

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Konsep Dasar Jaringan

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer, *software* dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan yang sama. Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan (*service*). Pihak yang meminta/menerima layanan disebut klien (*client*) dan yang memberikan/mengirimkan layanan disebut pelayan (*server*). Arsitektur ini disebut sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

##### 2.1.1 Jaringan *Wireless-LAN*

WLAN adalah jaringan komputer yang menggunakan frekuensi radio dan inframerah sebagai media transmisi data. *Wireless LAN* sering disebut sebagai jaringan nirkabel atau jaringan *wireless* (Jim Gieir, 2005).

*Wireless LAN* merupakan sebuah alternatif dimana untuk alternatif kabel LAN yang sulit atau tidak mungkin dibangun. Tempat-tempat seperti bangunan tua yang dilindungi atau ruang-ruang kelas. Instalasi WLAN juga murah karena hanya terdiri dari *Access Point* dan instalasi jaringan *Backbone*, bagian terakhir dari jaringan yang berada di udara.

Pada akhir 1970-an IBM mengeluarkan hasil percobaan mereka dalam merancang WLAN dengan teknologi IP (*infrared*), perusahaan lain seperti *Hwelett-Packard* (HP) menguji WLAN dengan RF (*Radio Frequency*). Kedua perusahaan tersebut hanya mencapai data rate 100 Kbps. Karena tidak memenuhi standar IEEE 802 untuk LAN yaitu 1 Mbps maka produknya pun tidak dipasarkan. Baru pada tahun 1985, *Federal Communication Commision* (FCC) menetapkan pita *Industrial, Scientific and Medical* (ISM band) yaitu 902-928 MHz, 2400-2483.5 MHz dan 5725-5850 MHz yang bersifat tidak terlisensi sehingga pengembangan WLAN secara komersial memasuki tahapan serius. Baru pada tahun 1990 WLAN dapat dipasarkan dengan produk yang menggunakan teknik spread spectrum pada pita ISM, frekuensi terlisensi 18-19 GHz dan teknologi IR

dengan data rate  $> 1$  Mbps. Pada tahun 1997, sebuah lembaga independen bernama IEEE membuat spesifikasi/standar WLAN pertama yang diberi kode 802.11. Peralatan yang sesuai dengan standar 802.11 dapat bekerja pada frekuensi 2.4 GHz dan kecepatan transfer data (*throughput*) teoritis maksimal 2 Mbps. Pada bulan Juli 1999, IEEE kembali mengeluarkan spesifikasi baru bernama 802.11b. Kecepatan transfer data teoritis maksimal yang dapat dicapai adalah 11 Mbps. Kecepatan data sebesar ini dengan Ethernet tradisional (IEEE 802.3 10 Mbps atau 10BaseT). Peralatan yang menggunakan standar 802.11b juga bekerja pada frekuensi 2.4 GHz.

Pada tahun 2002, IEEE membuat spesifikasi baru yang dapat menggabungkan kelebihan 802.11 dan 802.11a. Spesifikasi yang diberi kode 802.11g ini bekerja pada frekuensi 2.4 GHz dengan kecepatan transfer data teoritis maksimal 54 Mbps. Peralatan 802.11g compatible dengan 802.11b sehingga dapat saling dipertukarkan. Pada tahun 2006, 802.11n dikembangkan dengan menggabungkan teknologi 802.11b, 802.11g. Teknologi yang diusung dikenal dengan istilah MIMO (*Multiple Input Multiple Output*) merupakan teknologi Wi-Fi terbaru. MIMO dibuat berdasarkan spesifikasi Pre-802.11n. Kata “Pre-“ menyatakan “Prestandart *versions of 802.11n*”. MIMO menawarkan peningkatan *throughput*, keunggulan realibilitas dan peningkatan jumlah klien yang terkoneksi. Serta dapat menghasilkan kecepatan transfer data sebesar 108 Mbps. Secara umum perbandingan diantara keempat standar yang dimaksud dapat dijabarkan seperti pada Tabel 2.1:



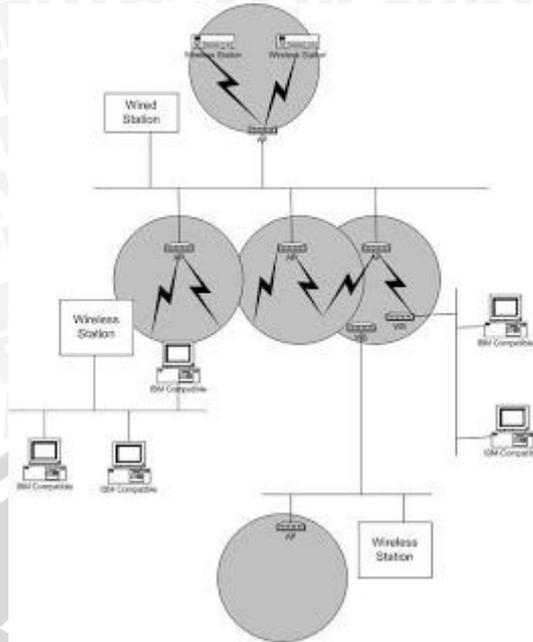
**Tabel 2.1** Perbandingan Standar *Wireless LAN*

	<b>802.11a</b>	<b>802.11b</b>	<b>802.11g</b>	<b>802.11n</b>
<b>Standar Approved</b>	July 1999	July 1999	June 2003	Not yet ratified
<b>Maximum Data Rate</b>	54 Mbps	11 Mbps	54 Mbps	Up to 100 Mbps
<b>Modulation</b>	OFDM	DSSS or CCK	DSSS or CCK or OFDM	DSSS or CCK or OFDM
<b>RF Band</b>	5 GHz	2.4 GHz	2.4 GHz	2.4 GHz or 5GHz
<b>Channel Width</b>	20 MHz	20 MHz	20 MHz	20 MHz or 40MHz

**Sumber:** (Nur Mardhiyah, 2011)

### 2.1.1.1 Topologi *Wireless-LAN*

*Wireless LAN* memungkinkan *workstation* untuk berkomunikasi dan mengakses jaringan dengan menggunakan propagasi radio sebagai media transmisi. *Wireless LAN* bisa menghubungkan LAN kabel yang telah ada sebagai sebuah *extensi* atau menjadi basis dari jaringan baru. *Wireless LAN* sangat mudah beradaptasi artinya dapat dirancang untuk lingkungan dalam ruangan dan juga untuk luar ruangan seperti menghubungkan gedung-gedung kantor, lantai produksi, rumah sakit dan Universitas.



