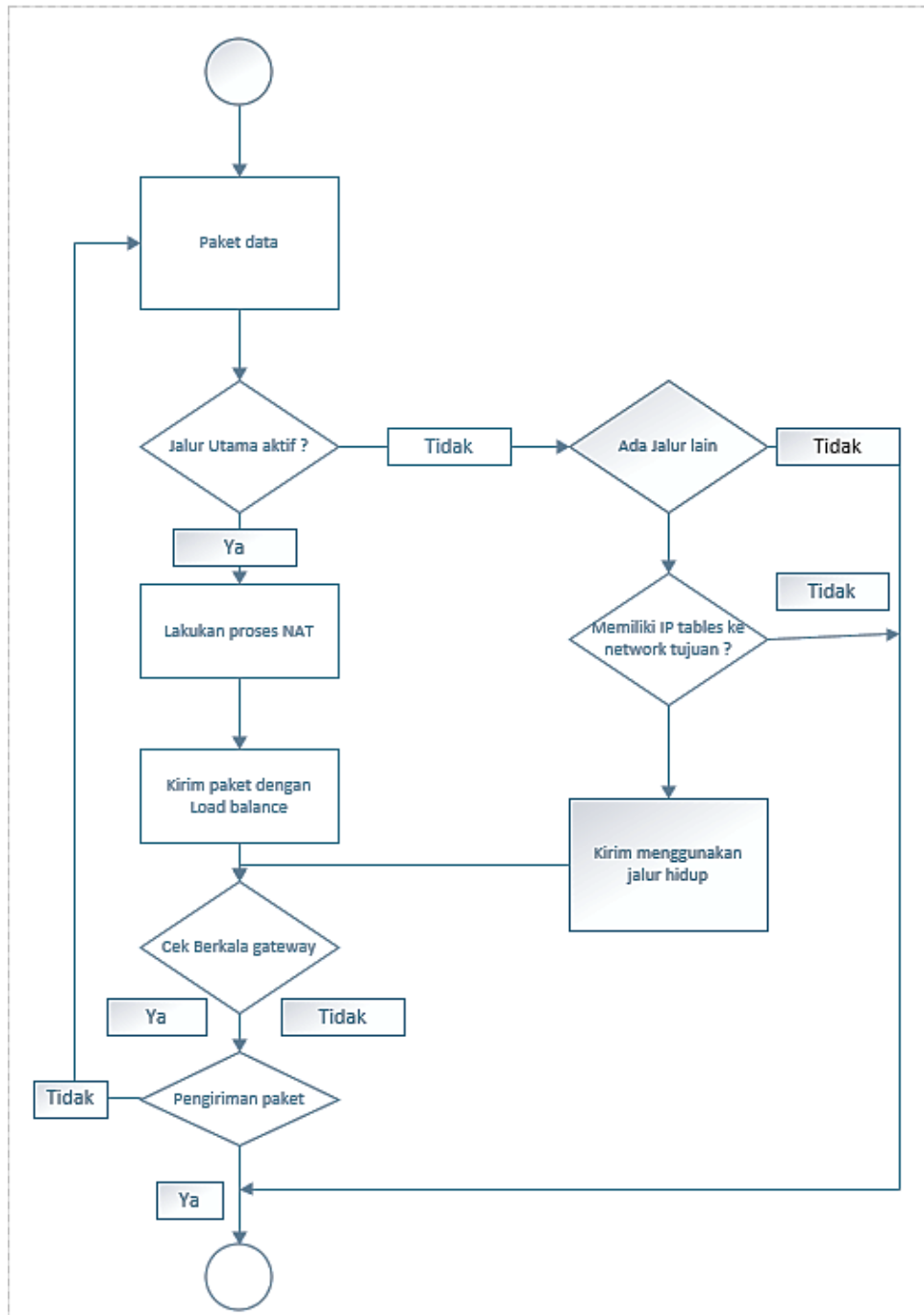
Pada Gambar 5 menceritakan tentang cara kerja sebuah *server* dimana *client* akan meminta *request* terhadap *server load balancer* lalu *server* akan melanjutkan proses *request* terhadap dua buah *server* dan *server* akan merespon dengan mengembalikan *request* terhadap *client*.

