BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Kajian pustaka yang sudah dilakukan yaitu peneliti menggunakan informasi dari penelitian sebelumnya tentang metode *load balancing* metode *Equal Cost Multi Path* (*ECMP*), *failover* dan *routing BGP*. Sebagai bahan pembanding dari yang sudah ada. Selain itu, peneliti juga menggali informasi dari buku-buku maupun penelitian skripsi dalam rangka mendapatkan informasi yang ada sebelumnya tentang teori-teori yang berkaitan dengan judul yang digunakan untuk untuk landasan teori ilmiah. Berikut ini beberapa penelitian yang terkait dengan *load balancing*, *failover*, *routing BGP*.

Pada referensi penelitian [3] yang memiliki tujuan penelitian mengimplementasikan metode *load balancing Equal Cost Multi Path (ECMP)* di kedua link koneksi internet pada perusahaan PT. Cyberindo Aditama tepatnya divisi *Customer Support Enginer* dan *Customer Accounting Executive*. Hasilnya dari penelitan tersebut metode *load balancing ECMP* mampu membuat *link* koneksi internet berjalan secara bergantian dengan begitu distribusi beban trafik terbagi secara rata di kedua *link*. *Load balancing ECMP* juga mampu memberikan efek *failover*, efek tersebut membuat kondisi apabila salah satu *link* terputus, secara otomatis beban trafik akan dialihkan ke *link* koneksi yang masih aktif. Dari percobaan yang dilakukan pada divisi *Customer Support Engineer* menghasilkan *throughput* 99,96% kategori sangat bagus, *delay* 1,07 ms dengan kategori sangat bagus dan *jitter* 0,02 ms dengan kategori sangat bagus. Sedangkan pada *divisi Customer Accounting Executive throughput* 99,92% kategori sangat bagus, *delay* 1,69 ms kategori sangat bagus dan *jitter* 0,01 ms dengan kategori sangat bagus.

Referensi penelitian [4]. Penelitian tersebut memiliki tujuan membandingkan dua *load balancing* yaitu metode *ECMP* dan *PCC* dengan parameter pengujian *QoS* berupa *throughput, packet loss, jitter dan RTT*. Pada penelitian dibangun sebuah jaringan dengan dua buah *gateway* dan mempunyai

bandwith yang berbeda. *Load balancing* akan berperan menyeimbangkan beban trafik pada kedua *gateway* dan menjalankan fungsi *fail over*. Jaringan komputer yang dibangun pada *virtual machine,* dengan perangkat yang digunakan berupa 1 *router*, 1 *server*, 1 *client*, 1 *load balancer* dan 2 buah *gateway*. Pengujian menggunakan aplikasi *Network Analyzer* dengan memberikan *request* dari *client* ke *server* menggunakan protokol *TCP* dan *UDP*, parameter yang diambil pada pengujian berupa *throughput, packet loss, jitter* dan *RTT*. Hasil dari penelitian tersebut mendapatkan kesimpulan bahwa *load balancing* metode *PCC* menghasilkan nilai *throughput* lebih baik jika dibandingkan dengan metode *ECMP*, karena memiliki nilai rata-rata *throughput* 1943 *Kbits/s*, sedangkan pada metode *ECMP* nilai *thoughputnya* 1942 *Kbits/s*. Metode *PCC* juga dianggap lebih baik ketahanannya ketika terjadi gangguan pada jaringan. Parameter *RTT* dari metode *PCC* juga lebih baik dibandingkan dengan *ECMP*, dengan nilai *RTT* 2,908 *ms* pada *PCC* dan 2,283 *ms* pada *ECMP*.

Pada peneltian [5] Penelitian dilakukan untuk membandingkan kinerja *load balancing ECMP* (*Equal Cost Mulit Path*), metode *PCC* (*Per Connection Classifier*) dan metode *load balancing Nth*. Penelitian dilakukan dengan menggunakan studi kasus pada jaringan Institut Sains dan Teknologi AKPRIND. Pengujian dilakukan menggunakan parameter pengujian berupa *delay, jitter, packet loss* dan *throughput*. Dari hasil pengujian yang dilakukan, metode *load balancing PCC* lebih baik dibandingkan 2 metode lainnya. Pada pengujian parameter *throughput PCC* nilai rata-ratanya 630 *Kbps*, *ECMP* 569 *Kbps*, dan *Nth* 586 *Kbps*. Parameter *packet loss*, metode *PCC* bernilai 0,20%, *ECMP* 0,42% dan *Nth* 0,50%. Pada *jitter*, *PCC* memiliki rata-rata nilai 12,94 *ms*, *ECMP* 15,37 *ms*, dan *Nth* memiliki rata-rata nilai 13,73 *ms*. Pada parameter *delay*, nilai rata-rata metode *PCC* 12,95 *ms*, *ECMP* 15,38 *ms*, dan *Nth* 13,77 *ms*. Pembagian jalur trafik metode *PCC* dan *Nth* juga lebih baik karena pembagian jalur dilakukan secara merata dan bisa menggunakan dua jalur secara bersamaan. Pada pengujian *CPU load*, dihasilkan *Nth* lebih baik dengan nilai beban pada *CPU load* sebesar 32%, *ECMP* sebesar 34% dan *PCC* mencapat 61%.

Pada penelitian [6] menggunakan studi kasus jaringan kampus IST AKPRIND Yogyakarta, jaringan kampus menggunakan tiga jalur *ISP* (*Internet*

Service Provider) yaitu *Excelindo*, *Biznet* dan *Astinet*. Masalah yang sering dihadapi adalah terjadinya kegagalan *link*. Solusinya dengan metode *fail over*, salah satu cara membuat *fail over* dengan menggunakan protokol *routing Border Gateway* Protokol *(BGP*). *Routing BGP* memiliki *hold time*, dimana semakin kecil *hold time* maka perpindahan *link* akan semakin cepat. Dari hasil penelitian tersebut didapatkan bahwa waktu konvergensi dipengaruhi oleh *hold time*. *Hold time* mencegah terjadinya *router oscillation*, atau koneksi tidak langsung terputus sampai *hold time expired* ketika *router* mati. Fitur *BGPtimer* pada *BGP* membuat kinerja *failover* lebih optimal. Dari pengujian, *ISP* dengan nilai konvergensi terendah memiliki kualitas kerja lebih optimal.