**Gambar 2.1** *Wireless-LAN connectivity*

**Sumber:** (<http://teknik-informatika.com/>)

### 2.1.1.2 *Wireless Frame*

Ketika *client* berada pada suatu jaringan, *client* yang lainnya yang berada pada suatu jaringan akan berkomunikasi dengan *passing frame* yang berseberangan dengan jaringan. Ada beberapa macam *wireless frame* yaitu:

#### 1. *Control Frame*

- *Request to Send (RTS)*, digunakan untuk cadangan oleh pemancar
- *Clear to Send (CTS)*, digunakan untuk merespon *access point RTS frame* yang berhubungan dengan pemancar
- *Acknowledgement (ACK)*, digunakan untuk memberitahu pemancar pada saat mengirimkan data

Penggunaan dari *acknowledgement (ACK)* bekerja dengan cara yang mudah. Ketika pemancar pengirim *wireless* mengirimkan paket, pemancar penerima mengirim balik ACK sekali setelah menerima paket. Jika pemancar penerima tidak mengirim ACK, pemancar pengirim beranggapan bahwa terjadi tabrakan antar paket-paket (*collision*) dan akan mengirim kembali data tersebut.

## 2. Interframe Spacing

*Interframe spacing* adalah syarat dalam menggunakan standarisasi penggunaan *space* yang digunakan dalam *wireless LAN 802.11*. ada dua utama *spacing interval* yaitu SIFS dan DIFS.

- *Short Interframe Space (SIFS)*, adalah *space interframe* yang pendek. SIFS digunakan sebelum dan sesudah dimana semua tipe dari pesan telah dikirim. Durasinya tergantung pada waktu propagasi pada kanal, waktu untuk menyampaikan informasi dari *layer PHY* ke *layer MAC*.
- *Distributed Coordination Function Interframe Space (DIFS)*, mempunyai *interframe space* yang lebih panjang dan digunakan dalam pemancar 802.11 yang berfungsi sebagai pendistribusi dan menunggu *medium idle* untuk memulai transmisi baru

## 3. Overhead

Overhead ini terdiri atas preamble dan header, dan besarnya merupakan penjumlahan dari preamble dan header. Preamble ini ada dua yaitu long preamble dan short preamble yang besarnya sesuai standar dari IEEE 802.11.

### 2.1.2 Protokol Jaringan

Protokol adalah sebuah aturan yang mendefinisikan beberapa fungsi yang ada dalam sebuah jaringan komputer, misalnya mengirim pesan, data, informasi dan fungsi lain yang harus dipenuhi oleh si pengirim (*transmitter*) dan si penerima (*reciever*) agar komunikasi dapat berlangsung dengan benar. Selain itu protokol juga berfungsi untuk memungkinkan dua atau lebih komputer dapat berkomunikasi dengan bahasa yang sama.

### 2.1.3 Model OSI

Model OSI atau *Open System Interconnection* menggambarkan bagaimana informasi dari suatu perangkat lunak aplikasi di sebuah komputer berpindah melewati sebuah media jaringan ke suatu perangkat lunak aplikasi di komputer lain. Model OSI dibuat untuk mengatasi masalah *internetworking* akibat perbedaan arsitektur dan protokol jaringan. Model OSI secara konseptual terbagi ke dalam tujuh lapisan dimana masing-masing lapisan memiliki fungsi jaringan yang spesifik seperti yang diperlihatkan pada Gambar 2.2.

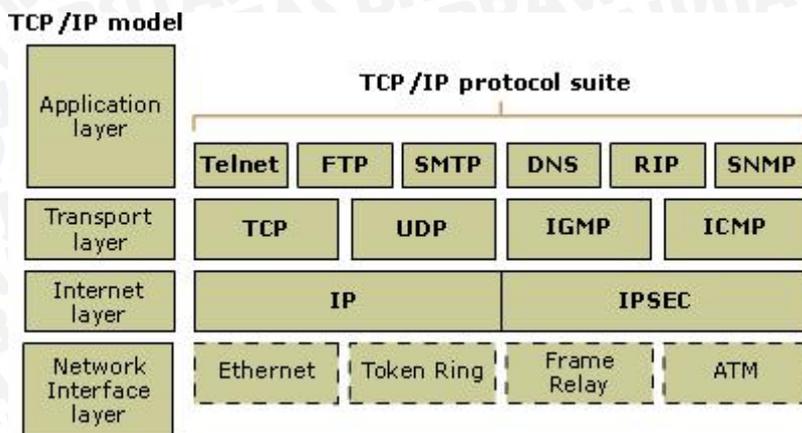


**Gambar 2.2** Model OSI

**Sumber:** (Iwan, 2008)

#### 2.1.4 TCP/IP

TCP/IP merupakan protokol jaringan komputer terbuka dan bisa terhubung dengan berbagai jenis perangkat keras dan lunak. Pada mulanya TCP/IP digunakan pada jaringan bernama ARPANET, namun saat ini telah menjadi protokol standar bagi jaringan yang lebih umum yang disebut internet. TCP terdiri dari beberapa *layer* atau lapisan yang memiliki fungsi tertentu dalam komunikasi data. Setiap fungsi dari *layer* selain dapat bekerja sama dengan *layer* pada tingkat yang lebih rendah atau lebih tinggi, juga bisa berkomunikasi dengan *layer* sejenis pada *remote host* (*peering*). Seperti pada Gambar 2.3 TCP/IP protokol terdiri dari empat *layers* yaitu *Application*, *Transport*, *Internetwork*, dan *Network Interface*.



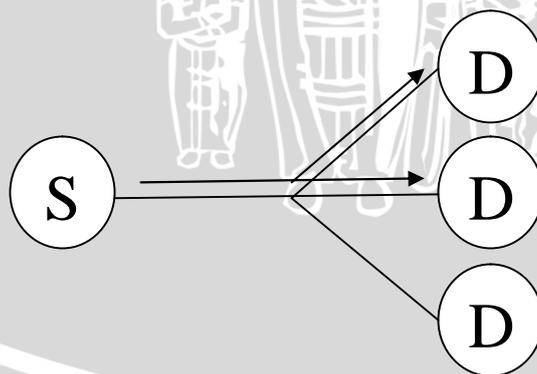
Gambar 2.3 TCP/IP layers

Sumber: (<https://technet.microsoft.com/>)

### 2.1.5 Alamat Multicast, Unicast

#### a. Multicast

*Multicast* merupakan bentuk komunikasi *one to many*, tetapi tidak seperti *broadcast*. Pada *multicast* informasi yang dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang menginginkan informasi tersebut. *Server* akan membuat *stream* satu kali kemudian *stream* ini di duplikasi dan dikirimkan ke setiap *client*. Setiap *client* akan menerima *stream* yang sama dengan *client* lainnya. Salah satu contoh penggunaan *multicast* adalah *live streaming video*.