D. Rancangan Skenario Pengujian



Gambar 6. Rancangan Skenario Pengujian

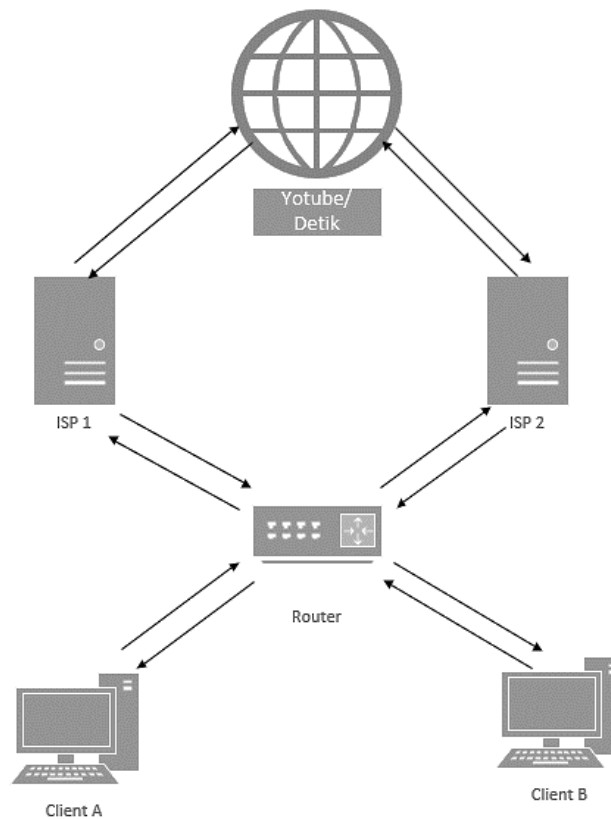
Pada Gambar 6 menjelaskan tentang rancangan scenario pengujian dimana *Server* akan melakukan *instalasi clear OS* lalu *menyetting IP external dari ISP dan IP internal untuk LAN*. Setelah itu *seting DHCP Server, setting DNS Server*. *Client* akan *men-setting IP* yang telah ditentukan untuk *LAN* dari *client OS*. *Server* *men-setting load balancing dan failover server*. *Setting bandwith management* lalu *menginstalasi dan mengkonfigurasi squid*.



Gambar 7. Proses Skenario Pengujian

Pada Gambar 7 menjelaskan tentang proses scenario pengujian. Ada beberapa proses yang bisa terjadi dalam rancangan skenario pengujian. Yang pertama adalah awal proses memeriksa paket data jalur utama aktif atau tidak. Jika tidak menggunakan jalur lain. Jika tidak menggunakan jalur lain maka proses berakhir. Proses kedua yaitu memeriksa paket data jalur utama aktif atau tidak. Jika tidak menggunakan jalur lain. Mengirim *ip tables* ke *network* tujuan. Mengirim menggunakan jalur yang aktif lalu mengecek *gateway*, mengirim paket dan proses selesai. Proses ketiga adalah awal proses memeriksa paket data jalur

utama aktif atau tidak. Jika aktif akan melakukan proses NAT lalu mengirim paket dengan *load balance*, mengecek *gateway* secara berkala paket terkirim dan proses selesai.



Gambar 8. Skenario Pengujian *Load Balancing*

Gambar 8 merupakan *scenario* pengujian *load balancing* yang akan diujikan. Dimana *client A* dan *B* meminta *request* pada router, lalu router membagi beban dimana *client A* mengakses Youtube/Detik melalui ISP 1 dan *client B* melalui ISP 2.

