

## BAB 2

### LANDASAN TEORI

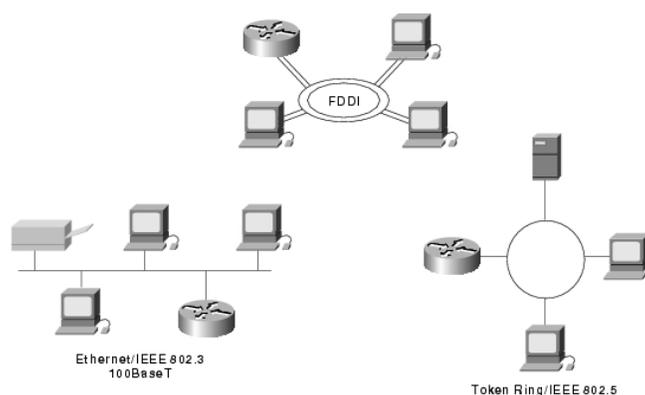
#### 2.1. Jaringan Komputer

Jaringan komputer adalah kombinasi *hardware*, *software* dan pengkabelan (*cabling*) yang secara bersama-sama memungkinkan berbagai piranti komputasi untuk berkomunikasi satu sama lain (Odom, 2004, p5).

##### 2.1.1. Tipe Jaringan Komputer

###### 2.1.1.1. Local Area Network (LAN)

LAN merupakan jaringan kecepatan tinggi pada suatu lingkungan *Local* tertentu, misalnya jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil (Arief, 2006, p8).



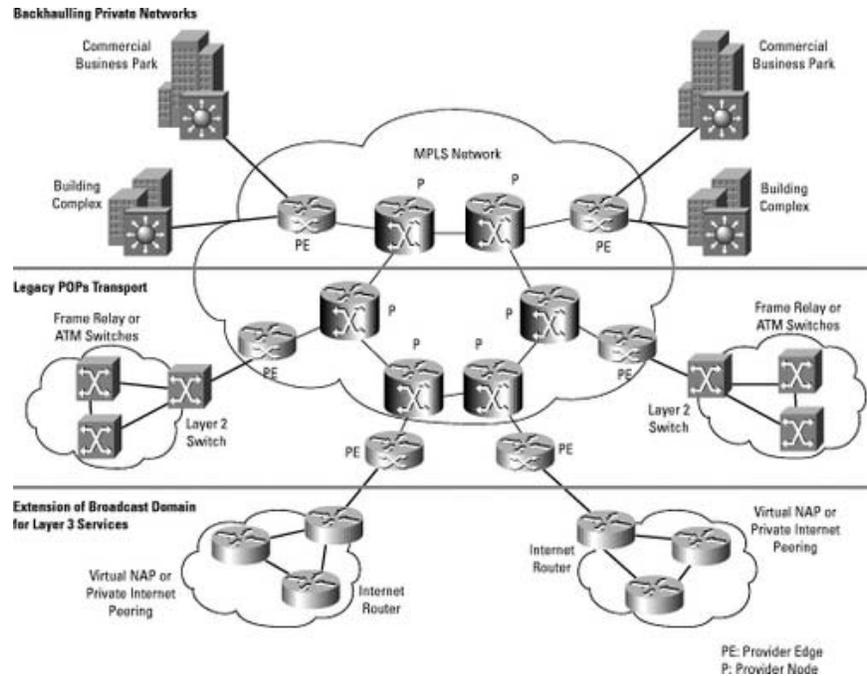
Gambar 2.1 *Local Area Network*

(Sumber :

[http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Intro-to-LAN\\_files/Intro-to-LAN-1.jpg](http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Intro-to-LAN_files/Intro-to-LAN-1.jpg))

### 2.1.1.2. Metropolitan Area Network (MAN)

Menurut Arief ( 2006, p8), MAN merupakan jaringan kecepatan tinggi untuk node yang terdistribusi dalam jarak jauh (biasanya untuk satu kota atau suatu daerah besar).



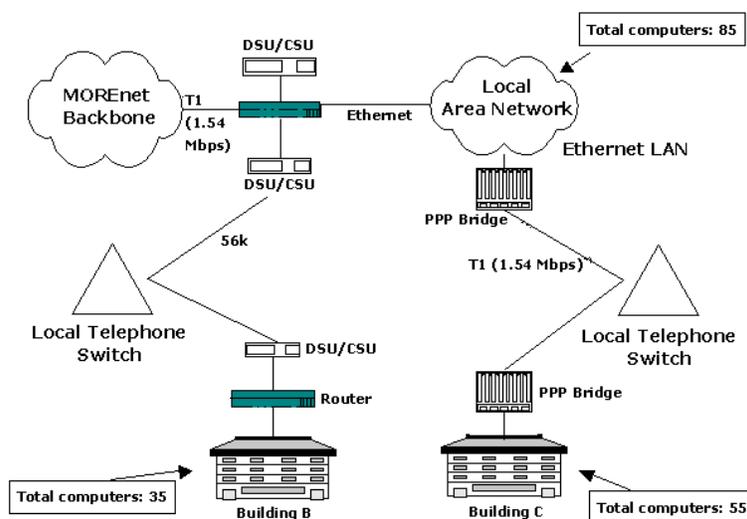
Gambar 2.2 Metropolitan Area Network

(Sumber : <http://product->

[image.tradeindia.com/00209645/b/Metropolitan-Area-Network-MAN-.jpg](http://product-image.tradeindia.com/00209645/b/Metropolitan-Area-Network-MAN-.jpg))

### 2.1.1.3. Wide Area Network (WAN)

Menurut Arief ( 2006, p9), WAN merupakan komunikasi untuk jarak yang sangat jauh. WAN dapat mencapai jarak ratusan ribu kilometer. Contoh dari WAN adalah jaringan antarpulau atau antarnegara. Internet merupakan contoh dari sebuah WAN yang mencakup seluruh dunia.



