

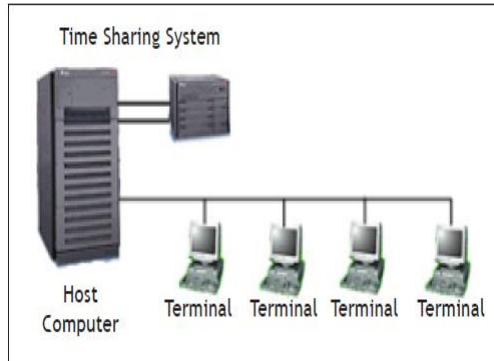
BAB II

DASAR TEORI

2.1 JARINGAN KOMPUTER

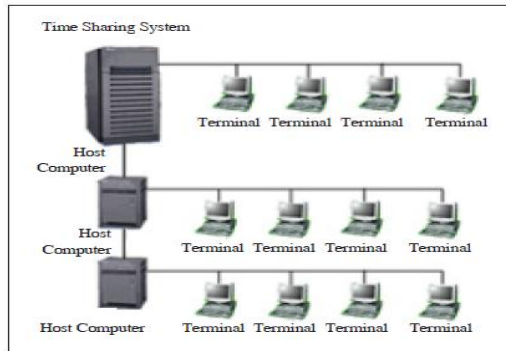
Konsep jaringan komputer lahir pada tahun 1940-an di Amerika pada sebuah proyek pengembangan komputer MODEL I di laboratorium Bell dan group riset Harvard University. Proyek ini dipimpin oleh profesor H. Aiken. Pada mulanya proyek ini hanyalah ingin memanfaatkan sebuah perangkat komputer yang harus digunakan bersama.

Pada tahun 1950-an ketika jenis komputer mulai membesar sampai terciptanya super komputer. Saat itu dikenalkan sebuah konsep untuk menghubungkan komputer yang saling tersebar. Konsep tersebut adalah konsep distribusi proses berdasarkan waktu yang dikenal dengan nama *Time Sharing System* (TSS), maka untuk pertama kali bentuk jaringan (*network*) komputer diaplikasikan. Pada sistem TSS beberapa terminal (komputer) terhubung secara seri ke sebuah *host* komputer. Dalam proses TSS mulai nampak perpaduan teknologi komputer dan teknologi telekomunikasi yang pada awalnya berkembang sendiri-sendiri.



Gambar 2.1 Jaringan Komputer Model TSS.

Memasuki tahun 1970-an, setelah beban pekerjaan bertambah banyak dan harga perangkat komputer besar mulai terasa sangat mahal, maka mulailah digunakan konsep proses distribusi (*Distributed Processing*). Seperti yang ada pada gambar 2.1, dalam proses ini beberapa host komputer mengerjakan sebuah pekerjaan besar secara paralel untuk melayani beberapa terminal yang tersambung secara seri disetiap host komputer. Dalam proses distribusi sudah mutlak diperlukan perpaduan yang mendalam antara teknologi komputer dan telekomunikasi, karena selain proses yang harus didistribusikan, semua *host* komputer wajib melayani terminal-terminalnya dalam satu perintah dari komputer pusat.



Gambar 2.2 Jaringan Komputer Model *Distributed* Prosesing.

Perjalanan sejarah komputer ukuran *hardware* komputer dari tahun ke tahun mengalami perkembangan pesat. Hal ini ditandai dengan kemampuan yang semakin tinggi dan ukuran yang semakin kecil. Saat ini komputer dan jaringannya sudah dapat menangani proses komunikasi antar komputer (*Peer to Peer System*) tanpa melalui komputer pusat. Untuk itu, mulailah berkembang teknologi jaringan lokal yang dikenal dengan sebutan LAN. Demikian pula ketika *Internet* mulai diperkenalkan.⁽¹⁾

2.1.1 Tujuan Membangun Jaringan Komputer

Jaringan komputer dibangun untuk membawa informasi secara tepat tanpa adanya kesalahan dari sisi pengirim

(*transmitter*) maupun sisi penerima (*receiver*) melalui media komunikasi. Kendala-kendala yang muncul adalah pada media komunikasi misalnya masih mahalnya fasilitas komunikasi yang tersedia dan bagaimana pemanfaatan jaringan komunikasi lebih efektif dan efisien, serta masih terdapatnya berbagai macam gangguan saat data ditransmisikan.⁽²⁾

2.1.2 Manfaat Jaringan Komputer

Manfaat jaringan komputer bagi manusia dapat dikelompokkan pada jaringan untuk perusahaan, dan jaringan untuk umum.

a. Jaringan Untuk perusahaan/organisasi

Dalam sebuah perusahaan atau organisasi ada beberapa manfaat yang diperoleh dari sebuah jaringan komputer ini seperti :

- ***Resource sharing*** bertujuan agar seluruh program, peralatan, khususnya data dapat digunakan oleh setiap orang yang ada pada jaringan tanpa terpengaruh oleh lokasi *resource* dan pemakai. jadi *resource sharing* adalah suatu usaha untuk menghilangkan kendala jarak.

Gambar 2.3 *Resource Sharing*

- **Reliabilitas tinggi** yaitu adanya sumber-sumber alternatif pengganti jika terjadi masalah pada salah satu perangkat dalam jaringan, artinya karena perangkat yang digunakan lebih dari satu jika salah satu perangkat mengalami masalah, maka perangkat yang lain dapat menggantikannya.

Gambar 2.4 *Reliabilitas Tinggi*

Komputer yang kecil memiliki rasio harga/kinerja yang lebih baik dibanding dengan komputer besar. Komputer *mainframe* memiliki kecepatan kurang lebih sepuluh kali lipat kecepatan komputer pribadi, akan tetapi harga

mainframe seribu kalinya lebih mahal. Dengan selisih rasio harga/kinerja yang cukup besar ini menyebabkan perancang sistem memilih membangun sistem yang terdiri dari komputer-komputer pribadi dibanding menggunakan *mainframe*.

- **Skalabilitas** yaitu kemampuan untuk meningkatkan kinerja sistem secara berangsur-angsur sesuai dengan beban pekerjaan dengan hanya menambahkan sejumlah prosesor. Pada komputer *mainframe* yang tersentralisasi, jika sistem sudah jenuh, maka komputer harus diganti dengan komputer yang mempunyai kemampuan lebih besar. Hal ini membutuhkan biaya yang sangat besar dan dapat menyebabkan gangguan terhadap kontinuitas kerja para pemakai.



Gambar 2.5 Skalabilitas

- **Media komunikasi** yang baik bagi para pegawai yang terpisah jauh. Dengan menggunakan jaringan, dua orang atau lebih yang tinggal berjauhan akan lebih mudah bekerja sama dalam menyusun laporan.