IV. IMPLEMENTASI

A. Implementasi *Load Balancing* dan *Failover* Menggunakan Metode PCC

1. Setting Ip Adress yang mengarah ke ISP 1 dan ke arah ISP 2
2. Setting IP Adress mengarah Ke Local/Lan menggunakan IP 192.168.10.1/24 yang menandakan bahwa ada 254 IP yang dapat digunakan untuk jaringan LAN
3. Setting DHCP-Server Untuk client local
4. Setting DNS Menggunakan Open DNS Google yaitu 8.8.8.8 atau dapat menceklis parameter Allow Remote Request agar menjadikan Gateway pada jaringan LAN menjadi DNS untuk HOST/Client.
5. Configuration *Load Balancing* Metode PCC.
 - Setting Firewall Address List untuk jaringan local
 - Setting Pre routing, accept trafik local dari interface ether4/Local
 - Accept output trafik local

- Membuat mark Connection, untuk manandakan koneksi dari ISP 1 dan ISP 2
 - Membuat firewall jump ke Chain firewall Client-ether4
 - Menandai koneksi yang sudah di tentukan berdasarkan classifier. Classifier 0 ISP 1, classifier 1 ISP 1, Classifier 2 ISP 2
 - firewall return mengulang untuk firewall sebelumnya
 - Membuat mark routing: Mark routing menuju ISP 1 dan ISP 2 dengan action Mark Routing dan isikan New Mark Routing ke con_ISP1 dan con_ISP2, yang akan digunakan untuk menentukan jalur routing.
Membuat Firewall return untuk mark routing.
6. Membuat Route Routing
Membuat routing ke ISP 1 dan ISP 2 yang sudah kita mark routing sebelumnya. Dengan tujuan 0.0.0.0/0 dengan gateway masing-masing ISP dengan parameter Routing Mark yang sudah dibuat sebelumnya.
7. Membuat Firewall Nat Ke masing masing ISP dengan parameter Out-Interface sesuai dengan interface 2 ISP, dimana berfungsi untuk mengubah IP local menjadi Public dan sebaliknya.

B. Implementasi Load Balancing dan Failover Menggunakan Metode NTH

1. Setting IP Address yang mengarah ke ISP 1 dan ke arah ISP 2
2. Setting IP Address untuk LAN menggunakan IP 192.168.10.1/24 , yang berarti akan ada 254 IP yang dapat digunakan untuk jaringan LAN. Gateway pada Jaringan LAN adalah IP yang di isikan pada interface LAN di Mikrotik
3. Melakukan pengujian ke masing-masing gateway ISP untuk memastikan jaringan sudah terhubung ke masing-masing gateway ISP
4. Konfigurasi DHCP-Server agar client mendapatkan IP secara otomatis.
5. a. Konfigurasi Mark connection ke ISP 1 dengan NTH jalur 1
b. New Connection mark to ISP 1 dan New Connection mark ISP 2
c. Membuat mark Routing ke ISP 1 dan ke ISP 2, lalu mengisikan parameter Connection mark yang sudah dibuat untuk masing masing ISP.
6. Membuat Firewall Nat Ke masing masing ISP dengan parameter Out-Interface sesuai dengan interface 2 ISP, dimana berfungsi untuk mengubah IP local menjadi Public dan sebaliknya.
7. Membuat routing ke ISP 1 dan ISP 2 yang sudah kita mark routing sebelumnya. Dengan tujuan 0.0.0.0/0 dengan gateway masing-masing ISP dengan parameter Routing Mark yang sudah dibuat sebelumnya.

C. Implementasi Load Balancing dan Failover Menggunakan Metode ECMP

1. Setting IP Address yang mengarah ke ISP 1 dan ke arah ISP 2
2. Melakukan pengujian ke masing-masing gateway ISP untuk memastikan jaringan sudah terhubung ke masing-masing gateway ISP
3. Melakukan setting DNS, apabila sudah mendapat DNS dari ISP maka bisa menggunakan DNS yang diberikan oleh ISP, jika tidak kita dapat menggunakan Open DNS Google yaitu 8.8.8.8 dan kita dapat memilih Parameter Allow remote Request agar menjadikan Gateway pada jaringan LAN menjadi DNS untuk HOST/Client
4. Setting Ip address mengarah ke LAN, dapat diisikan IP address 192.168.10.1/24, yang berarti akan ada 254 IP yang dapat digunakan untuk jaringan LAN. Gateway pada Jaringan LAN adalah IP yang diisikan pada interface LAN di Mikrotik