Gambar 2.4 Multicast

Sumber: (<http://jaringankomputer.wordpress.com/>)

## b. Unicast

*Unicast* adalah jenis transmisi dimana informasi dikirim hanya dari satu pengirim ke satu penerima. Dalam kata lain, *unicast* transmisi adalah satu-ke-satu *node*. Contoh dari transmisi *unicast* yaitu, HTTP, SMTP, Telnet, SSH, POP3 dimana permintaan informasi diarahkan dari satu pengirim ke satu penerima di ujung lainnya. Alamat *unicast* inilah yang harus digunakan oleh semua host TCP/IP agar dapat saling terhubung. Komponen alamat ini terbagi menjadi dua jenis, yakni alamat *host (host identifier)* dan alamat jaringan (*network identifier*). Contoh penggunaan *unicast* adalah *video on demand*.



**Gambar 2.5** *Unicast*

**Sumber:** (<http://jaringankomputer.wordpress.com/>)

## 2.2 *Internet Protocol (IP)*

IP merupakan salah satu protokol yang terdapat pada *Internet Layer*. Pada saat ini, ada dua macam *Internet Protocol* yang digunakan, yaitu IPv4 dan IPv6.

### 2.2.1 *IPv4 (Internet Protocol version 4)*

Protokol layer *network* dalam rangkaian protokol TCP/IP yang digunakan saat ini adalah IPv4. IPv4 dikembangkan pada awal tahun 1980-an dengan diperkenalkannya RFC 791, telah digunakan selama dua puluh tahun dan terbukti tangguh selama ini. IPv4 menyediakan komunikasi *host-to-host* antar sistem dalam internet. Meskipun IPv4 dirancang dengan baik, namun tetap mempunyai kekurangan yaitu pada keterbatasan alamat yang dimiliki, sehingga tidak cocok lagi digunakan untuk internet yang berkembang secara cepat.

### 2.2.2 *IPv6 (Internet Protocol version 6)*

IPv6 (*Internet Protocol version 6*) dirancang oleh IETF untuk mengatasi kekurangan yang terdapat pada IPv4. IPv6 juga dikenal sebagai IPng (*Internet Protocol next generation*). Perkembangan IPv6 dimulai pada tahun 1991. Inti dari IPv6 telah distandarisi pada tahun 1995 dan diperbaharui dalam RFC 2460 tahun 1998. Dalam IPv6, *Internet Protocol* dimodifikasi secara luas untuk mengakomodasi pertumbuhan

internet yang tidak dapat diprediksi. Format dan panjang alamat IP diubah, begitu juga format paketnya.

### 2.2.2.1 Kelebihan IPv6

Masalah keterbatasan alamat pada IPv4 dapat diselesaikan dengan adanya IPv6 yang alamatnya terdiri dari 128 bit sehingga dapat memberikan alamat sejumlah  $2^{128}$ . Selain itu, IPv6 mempunyai beberapa kelebihan dan layanan tambahan, antara lain:

1. Memiliki *header* yang lebih sederhana dan efisien untuk mengurangi beban penanganan paket dan memudahkan perluasan paket oleh *router*.
2. Adanya mekanisme konfigurasi alamat secara otomatis (*address auto configuration*) oleh setiap *node*.
3. Mempunyai kemampuan *flow labeling* yaitu kemampuan untuk menandai paket yang membutuhkan penanganan khusus seperti layanan *real time*.
4. Mendukung sistem autentifikasi untuk kerahasiaan data pada lapis IP (*IP Secure*).
5. Mampu berinteraksi dengan IPv4, sehingga perubahan dari IPv4 ke IPv6 dapat dilakukan secara bertahap.
6. Mendukung *mobile IP*.

Adanya IPv6 tidak membuang secara keseluruhan fungsi IPv4 pada jaringan, melainkan ada suatu mekanisme transisi tertentu yang memungkinkan kedua versi IP tersebut berkomunikasi.

### 2.2.2.2 Arsitektur IPv6

Header IPv6 didesain mempunyai *field* yang lebih sedikit dibandingkan dengan IPv4 yaitu sebesar 40 *byte*. Hal ini menyebabkan proses penjaluran atau aliran data menjadi lebih ringkas karena penanganan yang dilakukan oleh *router* terhadap tiap *field* menjadi lebih sedikit.

Secara garis besar, paket IPv6 dibagi menjadi 2 jenis *header* yaitu:

1. *Base Header* yaitu *header* yang selalu diberikan pada paket IPv6.



**Gambar 2.6** Format *base header* (*header* dasar)/*header* IPv6  
**Sumber:** Taufan Riza (2002:51)

### **Version**

*Version* adalah bit-bit yang ditambahkan oleh *header* IP untuk tiap data yang melewati protokol IP dan pada data tersebut diberikan informasi tentang versi IP yang digunakan. Dalam IPv6, *field* ini berisi 4 bit (0110 dalam biner) yang bernilai 6.

### **Traffic Class (kelas lalu lintas) / Different Service**

*Field* ini mempunyai fungsi yang hampir sama dengan *field type of service* pada *header* IPv4, yaitu untuk menentukan dan membedakan kelas-kelas atau prioritas yang berlainan dari paket IPv6. *Field* ini berukuran 8 bit.

### **Flow Label**

Untuk menandai aliran data tertentu yang membutuhkan penanganan khusus pada paket IPv6. Penanganan khusus tersebut dilakukan oleh *router* yang dilalui paket tersebut di dalam jaringan. Contohnya adalah pada penggunaan aplikasi tertentu misalnya aplikasi *real time*.

### **Payload Length**

Digunakan untuk menandai panjang *payload*. *Payload* adalah sisa paket setelah *header* IPv6 dalam bentuk oktet (di luar *header* IPv6). *Header* tambahan yang ada dihitung sebagai bagian dari *payload*, termasuk panjang *header* tambahan tersebut.

### **Next Header**

Merupakan *field* 8 bit yang menentukan *header* yang mengikuti *base header* dalam *datagram*. *Header* ini merupakan salah satu pilihan *extension header* yang digunakan oleh IP atau *header upper-layer protocol* seperti TCP atau UDP. Masing – masing *extension header* juga memuat *field* ini.

### **Hop Limit**

Untuk menandai maksimum *hop* yang dapat digunakan oleh IPv6 dalam lalu lintas internet.