Gambar 2.6 Media Komunikasi

b. Jaringan Untuk Umum

- **Access ke informasi yang berada di tempat yang jauh**

Ada bermacam-macam bentuk *access* ke informasi jarak jauh yang dapat dilakukan, terutama setelah berkembangnya teknologi *internet*, berita-berita di koran sekarang dapat di-*download* ke komputer kita melalui *internet*, dan tidak hanya itu sekarang kita dapat melakukan pemesanan suatu produk melalui *internet*, bisnis yang

dikenal dengan istilah *electronic commerce (e-commerce)*, ini sekarang sedang berkembang dengan pesat.

- **Komunikasi orang-ke-orang**

Dengan menggunakan *internet* kita juga dapat melakukan komunikasi orang ke orang, fasilitas *electronic mail (e-mail)*, *chatting* telah dipakai secara meluas oleh jutaan orang .

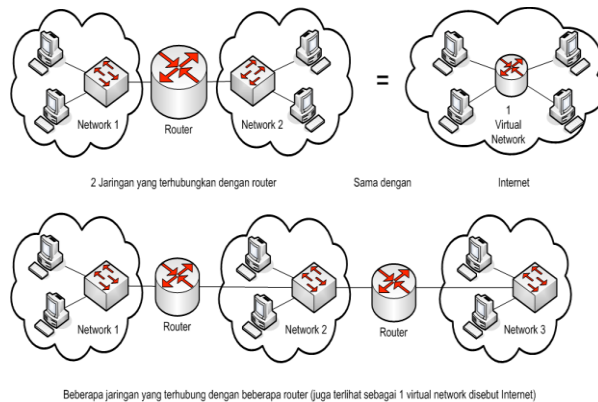
- ***Video conference*** atau pertemuan maya merupakan teknologi yang memungkinkan terjadinya komunikasi jarak jauh tanpa *delay*. Pertemuan maya ini dapat pula digunakan untuk keperluan sekolah jarak jauh, memperoleh hasil pemeriksaan medis seorang dokter yang berada di tempat yang jauh, dan sejumlah aplikasi lainnya. *Video on demand* merupakan jaringan komputer bagi orang per orang dimana kita dapat memilih *film* atau acara televisi dari negara mana saja dan kemudian ditampilkan di layar monitor kita.⁽³⁾

2.2 **INTERNETWORKING**

Transmission Control Protocol/Internet Protocol adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam melakukan proses tukar-menukar data dari

satu komputer ke komputer yang lain di dalam jaringan *internet*.

Tujuan dari TCP/IP adalah untuk membangun suatu koneksi antar jaringan (*network*), dimana biasa disebut *internetwork*, atau *internet*, yang menyediakan pelayanan komunikasi antar jaringan yang memiliki bentuk fisik yang beragam. Tujuan yang jelas adalah menghubungkan empunya (*hosts*) pada jaringan yang berbeda, atau mungkin terpisahkan secara geografis pada area yang luas. Pada gambar 2.7 merupakan contoh *internet*.



Gambar 2.7 Jaringan *Internet*

Internet dapat digolongkan menjadi beberapa group jaringan, antara lain:

- *Backbone* : Jaringan besar yang menghubungkan antar jaringan lainnya. Contoh : NSFNET yang merupakan jaringan backbone dunia di Amerika, EBONE yang merupakan jaringan *backbone* di Eropa, dan lainnya.
 - Jaringan regional, contoh: jaringan antar kampus.
 - Jaringan yang bersifat komersial dimana menyediakan koneksi menuju backbone kepada pelanggannya.
 - Jaringan lokal, contoh: jaringan dalam sebuah kampus.

Aspek lain yang penting dari TCP/IP adalah membentuk suatu standarisasi dalam komunikasi. Tiap-tiap bentuk fisik suatu jaringan memiliki teknologi yang berbeda-beda, sehingga diperlukan pemrograman atau fungsi khusus untuk digunakan dalam komunikasi. TCP/IP memberikan fasilitas khusus yang bekerja diatas pemrograman atau fungsi khusus tersebut dari masing-masing fisik jaringan. Sehingga bentuk arsitektur dari fisik jaringan akan tersamarkan dari pengguna dan pembuat aplikasi jaringan. Dengan TCP/IP, pengguna tidak perlu lagi memikirkan bentuk fisik jaringan untuk melakukan sebuah komunikasi. Sebagai contoh pada

Gambar 2.7, untuk dapat berkomunikasi antar 2 jaringan, diperlukan komputer yang terhubung dalam suatu perangkat yang dapat meneruskan suatu paket data dari jaringan yang satu ke jaringan yang lain. Perangkat tersebut disebut *Router*. Selain itu *router* juga digunakan sebagai pengarah jalur (*routing*). Untuk dapat mengidentifikasi *host* diperlukan sebuah alamat, disebut alamat IP (*IP address*). Apabila sebuah *host* memiliki beberapa perangkat jaringan (*interface*), seperti *router*, maka setiap interface harus memiliki sebuah IP address yang unik. ⁽⁴⁾

2.3 TIPE-TIPE JARINGAN

Jaringan terdiri dari perpaduan dari banyak *interconnected network*, jaringan mereka saling terintegrasi untuk menyediakan sebuah media komunikasi data tunggal, disebut sebagai *internetworks*. *Internet* adalah *prototypical* dari *internetwork*, biasanya terdiri dari ratusan ribu dari *local*, metropolitan dan *wide area networks*.

2.3.1 *Local Area Network* (LAN)

Local Area Network seringkali disebut LAN, merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran beberapa kilo meter. LAN sering digunakan untuk menghubungkan komputer-komputer pribadi dan

wrokstation dalam kantor perusahaan atau pabrik-pabrik untuk pemakaian *resource* bersama (misalnya, printer) dan saling bertukar informasi. LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik yakni: ukuran, teknologi transmisinya, dan topologinya.

LAN mempunyai ukuran terbatas, yang berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Dengan mengetahui keterbatasannya menyebabkan adanya kemungkinan untuk menggunakan jenis tertentu. Hal ini juga memudahkan manajemen jaringan.

Seperti halnya saluran telepon yang dipakai di daerah pedesaan, LAN sering menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai dari 10 sampai 100 Mbps dengan *delay* rendah (puluhan mikrosekon) dan mempunyai faktor kesalahan yang kecil. LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan megabit/detik.