Gambar 2.3 Wide Area Network

(Sumber :

<http://www.more.net/technical/netserv/diagrams/images/wanpoint>

.gif)

#### 2.1.1.4. *Wireless Network*

Menurut Arief ( 2006, p10), *Wireless Network* merupakan peralatan end-user untuk mengakses jaringan dengan menggunakan transmisi radio pendek atau sedang.

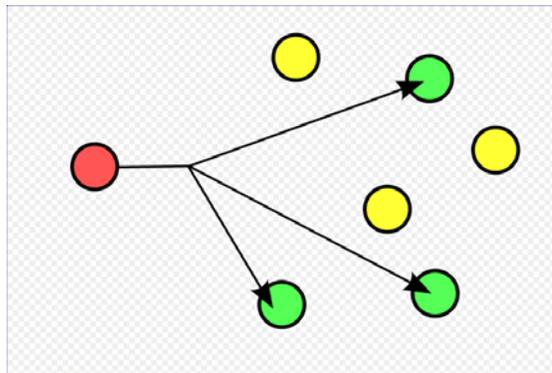
- *Wireless* WAN : GSM (sampai 20 Kbps).
- *Wireless* LAN/MAN : WaveLAN (2-11 Mbps, sampai 150 m).
- *Wireless* PAN (Personal Area Network) : bluetooth (sampai 2 Mbps, jarak < 10 m).

Tabel 2.1 *Range, bandwidth (Mbps), latency (ms)* pada Jaringan Komputer.

|                        | Range        | <i>Bandwidth</i><br>(Mbps) | <i>Latency</i> (ms) |
|------------------------|--------------|----------------------------|---------------------|
| LAN                    | 1-2 kms      | 10-1000                    | 1-10                |
| WAN                    | Worldwide    | 0.010-600                  | 100-500             |
| MAN                    | 2-50 kms     | 1-150                      | 10                  |
| <i>Wireless</i><br>LAN | 0,15-1.5 kms | 2-11                       | 5-20                |
| <i>Wireless</i><br>WAN | Worldwide    | 0.010-2                    | 100-500             |
| Internet               | Worldwide    | 0.010-2                    | 100-500             |

### 2.1.2. *Multicast*

Menurut Agung (2003, p4), *Multicast* adalah metode transmisi data secara *connectionless*, yang berarti *client* menerima aliran data tapi tidak terhubung secara langsung ke *server*. Tidak ada informasi kontrol dan feedback yang dikirim ke *server*. Metode ini menghemat *bandwidth* jaringan karena hanya satu aliran data yang dibangkitkan oleh *server*.



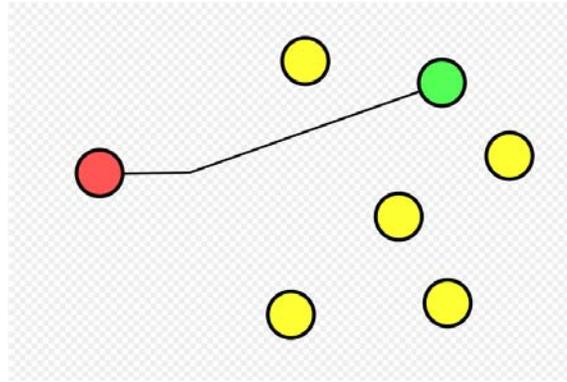
Gambar 2.4 *Multicast*

(Sumber : <http://en.wikipedia.org/wiki/Image:Multicast.svg>)

### 2.1.3. *Unicast*

*Unicast* adalah cara untuk pengiriman paket informasi pada suatu tujuan tunggal. *Unicast* biasa digunakan pada jasa *streaming* content provider. *Server unicast* menyediakan aliran data pada pengguna tunggal pada saat yang sama. Keuntungannya adalah adanya hubungan dua arah dengan *server*, sehingga memungkinkan mengirim informasi kontrol dan feedback ke *server* yang dapat digunakan untuk *error correction* dan adaptasi terhadap kondisi jaringan. Kerugiannya adalah jika *unicast*

*streaming* melayani *client* yang sangat banyak, akan mempengaruhi *bandwidth* yang digunakan (Anonymous, 2007).



Gambar 2.5 *Unicast*

(Sumber : <http://en.wikipedia.org/wiki/Image:Unicast.svg>)

Kelemahan dari metode ini adalah :

- Beban kerja host akan meningkat.
- Butuh *bandwidth* yang sangat besar untuk komunikasi multi-point yang melibatkan jumlah host yang besar.

Keuntungan dari metode ini adalah :

- Merupakan desain yang paling sederhana untuk diimplementasi.
- Tidak member beban pada host yang bukan merupakan tujuan datagram.