Pada penelitian [7] menganalisa *Link Balancing* dan *Failover* 2 *Provider* Dengan *routing Border Gateway* Protokol (*BGP*) pada *Router Cisco 7606s*. Pengujian dilakukan dengan menggunakan perangkat 4 buah *router*, yaitu satu *router Cisco* 7606s sebagai *main gateway*, dua *router Mikrotik RB951G 2HND* sebagai jalur dua *ISP*, dan *router Mikrotik CCR1036*-12G-*4S* sebagai *gateway FTP server*. *Router* yang digunakan memiliki *Autonomous System Number* yang berbeda dan terhubung menggunakan melalui metode *routing Border Gateway* Protokol (*BGP*). Berdasarkan hasil penelitian tersebut penggunaan *link balancing* dapat membuat penggunaan *bandwith* lebih optimal, terbukti pada penggunaan *link balancing* yang diberikan *bandwith* sebesar 10 *mb*/s mampu menghasilkan nilai *throughput* sebesar *794 KB*/s. Sedangkan yang tidak diimplementasikan *link balancing* walaupun diberikan nilai *bandwith* sebesar 20 *mb*/s nilai *throughput* hanya sebesar 136 *KB*/s. *Link balancing* juga memberikan perbedaan pada saat *traffic* padat, nilai *throughput* dengan *link balancing* memiliki *nilai 8,21 MB*/s, sedangkan yang tidak menggunakan *link balancing* memiliki nilai *throughput* sebesar 1,25 *MB*/s. *Delay* menggunakan *link balancing* sebesar 1.24 *ms*, sedangkan tanpa *link balancing delay* bernilai 8,19 *ms*. Nilai *latency* juga berbeda, dengan *link balancing* memiliki nilai 12 *ms*, tanpa *link balancing* mencapat 1896 *ms*.

Pada penelitian [8] melakukan analisa dan simulasi *routing BGP* antar *autonomous system* menggunakan *FRR Routing* penelitian tersebut menggunakan 2 topologi, dengan topologi pertama menggunakan 4 *FRR* dan topologi kedua

menggunakan 6 *FRR* dengan menggunaakn protokol *UDP*. Nilai ukuran data yang diujikan bervariasi, yaitu 10 MB, 20 MB, 30 MB, 40 MB dan 50 MB menggunakan skenario dengan *fail-over* dan tanpa *fail-over*. Hasilnya pada topologi pertama tanpa *fail-over* nilai *througputnya* memiliki nilai 10,805 *Mbps*, *delay* 3,441 *ms*, *jitter* 0,747 *ms* dan *packet loss* 0%, sedangkan topologi pertama dengan *fail-over* nilai *throughputnya* 10,758 *Mbps*, *delay* 100,039 *ms*, *jitter* 0.928 *ms* dan *packet loss* 0,349 %. Pengujian topologi kedua tanpa *fail-over* nilai *throughput*nya 10,797 *Mbps*, *delay* 124,786 *ms*, *jitter* 1,132 *ms*, *packet loss* 0%, sedangkan topologi 2 dengan *fail-over* memiliki nilai *throughput* 10,734 *Mbps*, *delay* 445,864 *ms* dan *packet loss* 0,341%.

Berikut ini pemetaan dengan penelitian sebelumnya dan penelitian yang akan dilakukan pada Tabel 2.1.

Tabel 2. 1 Pemetaan Dengan Penelitian Sebelumnya

Penelitian Oleh	*Load Balancing ECMP*	*Routing BGP*	Parameter *QoS*	*Open Network*
Muhammad Khoirul Anwar,da Nurhaida	✓		✓	
Muhammad Iqbal Firdaus	✓		✓	
Reza Pakiding, Catur Iswahyudi, Rena Yanwastika Ariyana	✓		✓	
Asih Meirani, Yuliana Rachmawati, Muhammad Sholeh		✓		
Darmawan, Teguh Imanto		✓	✓	
Muhammad Sahal Nurhidayah, Reni Dyah Wahyuningrum, Dadiek Pranindito		✓	✓	✓
Muhammad Arli Fauzi	✓	✓	✓	✓

2.2 DASAR TEORI

Beberapa teori yang mendukung penelitian yang dilakukan juga menjadi dasar dari penelitian.

2.1.1 *IP Address*

IP Address atau Internet Protocol Address, merupakan alat identifikasi untuk setiap komputer dalam jaringan internet berbasis TCP/IP. IP address berisi deretan angka biner. *IP* versi 4 (IPv4) menggunakan deretan angka biner sepanjang 32-*bit*. Sedangkan *IP* versi 6 (IPv6) menggunakan deretan angka biner sepanjang 128-*bit*. *IP Address* memiliki fungsi sebagai identifikasi host dan alamat lokasi jaringan. Fungsi *IP address* sebagai alamat lokasi jaringan dapat diilustrasikan sebagai sebuah nama untuk mempermudah mengingat kata daripada mengingat deretan angka yang menunjuk pada suatu *website*, sebuah alamat untuk mengetahui di mana *website* tersebut, dan juga sebuah rute agar dapat mencapai alamat tersebut.[9]

IP Address versi 4 juga memiliki klasifikasi kelas *IP*, setiap kelas *IP* memiliki ukuran dan jumlah alamat yang berbeda

a. *IP* kelas A

IP kelas A umumnya digunakan pada jaringan dengan skala besar karena kapasitasnya yang memadai, tetapi jika dibandingkan dengan IP kelas lainnya, kelas A cenderung memiliki jumlah pembentukan jaringan yang lebih sedikit