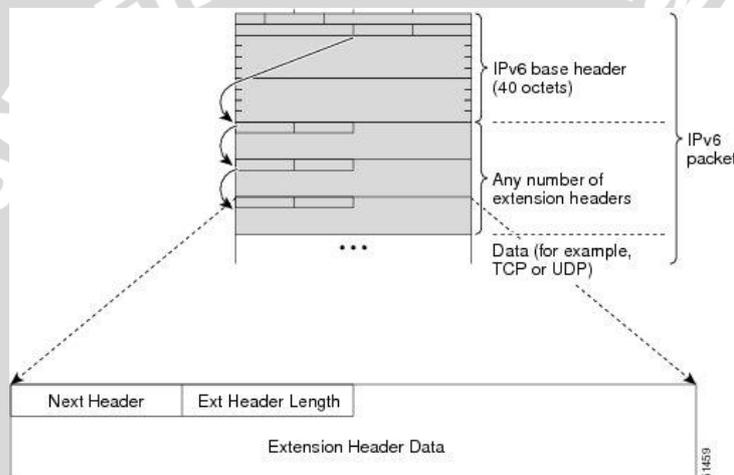
### Source Address

Digunakan untuk menyimpan alamat IPv6 dari *host* asal. Ukuran *field* ini 128 bit.

### Destination Address

Digunakan untuk menyimpan alamat IPv6 dari *host* tujuan. Ukuran *field* ini 128 bit.

2. *Extension Header* (*header* tambahan/pilihan) yaitu *header* pilihan yang tidak selalu diberikan pada paket IPv6, diletakkan antara *base header* IPv6 dan *header* protokol *upper layer* (TCP/UDP), seperti yang terlihat pada Gambar 2.2. Masing – masing *extension header* panjangnya 8 byte (64 bit). Seluruh *extension header* secara bersama–sama membentuk daftar rangkaian *header*. *Extension header* tersedia untuk sejumlah fungsi yang berbeda, tetapi tidak dianggap sebagai bagian *header* IPv6 itu sendiri.



**Gambar 2.7** Format *extension header* IPv6

**Sumber:** Taufan Riza (2002:53)

**Tabel 2.2** *Extension Header*

Extension Header	Next Header Value	Description
Hop-by-Hop Options header	0	read by all devices in transit network
Routing header	43	contains methods to support making routing decision
Fragment header	44	contains parameters of datagram fragmentation
Destination Options header	60	read by destination devices
Authentication header	51	information regarding authenticity
Encapsulating Security Payload header	50	encryption information

**Sumber:** (<http://www.tutorialspoint.com>)

Seperti yang dijabarkan pada Tabel 2.2 *Extension header* meliputi:

### ***Hop by Hop Header***

*Header* ini membawa informasi tambahan (bila ada) serta harus diproses oleh tiap *node* dan *router* sepanjang jalur pengiriman. *Header* ini terletak tepat setelah IPv6 *base header*.

### ***Destination Option Header***

Membawa informasi pilihan yang secara khusus ditujukan untuk alamat tujuan paket.

### ***Routing Header***

*Routing header* mendorong proses *routing* melalui sejumlah *intermediate router*. *Header* ini berisi daftar *node* yang harus dilewati dalam perjalanan menuju tujuan. Masing–masing *node* dalam daftar memproses *header* dan mengubah alamat tujuan ke *router* selanjutnya dalam daftar. *Header* ini serupa dengan *option loose source route* pada IPv4.

### ***Fragment Header***

Digunakan oleh sumber ketika paket difragmentasi, paket dengan ukuran lebih besar akan dipecah sesuai dengan *path MTU (Maximum Transmission Unit)*. Fragmentasi hanya dilakukan oleh sumber paket, bukan *router*. Posisi fragmen tertentu dalam paket asli dan digunakan oleh *node* tujuan untuk menyusun kembali paket.

### ***IPSec Authentication Header***

Menyediakan kerahasiaan, integritas, dan autentikasi sumber.

### ***IPSec ESP Header***

Menyediakan layanan yang sama seperti *authentication header*, tetapi menambahkan fungsi *enkripsi* (proses perubahan data dalam bentuk sandi untuk keamanan informasi).

### 2.2.2.3 Maximum Transmission Unit (MTU) IPv6

*Maximum Transmission Unit* IPv6 adalah ukuran maksimum paket IPv6 sebelum menuju lapisan *interface*. Besarnya MTU IPv6 tidak kurang dari 1280 byte dan dapat lebih besar tergantung jenis lapisan *interface* yang digunakan. Misalnya jika lapisan *interface* menggunakan *Ethernet* maka besar MTU adalah 1500 *byte*.

## 2.3 Video Streaming

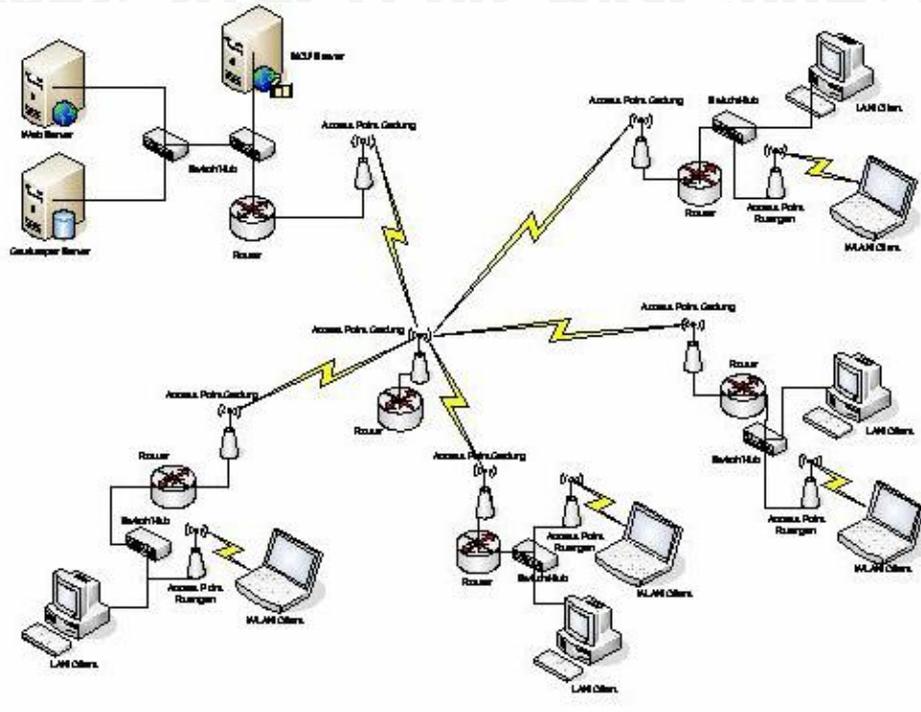
*Video Streaming* adalah informasi yang berisi gabungan gambar yang ditampilkan berurutan dengan kecepatan tertentu yang kemudian di kompresi menjadi file yang lebih kecil kemudian dikirimkan melalui jaringan internet dan ditampilkan oleh *media player*. *Media player* adalah aplikasi yang dapat melakukan proses *coding* dan *decoding* video agar video dapat ditampilkan ke layar monitor (Austerberry, 2005). Ada dua jenis tipe *video streaming* yaitu:

1. *Live Streaming*, dimana tayangan yang ditampilkan merupakan siaran langsung.
2. *VoD (Video on Demand)* dimana video yang ditampilkan sudah terlebih dahulu direkam (*pre-encoded*) atau disimpan dalam *server*.