## 2.2. Referensi Model Jaringan

### 2.2.1. OSI Model

Menurut Stallings (2000, pp 35-38), arsitektur jaringan menurut OSI (*Open System Interconnection*) dibagi menjadi 7 *layer*, yaitu :

1. *Application*, berfungsi untuk menyediakan akses ke lingkungan OSI bagi user serta menyediakan layanan informasi terdistribusi.
2. *Presentation*, berfungsi untuk menyediakan keleluasaan terhadap proses aplikasi untuk bermacam-macam representasi data (sintaks).
3. *Session*, berfungsi untuk menyediakan struktur kontrol untuk komunikasi diantara aplikasi-aplikasi. Selain itu, *layer* ini juga berfungsi untuk menentukan, menyusun, mengatur, dan mengakhiri koneksi sesi diantara aplikasi-aplikasi yang sedang beroperasi.
4. *Transport*, berfungsi untuk menyediakan transfer data yang handal dan transparan diantara titik ujung dan juga menyediakan perbaikan *end-to-end error* dan *flow control*.
5. *Network*, berfungsi untuk melengkapi lapisan yang lebih tinggi dengan keleluasaan dari transmisi data dan teknologi-teknologi *switching* yang dipergunakan untuk menghubungkan sistem. Selain itu, *layer* ini bertugas untuk menyusun, mempertahankan, serta mengakhiri koneksi.
6. *Data-link*, berfungsi untuk menyediakan transfer informasi yang andal melewati *link* fisik dan mengirim blok (*frame*) dengan sinkronisasi yang diperlukan, *control error*, dan *flow control*.

7. *Physical*, berkaitan dengan transmisi, *bit stream* yang tidak terstruktur sepanjang *physical medium* dan berhubungan dengan karakteristik procedural, fungsi, elektrik, dan mekanis untuk mengakses *physical medium*.

### 2.2.2. TCP/IP Model

Menurut Greenlaw (1999, p92), TCP/IP merupakan protocol berorientasi koneksi. Arsitektur TCP/IP dibagi menjadi 4 layer, yaitu :

1. Layer 4 (*Application Layer*)

Lapisan ini bertanggung jawab dalam rangka menyediakan akses kepada aplikasi terhadap jaringan TCP/IP. Protokol-protokol yang berjalan pada lapisan ini adalah protokol Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Telnet, Simple Mail Transfer Protocol (SMTP), Simple *Network* Management Protocol (SNMP), dan lain-lain.

2. Layer 3 (*Transport layer*)

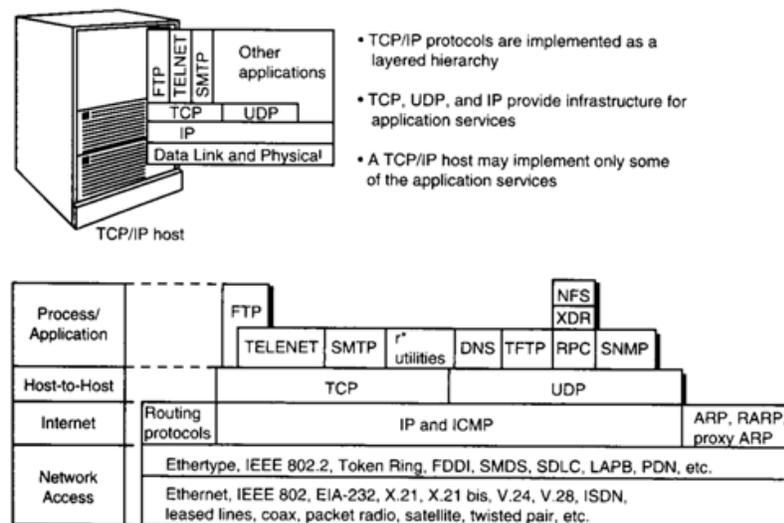
Lapisan ini bertanggung jawab dalam rangka membuat komunikasi antar dua host, dengan menggunakan cara membuat sebuah sesi connection-oriented atau menyebarkan sebuah connectionless broadcast. Protokol-protokol yang berjalan pada lapisan ini adalah protokol Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP).

### 3. Layer 2 (*Internet layer*)

Lapisan ini bertanggung jawab dalam melakukan routing dan pembuatan paket IP (dengan menggunakan teknik encapsulation). Protokol-protokol yang berjalan pada lapisan ini adalah Internet Protocol (IP), Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), serta Internet Group Management Protocol (IGMP).

### 4. Layer 1 (*Network access layer*)

Lapisan ini bertanggung jawab dalam meletakkan frame-frame data di atas media jaringan. Protokol yang berjalan dalam lapisan ini adalah beberapa arsitektur jaringan lokal (seperti halnya Ethernet atau Token Ring), serta layanan teknologi WAN (seperti Plain Old Telephone Service (POTS), Integrated Services Digital *Network* (ISDN), Frame Relay, dan Asynchronous Transfer Mode (ATM)).

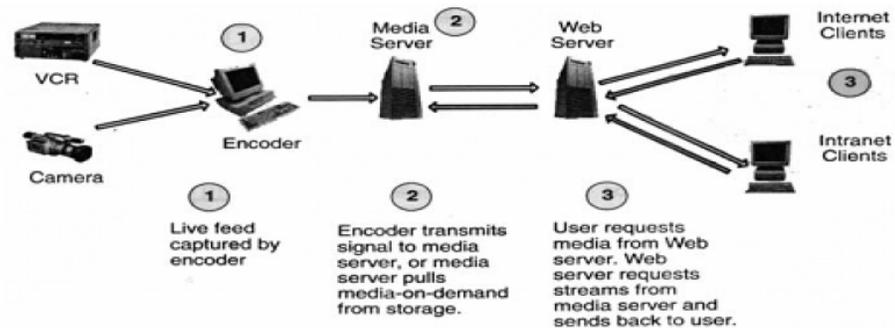


Gambar 2.6 Penerapan Protokol TCP/IP

### 2.3. Multimedia Streaming

*Streaming* adalah suatu teknologi yang mampu mengirimkan file audio dan video digital secara real time pada jaringan komputer (Austerberry, 2005, p7).