Rentang *IP* kelas A : 0.0.0.0 – 127.255.255.255

Jumlah alamat *IP* maksimal : 16.777.216

Jumlah jaringan maksimal : 128

b. *IP* Kelas B

IP kelas B diperuntukan untuk jaringan dengan skala menengah ke atas dan memiliki maksimal jaringan 16.384 dengan masing masing jaringan bisa menampung 65.534 *host*

Rentang *IP* kelas A : 128.0.0.0 – 191.255.255.255

Jumlah alamat IP maksimal : 1.048.576

Jumlah jaringan maksimal : 16.384

c. *IP* Kelas C

IP kelas C digunakan untuk skala jaringan yang jauh lebih kecil karena hanya mampu menampung 254 *host* saja, umumnya digunakan pada rangkaian *internet* protokol *private* seperti sekolah, universitas, dan lainnya.

Rentang *IP* kelas A : 192.0.0.0 – 223.255.255.255

Jumlah alamat IP maksimal : 65.536

Jumlah jaringan maksimal : 2.097.152

d. *IP* Kelas D

IP kelas D digunakan untuk *multicasting*, alamat IP kelas D memiliki 4 *bit* pertama bernilai 1110 sekaligus sebagai *network identifier* dan 28 *bit* berikutnya untuk *host identifier*. Range IP kelas D dimulai dari 224.0.0.0 hingga 239.255.255.255.

e. *IP* Kelas E

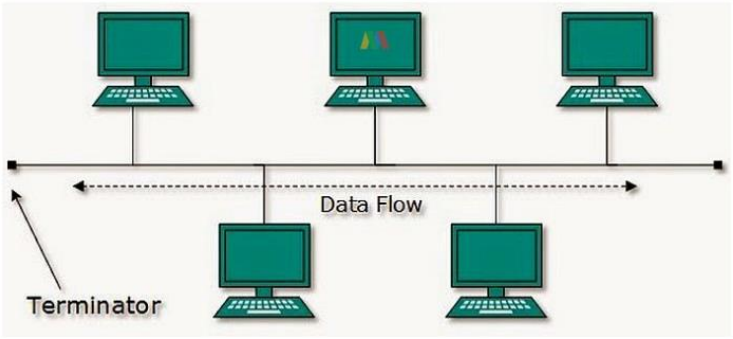
IP kelas E digunakan untuk eksperimen, 4 *bit* pertama pada oktet pertama kelas E bernilai 1111 yang berfungsi sebagai *network* identifier, 28 bit berikutnya berfungsi sebagai *host identifier*. *Range IP* kelas E dimulai dari 240.0.0.0 hingga 254.255.255.255.[10]

2.1.2 Topologi Jaringan

Topologi jaringan merupakan struktur atau metode yang digunakan untuk menghubungkan satu komputer dengan komputer lainnya menggunakan kabel maupun nirkabel, dengan adanya topologi jaringan akan memudahkan dalam menentukan cara pengiriman data, menentukan kinerja dari jaringan, mempermudah diagnosis apabila terjadi masalah pada jaringan, membantu pemilihan teknologi yang akan digunakan pada jaringan, dan membantu perentanaan dan juga pemeliharaan yang akan dilakukan pada jaringan. Ada beberapa jenis topologi, yaitu:

a. Topologi Bus

Gambar 2.1 merupakan topologi bus yang menghubungkan semua terminal ke satu jalur komunikasi yang kedua ujungnya ditutup dengan terminator.



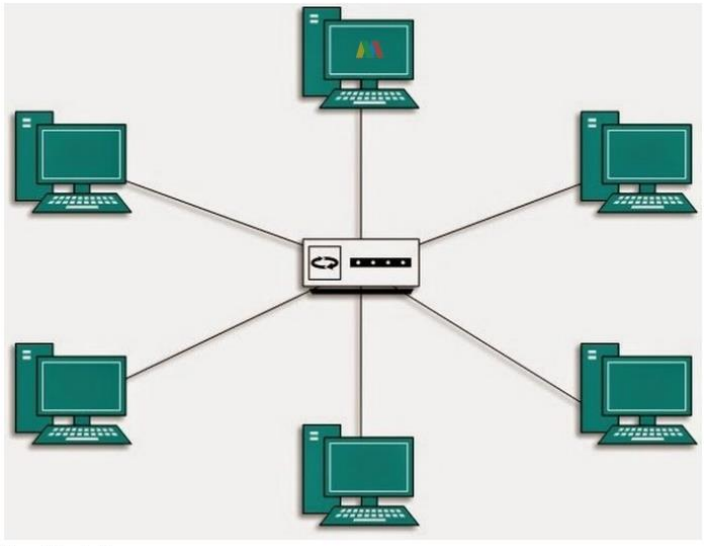
Gambar 2. 1 Topologi Bus[11]

Karakteristik topologi bus, yaitu :

- Node dihubungkan secara serial sepanjang kabel.
- Sangat sederhana.
- Paket data saling bersimpangan pada sebuah kabel.
- Tidak memerlukan hub, tetapi banyak memerlukan *T connector*.

b. Topologi *Star*

Merupakan topologi dimana setiap node terhubung ke jaringan melalui hub, atau *concentrator*, seperti yang digambarkan pada Gambar 2.2.



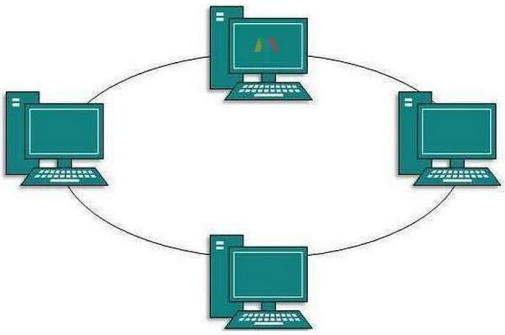
Gambar 2. 2 Topologi *star*[12]

- Setiap node berkomunikasi langsung dengan hub.
- Kinerja jaringan akan menurun apabila setiap data yang masuk di*broadcast* ke semua node.