2.3.2 *Wide Area Network* (WAN)

Wide Area Network atau WAN, mencakup daerah geografis yang luas, seringkali mencakup sebuah negara atau benua. WAN terdiri dari kumpulan mesin yang bertujuan untuk menjalankan program-program (aplikasi)

pemakai. Kita akan mengikuti penggunaan tradisional dan menyebut mesin-mesin ini sebagai *host*. Istilah *end system* kadang-kadang juga digunakan dalam literatur. *Host* dihubungkan oleh sebuah *subnet* komunikasi, atau cukup disebut *subnet*. Tugas *subnet* adalah membawa pesan dari satu *host* ke *host* lainnya, seperti halnya sistem telepon yang membawa isi pembicaraan ke pendengar. Dengan memisahkan aspek komunikasi murni sebuah jaringan (*subnet*) dari aspek-aspek aplikasi (*host*), rancangan jaringan lengkap menjadi jauh lebih sederhana.

Pada sebagian besar WAN, *subnet* terdiri dari dua komponen: kabel transmisi dan *element switching*. Kabel transmisi (disebut juga sirkuit, kanal, atau *trunk*) memindahkan bit-bit dari satu mesin ke mesin lainnya.

Element Switching adalah komponen khusus yang dipakai untuk menghubungkan dua kabel transmisi atau lebih. Saat data sampai ke kabel penerima, *element switching* harus memilih kabel pengirim untuk meneruskan pesan-pesan tersebut.

2.3.3 Metropolitan Area Network (MAN)

Metropolitan area Network (MAN) pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN

dapat mencakup kantor-kantor perusahaan yang berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, dan bahkan dapat berhubungan dengan jaringan televisi kabel. Man hanya memiliki sebuah atau dua buah kabel dan tidak mempunyai elemen *switching*, yang berfungsi untuk mengatur paket melalui beberapa kabel *output*. Adanya elemen *switching* membuat rancangan menjadi lebih sederhana.

Alasan untuk memisahkan MAN sebagai kategori khusus adalah telah ditentukannya standar untuk MAN, dan standar ini sekarang sedang diimplementasikan. Standar tersebut disebut *Distributed Queue Dual Bus* (DQDB) atau dalam nomor disebut 802.6 (nomor yang telah ditentukan oleh IEEE).

Aspek penting dari sebuah MAN adalah adanya sebuah medium *broadcast* (untuk 802.6, dua buah kabel) tempat semua kabel dihubungkan. Medium ini mampu menyederhanakan rancangan jaringan dibandingkan dengan jenis-jenis jaringan lainnya.⁽⁵⁾

2.4 TOPOLOGI JARINGAN⁽⁶⁾

Topologi (dari bahasa Yunani *topos*, "tempat", dan *logos*, "ilmu") merupakan cabang matematika yang

bersangkutan dengan tata ruang yang tidak berubah dalam deformasi dwikontinyu (yaitu ruang yang dapat ditebuk, dilipat, disusut, direntangkan, dan dipilin tetapi tidak diperkenankan untuk dipotong, dirobek, ditusuk atau dilekatkan). Topologi muncul melalui pengembangan konsep dari geometri dan teori himpunan, seperti ruang, dimensi, bentuk, transformasi.

Topologi jaringan komputer adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Dalam suatu jaringan komputer jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu dicermati kelebihan/keuntungan dan kekurangan/kerugian dari masing-masing topologi berdasarkan kateristiknya.

Topologi pada dasarnya adalah peta dari sebuah jaringan. Topologi jaringan terbagi lagi menjadi dua yaitu topologi secara fisik (*physical topology*) dan topologi secara logika (*logical topology*). Topologi secara fisik menjelaskan bagaimana susunan dari label, komputer dan lokasi dari semua komponen jaringan. Sedangkan topologi secara logika menetapkan bagaimana informasi atau aliran data dalam jaringan.

Arsitektur topologi merupakan bentuk koneksi fisik untuk menghubungkan setiap node pada sebuah jaringan.

Berikut jenis-jenis topologi topologi: Topologi *Bus*, Topologi *Star* (Bintang), Topologi *Ring* (Cincin), Topologi *Tree* (Pohon), Topologi *Mesh* (Tak Beraturan).

2.4.1 Topologi *Bus*

Topologi *Bus* terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server*, *workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*). Jaringan-jaringan *ethernet* dan *Local Talk* biasanya menggunakan topologi ini.



Gambar 2.8 Topologi *Bus*

Kelebihan dari topologi *Bus* adalah :

- Mudah dalam mengkonfigurasi komputer atau perangkat lain ke dalam sebuah kabel utama.

- Tidak terlalu banyak menggunakan kabel dibandingkan dengan topologi *Star*/bintang.

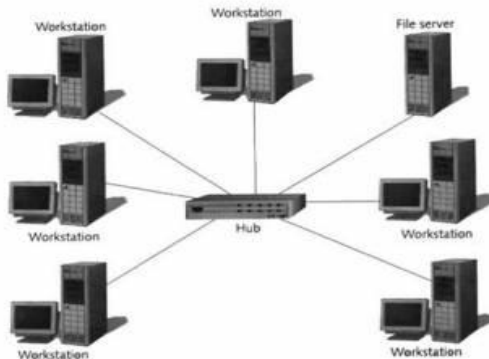
Kekurangan dari topologi *Bus* ini adalah :

- Seluruh jaringan akan mati jika ada kerusakan pada kabel utama (*backbone*).
- Membutuhkan terminator pada kedua sisi dari kabel utamanya.
- Sangat sulit mengidentifikasi permasalahan jika jaringan sedang jatuh atau rusak.
- Sangat tidak disarankan dipakai salah satu solusi pada penggunaan jaringan di gedung besar.

2.4.2 Topologi *Star*⁽⁷⁾

Topologi model ini dirnacang, yang mana setiap *nodes* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*.

Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai *repeater* (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel *Twisted Pair*, dan dapat digunakan pula kabel *coaxial* atau kabel *fibre optic*.

Gambar 2.9 Topologi *Star*⁽⁸⁾

Kelebihan dari topologi *Star* adalah :

- Mudah didalam pemasangan dan pengkabelan.
- Tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya.
- Mudah untuk mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

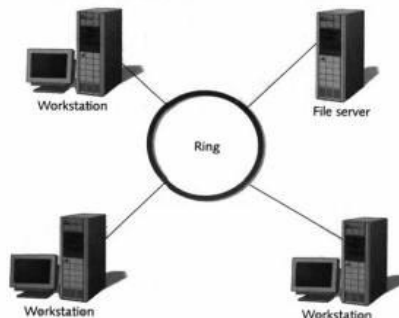
Kekurangan dari topologi *Star*

- Membutuhkan lebih banyak kabel daripada topologi *Bus*
- Membutuhkan *concentrator*, dan bilamana *concentrator* tersebut rusak maka semua *node* yang terkoneksi tidak dapat terdeteksi.

- Lebih mahal daripada topologi *Bus*, karena biaya untuk pengadaan *concentrator*.