### 2.3.1 Video Conference

*Video conference* adalah seperangkat teknologi telekomunikasi interaktif yang memungkinkan dua pihak atau lebih di lokasi berbeda dapat berinteraksi melalui pengiriman dua arah *audio* dan *video* secara bersamaan. *Video conference* dapat memudahkan orang untuk berkomunikasi jarak jauh dengan menggunakan komputer dan media internet.

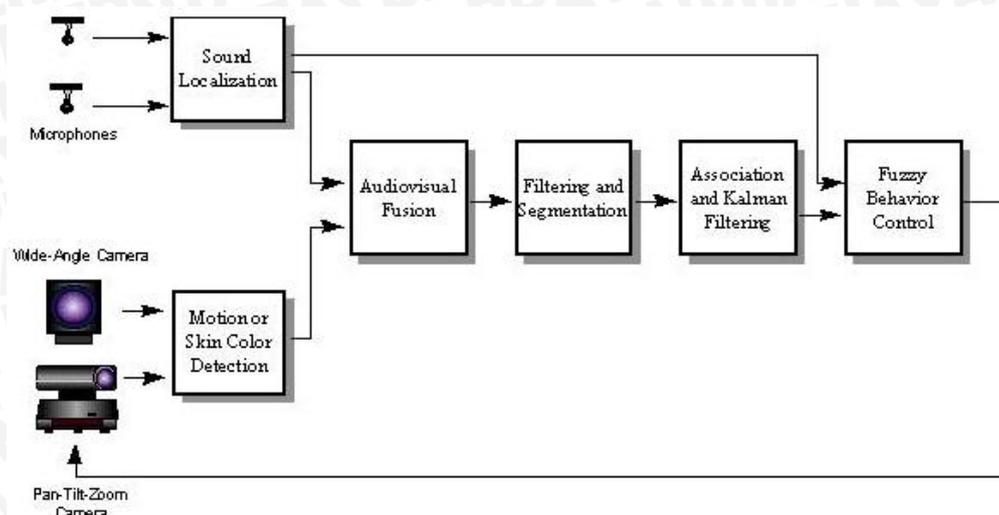
Dalam proses pengiriman paket data, digunakan perangkat komunikasi data digital seperti misalnya teknologi internet *broadband*. Teknologi tersebut dapat mengirimkan dan menerima paket data dalam jumlah yang sangat besar.



**Gambar 2.8** Arsitektur *Video Conference*

**Sumber:** (<https://technet.microsoft.com/>)

Gambar 2.8 merupakan arsitektur dari video conference, selain arsitektur terdapat pula blok diagram sistem dari video conference. Blok diagram dari langkah-langkah pengolahan yang dilakukan oleh sistem ditunjukkan pada Gambar 2.9. Pengolahan suara dilakukan pada mikrofon yang berkorelasi dengan sinyal suara untuk membuat histogram yang berkaitan dengan kekuatan suara. Sementara itu, CPU host menangkap video dari kamera yang mendeteksi gerak dan warna. Kemudian tugas sensor fusi menggabungkan informasi warna dengan histogram suara. Gambar yang dihasilkan tersegmentasi, dan wilayah dilacak menggunakan Kalman filtering di koordinat Cartesian global. Terakhir, sistem kontrol berbasis Fuzzy menafsirkan suara dan data pelacakan untuk mengontrol kamera pan-tilt-zoom.



**Gambar 2.9** Blok Diagram *Video Conference*

**Sumber:** (<http://www.ise.ncsu.edu/kay/msf/intro.htm>)

### 2.3.2 Kompresi *Video Conference*

Kompresi adalah pengubahan data kedalam bentuk yang memerlukan bit yang lebih sedikit, biasanya dilakukan agar data dapat disimpan atau dikirimkan dengan lebih efisien.

Dalam melakukan perencanaan sebelum menggelar fasilitas konferensi *video* melalui Internet, kita perlu memperhitungkan kebutuhan *bandwidth* untuk kesuksesan sebuah konferensi *video*. Secara umum ada dua kebutuhan *bandwidth* yang perlu di penuhi, yaitu:

- Kebutuhan *bandwidth* untuk mengirimkan sinyal gambar / *video*. Sebuah kanal *video* yang baik tanpa dikompresi akan mengambil *bandwidth* sekitar 9Mbps.
- Kebutuhan *bandwidth* untuk mengirimkan sinyal suara / *audio*. Sebuah kanal suara (*audio*) yang baik tanpa dikompresi akan mengambil *bandwidth* sekitar 64Kbps

#### 2.3.2.1 Teknik Kompresi Suara

Terdapat beberapa teknik kompresi suara yang sering digunakan dengan beberapa parameter yang mencerminkan kinerja dari teknik kompresi suara tersebut, diantaranya adalah sebagai berikut:

**Tabel 2.3** Teknik kompresi suara

Kompresi	Bandwidth (Kbps)	Delay (ms)
G.711PCM	64	0.125
G.726 ADPCM	32	0.125
G.728 LD-CELP	16	0.625
G.729 CS-ACELP	8	10
G.729 x2 <i>Encoding</i>	8	10
G.729 x3 <i>Encoding</i>	8	10
G.729a CS-ACELP	8	10
G.723.1 MPMLQ	6.3	30
G.723.1 ACELP	5.3	30

**Sumber:** (Onno W Purbo, 2002)

### 2.3.2.2 Teknik Kompresi Video

Kompresi *video* merupakan proses yang digunakan untuk mengkompres *video* sementara harus tetap menjaga kualitas *video* untuk aplikasi yang ditunjuk. Salah satu teknik kompresi *video* yang digunakan adalah CODEC H.264. H.264 menawarkan efisiensi kompresi yang lebih baik yakni kompresi *video* yang lebih berkualitas dan fleksibilitas yang lebih besar dalam melakukan kompresi, transmisi dan penyimpanan *video*. *Video encoder* pada H.264 dapat melakukan prediksi, transformasi dan proses *encoding* untuk menghasilkan kompresi *bitstream* H.264. Sedangkan *video decoder* H.264 dapat melakukan proses *decoding* secara lengkap. Dibandingkan dengan standar seperti MPEG-2 dan MPEG-4 *Visual*, H.264 memiliki kelebihan antara lain:

- a. Kualitas gambar yang lebih baik pada bitrate kompresi yang sama
- b. Kecepatan bit kompresi yang lebih rendah untuk kualitas gambar yang sama.

Standar H.264 menawarkan fleksibilitas yang lebih besar dari segi kompresi dan transmisi.