- Jika salah satu *port* rusak, keseluruhan jaringan masih bisa berjalan.

c. Topologi *Ring*

Merupakan topologi dimana setiap *node* terhubung dan membentuk pola lingkaran, dan tiap *workstation* akan menerima dan melewatkan informasi dari satu komputer ke komputer lainnya. Seperti yang digambarkan pada gambar 2.3.



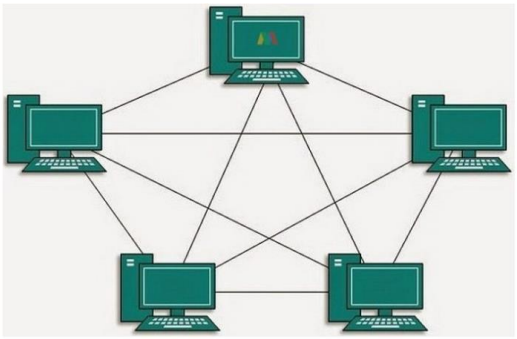
Gambar 2. 3 Topologi *Ring*[13]

karakteristik dari topologi *ring*, yaitu:

- Node dihubungkan secara serial sepanjang kabel.
- Paket data mengalir searah.
- Jika salah satu node rusak, maka semua *node* tidak bisa berkomunikasi.

d. Topologi *Mesh*

Topologi mesh menjadi topologi yang cukup banyak digunakan karena ketersediannya sangat baik. Seperti yang digambarkan pada gambar 2.4.



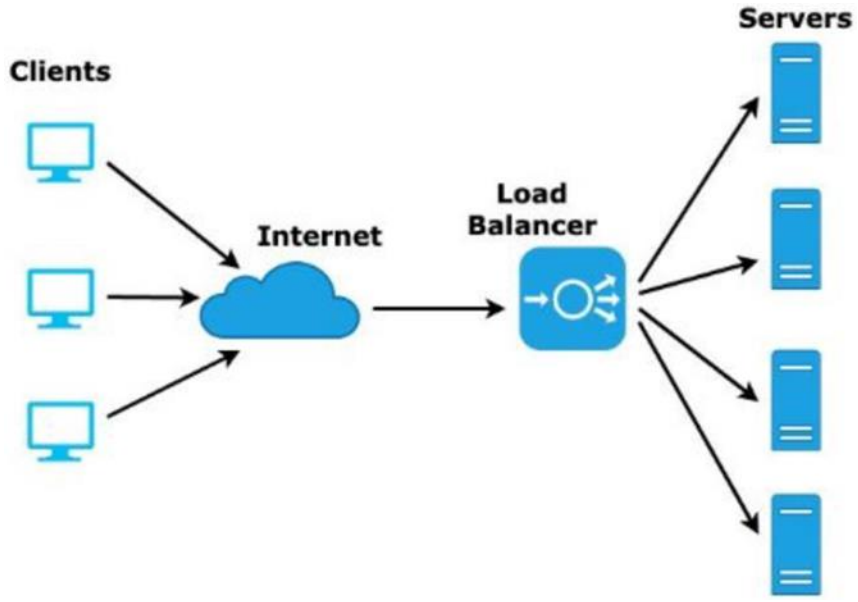
Gambar 2. 4 Topologi *Mesh*[14]

Merupakan topologi yang susunannya saling terhubung ke semua node perangkat, karakteristik dari topologi *ring*, yaitu:

- Jika perangkatnya banyak akan sulit di *maintenance*.
- Setap *node* terhubung satu sama lain ke semua *node*.[15]

2.1.3 *Load Balancing*

Load balancing merupakan metode pembagian beban traffic server menjadi dua atau lebih secara sama atau seimbang [15]. Beberapa manfaat yang didapatkan jika menggunakan metode load balancing, diantaranya meningkatkan kecepatan respon, menghindari overload hingga memperkecil kemungkinan terjadinya downtime. Gambaran load balancing dijelaskan dalam ilustrasi pada gambar 2.5 berikut ini :



Gambar 2. 5 Ilustrasi *Load Balancing*

Dari ilustrasi tersebut, menjelaskan bahwa *traffic* yang masuk akan melalui *load balancer*, selanjutnya load balancer akan mendistribusikan beban *traffic* ke semua *server* secara merata [16]. Terjadinya load balancing tidak lepas dari beberapa syarat yang harus dipenuhi, yaitu:

- Ada dua atau lebih *resource* yang bisa dikelola dan dibagikan beban kerjanya.
- *Workload* bervariasi, beban kerja yang bervariasi akan membuat *load balancing* akan bekerja lebih efisien.
- Algoritma yang digunakan, perlu ditentukan algoritma yang akan digunakan untuk membagi dan mendistribusikan beban kerja.

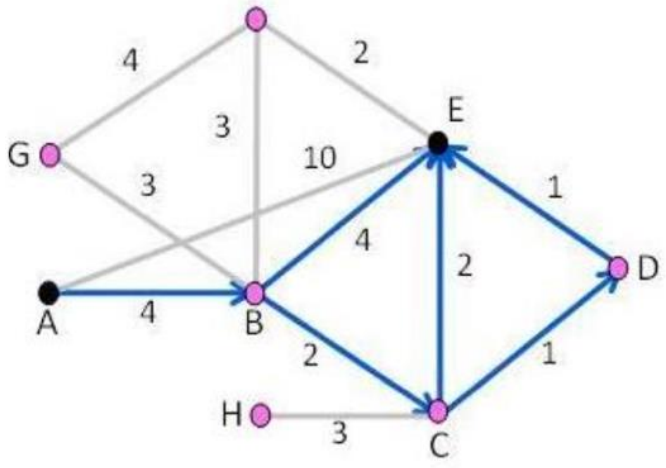
- *Fault tolerance*, sistem harus bisa mengatasi kegagalan sumber daya.
- Antar sumber daya harus terhubung baik dengan jaringan yang sama atau jaringan yang berbeda.