2.4.3 Topologi *Ring*

Topologi *Ring* menggunakan teknik konfigurasi yang sama dengan topologi *Star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai satu lingkaran tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.

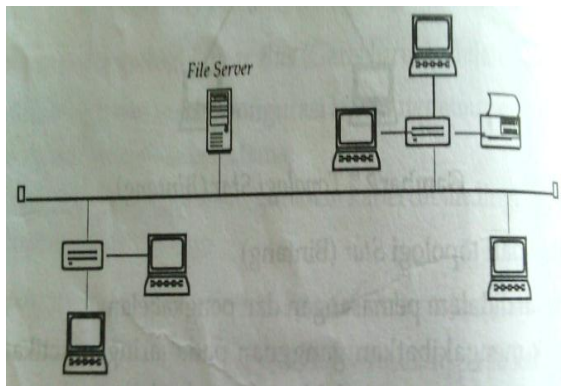


Gambar 2.10 Topologi *Ring*⁽⁸⁾

2.4.4 Topologi *Tree* (Pohon)

Topologi model ini merupakan perpaduan antara topologi *Bus* topologi *Star* yang mana terdiri dari kelompok-kelompok dan *workstation* dengan konfigurasi *Star* yang terkoneksi dengan kabel utama yang

menggunakan topologi *Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2.11 Topologi *Tree*

Kelebihan dari topologi *Tree* adalah :

- Proses konfigurasi jaringan dilakukan dari titik ke titik pada masing-masing segmen.
- Didukung oleh banyak perangkat keras dan perangkat lunak.

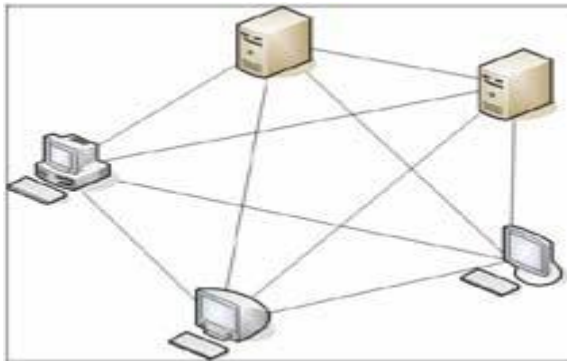
Kekurangan dari topologi *Tree* adalah :

- Keseluruhan panjang kabel pada tiap-tiap segmen dibatasi oleh tipe kabel yang digunakan.

- Jika jaringan utama (*backbone*) rusak, maka keseluruhan segmen ikut rusak juga.
- Sangat relatif sulit untuk di konfigurasi dan proses pengkabelannya dibandingkan topologi jaringan yang lain.

2.4.5 Topologi *Mesh*

Topologi *Mesh* dibangun dengan memasang banyak link pada setiap komputer. Hal ini dimungkinkan karena pada setiap komputer terdapat lebih dari satu NIC.



Gambar 2.12 Topologi *Mesh*.⁽⁸⁾

Topologi ini secara teori memungkinkan akan tetapi tidak praktis dan biayanya cukup tinggi. Topologi Mesh memiliki tingkat redundancy yang tinggi.

Topologi *Mesh* memiliki beberapa kelebihan seperti :

- Hubungan *dedicated links* akan menjamin data yang langsung dikirimkan ke titik (komputer) tujuan tanpa harus melalui komputer yang lain. Dengan kondisi seperti ini, maka komunikasi akan lebih cepat dari satu komputer ke komputer tujuan.
- Memiliki sifat *Robust*, yakni apabila terjadi kerusakan *link* pada salah satu komputer maka kerusakan tersebut tidak akan mengganggu konektivitas komputer yang lain.
- Dalam hal *security* dan kerahasiaan pada topologi ini lebih terjamin, karena konektivitasnya anantara satu komputer dengan komputer tujuan tidak dapat di masukin komputer yang lain.

2.5 ROUTING

Routing adalah suatu protokol yang digunakan untuk mendapatkan *route* dari satu jaringan ke jaringan yang lain. *Route* ini, disebut dengan route dan informasi route secara dinamis dapat diberikan ke *router* yang lain ataupun dapat diberikan secara statis ke *router* lain.

Algoritma *routing* merupakan bagaian perangkat lunak dan lapisan *network* yang bertanggung jawab terhadap

saluran keluaran bagi paket masuk dan harus ditransmisikan. Proses *routing* paket data diperlukan syarat berikut:

- Alamat tujuan yang jelas dan memilih jalur-jalur terbaik dan tercepat.
- Mengidentifikasi informasi dan sumber yaitu *router* mempelajari dan mana informasi berasal, jalur-jalur yang dipilih selanjutnya.
- Menentukan jalur-jalur yang mungkin dilewati yaitu lalu lintas yang dapat ditempuh untuk sampai tujuan.
- Mengatur dan mengkonfirmasi informasi *routing* yaitu jalur yang akan digunakan bisa terpercaya atau tidak.

Routing ideal berkriteria ketepatan, kesederhanaan, ketangguhan, stabilitas, keadilan, dan optimalisasi. Tujuan utama *routing* adalah *router*-muter tidak mempelajari jalur-jalur terhubung langsung dengannya, tetapi mengatur bagaimana meneruskan paket data ke jalur yang terhubung tidak langsung.

Routing harus mampu mengatasi perubahan topologi jaringan serta lalu lintas jalur tanpa pembatalan proses pada

host, selain itu jaringan tidak memerlukan *reboot* jika *router* mengalami tabrakan.⁽⁷⁾

2.5.1 *Routing Statis*

Routing Statis adalah jenis *routing* yang dikonfigurasi *admin/user* untuk melakukan konfigurasi informasi tentang jaringan yang dituju secara manual. Ciri-ciri dari *routing statis* ini adalah: jalur spesifik jaringan yang akan digunakan ditentukan oleh *admin/user*, selain penentuan jalur yang manual penentuan tabel *routing* juga dilakukan secara manual, *routing statis* biasanya digunakan dalam jaringan yang kecil. Cara kerja *routing* statis dapat dibagi menjadi tiga bagian, yaitu :

- Administrator jaringan yang mengkonfirmasi *router*
- *Router* melakukan *routing* berdasarkan informasi dalam tabel *routing*.
- *Routing* statis digunakan untuk melewatkan paket data.

Routing statis memiliki beberapa keuntungan seperti :

- Pemeliharaan *bandwidth network* karena peng-update-an informasi *router* membutuhkan broadcasts yang terus menerus.

- Keamanan network karena *statis routing* hanya mengandung informasi yang telah dimasukkan secara manual.