Tabel 2.4 Audio dan video coding

Audio Codec	Bit Rate	Delay Codec
G.711	64 Kbps	0.75 ms
G.26	32 Kbps	1 ms
G.723.1	5,3/6,3 Kbps	30 ms
G.728	16 Kbps	3-5 ms
G.729	8 Kbps	10 ms
Video Codec	Bit Rate	Delay Codec
H.261	P x 64 Kbps	20-150 ms
H.263	10-384 Kbps	20-150 ms
H.264/AVC	64-384 Kbps	150-300 ms

Sumber: (Onno W Purbo, 2002)

## 2.4 Parameter Kinerja Jaringan Aplikasi Video Conference

*Quality of Service* (QoS) adalah parameter yang menyatakan kinerja suatu jaringan berdasarkan beberapa parameter. Sedangkan menurut ITU-T E.800, QoS adalah sekumpulan efek performansi yang menentukan derajat kepuasan pengguna terhadap *service* yang diberikan oleh jaringan. Pada jaringan *Wireless-LAN*, QoS dapat dilihat dari parameter *throughput*, *packet loss*, dan *delay* yang dilihat dari sisi pengguna.

### 2.4.1 Throughput

*Throughput* adalah parameter yang menunjukkan jumlah data yang diterima oleh pengguna dengan benar setelah melalui media transmisi. Nilai *throughput* diukur dengan satuan bps (bit per second). *Throughput* erat kaitannya dengan *bitrate*. Perbedaannya adalah *bit rate* merupakan rata-rata nilai bit informasi yang diperlukan *file* atau data untuk ditransmisikan dalam satuan waktu. Untuk mengetahui nilai *throughput* maka diperlukan data *bit rate file video* yang akan di transmisikan. Nilai *throughput* akan sama dengan nilai *bitrate* jika seluruh data yang ditransmisikan diterima dengan benar oleh pengguna. Berikut adalah persamaan untuk menghitung *throughput* dengan waktu transmisi paket data.

Rumusan secara teori waktu mentransmisikan data:

$$t_T = DIFS + Overhead + \frac{Data}{Rate} + SIFS + ACK \quad (2.1)$$

Keterangan:

- $t_T$  = Waktu transmisi
- DIFS = DCF *Interframe Space* ( $\mu$ s)
- Overhead = Terdiri atas *preamble* dan *header* ( $\mu$ s)
- Data = WLAN data yang di transmisikan

Rate	= Data rate pada WLAN
SIFS	= Short Interframe Space ( $\mu\text{s}$ )
ACK	= Acknowledgement frame ( $\mu\text{s}$ )

Sehingga *throughput* dapat diketahui dengan menggunakan persamaan berikut ini:

$$\text{Throughput} = \frac{\text{Data}}{t_T} \quad (2.2)$$

Keterangan:

Data = WLAN data yang di transmisikan

$t_T$  = Waktu transmisi

Jumlah gerakan dalam *video* dikategorikan menjadi tiga urutan yang disebut *motion rank*. Tiga jenis tersebut adalah *low motion*, *medium motion* dan *high motion*.

- *Low motion* adalah *video* yang memiliki gerakan yang tidak terlalu banyak seperti saat melakukan wawancara atau melakukan *review* suatu produk. Nilai dari *low motion* = 1.
- *Medium motion* adalah *video* yang memiliki gerakan-gerakan yang dinamis dengan beberapa perubahan *scene* atau adanya perubahan *background* secara cukup drastis. Nilai dari *medium motion* = 2.
- *High motion* adalah *video* yang memiliki gerakan yang sangat dinamis seperti *action movies* dengan pergerakan dan perubahan *scene* yang sangat cepat. Nilai dari *high motion* = 4.

#### 2.4.2 Packet Loss

*Packet loss* merupakan parameter yang menunjukkan banyaknya jumlah paket yang hilang atau tidak sampai ke tujuan ketika melakukan pengiriman data dari sumber ke tujuan. Pada beberapa situasi jika *packet loss* terjadi terdapat penurunan performansi secara signifikan sehingga dibutuhkan upaya untuk meminimalkan terjadinya *packet loss* ini. Berdasarkan ITU-T G.1010, Aplikasi *streaming* memiliki persyaratan *packet loss* yang berbeda-beda berdasarkan media yang dikirimkan. Berikut Tabel 2.5 menunjukkan syarat *packet loss* berdasarkan ITU-T G.1010,

**Tabel 2.5** Packet Loss Standart by ITU-T G.1010 for Streaming Application

Media	Aplikasi	Derajat Simetri	Informasi Rugi
Audio	Conversational Voice	Two-way	< 3% Packet Loss Ratio (PLR)
Audio	Voice Messaging	One-way	< 3% PLR
Audio	High Quality Audio Streaming	One-way	< 1% PLR
Video	Videophone	Two-way	< 1% PLR
Video	Streaming	One-way	< 1% PLR

Sumber: ITU-T G.1010, 2002

Persamaan untuk *Packet Loss* ditentukan sebagai berikut (Mischa Schwartz, 1987),

$$\text{Packet Loss} = \frac{N_{\text{lost packet}}}{N_{\text{packet}} - N_{\text{lost packet}}} \times 100\% \quad (2.3)$$

Keterangan:

*Packet Loss* = Prosentase *Packet Loss* (%)

$N_{\text{lost packet}}$  = Jumlah paket data yang hilang (paket)

$N_{\text{packet}}$  = Jumlah paket data yang diterima dengan benar (paket)

Untuk probabilitas packet loss ditentukan dengan persamaan berikut:

$$\text{Probabilitas}_{PL} = \frac{\text{Traffic Sent} - \text{Traffic Received}}{\text{Trafik Sent}} \quad (2.4)$$

Keterangan:

*Traffic Sent* = Paket yang dikirim (pps)

*Traffic Received* = Paket yang diterima (pps)

Semakin kecil paket data yang hilang dalam suatu pentransmisian, maka akan semakin kecil nilai *Packet Loss*-nya dan semakin baik kualitas jaringannya.

### 2.4.3 Delay

*Delay* total merupakan penjumlahan dari *delay* codec dan *Delay end-to-end*. *Delay global (Delay end-to-end)* adalah waktu yang dibutuhkan untuk mengirimkan paket data dari sumber sampai ke tujuan. *Delay end-to-end* merupakan penjumlahan antara *delay* transmisi, *delay* propagasi, *delay* antrian dan *delay* proses (Mellouk, 2013). *Delay end-to-end* pada paket telah banyak digunakan pada penelitian untuk mengkaji performansi jaringan pada aplikasi dan pengembangannya (Vivanco, 2008)

Berikut adalah penjelasan macam-macam *delay* yang menyusun *delay end-to-end*:

#### 1. Delay Propagasi

*Delay* propagasi adalah waktu yang dibutuhkan oleh paket data untuk merambat dari *server* menuju *client* melalui media transmisi. Kecepatan propagasi berbagai jenis kabel (media) ditunjukkan pada Tabel 2.6.