Dari sudut pandang prosesnya, *streaming* berarti sebuah teknologi pengiriman *file* dari *server* ke *client* melalui jaringan *packet-based* semisal Internet. *File* tersebut berupa serangkaian paket *time-stamped* yang disebut *stream* (Donny, 2002).



Gambar 2.7 Proses Pengiriman Data *Streaming*

Secara umum, arsitektur *streaming* terdapat 4 komponen (Austerberry, 2005, p138) yaitu :

1. *Capture* dan *encoding*.
2. Pelayanan oleh *server* (*serving*).
3. Distribusi pengiriman.
4. *Media player*.

*Capture* dan *encoding* adalah proses pengambilan data audio dan video dari mikropon dan kamera serta memprosesnya menjadi sebuah *file* yang terkompresi. *File* ini akan disimpan pada *server* penyimpanan data yang mempunyai perangkat lunak khusus untuk bisa mengontrol pengiriman data *stream* secara *real time*.

Pada proses *servicing*, *file* yang telah di-*encode*-kan di kirim ke *server* untuk didistribusikan melalui jaringan. *Server* pada *streaming* yang berbasis *web* memiliki dua fungsi pokok, pertama berfungsi sebagai *web server* yang mengatur komunikasi antara *client* dengan *server streaming*. Kedua, berfungsi untuk mengontrol pengiriman data *stream* melalui jaringan.

Jaringan distribusi menghubungkan video *player* dengan *server* yang beroperasi pada jaringan tersebut. Proses ini akan melibatkan banyak interkoneksi pada jaringan tersebut. Proses ini akan melibatkan banyak interkoneksi jaringan dan *buffer* pada *server*. Koneksi jaringan berasal dari jaringan LAN menuju jaringan Internet.

Video *player* biasanya sebagai bahan (*plug-in*) pada *web browser*, yang berfungsi untuk menerima data *stream* dan melakukan dekompresi kembali ke format *audio video* semula, dimana data tersebut agar bisa dijalankan.

### **2.3.1. Download dan Streaming**

#### **2.3.1.1. Download**

*Download* adalah proses menerima data (umumnya berbentuk berkas) dari sebuah sistem seperti *server web*, *FTP server*, *server mail* atau sistem serupa lainnya.

Kelebihan *download* :

- (+) *Download* dan simpan file dalam HD sehingga dapat dinikmati pada saat offline.
- (+) Dapat dilihat berkali-kali.
- (+) Standard file (bisa dibaca oleh semua jenis mesin).
- (+) Kualitas bagus.

Kekurangan *download* :

- (-) Waktu *download* lama.

### **2.3.1.2. Streaming**

Kelebihan *streaming* :

- (+) Dapat dilakukan pada bandwidth dengan kecepatan rendah.
- (+) *Web* master tidak perlu risau dengan bandwidth.
- (+) *Web* master tidak dibatasi oleh besar file.

Kekurangan *streaming* :

- (-) Hanya dapat dilihat pada saat online.
- (-) Kualitas gambar jelek.

## **2.3.2. Protokol Streaming**

### **2.3.2.1. Real-Time Streaming Protocol (RTSP)**

Menurut Agung (2003, p5), RTSP adalah protokol pada tingkat aplikasi untuk mengontrol penyampaian data secara *real-time*. Dengan RTSP *client* dapat mengontrol jalannya presentasi,

misalnya melakukan *rewind*, *play*, *fast-forward*, *pause*, *resume*, dan *stop* terhadap aliran data. Sumber aliran data dapat meliputi keduanya, *webcast* dan *on-demand*.

RTSP memiliki empat buah perintah. Perintah ini dikirim dari klien ke sebuah *server streaming* RTSP. Keempat perintah tersebut adalah :

1. Setup, *server* mengalokasikan sumber daya kepada sesi klien.
2. Play, *server* mengirim sebuah stream ke sesi klien yang telah dibangun dari perintah setup sebelumnya.
3. Pause, *server* menunda pengiriman stream namun tetap menjaga sumber daya yang telah dialokasikan.
4. Teardown, *server* memutuskan koneksi dan membebaskan tugas sumber daya yang sebelumnya telah digunakan.



Gambar 2.8 Finite-State Machine Yang Merepresentasikan RTSP

### 2.3.2.2. Real Time Messaging Protocol (RTMP)

Menurut Joachim Bauch (2007), RTMP adalah sebuah protokol yang dikembangkan oleh Adobe Systems untuk *streaming* audio, video dan data melalui Internet, antara Flash player dan *server*.

RTMP memiliki 3 variasi :

- Plain protokol berjalan pada tcp menggunakan port 1935.
- RTMPT yang di enkapsulasi dalam http meminta firewall traverse.
- RTMPS bekerja sama dengan RTMPT didalam HTTPs koneksi.

Contoh ActionScript untuk menghubungkan dan menjalankan *streaming* dengan RTMP :

```
var videoInstance: Video = your_video_instance;
var nc: NetConnection = baru NetConnection ();
var connected: Boolean = nc. menghubungkan ( "rtmp: / localhost / myapp");
var ns: NetStream = baru NetStream (nc);
videoInstance. attachVideo (ns);
ns. memutar ( "flvName");
```

default port 1935.

### 2.3.3. *Bandwidth*

Bandwidth adalah luas atau lebar cakupan frekuensi yang digunakan oleh sinyal dalam medium transmisi ( Brian, Stacey, 2005, p41).