Selain itu, *routing statis* juga memiliki kerugian seperti :

- Tidak ada toleransi kesalahan. Jika suatu *router* down, maka *statis* tidak akan memperbaharui informasi dan tidak akan menginformasikan ke *router* yang lain.
- Pengembangan *network*. Jika suatu *network* ditambah atau dipindahkan maka *statis routing* harus diperbaharui oleh administrator

2.5.2 ROUTING DINAMIS

Routing dinamis merupakan jenis *routing* yang dapat bekerja secara otomatis sesuai dengan kondisi yang diinginkan dengan mengacu pada parameter tertentu sesuai dengan protokolnya. Biasanya *routing* dinamis ini diterapkan pada sebuah Personal Computer (PC) yang dimana PC tersebut berfungsi sebagai *router* dan dibutuhkan *router* yang lain yang sama-sama menggunakan *routing* ini. Pada *dynamic routing* memiliki beberapa jenis *routing* protokol.⁽⁹⁾

2.5.2.1 Algoritma Vektor Jarak (*Distance Vector Algorithm*)

Algoritma vektor jaringan mengirim informasi secara periodik dan tabel *routing* salah satu *router* ke tabel lain. Algoritma ini tidak memperbolehkan *router* untuk mengetahui topologi seluruh interkoneksi jaringan.

Perubahan topologi jaringan yang terjadi adalah perubahan yang harus dikomunikasikan antar *router*. Informasi pembaharuan akan disesuaikan serta dibandingkan dengan tabel miliknya, juga memberi hak pada setiap *router* untuk menjaga tabel sendiri untuk menghasilkan jalur terbaik yang bisa dijangkau ke tujuan. Perubahan pengukuran ditulis pada tabel *routing* dan diperbaharui oleh *router* itu sendiri.

Setiap *router* dapat mengatur sebuah tabel *routing* yang dirangking oleh *router* itu sendiri berdasarkan informasi mengenai perhitungan jarak atau penggunaan antarmuka dan disimpan pada setiap *router* dan *subnet*.

Ciri protokol vektor jarak adalah menemukan jalur terbaik ke tujuan yang tersambung tidak langsung berdasar *matric* akumulasi *router* tetangga, oleh karena itu pengetahuan diperoleh dari informasi tetangganya. *Router* menambah biaya pengantaran ke *router* tetangga

agar jarak ke tujuan dilaporkan oleh *router* sebelahnya sehingga menghasilkan ukuran *matric* baru.

2.5.2.2 Algoritma Keadaan *Link* (*Link-State Algorithm*)

Algoritma keadaan *link* prosesnya memakai sistem perencanaan secara hirarki. *Router* yang bekerja pada suatu wilayah harus mengetahui detail aturan menentukan jalur paket ke tujuan pada wilayahnya, akan tetapi sebuah *router* tidak diperbolehkan mengetahui struktur internal *router* wilayah lain.

Protokol *routing* yang menggunakan konsep link state akan bekerja membuat tabel *routing* menurut perhitungannya sendiri dan tidak tergantung perhitungan *router* yang lain. Beberapa hal yang harus Anda pertimbangkan dalam menggunakan konsep ini adalah : Kemampuan processor yang lebih tinggi, karena perhitungannya menggunakan algoritma SPF (*Short Path First*), memori yang besar untuk menampung paket link state, bandwidth yang sedikit lebih besar untuk melakukan proses penerimaan, penyalinan, dan pengiriman paket link state.

Protokol-protokol yang bekerja menggunakan konsep link state antara lain OSPF (*Open Short Path*

First) yang biasa digunakan dalam sebuah jaringan besar.⁽¹⁰⁾

2.6 **ROUTING PROTOCOL**⁽¹¹⁾

Tiap *router* melakukan pemilihan jalan/cara terbaik melalui proses pertukaran informasi antara *router* tersebut dengan *router* yang lain. Prosedur untuk memilih *route* yang terbaik ini disebut *routing protocol*. *Routing table* dibangun oleh *router* untuk mempermudah data dikirim ke tujuannya. *Routing table* digunakan untuk meneruskan paket data pada jaringan tiap segmen.

2.6.1 **Routing Information Protocol (RIP)**

Routing Information Protocol (RIP) merupakan protokol *distance-vector* yang digunakan untuk dalam *domain*. RIP ditujukan untuk jaringan yang kecil dan batas memiliki hopnya dibatasi 16 hop.

Routing ini berdasarkan jumlah *hop* dan tidak berdsarkan *bandwidth* yang ada pada *link* tertentu. Karena menggunakan algoritma *distance vector*, maka metodenya seperti yang terjadi pada algoritma *distance vector*. Merupakan *protocol routing* yang digunakan secara luas di *internet* dan dapat memanfaatkan *broadcast address* untuk distribusi informasi *routing* serta dapat menentukan *route*

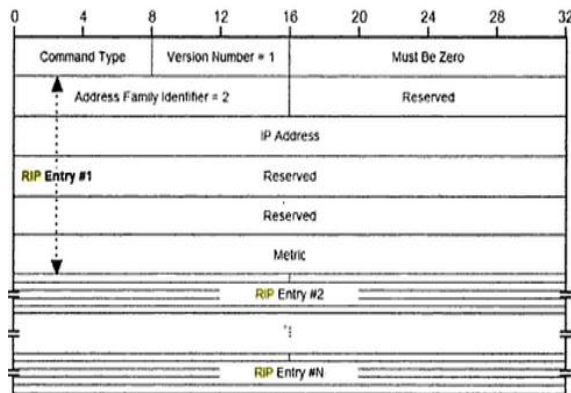
terbaik dengan “*hop count*” terkecil. Biasanya *update routing* dilakukan terus menerus.

Cara kerja dari *Routing Information Protocol* yaitu *host* akan mendengar pada alamat *broadcast* jika ada *update routing* dari *gateway*. Selanjutnya *host* akan memeriksa terlebih dahulu *routing table* lokal jika menerima *update routing*. Jika *rute* belum ada, informasi segera dimasukkan ke *routing table*. Tetapi jika *rute* sudah ada, matrik yang terkecil akan diambil sebagai acuan. Selanjutnya *rute* melalui suatu *gateway* tersebut dalam waktu tertentu.

Routing RIP merupakan jenis *routing* yang menggunakan protokol *distance vector*. Pada dasarnya protokol ini memiliki kelemahan yang akan terlihat pada saat adanya *link* yang putus dalam suatu jaringan. Dua kemungkinan kegagalan yang mungkin terjadi adalah efek *bouncing* dan kasus “menghitung sampai tak hingga” (*counting to infinity*). Efek *bouncing* dapat terjadi pada jaringan yang menggunakan metrik yang berbeda pada minimal sebuah *link*. *Link* yang putus menyebabkan *routing loop*, sehingga datagram yang melewati *link* tentu hanya berputar-putar diantara beberapa router sampai umur (*time to live*) datagram tersebut habis.