**Tabel 2.6** Kecepatan Propagasi

Jenis Kabel	Kecepatan Propagasi
Thick Coax	0,77c
Thin Coax	0,65c
Twisted Pair	0,59c
Fiber Optic	0,66c
UTP	0,64c

**Sumber:** (<http://stason.org/TULARC/networking/lans-ethernet/3-11-What-is-propagationdelay-EthernetPhysical-Layer.html>)

#### 2. Delay Enkapsulasi

*Delay* enkapsulasi adalah waktu yang dibutuhkan untuk proses pemaketan data dengan menambahkan *header* pada paket data, sehingga paket data tersebut dapat dikirimkan ke tujuan.

#### 3. Delay Dekapsulasi

*Delay* Dekapsulasi adalah waktu yang dibutuhkan untuk proses pembacaan keseluruhan *header* dari sebuah paket yang diterima suatu *node*.

#### 4. *Delay* antrian

*Delay* antrian adalah lamanya waktu yang diperlukan maupun keterlambatan yang dialami oleh paket data dalam kurun waktu tertentu sebagai akibat terjadinya pemberlakuan antrian untuk pemrosesan di dalam jaringan komputer.

#### 5. *Delay* codec

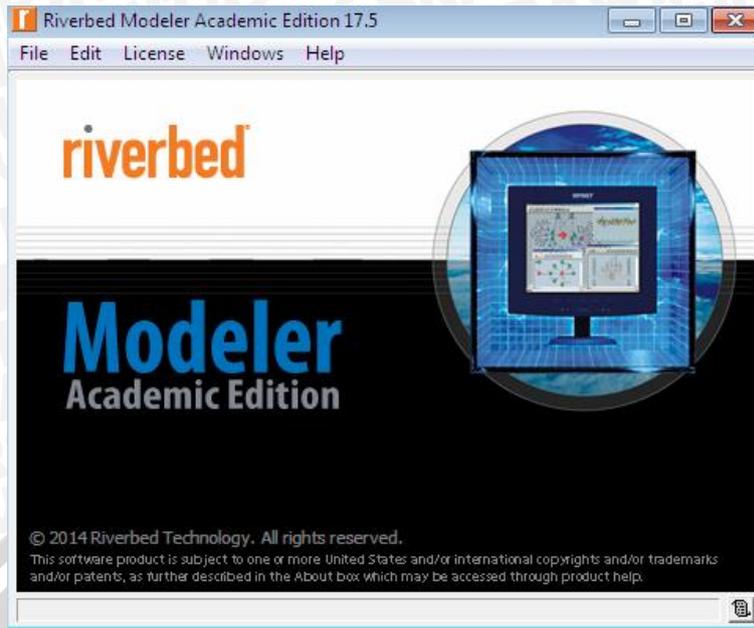
*Delay* ini terjadi pada sisi *encoder* terdiri dari waktu untuk mengakumulasi sample suara ke dalam *frame* suara, waktu untuk mengkompresi paket suara, waktu untuk memuat *frame* suara ke dalam paket dan mentransfer paket tersebut ke jaringan *transport* dan *delay hardware* yang bersifat tetap. Atau waktu yang dibutuhkan untuk mengkompresi satu blok *sample* PCM.

#### 6. *Delay* Transmisi

*Delay transmisi* adalah waktu yang dibutuhkan untuk meletakkan sebuah paket multimedia ke media transmisi dan dipengaruhi oleh ukuran paket serta kapasitas media transmisi.

### 2.5 *Riverbed Modeler*

*Riverbed Modeler* adalah sebuah *network simulator* yang dirancang untuk membuat model dan menjalankan simulasi protokol komunikasi. Dengan menggunakan simulasi, *network designers* dapat mengurangi biaya penelitian dan memastikan kualitas produk yang optimal. Teknologi terbaru *Riverbed Modeler* menyediakan sebuah lingkungan untuk mendesain protokol dan teknologi, juga menguji dan mendemonstrasikan dengan skenario yang realistik sebelum diproduksi.



**Gambar 2.10** Tampilan awal *Riverbed Modeler – Academic Edition*  
**Sumber :** (<http://www.riverbed.com>)

Di dalam simulasi jaringan berbasis IP khususnya IPv6 dengan mempergunakan *simulator Riverbed*, hal-hal yang perlu dilakukan antara lain:

a. Konfigurasi Jaringan

Di dalam *simulator Riverbed* harus dilakukan penggambaran model jaringan yang akan disimulasikan. Konfigurasi yang digambarkan disesuaikan dengan model jaringan yang akan disimulasikan. Secara umum untuk menggambarkan suatu jaringan berbasis IP antara lain terdapat: *router*, *bridge/switch*, *hub*, LAN, *link* baik yang dipergunakan untuk menghubungkan antar *router* ataupun hubungan ke *user*, *workstation*, *application server*, dan lain-lain. Kelengkapan suatu model akan tergantung kepada kebutuhan dan kerumitan yang diinginkan.

b. *Profile User*

Dipergunakan untuk menggambarkan *profile* dari *user* yang disimulasikan di dalam model tersebut. Sebagai contoh *profile* karyawan akan memiliki *profile* sesuai dengan kondisi karyawan di suatu perusahaan apakah dia memiliki aksesibilitas untuk menjalankan semua aplikasi di dalam jaringan perusahaan tersebut atau terbatas.

c. Layanan

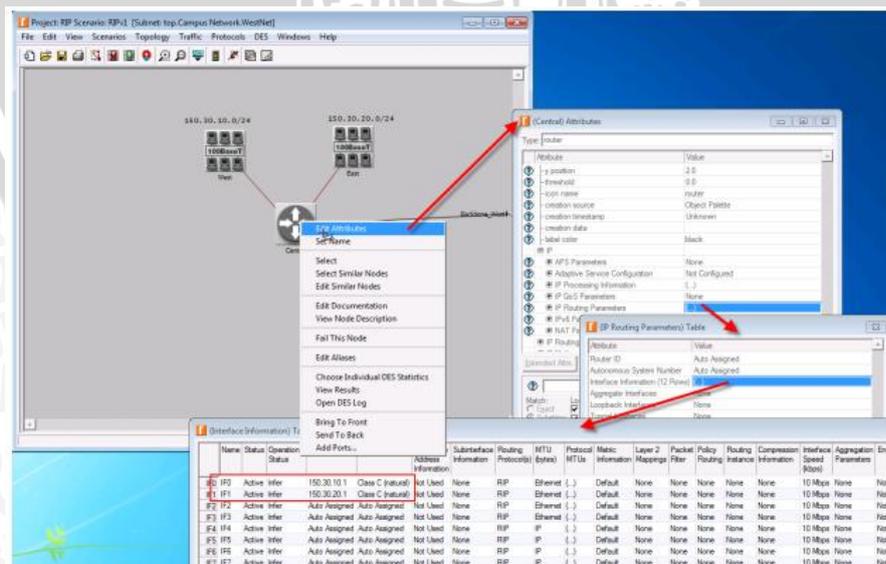
Dipergunakan untuk menggambarkan aplikasi/layanan apa saja yang dijalankan di dalam jaringan tersebut. Di dalam *simulator Riverbed*, aplikasi yang dapat dijalankan antara lain *Email*, TELNET, *Database*, FTP, Print, VoIP, *remote login*, *video conference* ataupun aplikasi lain yang disesuaikan dengan kebutuhan pelanggan.