Ada 4 metode *load balancing* sering digunakan, yaitu: *Equal Cost Multi Path* (*ECMP*), *Per Connection Classifier* (*PCC*), dan *Nth*.

2.1.4 *Equal Cost Multi Path (ECMP)*

Equal Cost Multi Path merupakan metode *load balancing* dimana untuk jalur yang dipilih dilakukan secara bergantian pada *gateway*.[17] Nilai *equal cost* sendiri dapat didefinisikan secara asimetris atau tidak seimbang pada saat terjadi proses *routing*.

Cara kerja *load balancing ECMP* adalah dengan mengkonfigurasi beberapa default *gateway* dengan *distance* yang sama, nantinya *router* akan menggunakan *gateway-gateway* tersebut secara bersamaan. *Load Balancing ECMP* juga memungkinkan penggunaan lebih dari 1 jalur yang berperan sebagai *exit interface*, pada *load balancing ECMP* juga tidak dapat diprediksi *link* yang digunakan karena *router gateway* selalu ditentukan secara acak



Gambar 2. 6 Ilustrasi *Load Balancing ECMP*

Ilustrasi load balancing ECMP digambarkan pada Gambar 2.6. Pada *ECMP*, setiap *router* menggunakan *hash function* pada setiap *header* trafik, contohnya seperti *IP* pengirim dan *IP* penerima, tujuannya agar trafik dapat dipetakan ke dalam beberapa jalur yang tersedia yang terimplementasi pada jaringan [18].

Dari ilustrasi pada Gambar 2.6 antar rute memiliki *cost* yang berbeda beda. Pada metode *ECMP*, apabila akan menempuh jarak dari A ke E, maka jalur terbaik melalui ABE karena memiliki *cost* 8, ABCE karena memiliki nilai *cost* 8, dan ABCDE karena memiliki nilai *cost* 8, dimana masing masing jalur memiliki total *cost* yang sama dan memiliki total *cost* terendah dibandingkan jalur lainnya. Penentuan *cost* dari *ECMP* dipengaruhi oleh beberapa faktor diantaranya: Jarak ke tujuan, kecepatan *link*, keandalan *link*, dan kualitas layanan, cost *ECMP* yang lebih rendah akan memiliki prioritas lebih tinggi. beberapa *routing* yang *support* dengan *ECMP* diantaranya: *Intermediate System to Intermediate System (ISIS)*, *Enchanced Interior Gateway Routing* Protokol *(EIGRP)*, *Open Shortest Path First (OSPF)*, dan *Border Gateway* Protokol *(BGP)* [19].

Penggunaan atau pemilihan metode *Load balancing ECMP* juga tidak lepas dari kelebihan dan kekurangannya, kelebihannya *ECMP* adalah mampu membagi beban jaringan, meningkatkan keandalan dari jaringan karena *ECMP* akan memilih jalur terbaik dan apabila salah satu jalur gagal maka akan dialihkan ke jalur lain yang masih aktif. Selain itu kekurangan pada *ECMP* adalah sering terjadi *disconnect* yang disebabkan *restartnya routing table* yang terjadi otomatis, *ECMP* tidak murah karena semakin banyak jalur semakin banyak biaya yang dibutuhkan [20].

2.1.5 *Routing Protocol*

Routing protocol merupakan protokol yang ada pada roouting dinamik, fungsinya adalah menentukan jalur terbaik yang akan dilewati serta memperbarui informasi *table routing* apabila terjadi perubahan jaringan.[21]

Ada 2 pembagian *protocol routing*, yaitu:

1. ***IGP (Interior Gateway Protocol)***

   Protokol *routing* ini melakukan pertukaran informasi dalam satu *Autonomous System* (*AS*) yang sama. Contoh dari *IGP*, yaitu:

   - *RIP (Routing Information Protocol)*
   - *IGRP (Interior Gateway Routing Protocol)*
   - *OSPF (Open Shortest-path First)*
   - IS-IS (*Intermediate System to Intermediate System*)

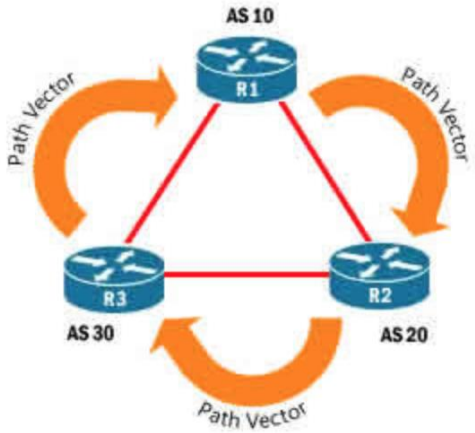
2. ***EGP* (*Exterior Gateway Protocol*)**

Prototokol routing *EGP* merupakan protokol *routing* yang digunakan untuk menghubungkan jaringan antar *Autonomous System* (*AS*). Contoh dari *EGP* adalah routing *BGP* (*Border Gateway Protocol*)

2.1.6 *Border Gateway Protocol (BGP)*

Border Gateway Protocol atau *BGP* bisa disebut sebagai *routing* utama dari internet. *Routing BGP* masuk dalam jenis protokol routing *EGP (Exterior Gateway Protocol)*. Definisi dari *BGP* merupakan sebuah protokol *routing* yang berfungsi melakukan pertukaran *table routing* antar *Autonomous System* (*AS*). *BGP* memiliki kelebihan mampu mengindari *loop path selection*, *loop path selection* sendiri berarti pengiriman paket data dilakukan secara terus menerus dengan rute yang berputar-putar [22].