5. Konfigurasi DHCP untuk Client agar mendapatkan IP secara otomatis
6. Membuat Firewall Nat ke masing masing ISP dengan parameter Out-Interface sesuai dengan interface 2 ISP, dimana berfungsi untuk mengubah IP local menjadi Public dan sebaliknya.
7.
 - a. Membuat Firewall Mark connection untuk menandakan koneksi yang masuk dari ISP1 dan ISP 2
 - b. Isi parameter action dengan Mark connection, lalu isi parameter New Connection Mark dengan nama ISP-A-Conn dan nama ISP-B-Conn
8.
 - a. Membuat Mark Routing ke luar menuju ISP 1 dengan connection yang sudah di tandai. Dengan action Mark Routing dan isikan New Mark Routing Ke_ISPA dan ISP B yang sudah kita buat sebelumnya, yang akan digunakan nanti untuk menentukan jalur Routing
9. Membuat routing ke ISP 1 dan ISP 2 yang sudah kita mark routing sebelumnya. Dengan tujuan 0.0.0.0/0 dengan gateway masing-masing ISP dengan parameter Routing Mark yang sudah dibuat sebelumnya.

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil pengujian, penggunaan metode PCC memberikan hasil terbaik dibandingkan dengan NTH dan ECMP karena:

- Pada metode *load balance* PCC dengan parameter uji trafik, menunjukkan bahwa metode ini menggunakan sumberdaya pada 1 jalur dengan *bandwidth* yang lebih besar, sedangkan pada metode *load balance* Nth bekerja dengan membagi pemanfaatan sumberdaya di kedua jalur, selain itu metode ECMP membagi sumberdaya di kedua jalur menggunakan sistem perbandingan kecepatan *bandwidth*.
- Pada metode *load balance* Nth dan ECMP dengan parameter uji *packet loss*, menunjukkan bahwa nilai parameter *packet loss* lebih besar dari pada metode PCC dikarenakan metode *load balance* NTH dan ECMP menggunakan algoritma *round robin*, berbeda dengan metode *load balance* PCC yang menggunakan algoritma *hashing*.
- Pada metode *Load balance* Nth dengan parameter uji trafik, menunjukkan bahwa metode ini membagi beban sesuai perbandingan *bandwidth*, dan metode ini menggunakan algoritma *round robin*.
- Pada Ketiga metode di atas dapat disimpulkan bahwa Metode Nth dan ECMP dapat membagi *bandwidth* setiap *user* menjadi merata namun tingkat kestabilannya dalam litensi dan pengiriman *packet* masih terdapat *packet loss*, jika menggunakan PCC *bandwidth* lebih dominan kepada jalur atau *bandwidth* yang lebih besar yang di sediakan oleh ISP, jika sudah ramai trafik maka akan dibagi beban kepada ISP ke 2.
- Metode PCC direkomendasikan karena mengelompokkan trafik koneksi yang keluar masuk router menjadi beberapa kelompok. Pengelompokkan ini bisa dibedakan berdasarkan *src-address*, *dst-address*, *src-port* dan *dst-port*. Mikrotik akan mengingat-ingat jalur gateway yang telah dilewati di awal trafik .koneksi. Sehingga paket data selanjutnya yang masih berkaitan akan lewat pada jalur gateway yang sama dengan paket data sebelumnya yang sudah dikirim. Jadi jika mengakses suatu website yang sama di beberapa komputer berbeda maka akan dilewatkan ke jalur ISP yang sebelumnya sudah di tandai.
- Dan metode ECMP memberikan hasil performa yg paling rendah dengan latency 71.3 saat diujikan pada bandwidth 20.6

B. Saran

Saran yang didapat setelah mengerjakan “Implementasi dan Evaluasi Metode PCC, NTH, ECMP Untuk Load Balancing dan Failover” untuk memberikan nilai availability pada production level, metode PCC baik untuk digunakan karena telah teruji untuk menangani pembagian beban trafik secara merata dan handal terhadap efek failover. Selain itu, metode PCC lebih stabil dibandingkan dengan metode lainnya yaitu NTH dan ECMP.