Paket-paket *routing* RIP secara normal dikirim setiap 30 detik atau lebih cepat jika terdapat *triggered updates*. Jika

dalam 180 detik sebuah *route* tidak diperbaharui, *router* menghapus *entri route* tersebut dari *forwarding table*. Format paket RIP seperti gambar 2.13.



Gambar 2.13 Format Paket RIP

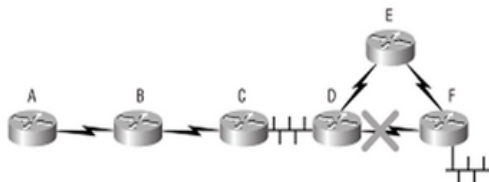
Setiap *host* yang menjalankan RIP memiliki tabel *routing* yang setidaknya berisi :

- IP address tujuan
- IP address *router* yang dilalui
- Metrik yang menunjukkan biaya total untuk mencapai tujuan tersebut dari *host*.
- Sebuah tanda unik menunjukkan apakah rute baru saja berubah.
- Beberapa *timer* (penunjuk waktu).

RIP mengasumsikan bahwa sebuah jaringan dibagi menjadi *subnet* dengan ukuran yang sama, misal sebuah jaringan kelas C dibagi menjadi 8 *subnet* dengan *netmask* 255.255.255.224, maka format paket RIP hanya memuat informasi IP address tanpa memulai informasi *netmask* yang digunakan pada *interface router* atau *host*.

2.6.1.1 Konvergensi Pada RIP⁽¹⁸⁾

Pada gambar 2.14 akan membantu untuk menjelaskan proses dari convergence pada RIP. Daftar berikut menjelaskan RIP convergence ketika problem terjadi. Pada gambar 2.14 WAN diantara roter D dan F down. Link ini berada pada path dari router A sampai D, ketika mengirimkan packet ke segment Ethernet pada router F off. Sekarang, 4 router ini termasuk router D harus mempelajari path atau lintasan melalui router F, tetapi setiap router dari keempat router tersebut akan menegetahui adanya hop tambahan pada network. Berikut apa yang terjadi :



Gambar 2.14 Konvergensi *routing* RIP

- Router D menyebarkan hal ini secara langsung ke rute-rute yang langsung terhubung sesuai dengan routing tabelnya, menghapusnya, dan mengirim trigger update ke router E dan C. setiap rute dengan interface Router D yang linknya down atau alamat router F pada link tersebut sebagai next hop juga akan disebarkan. Hal ini juga termasuk pada segment off pada Router F.
- Router C menggunakan path ini untuk mengakses Ethernet router F, tetapi router E secara langsung terhubung melalui router F. jadi, effect yang terjadi pada router E adalah penyebaran dan penghapusan link WAND dari routing tabelnya. Akan tetapi Router C, meracuni kedua rute, menghapus keduanya dari routing tabelnya. Router C mengirim trigger update ke router B, yang mengirim sinyal trigger update ke router A, setiap router menyebarkan/ meracunin dan menghapus rute secara local. Router F juga mengirim trigger update ke router E.
- Trigger update yang diterima Router E dari router D dan router F secepatnya mengirimkan trigger ke setiap arah. Update ini mengatakan kepada kedua neighborsnya bahwa semua rute yang biasa digunakan masih accessible melalui Router E. Router D

memasukkan rute baru untuk Ethernet Router F melalui Router E kedalam routing tabelnya.

- Router D menerima rute baru untuk network tujuannya, sehingga tidak ada holdtime karena baru saja dibersihkan; sebuah rute harus berada pada routing table untuk berada pada status holddown. Karena rute penyebarang sedang digunakan, situasi yang sama juga terjadi pada router yang lain. Dengan cepat router tersebut akan menyetujui metric baru.
- Router D mengadvertise metric baru ke router C, dan walaupun jika router C tidak disebarkan dan dihapus rutenya dari routing tabelnya, tetapi secara langsung menempatkan rute ini pada holddown. Itu akan menerima metric baru, karena berasal dari sumber advertise yang asli yaitu router D.
- Dampak yang sama berlanjut pada Router C, B dan A dan mengtrigger update karena penyebaran, dan berikutnya mentrigger update. Jadi, tidak ada router yang memiliki route yang gagal pada routing tabelnya, maupun dapat melewati rute yang lain yang masih accessible melalui link yang gagal, termasuk Router F Ethernet network, yang mengizinkan routing tabelnya dimabahkan sebagai update secepatnya dengan metric

Tanpa triggered update dan route poisoning, waktu yang diperlukan untuk router konvergensi akan waktu deteksi, ditambah kali holdtime, dan lain waktu update. konvergensi lengkap ke router A bisa lebih dari 240 detik. dengan mekanisme ini bekerja untuk Anda, namun, konvergensi akan terjadi di internetwork ukuran ini di sekitar 15 detik. itu akan menguntungkan Anda, meskipun, untuk memahami logika di balik skenario results itu dalam waktu kira-kira 240 detik

2.6.2 Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) adalah protokol *routing* berbasis *link-state*, dimana *router* akan melakukan pemetaan topologi dari jaringan (*autonomous system*). *Autonomous system* dapat berupa struktur yang punya hirarki menjadi jaringan yang lebih kecil.

Didalam OSPF terdapat metode penggabungan *database link* melalui penggunaan perbedaan *subnet mask*, penggabungan beberapa *route-rute* menjadi masukan *route* didalam *database*.

Didalam konfigurasi OSPF itu sendiri terdapat semacam area seperti *autonomous system* sebagai level tingkatan yang tidak digunakan pada protokol. *Router* yang semua *interface*-nya terhubung ke dalam satu area dinamakan

router internal. *Router* yang hanya terhubung dengan *backbone* dinamakan *router backbone*. *Router* yang terhubung dengan area yang berbeda disebut *router* batas area.

2.6.3 Interior Gateway Routing Protocol (IGRP)

Interior Gateway Routing Protocol (IGRP) adalah vektor jarak-protokol yang dirancang untuk digunakan pada *interior gateway router* dalam *autonomous system*. Vektor jarak protokol akan *memforward* secara teratur semua atau sebagian dari tabel *routing router* itu sendiri maupun dari *router* tetangga. Proses ini akan berulang secara berkala, dan secara rekursif seluruh AS sampai semua *node* yang terdapat dalam sistem ini menyepakati topologi dan jarak tujuan yang efektif.

IGRP memiliki kemampuan yang diperluas dari *routing* RIP dan protokol vektor jarak lainnya. Salah satu fitur revolusionernya adalah cara penghitungan vektor jaraknya. Tidak seperti sebelumnya penghitungan vektor jarak menggunakan satu matrik saja, sedangkan IGRP memiliki serangkaian matrik untuk menghitung vektor jaraknya.