Tabel 2.2 Unit *Bandwidth*

| Unit-unit Lebar Pita      | Akronim | Persamaan                                |
|---------------------------|---------|--|
| <i>Bit per second</i>     | bps     | 1 bps = ukuran dasar<br><i>bandwidth</i> |
| <i>Kilobit per second</i> | Kbps    | 1 Kbps = 1.000 bps                       |

|                           |      |                            |
|---------------------------|------|----------------------------|
| <i>Megabit per second</i> | Mbps | 1 Mbps = 1.000.000 bps     |
| <i>Gigabit per second</i> | Gbps | 1 Gbps = 1.000.000.000 bps |

*Bandwidth* sangat berpengaruh terhadap kualitas presentasi suatu data stream. Disamping kondisi jaringan juga mempengaruhi *bandwidth*, hal yang perlu diperhatikan adalah ukuran data *stream* harus sesuai dengan kapasitas *bandwidth* jaringan. Untuk mengatasinya, digunakan kompresi data dan penggunaan buffer (Agung, 2003, p8).

#### 2.3.4. Teknologi Video dan Audio Digital

Kekurangan utama video analog adalah penurunan kualitas ketika dilakukan penyimpanan atau produksi ulang, cacat seperti snow pada gambar karena noise atau efek interferensi. Sebaliknya, video digital merepresentasikan informasi sebagai angka digital, dibuat dengan sampling dan kuantisasi sinyal. Kualitas video digital dapat dipengaruhi 3 faktor :

1. *Frame Rate*, yaitu banyak gambar ditampilkan / detik pada sisi receiver (frame per second / fps). Frame rate digunakan untuk sinkronisasi gambar-suara dan kecepatannya tergantung pada kecepatan koneksi internet. Standar National Television Standards Committee (NTSC) untuk full motion video adalah 30 fps, program TV memiliki frame rate 24 fps dan video digital pada internet memiliki frame rate 15 - 30 fps.

2. *Color Depth*, yaitu banyak bit / pixel sebagai representasi warna, contoh, 24 bit merepresentasi 16.7 juta warna, 16 bit sekitar 65,535 warna, dan 8 bit hanya 256 warna.
3. *Frame Resolution*, yaitu besaran lebar dan tinggi dalam pixel, contoh, full screen PC display memiliki resolusi 640 x 480.

Variabel lain dalam video adalah Image Size, Bit Depth dan Data Rate / Bit Rate. Bit Depth adalah banyak warna pada gambar yang dapat disajikan pada waktu tertentu. Makin tinggi Bit Depth maka makin besar ukuran video.

Data Rate / Bit Rate adalah ukuran banyak data dikompres yang dikirimkan / detik (bits per second / bps), sangat dipengaruhi oleh Image size, Bit Depth dan Frames / Second. Penggunaan Bit Rate dan Codec yang tepat dapat menambah kualitas dari data *streaming*.

Bit Rate dapat ditentukan dengan strategi berikut :

1. *Constant Bit Rate / CBR* adalah strategi memaksimalkan kualitas visual video dengan mengkonstantakan bit rate / frame, buffer dan feedback pada encoding. Strategi ini digunakan pada real-time dan non-buffered video *streaming* karena ketersediaan *bandwidth* tetap namun kapasitas channel terbatas, contoh *video conferencing* pada channel dengan fixed *bandwidth*.
2. *Variable Bit Rate / VBR* adalah strategi memaksimalkan kualitas visual video dengan meminimalisir Bit Rate, karena hal ini berbanding lurus dengan distorsi yang terjadi pada setiap frame. Pada

fast motion scene, VBR menggunakan lebih banyak bit dari pada slow motion scene pada durasi yang sama untuk mengkonsistenkan visual quality. contoh DVD.

Seperti video, juga terdapat beberapa variabel yang mempengaruhi ukuran pada kompresi audio, yaitu :

1. *Sample Rate*, yaitu frekuensi bit data yang diambil secara acak (Kilohertz / KHz). Nilai umum untuk variabel ini adalah 11.025 kHz, 22.050 kHz, 44.1 kHz and 48 kHz. Makin tinggi sample rate maka makin baik kualitas audio, namun makin sulit mengatur sinkronisasi audio dan video.
2. *Sample Size*, yaitu banyak bit yang merepresentasikan amplitudo sampel audio. Nilai umum untuk audio codex menggunakan 16-bit - 8-bit sample size.
3. *Channel*, yaitu banyak jenis sinyal audio, Mono atau Stereo.

## **2.4. Perangkat Lunak**

### **2.4.1. Linux**

Linux adalah sistem operasi *open source* yang cepat dan stabil untuk komputer PC dan workstation yang mampu menyediakan layanan *internet* level profesional, peralatan pengembangan intensif, dipenuhi dengan fungsi *graphical user interfaces* (GUI), dan jumlah aplikasi yang banyak dari aplikasi *office* sampai multimedia Linux dikembangkan pada awal tahun 1990 oleh Linus Torvalds, bersama dengan para *programmer*

di seluruh dunia. Sebagai sebuah sistem operasi, linux menampilkan fungsi yang sama dengan Unix, Macintosh, Windows dan Windows NT. Meskipun begitu, Linux berbeda karena kemampuan dan fleksibilitasnya, dan juga karena tersedia secara gratis (Peterson, 2004).