Aplikasi tersebut merupakan aplikasi yang *default* telah disediakan oleh *Riverbed*, dan masing-masing terdiri atas aplikasi yang dijalankan secara umum, rendah, berat ataupun dapat di *setting* sesuai kebutuhan

### 2.5.1 Kelebihan *Riverbed Modeler*

Berikut adalah beberapa alasan digunakannya *Riverbed Modeler* dalam penelitian ini:

- *Riverbed* menyediakan *Graphic User Interface* (GUI) yang sangat nyaman dan mudah untuk dipelajari,
- *Riverbed* dapat digunakan untuk membuat model pada seluruh jaringan, termasuk *router*, *switch*, *protokol*, *server*, dan aplikasi lain yang mendukung. Berbagai macam sistem komunikasi mulai LAN tunggal hingga antar jaringan global dapat didukung,
- *Software Riverbed* (dengan model *source code*) tersedia secara gratis untuk penelitian akademis dan mengajar masyarakat,
- *Discrete event simulation* (DES) engine pada *Riverbed* untuk simulasi jaringan adalah tercepat. Biasanya diperlukan waktu beberapa menit untuk menyelesaikan simulasi percobaan laboratorium,
- *Riverbed* memiliki komunitas pengguna yang besar. *Riverbed* telah digunakan lebih dari 500 perusahaan, penyedia layanan, dan organisasi pemerintah di seluruh dunia. siswa yang memiliki pengalaman dengan simulator *Riverbed* akan memiliki peluang kerja jauh lebih baik di masa depan dalam industri.



**Gambar 2.11** Tampilan proses simulasi dengan menggunakan *Riverbed Modeler*  
**Sumber :** (<http://www.riverbed.com>)

## 2.5.2 Discrete Event Simulation (DES)

Simulasi adalah proses mendesain suatu model yang dikomputerisasi dari sebuah sistem atau sebuah proses dan menghubungkan percobaan dengan model tersebut untuk tujuan memahami perilaku dari sistem atau mengevaluasi strategi yang beragam dari operasi pada sistem tersebut.

Kelebihan dari proses simulasi adalah :

- dapat mengendalikan percobaan, dengan mengubah parameter tertentu tanpa harus mengubah atau mengganggu system yang telah ada,
- dapat lebih menghemat waktu,
- merupakan perangkat yang efektif digunakan dalam program pelatihan

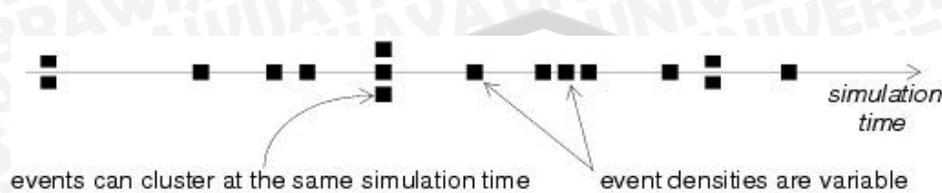
*Discrete event simulation* (DES) merupakan suatu model simulasi dimana perubahan statusnya terjadi pada titik-titik diskrit dalam waktu yang dipicu oleh suatu kejadian (*event*). Dalam titik waktu tersebut akan terjadi suatu *event*, dimana *event* didefinisikan sebagai suatu kejadian yang dapat mengubah kondisi suatu sistem. Contohnya operator yang sebelumnya menganggur (*idle*), setelah kedatangan suatu pelanggan akan menjadi sibuk. Atau sebaliknya dari kondisi sibuk ke kondisi *idle*.

Berikut adalah beberapa kejadian yang dapat dimodelkan sebagai sebuah *event*:

- menerima sebuah paket atau sebuah perintah dalam suatu proses,
- mulai atau berakhirnya transmisi atau penerimaan paket-paket pada suatu *link*,
- pembangkitan sebuah pesan baru, perintah, atau tugas oleh sebuah proses aplikasi,
- kegagalan (atau *recovery* dari kegagalan) pada suatu perangkat.

Ada dua pendekatan pokok yang digunakan untuk meningkatkan waktu simulasi yaitu mempercepat waktu *event* berikutnya dan mempercepat interval waktu tetap. Dengan pendekatan mempercepat *event* berikutnya, waktu simulasi diinisialisasi dengan (nol) 0 kemudian banyaknya kejadian dari suatu *event* di masa depan ditentukan. Waktu simulasi kemudian dipercepat ke waktu kejadian *event* paling dekat di masa depan, kemudian titik-titik kondisi sistem diperbarui ke setiap kejadian yang telah terjadi sehingga banyaknya *event* di masa depan juga diperbarui. Kemudian waktu simulasi dipercepat, kondisi sistem diperbarui, banyaknya *event* masa depan ditentukan dst. Proses dari waktu simulasi yang dipercepat ini dari waktu *event* satu ke yang lain kontinu sampai beberapa kondisi berhenti dan telah terpenuhi. Sejak semua perubahan kondisi hanya terjadi pada *discrete event simulation*, dimana periode tidak aktif dilompati dari satu

waktu *event* ke waktu *event* lain. Waktu simulasi hanya berubah ketika *event* baru terjadi, selama satu *event* terjadi, waktu tidak dapat berubah. Waktu simulasi sekarang didefinisikan sebagai waktu *event* saat ini dieksekusi. Waktu simulasi tidak perlu dipercepat dengan besar yang sama pada setiap *event*, namun melompat pada waktu *event* selanjutnya.



**Gambar 2.12** Distribusi *event* pada *timeline* simulasi

Sumber : (*Riverbed* tutorial.pdf)

*Riverbed Modeler* mendukung pemodelan dalam sistem terdistribusi, dan *Riverbed* juga memperbolehkan dilakukannya *multiple-event* terjadi secara simultan dalam simulasi dan mempengaruhi komponen-komponen yang berbeda dari sistem. Oleh karena itu, dimungkinkan untuk terjadinya beberapa *event* berbeda dalam waktu simulasi yang sama.