Hal ini memungkinkan *administator* jaringan dapat menyesuaikan perhitungan *route* algoritma menurut kebutuhan spesifiknya. Selain itu, IGRP mendukung untuk

menggunakan *multipath routing*. IGRP dapat mengingat hingga empat *route* ke tujuan tertentu.

2.6.4 *Enhanced Interior Gateway Routing (EIGRP)*⁽¹²⁾

EIGRP adalah sebuah *routing* protokol milik Cisco yang bekerja pada *router* Cisco dan pada prosesor-prosesor route internal yang terdapat pada *switch layer core* dan *switch layer* distribusi Cisco. EIGRP merupakan protokol distance vector yang *classes* dan yang sudah ditingkatkan (*enhanced*).

EIGRP memiliki karakteristik sebagai berikut :

- Menggunakan *cost load balancing* yang tidak sama.
- Menggunakan algoritma kombinasi antara *distance vector* dengan *link state*.
- Menggunakan *Diffusing Update Algorithm (DUAL)* untuk menghitung jalur terpendek.
- Mendukung IP, IPX, dan *AppleTalk* melalui modul-modul yang bersifat *protocol dependent*.
- Komunikasi melalui *Reliable Transport Protocol (RTP)*

Untuk memelihara informasi *route* yang diperoleh, *router* EIGR menyimpannya ke dalam tiga jenis tabel *routing*, antara lain:

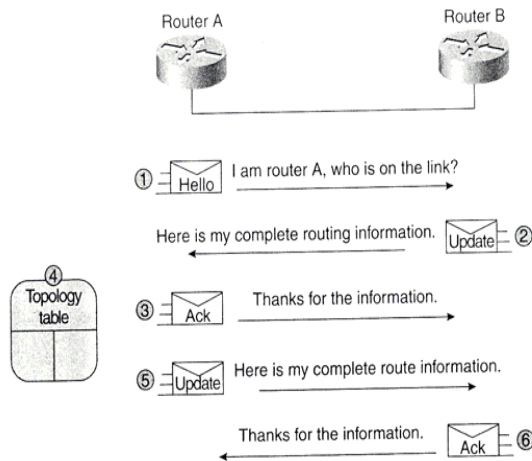
- Table *neighborship*, tabel ini merekam informasi tentang *router-router* yang telah membentuk hubungan bertetangga (*neighborship*).
- Tabel topologi, tabel ini menyimpan pengumuman (*advertisements*) tentang semua *route* di *internetwork* yang diterima dari setiap tetangga.
- Tabel *routing*, EIGRP memilih *route* terbaik ke tujuan dari tabel topologi dan menempatkan *route* dalam tabel *routing*. *Router* mempertahankan satu tabel *routing* untuk setiap protokol jaringan.

EIGRP menggunakan konsep *autonomous system* untuk menggambarkan sekelompok *router* yang beroperasi dengan menggunakan protokol yang sama dan berbagi informasi *routing* yang sama. EIGRP juga memiliki tipe paket⁽²⁰⁾. Adapun tipe paket pada EIGRP adalah :

- *Hello*
Merupakan paket data yang dikirimkan secara *multicast* untuk mencari *router* tetangga.

- *Update* (Pembaharuan)
Update akan dikirim secara *multicast* ketika sebuah rute baru saja ditemukan dan ketika konvergensi selesai.
- *Queries* (Permintaan)
Paket *query* akan dikirim oleh *router* ke *router* tetangganya untuk menanyakan apakah tetangganya memiliki *feasible successor* ke tujuan. hal ini dilakukan ketika *router* sedang melakukan perhitungan rute dan tidak dapat menemukan *feasible successor*. *Query* akan dikirimkan secara *multicast*.
- *Replay* (Balasan)
Replay merupakan paket balasan yang dikirim sebagai respon dari paket *query*.
- ACK
Merupakan paket *hello* yang yang dikirimkan secara *unicast* digunakan untuk mengetahui *update*, *query* dan *replay*.

EIGRP menggabungkan proses menemukan tetangga dan mempelajari rute dalam satu langkah proses. Proses awal pencarian rute EIGRP akan ditunjukkan pada gambar 2.13



Gambar 2.15 Pencarian Rute EIGRP

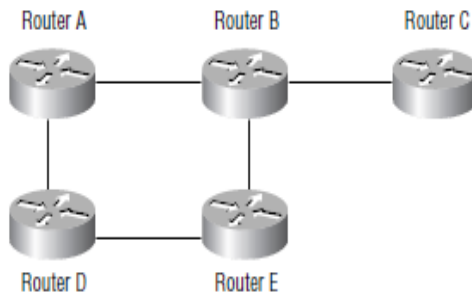
- Pada gambar 2.15 *router* A bertindak sebagai *router* baru yang ada pada *link* tersebut dan mengirimkan paket *hello* melalui semua *interface*-nya. Pada kondisi tersebut *router* B bertindak sebagai penerima paket *hello* yang disebarkan oleh *router* A.
- Router B menerima paket *hello* dalam satu *interface* lalu membalas dengan mengirimkan paket *update* yang berisikan semua rute yang ada dalam tabel *routing*-nya. Berbeda dengan OSPF, *router* B tidak akan mengirimkan paket *hello* kembali ke *router* A. sebaliknya, paket *update* menetapkan hubungan

tetangga diantara perangkat komunikasi. Paket *update* berisikan informasi tentang rute yang diketahui oleh tetangga termasuk metrik yang akan digunakan tetangga untuk setiap tujuan.

- *router* A membalas dengan mengirimkan paket ACK kepada masing-masing tetangga yang terhubung dengannya yang berarti bahwa *router* A telah menerima informasi *update*.
- Setelah proses tersebut terjadi, maka *router* A memasukkan informasi paket *update* yang telah diterimanya kedalam tabel topologi. Tabel topologi berisi tentang semua tujuan yang diumumkan oleh *router* tetangga. Hal ini dilakukan agar setiap tujuan terdaftar, sehingga semua tetangga bisa sampai ketujuan terkait.
- *Router* A melakukan tukar-menukar paket *update* dengan tetangga-tetangganya.
- Hal yang dilakukan setiap *router* setelah menerima paket *update* adalah mengirimkan sebuah paket ACK. Pada gambar 2.15 *router* B atau setiap *router* yang menerima paket *update* akan kembali mengirimkan paket ACK. Ketika semua *update* telah diterima,

router diap untuk memilih rute utama dan cadangan untuk disimpan kedalam tabel topologi⁽¹²⁾.