#### **2.4.2. Java**

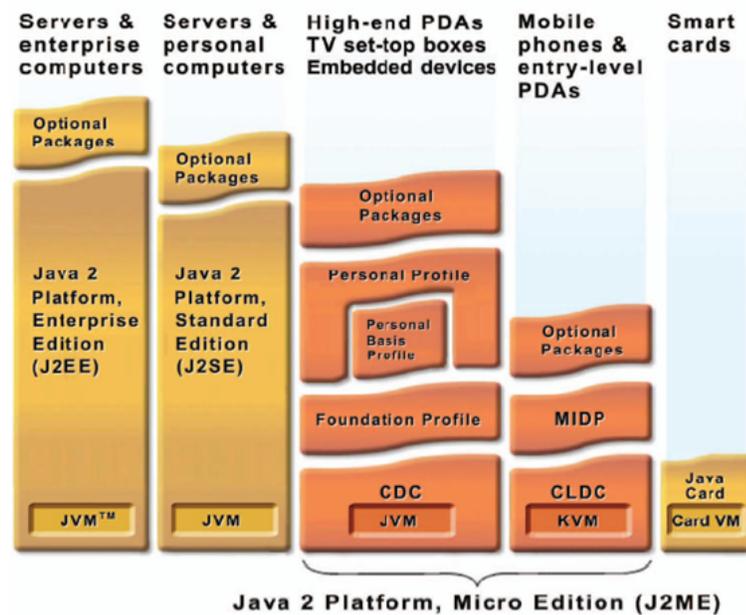
Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java merupakan bahasa pemrograman berorientasi objek dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat neutral architecture, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai Java Bytecode.

Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu :

- Java Virtual Machine (JVM).
- Java Application Programming Interface (Java API).

Sun membagi arsitektur Java menjadi tiga bagian, yaitu :

- Enterprise Java (J2EE) untuk aplikasi berbasis *web*, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan superset dari Standar Java.
- Standar Java (J2SE), ini adalah yang biasa dikenal sebagai bahasa Java.
- Micro Java (J2ME) merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk *wireless device* / mobile device.



Gambar 2.9 Java Platform

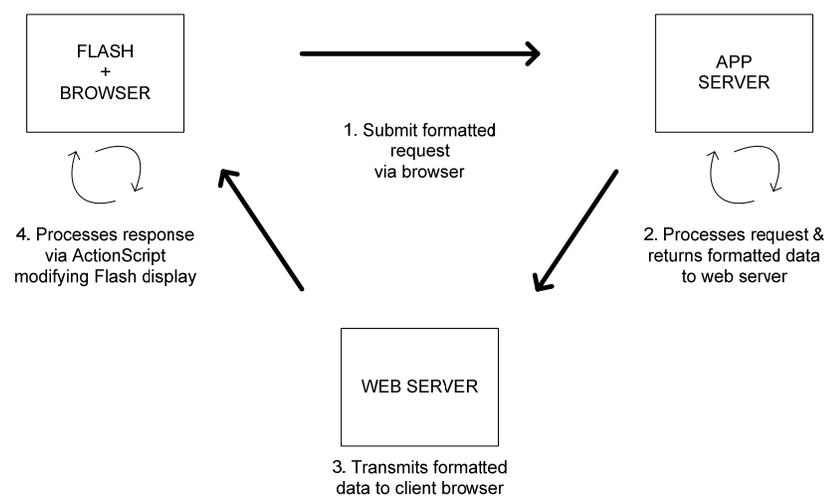
### 2.4.3. Apache Ant

Menurut Steve Loughran, Erik Hatcher (2007), Apache Ant adalah Java build tool yang disusun dalam format XML untuk melakukan proses kompilasi dan build class-class Java dan jar.

### 2.4.4. Flash

Menurut Wiwit (2006, p2), Flash adalah program animasi interaktif berbasis vector yang sering digunakan pada *web* desain. Flash memiliki kelebihan dari program animasi lainya karena adanya fasilitas *action script* sehingga animasi bisa menjadi lebih interaktif.

Out put dari Flash adalah dalam format flash movie file (\*.swf), dalam file ini sound dapat di kompresikan menjadi ukuran yang lebih kecil dengan format MP3 sehingga file flash tetap dalam ukuran kecil tanpa merubah kualitas suara.



Gambar 2.10 Aliran Informasi Aplikasi Flash

#### 2.4.5. *Red5 Server*

*Red5 Server* adalah sebuah aplikasi *server* yang memungkinkan pengiriman data aktual dari *client* ke *client* yang lainnya. *Red5 Server* menyediakan kemampuan media *streaming* yang powerfull dan lingkungan pengembangan yang fleksibel. *Red5 Server* merupakan open source Flash RTMP *server* berbasis Java yang mendukung :

- *Streaming* Audio/Video (FLV and MP3).
- Recording *Client* Streams (FLV only).
- Shared Objects.
- Live Stream Publishing.
- Remoting (AMF).

##### 2.4.5.1. *Komponen Red5 Server*

Dalam perancangan dan pengembangan ini, aplikasi dibangun dengan menggunakan aplikasi *Red5 Server* yang berfungsi sebagai *stream server* dan menangani aksi-aksi yang diberikan pada *client* yang terhubung dengan *Red5 Server*. Sedangkan dalam membangun aplikasi *client*-nya menggunakan *tools Macromedia Flash*, *Macromedia Dreamweaver*, dan fungsi-fungsi dari komponen *Red5 Server*.

Protokol HTTP hanya digunakan untuk memberikan akses kepada pengguna untuk memperoleh *file* aplikasi sehingga aplikasi dapat di jalankan pada *web browser*. Ketika aplikasi

*client* dijalankan, aplikasi akan terhubung dengan *Red5 Server* dan berkomunikasi dengan menggunakan protokol RTMP dalam mengakomodasikan hubungan antar aplikasi *client*.