Cara kerja dari *BGP* adalah dengan mentranslasikan *IP network* dengan *path vector*, tabel *routing* akan dikirim ke setiap *neighbor* kemudian akan ada pemberitahuan *update* tabel *routing* otomatis [23]. Penentuan rute dari *BGP* menggunakan algoritma *path vector*, dimana *path vector* akan mengumpulkan informasi jalur tujuan dari *router* lain. Berikut ini ilustrasi dari cara kerja *BGP* digambarkan pada Gambar 2.7:



Gambar 2. 7 Cara Kerja *BGP*

Routing BGP memiliki kelebihan dan kekurangan, kelebihannya diantaranya *routing BGP* sangat efisien, dapat diterapkan dalam jaringan skala besar, cocok digunakan pada *ISP* maupun jaringan dengan koneksi *host* yang banyak. Kekurangan dari *routing BGP* diantaranya, apabila terjadi salah

konfigurasi memungkinkan terjadinya memilih jalur ke jaringan yang tidak terkait, sangat terbatas dalam penggunaan topologi.

Routing BGP sendiri dibagi menjadi 2 jenis, yaitu:

a. *iBGP (Internal BGP*) merupakan pertukaran informasi *routing network* dari *Autonomous System* (*AS*) yang sama.
b. *eBGP (External BGP)* merupakan pertukaran informasi *routing network* dari *Autonomous System* yang berbeda [24].

Routing BGP juga memiliki beberapa karakteristik, yaitu:

a. Menggunakan algoritma *routing distance vector.*
b. Digunakan pada *ISP* dengan *ISP* dan *client*.
c. Digunakan untuk merutekan trafik *autonomous system* (AS).
d. Membuat dan menjaga koneksi antar *peer* dengan *port* 179.
e. Jika terjadi perubahan topologi *table routing* akan diupdate antar *router* yang berhubungan [25].

2.1.7 *External Border Gateway Protocol (eBGP*)

eBGP (External Border Gateway Protocol), digunakan untuk bertukar informasi *routing network* antar *Autonomous System* (*AS*) yang berbeda, karena itu *eBGP* masuk dalam kategori *routing EGP (External Gateway Protocol)*. Dengan pembagian informasi *routing* memungkinkan *AS* berbagi informasi untuk menentukan jalur terbaik. Tujuan dari *eBGP* adalah untuk menjaga pemisahan antar *AS* dan memberikan fasilitas pertukaran informasi antar *AS*

Penggunaan *EBGP* pada penelitian ini tidak lepas dari kelebihan dari *eBGP* yaitu mampu menghubungkan antar *Autonomous System* (*AS*) yang berbeda, mampu menentukan rute terbaik ke tujuan, cocok diterapkan pada jaringan dengan skala yang besar.[26]

Dari hal tersebut, penerapan *eBGP* sangat cocok digunakan dengan *load balancing ECMP*. *eBGP* yang banyak digunakan pada jaringan skala besar seperti *provider* dimana *provider* memiliki *Autonomous System Number (ASN)* yang berbeda, membuat *load balancing ECMP (Equal Cost Multi Path)* bisa dimanfaatkan untuk membagi beban trafik dari jaringan yang memiiki *Autononous System* yang berbeda.

2.1.8 *Autonomous System (AS)*

Autonomous System merupakan satu jaringan yang atau sekumpulan jaringan yang berada di bawah kontrol satu administrasi. AS terdiri dari sekelompok *router* yang membentuk jaringan yang masih berada dalam satu hak administrasi, satu kepemilikan, satu kepentingan dan menggunakan *policy* yang sama. Biasanya dalam satu *AS* diberikan kebebasan untuk bertukar informasi.

Autonomous system memiliki nomor identifikasi yang berbeda-beda, nomor ini diberikan dan diatur oleh *Internet Assigned Number Authority (IANA)*, pemberian nomor *AS* antara 1 sampai 65.535. Untuk *private number AS* berada antara 64.512 hingga 65.535, penggunaan *private number* nomor *AS* tidak boleh sampai keluar jaringan dari *AS* tersebut, karena bisa mengacaukan sistem pengalamatan *AS*.[27]

2.1.9 *Metric Routing*

Metric pada routing merupakan unit yang dihitung dengan algoritma *routing* untuk memilih jalur untuk lalu lintas data. *Metric routing* dihitung dengan algoritma *routing* pada saat menentukan rute optimal. Beberapa parameter yang digunakan untuk mempertimbangkan nilai dari *metric routing*, yaitu:

- *Hop count*
- Keandalan jalur
- Kecepatan jalur
- Beban
- *Bandwith*
- Latensi[28]

2.1.10 *Open Network*

Open network mereferensikan jaringan yang memisahkan *hardware* dan juga *software*. Tujuan dari *open network* untuk menyediakan jaringan yang andal, fleksibel dan dapat dikustomisasi sesuai dengan kebutuhan *developer* jaringan. Beberapa kelebihan dari penggunaan dan implementasi dari *open network* diantaranya:

- **Fleksibel**

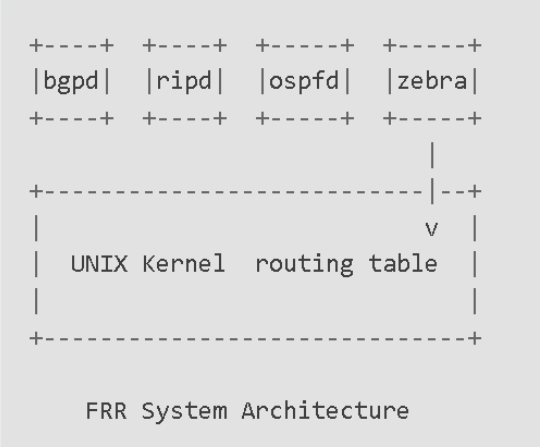
  Open network memiliki kelebihan lebih fleksibel dibandingkan dengan *close vendor*, memungkinkan banyak opsi pada saat penerapan di suatu instansi. *Open network* membuat anda bebas menerapkan topologi, dan arsitektur yang disesuaikan dengan kebutuhan. Selain itu *open network* memungkinkan *developer* memprogram jaringan secara langsung dan tidak terikat dengan protokol seperti *close vendor*.
- **Ketersediaan**