DAFTAR PUSTAKA

- [1] T. S. F. U. Mohammad Faruq Afif, “Implementasi Disaster Recovery Plan Dengan Sistem Fail Over Menggunakan DRBD dan Heartbeat Pada Data Center FKIP UNS,” vol. 2, April 2013.
- [2] S. S. S. K. Ryan Ramadito, “Analisis Performance Jaringan Komputer Dengan Mekanisme Load Balancing-Failover,” vol. 3, Desember 2010.
- [3] T. P. N. H. Suryanto, “Implementasi Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Dengan Failover Berbasis Mikrotik Router,” *Seminar Nasional Inovasi dan Tren 2018*, 2018.
- [4] M. R. U. Hary Nugroho, “Simulasi Management Bandwidth dan Load Balancing Server Menggunakan Clear OS Pada Virtual Box,” *Jurnal ICT Penelitian dan Penerapan Teknologi*.
- [5] J. P. W. S. R. Jery Alvonsius Rabu, “Implementasi Load Balancing Web Server Menggunakan Metode LVS-NAT,” Vol. %1 dari %28,.No. 2, November 2012.
- [6] A. M. A. A. R. L. R. R. Arief Setyawan, “Analisis Mekanisme Multi Server Load Balancing pada Server SIAKAD Universitas Brawijaya,” *EECCIS*, vol. 8, Juni 2014.
- [7] W. Y. S. R. A. Riski Julianto, “Implementasi Load Balancing di Web Server Menggunakan Metode Berbasis Sumber Daya CPU Pada Software Defined Networking,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, 9 Juni 2017.
- [8] A. A. Indra Warman, “Analisis Kinerja Load Balancing Dua Line Koneksi Dengan Metode NTH,” vol. 5, April 2017.
- [9] I. R. Fiki Justisia Bhayangkara, “Implementasi Proxy Server dan Load Balancing Menggunakan Metode Per Connection Classifier (PCC) Berbasis Mikrotik,” *Jurnal Sarjana Teknik Informatika*, vol. 2, Juni 2014.
- [10] Febriyanti, “Analisis Perbandingan Load Balancing Dengan Metode Cost Multipath (ECMP) dan Per Connection Classifier (PCC) Menggunakan Mikrotik Routerboard,” *Makalah Tugas Akhir*, pp. 7-48, 2016.
- [11] N. F. Zamzami, “Implementasi Load Balancing dan Failover Menggunakan Mikrotik Router OS Berdasarkan Multihomed Gateway Pada Warung Internet “DIGA”,” 2015.
- [12] D. S. Andey Krishnaji, “Implementing Advanced Internet Download Manager,” vol. 3, no. 8, Agustus 2013.
- [13] M. F. Adani, “Analisis Perbandingan Metode Load Balance PCC dengan NTH Menggunakan Mikrotik,” *Tugas Akhir*, 2016.
- [14] A. Evin Nofia Delta, “Performance Test dan Stress Website Menggunakan Open Source Tools,” *Jurnal Manajemen Informatika*, vol. 6, 2016.
- [15] A. Y. Dimara Kusuma Hakim, “Implementasi Load Balancing Menggunakan Jaringan Indihome dan Telkomsel Pada Mikrotik Router Dengan Metode NTH,” *Jurnal Teknik Informatika*, 2015.
- [16] S. Novari, “Penerapan Manajemen Bandwith Pada Amik Akmi Baturaja Menggunakan Mikrotik,” *JIK*, vol. 9, no. 2, Desember 2018.