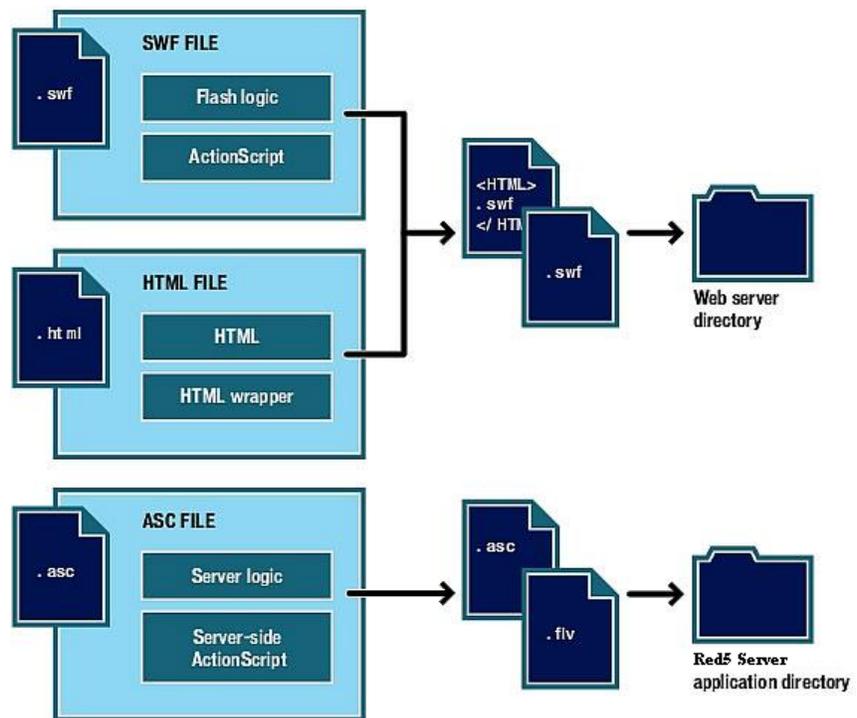
Hubungan antara *client (Flash Player)* dan *Red5 Server* menggunakan RTMP yang berbeda dari HTTP karena menyediakan sebuah koneksi secara terus-menerus untuk berkomunikasi dua arah antara *Flash Player* pada *Client* dengan *Red5 Server*.

Pada system *media streaming*, media melalui *Red5 Server* yang kemudian dikirimkan ke *Flash Player* pada *client*. Ketika proses *streaming*, *Flash Player* pada *client* menciptakan sebuah koneksi dengan *server* yang menyediakan sebuah aliran informasi, yang disebut dengan *Network Stream (NetStream)*.

#### **2.4.5.2. Prinsip kerja *Red5 Server***

Prinsip kerja *Red5 Server* dibangun dari dua bagian yang terpisah, dimana *file server (.ASC)* dirancang untuk mendukung tujuan dari komunikasi yang ingin dicapai sehingga mampu untuk mendukung tujuan dari komunikasi yang dicapai sehingga mampu merespon interaksi yang diberikan dari aplikasi *client* dan *file (.SWF)* merupakan aplikasi *flash* yang memberikan serta menyediakan tampilan layar untuk pengguna akhir. Komunikasi yang terjadi melalui aplikasi *Red5 Server* yang dikirimkan kepada aplikasi *client (Flash Player)*, ketika aplikasi *client* menggunakan

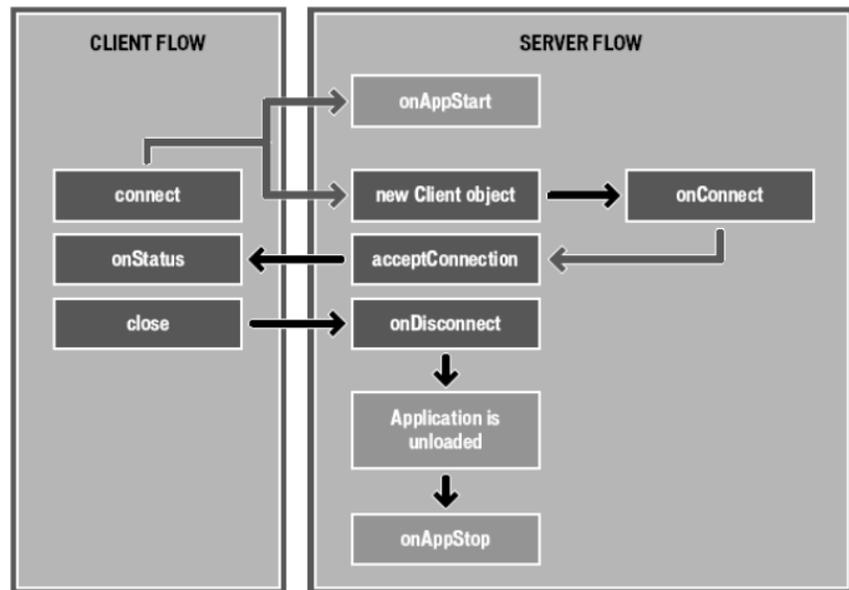
komponen *Red5 Server* dan hubungan terjalin dengan *stream server Red5 Server* untuk menyediakan aliran informasi yang diinginkan. Sedangkan *file HTML* digunakan sebagai penampung dari aplikasi *client (Flash Player)* untuk dijalankan. Seperti terlihat pada gambar dibawah ini



Gambar 2.11 Aliran Kerja *Red5 Server*

Ketika *client* terhubung dengan aplikasi *server*, *server* menggunakan *method onAppStart* jika aplikasi belum dijalankan, kemudian *method on Connect* dijalankan untuk menangani objek *client* yang tercipta. Secara logika *method* ini yang menentukan untuk menerima atau menolak koneksi aplikasi *client*. Yang

dikembalikan pada aplikasi *client* dengan menggunakan *method onStatus* untuk memberikan laporan mengenai koneksi apakah diterima atau ditolak. Ketika aplikasi *client* menutup koneksi, *method onDisconnect* pada aplikasi *server*. Seperti terlihat pada gambar dibawah ini.