  Open network dikelola dengan antarmuka dan protokol yang umum, memudahkan pengujian, validasi, dan penerapan jaringan lebih cepat. *Open* network juga memudahkan dalam memaksimalkan *resource* jaringan yang ada.
- **Efisiensi biaya**

  Biaya yang dibutuhkan untuk membangun *open network* relatif lebih murah karena *developer* memiliki kebebasan dalam menentukan perangkat yang akan digunakan.[29]

2.1.11 *FRRouting*

Free Range Routing (FRRouting) adalah *IP routing suite* yang memiliki banyak fitur, memiliki performa yang tinggi dan memiliki sifat *open source*. Arsitektur FRRouting digambarkan pada Gambar 3.1.



Gambar 3. 1 Arsitektur *FRRouting*

FRRouting mampu diimplementasikan dengan semua protokol *dynamic* seperti *OSPF, IS-IS, RIP* hingga *BGP*. *FRR* perangkat yang cocok digunakan pada perangkat mulai dari *SBC* murah hingga kelas komersial. *FRR* berfungsi sama seperti *close vendor* seperti *Cisco, Mikrotik*, dan sebagainya, yaitu sebagai *routing stacks* yang berfungsi untuk mengelola rute dari jaringan. Penerapan *FRR* banyak digunakan di universitas, laboratorium hingga perusahaan [30]. Beberapa kelebihan dari *FRRouting* diantaranya, sudah banyak digunakan di banyak perusahaan IT, konfigurasi mirip dengan *vendor Cisco*, berjalan pada *OS Linux* [31].

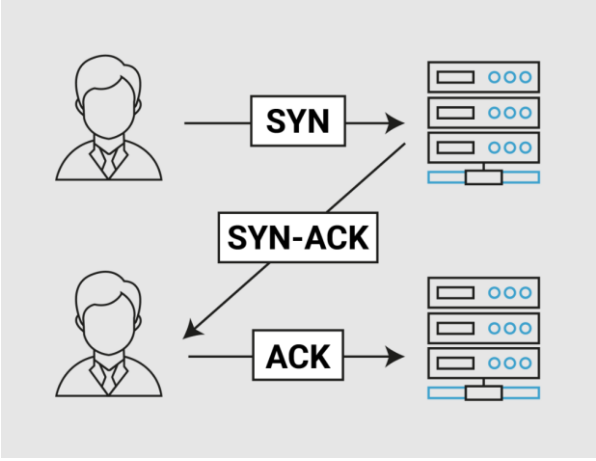
FRRouting bekerja menggunakan *kernel UNIX* dengan beberapa kumpulan daemon yang bekerja sama membangun sebuah *table routing*.

2.1.12 Protokol Komunikasi Data

Pada jaringan, ada 2 jenis protokol yang umum digunakan, yaitu *TCP (Transmission Control Protocol)* dan *UDP (User Datagram Protocol)*, kedua protokol komunikasi data tersebut memiliki karakteristiknya masing-masing.

a. *Transmission Control* Protokol *(TCP)*

Transmission Control Protokol *(TCP)* adalah standar komunikasi data yang digunakan pada untuk bertukar data dari satu komputer ke komputer lain di *internet* [32]. Dalam proses pengiriman data, protokol *TCP* memberikan nomor urut pada setiap paket data yang dikirim, ketika paket diterima penerima akan mengirimkan pemberitahuan pada pengirim, setelah itu pengirim dapat mengrimkan paket lainnnya.

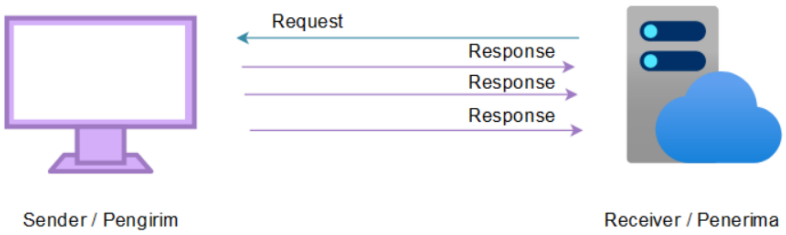


Gambar 2. 8 *Three Way Handshake* pada Komunikasi Protokol *TCP*

Gambar 2.8 menggambarkan pengiriman data yang dilakukan secara urut meminimalisir terjadi pengiriman data yang macet, hanya saja pengiriman dengan protokol *TCP* memerlukan waktu lebih lama, karena banyak terjadi komunikasi dua arah[33].

b. *User Datagram* Protokol *(UDP)*

UDP atau *User Datagram* Protokol merupakan protokol yang mendukung komunikasi yang tidak andal (*unreliable*). Ilustrasi UDP digambarkan pada Gambar 2.9.