2.6.4.1 Konvergensi Pada EIGRP⁽¹⁹⁾



Gambar 2.16 Konvergensi *Routing* EIGRP

Dalam gambar 2.16, kita akan menganggap bahwa metrik hanya didasarkan pada jumlah hop. Dengan demikian, Router B adalah satu hop dari Router C, dan Router D adalah tiga hop dari Router C. Tujuan dalam contoh ini adalah Router C, dan router kita prihatin dengan A, yang dua hop jauhnya. Router A, dengan asumsi semua link aktif, akan tempat ke routing tabel rute melalui B ke C- ini adalah jelas jalan terpendek melalui jaringan. Namun, itu tidak akan menempatkan rute pengganti yang layak dalam tabel yang menggunakan rute A-D-E-B-C. Dalam hal terjadi

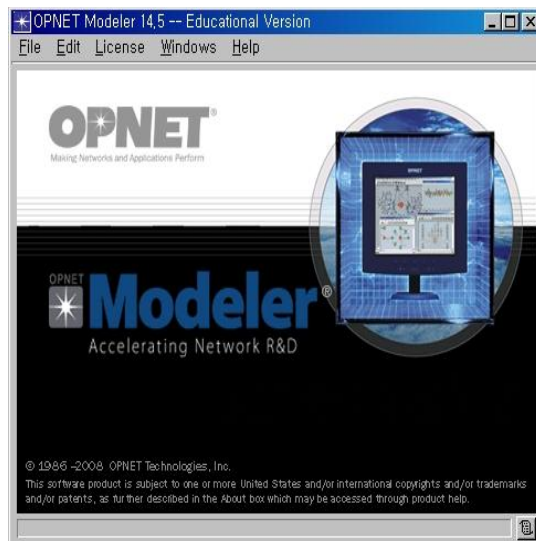
kegagalan link antara A dan B, router harus menghitung ulang path ke C.

Alasan adalah bahwa dalam rangka untuk rute untuk menjadi layak, itu harus memiliki biaya yang lebih rendah ke tujuan daripada metrik routing saat ini pada router itu sendiri. Sebagai contoh, D akan mempertimbangkan D-A-B-C untuk menjadi layak apabila terjadi kegagalan link-di biaya untuk C adalah satu hop kurang D's Perilaku layak penerus berkaitan dengan tujuan utama protokol-loop tidak mungkin ada dalam topologi setiap saat. Dengan selalu memilih sebuah router dengan metrik yang lebih rendah, protokol menghindari skenario seperti ini, meskipun ini mungkin menghambat konvergensi. Kebanyakan EIGRP konvergensi skenario lengkap dalam satu detik; Namun, dalam kasus terburuk bekerja dengan benar

Proses EIGRP akan memakan waktu 16 detik. Perkiraan konvergensi ini didasarkan pada deteksi link kegagalan dan waktu yang diperlukan untuk merespon dengan perhitungan rute baru. Selain itu, EIGRP menyediakan untuk beberapa pengganti yang layak, yang didefinisikan sebagai satu set, dan hingga empat varian jalan mungkin akan loadbalanced jika dikonfigurasi.

2.7 OPNET MODELER v 14.5

OPNET adalah perangkat lunak yang dapat dimasukkan dalam komputer yang berfungsi sebagai pemodelan, mendesain atau simulasi jaringan. Pada OPNET memiliki banyak modul-modul yang disesuaikan dengan kebutuhan pengguna. Modul-modul yang banyak inilah yang menjadi mempermudah para pengguna untuk melakukan pemodelan, mendesain sebuah jaringan. Tampilan dari aplikasi OPNET MODELER ini seperti yang ada pada gambar 2.17.



Gambar 2.17 Tampilan Simulator OPNET Modeler v 14.5⁽⁸⁾

2.7.1 Parameter-Parameter Pada OPNET

OPNET Modeler v 14.5 akan membantu untuk proses simulasi jaringan dengan menggunakan kedua *routing* seperti yang sudah ditentukan pada judul skripsi ini. Pada OPNET ada beberapa parameter yang akan diamati seperti waktu konvergensi, *jiiter*, *packet delay*, dan *packet loss*

- Waktu konvergensi merupakan total waktu yang dibutuhkan oleh sebuah *router* selesai melakukan konvergensi, diantaranya menghitung jalur terbaik dan memperbarui *routing table*. Nilai konvergensi dapat diketahui ketika terdapat perubahan pada jaringan.
- *Packet loss* adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Paket hilang terjadi ketika satu atau lebih paket data yang melewati suatu jaringan gagal mencapai tujuannya. Besaran nilai *packet loss* tersebut merupakan besaran standarisasi yang ditetapkan oleh badan internasional bidang telekomunikasi yakni ITU-T Y1541

Tabel 2.1 Standarisasi Penilaian *Packet Loss*⁽¹⁵⁾

Nilai Packet Loss (%)	Keterangan
0 – 0,5	Merupakan rentang <i>packet loss</i> yang dapat diterima oleh pengguna secara umum (Baik)
0,5 – 1,5	Merupakan rentang <i>packet loss</i> pada jaringan yang bersifat global tetapi masih dapat ditolerir oleh pengguna (sedang)
> 1,5	Merupakan rentang <i>packet loss</i> yang tidak dapat ditolerir pada jaringan dimana pengguna mengalami gangguan berkomunikasi (jelek)

- *Jitter* adalah variasi *delay*, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan. *Jitter* dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar

Tabel 2.2 Standarisasi Penilaian *Jitter*⁽¹⁷⁾

Kategori Latensi	Besar <i>Jitter</i>	Indeks
Sangat Bagus	0 ms	4
Bagus	0 s/d 75 ms	3
Sedang	75 s/d 125 ms	2
Jelek	125 s/d 225 ms	1

Besaran nilai *jitter* tersebut merupakan besaran standarisasi yang di tetapkan oleh *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON).

- *Delay* adalah waktu yang dibutuhkan oleh sebuah paket data terhitung dari saat pengiriman oleh *transmitter* sampai saat diterima oleh *receiver*. Rumus perhitungan *delay* adalah sebagai berikut.⁽¹⁴⁾

Delay memiliki Standarisasi seperti pada tabel 2.3 berikut ini.

Tabel 2.3 Standarisasi Penilaian *Delay*⁽¹⁶⁾

Nilai Packet Loss (%)	Keterangan
0 – 0,5	Merupakan rentang <i>packet loss</i> yang dapat diterima oleh pengguna secara umum (Baik)
0,5 – 1,5	Merupakan rentang <i>packet loss</i> pada jaringan yang bersifat global tetapi masih dapat ditolerir oleh pengguna (sedang)
> 1,5	Merupakan rentang <i>packet loss</i> yang tidak dapat ditolerir pada jaringan dimana pengguna mengalami gangguan berkomunikasi (jelek)