Gambar 2.12 Aliran Kerja Aplikasi

#### 2.4.6. Helix Server

*Helix Server* adalah salah satu software streaming server ( streaming audio dan video ) dengan dukungan terhadap berbagai macam format dan beroperasi dengan optimum pada sistem operasi Linux dibandingkan dengan sistem operasi yang lain. *Helix Server* menggunakan protokol RTSP.

Keuntungan dari penggunaan *Helix Server* sebagai streaming server yaitu :

- Memberikan fleksibilitas untuk beradaptasi terhadap format-format baru di masa yang akan datang.
- Memberikan fleksibilitas untuk mengubah platform streaming tanpa mengganti infrastruktur dari waktu ke waktu
- Proxy *Helix Server* melindungi ip address internal user.

## **2.4.7. Pemrograman Web**

### **2.4.7.1. Web**

*Web* merupakan dokumen yang ada di internet, yang dapat diakses dengan mengetik URL (*Uniform Resource Locator*). Halaman *web* terbagi dua, yaitu statis dan dinamis. Perbedaannya adalah pada halaman statis, halaman *web* tersebut tidak akan berubah, semua *user* akan melihat halaman yang sama bagaimanapun cara mengaksesnya. Sedangkan halaman dinamis adalah halaman *web* yang dapat berubah sesuai permintaan *user*, perubahan tersebut dikarenakan pada halaman *web* tersebut telah diprogram untuk menghasilkan halaman sesuai *input user*. Ekstensi file-file tersebut umumnya adalah html, htm, asp, php, jsp, aspx, dan sebagainya.

#### **2.4.7.2. Web Server**

*web server* adalah perangkat lunak yang berjalan pada komputer yang dapat menerima permintaan dari *client* untuk melihat halaman *web* berbentuk HTML (*Hyper Markup Language*), dimana halaman *web* diakses menggunakan HTTP (*Hyperext Transfer Protocol*). Contoh sebuah *web server open source* adalah Apache pada *platform* Linux. Sebelum halaman *web* dapat di akses, *file* HTML harus disimpan pada *folder* direktori *root* (*/var/www/html*) terlebih dahulu. Kemudian alamat URL (*Uniform Resource Locater*) dimasukan dalam *web browser* untuk menampilkan halaman *web* yang disimpan tadi.

#### **2.4.7.3. Web Browser**

*Web browser* adalah sebuah perangkat lunak yang memungkinkan pengguna untuk menampilkan dan berinteraksi dengan *text*, gambar, dan informasi yang ada pada halaman *web* pada sebuah di *World Wide Web* (WWW) atau *Local Area network* (LAN). Sebagian *browser modern* dapat mempresentasikan multimedia, termasuk suara dan video, meskipun memerlukan beberapa *plug-in* untuk format tertentu (Sufandi, 2003).

### 2.4.8. Codec (Coder Decoder)

*Codec* digunakan untuk meng-*encode* dan meng-*decode* berbagai tipe data *file* dari sinyal analog ke sinyal digital dan sebaliknya. Proses ini mengkonversi sinyal audio menjadi data digital yang didapatkan (kompresi) untuk kemudian dikirim lewat jalur internet. Di titik lain, data dikembangkan lagi (dekompresi), dan diubah menjadi sinyal analog. *Codec* juga bekerja menggunakan algoritma tertentu untuk membantu memecah, mengurutkan, mengkompresi, dan merakit ulang audio data yang ditransmisikan.

#### 2.4.8.1. Format Video

*Video compression* berfungsi untuk mengurangi besarnya data yang dipakai untuk menampilkan video tanpa mengurangi mutu secara berlebihan. *Video compression* juga dapat mengurangi jumlah bit yang diperlukan untuk menyimpan atau mentransfer media digital. Video tekompresi dapat dikirimkan lebih “ekonomis” melalui *carrier* yang lebih kecil.

*Flash Video* (FLV) adalah bentuk format *file* yang digunakan untuk mengirimkan data video melalui Internet dengan menggunakan *Adobe Flash Player* (biasa dikenal dengan *Macromedia Flash Player*) versi 6, 7, 8, atau 9. Didalam *Flash Video* berisi (*embedded with*) *file SWF* (*shockwave file* milik Flash). *Flash Video* dapat dijalankan pada hampir semua system

operasi melalui *Adobe Flash Player* dan *web browser plugin* (Donaldarkula, 2007).

#### **2.4.8.2.Format Audio**

Format audio didalam *flash* video umumnya berbentuk .mp3. Sama halnya dengan codec untuk video, untuk audio *flash* video menggunakan audio *codec* milik *Nellymoser codec*.

*Nellymoser* adalah format audio *single – channel* (mono) untuk mengoptimalkan transmisi audio dengan bit rendah (low-bit rate). Pada proses pengkodean audio, data suara terdiri atas sebuah grup yang berisi *frame-frame* dengan 256 sample data. Tiap *frame* diubah menjadi beberapa frekuensi yang signifikan, dan hanya satu dari frekuensi signifikan dari setiap *frame* itu yang dikodekan (*encode*) sementara yang lainnya diabaikan (Donaldarkula, 2007).