Gambar 2. 9 *Flow* Komunikasi dengan Protokol *UDP*

Pada proses pengiriman data dengan prokotol *UDP*, tidak diberikan nomor urut seperti *TCP*, protokol *UDP* hanya melakukan pengiriman tanpa melakukan koreksi pengiriman. Hasilnya *UDP* rentan terjadinya kesalahan, tetapi pengiriman data dengan protokol *UDP* lebih cepat dibandingkan [34].

c. Perbedaan protokol *TCP* dan *UDP*

Ada beberapa perbedaan antara protokol *TCP* dan *UDP*, diantaranya:

- Protokol TCP berorientasi pada koneksi, sedangkan protokol *UDP* tidak berorientasi pada koneksi.
- *TCP* menjamin pengiriman data, sedangkan *UDP* tidak menjamin pengiriman data.
- Protokol *TCP* mengikuti *flow control*, protokol *UDP* tidak mengikuti mekanisme *flow control*.
- Protokol *TCP* mengirimkan data secara urut, protokol *UDP* dikirim dalam urutan apapun.
- Kecepatan pengiriman data protokol *TCP* cenderung lebih lambat dibandingkan dengan protokol *UDP*. [35]

2.1.13 Parameter *QoS (Quality of Service*) *Throughput*

Throughput adalah *bandwidth* actual dan terukur dalam interval waktu tertentu dalam mengirim file. *Throughput* bisa didefinisikan jumlah kedatangan paket yang berhasil pada interval waktu dibagi dengan durasi interval waktu tersebut. Berikut ini persamaan nilai *throughput* [36].

$$\boldsymbol{Throughput} = \frac{\textbf{Jumlah data yang dikirim}}{\textbf{Waktu Pengiriman}} \quad (1)$$

Berikut ini standarisasi *throughput* berdasarkan *TIPHON:* [24]

Tabel 2. 2 Indeks Kategori *Throughput*

Kategori *Throughput*	Nilai *Throughput*	Indeks
Sangat Baik	> 2,1 *Mbps*	4
Baik	1200 Kbps – 2,1 Mbps	3
Cukup	700 Kbps – 1200 *Kbps*	2
Kurang Baik	338 Kbps – 700 *Kbps*	1
Buruk	0 – 338 *Kbps*	0

Sesuai pada Tabel 2.2 *throughput* akan mendapat nilai indeks 4 dan masuk kategori sangat baik apabila nilai *throughput*nya lebih dari 2,1 *Mbps*, dan akan mendapatkan nilai indeks 1 dengan kategori buruk apabila nilai *throughput* kurang dari 338 *Kbps*. Semakin kecil nilai *throughput* maka semakin kecil juga nilai indeksnya.

2.1.14 Parameter *QoS (Quality of Service*) *Delay*

Delay atau latensi merupakan salah satu parameter dari *QoS, delay* bisa didefinisikan arti waktu yang dibutuhkan data dari sumber hingga tujuan. Beberapa faktor yang mempengaruhi *delay* diantaranya media yang digunakan, jarak yang ditempuh, kongesti dan waktu proses. Berikut ini persamaan nilai *delay* [37]:

$$\boldsymbol{Delay\ (ms) rata - rata\ per\ packets} = \frac{\textbf{Delay rata−rata}}{\textbf{Jumlah paket yang dikirimkan}} \quad (2)$$

Berikut ini standarisasi *delay* berdasarkan *TIPHON:* [25]

Tabel 2. 3 Indeks Kategori *Delay*

Kategori *Delay*	Nilai *Delay*	Indeks
Sangat Baik	< 150 *ms*	4

Kategori *Delay*	Nilai *Delay*	Indeks
Baik	150 ms – 300 *ms*	3
Cukup	300 ms – 450 *ms*	2
Buruk	> 450 *ms*	1

Sesuai pada Tabel 2.3 *Delay* akan mendapatkan nilai indeks 4 atau kategori sangat baik apabila nilai *delay*nya kurang dari 150 *ms*, akan mendapatkan nilai buruk dengan indeks 1 apabila nilai *delay*nya lebih dari 450 *ms*. Semakin besar *delay* maka nilai indeks akan semakin kecil.

2.1.15 Parameter *QoS (Quality of Service*) *Packet Loss*

Packet loss merupakan salah satu parameter *QoS* yang menunjukkan persentase paket data yang hilang pada saat mentransmisikan data. Beberapa penyebab terjadinya *packet loss*, diantaranya: kualitas sinyal, kesalahan pada *hardware* dan pengaruh lingkungan sekitar [36]. Berikut persamaan nilai *packet loss:*

$$\boldsymbol{Packet\ Loss}\ (\%) = \frac{(\textbf{Paket data dikirim} - \textbf{paket data diterima})}{\textbf{Paket data dikirim}}\ \textbf{x}\ \mathbf{100}\% \qquad (3)$$

Berikut ini standarisasi *packet loss* berdasarkan *TIPHON:* [24]

Tabel 2. 4 Indeks Kategori *Packet Loss*

Kategori *Packet Loss*	Nilai *Packet Loss*	Indeks
Sangat Baik	0 % - 3 %	4
Baik	3 % - 15%	3
Cukup	15 % - 25%	2
Buruk	> 25%	1

Sesuai pada Tabel 2.4 *Packet loss* akan mendapatkan indeks 4 dengan kategori sangat baik apabila nilai *packet loss* nya kurang dari 3%, dan akan mendapat nilai indeks 1 apabila nilai *packet loss* nya lebih dari 25%. Semakin besar nilai *packet loss* akan kecil nilai indeksnya.

2.1.16 *GNS3*

GNS3 atau *graphical network* simulator merupakan aplikasi yang biasa digunakan untuk mensimulasikan topologi jaringan, aplikasi ini juga bersifat *open*

source. *GNS3* support beberapa sistem operasi, diantaranya: *Linux*, *Windows* bahkan *Mac* OS. Beberapa komponen dari *GNS3*, yaitu:

a. *Dynamips*.
b. *Dynagen*.
c. *Qemu*.
d. *WinPCAP* [38].

2.1.17 *Wireshark*

Wireshark merupakan aplikasi yang digunakan sebagai media untuk menganalisis perangkat jaringan, perangkat lunak. *Wireshark* sering digunakan *network administrator* untuk mengontrol trafik jaringan dan menganalisa kinerja jaringan. *Wireshark* sendiri memiliki kekurangan, yaitu kesulitan saat pendeteksian drive bersifat *wireless*, *wireshark* hanya mampu mengenali *driver* dengan nama *Microsoft* atau hanya sebatas protokol *WLAN* [39].