BAB II

TINJAUAN PUSTAKA

2.1 Multimedia

2.1.1 Definisi Multimedia

Multimedia berasal dari kata MULTI yaitu banyak; bermacam-macam dan MEDIUM yaitu sesuatu yang dipakai untuk menyampaikan atau membawa sesuatu, sehingga multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk text, audio, grafik, animasi, dan video. [4]

Beberapa definisi menurut beberapa ahli :

1. Kombinasi dari komputer dan video *(Rosch, 1996)*
2. Kombinasi dari tiga elemen: suara, gambar, dan teks *(McComick, 1996)*
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa audio (suara, musik), animasi, video, teks, grafik dan gambar *(Turban dan kawan-kawan, 2002)*
4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan video *(Robin dan Linda, 2001).*
5. Multimedia dalam konteks komputer menurut Hofstetter 2001 adalah: pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, video, dengan menggunakan *tool* yang memungkinkan pemakai berinteraksi, berkreasi, dan berkomunikasi.

2.1.2 Sistem Multimedia

Sebuah sistem multimedia adalah setiap sistem yang mendukung lebih dari satu jenis media. Suatu sistem dapat dikatakan sebagai sistem multimedia apabila memenuhi syarat sebagai berikut, antara lain :

1. Kombinasi Media

   Sistem disebut sistem multimedia jika kedua jenis media (kontinu/ dikrit) dipakai. Contoh media diskrit : teks dan gambar, dan media kontinu adalah audio dan video.

2. Independence

   Aspek utama dari jenis media yang berbeda adalah keterkaitan antar media tersebut. Sistem disebut sistem multimedia jika tingkat ketergantungan/ keterkaitan antar media tersebut rendah.

3. Computer-supported Integration

   Sistem harus dapat melakukan pemrosesan yang dikontrol oleh komputer. Sistem dapat diprogram oleh sistem programmer/ user. Sistem Multimedia dapat dibagi menjadi :

   a. Sistem Multimedia *Stand Alone*

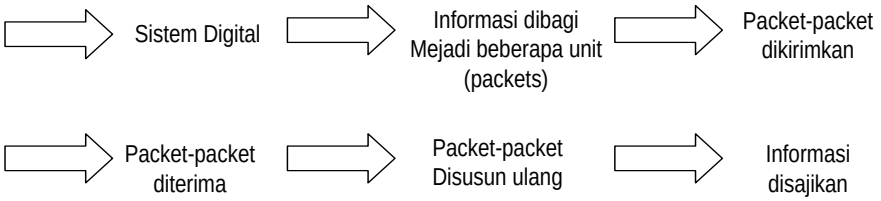
      Sistem ini berarti merupakan sistem komputer multimedia yang memiliki minimal *storage* (*harddisk*, *CD-ROM/DVD-ROM/CD-RW/DVD-RW*), alat input (*keyboard*, *mouse*, *scanner*, *mic*), dan output (*speaker*, monitor, *LCD* Proyektor), *VGA* dan *Soundcard*.

   b. Sistem Multimedia Berbasis Jaringan

      Sistem ini harus terhubung melalui jaringan yang mempunyai bandwidth yang besar. Perbedaannya adalah adanya *sharing* sistem

dan pengaksesan terhadap sumber daya yang sama. Contoh: video *converence* dan video *broadcast* Permasalahan: bila *bandwidth* kecil, maka akan terjadi kemacetan jaringan, *delay* dan masalah infrastruktur yang belum siap.

Dalam sistem multimedia terdistribusi (*data stream*), data di transmisikan (*time dependent*) dan terjadi pertukaran informasi



Gambar 2.1 Distribusi sistem multimedia

Transmisi informasi dapat dikategorikan :

1. Berdasarkan mode transmisi
   a. Aynschronous Trasmission Mode
      1. Komunikasi tanpa batas waktu.
      2. Packets mencapai penerima secepat mungkin.
      3. Paket yang dikirm cepat karena tidak perlu adanya sinkronisasi.
      4. Informasi untuk discrete media dapat ditransmisikan sebagai aynchronous data stream.
      5. Contoh : transmisi *e-mail*
   b. Synchronous Transmission Mode
      1. Terdapat batas waktu tunda maksimal untuk setiap paket dari suatu data *stream*.

2. Butuh sinkronisasi.
3. Penerima butuh buffer untuk menyimpan data sementara sambil menunggu paket lengkap.

c. Isochronous Transmission Mode

1. Terdapat batas waktu tunda maksimal dan minimal.
2. Melakukan garansi paket diterima dengan baik.
3. *Client* memberikan informasi kepada server tentang statusnya.
4. Membutuhkan buffer yang sangat besar.

2. Berdasarkan periode *streaming*

a. Strongly Periodic Stream

Interval waktu antara dua paket yang berurutan tetap.

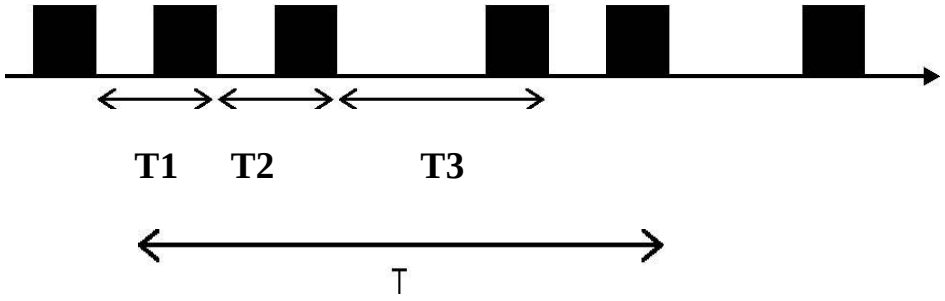
Contoh : PCM *coded*



Gambar 2.2 Strongly Periodic *Stream*

b. Weakly Periodic Stream

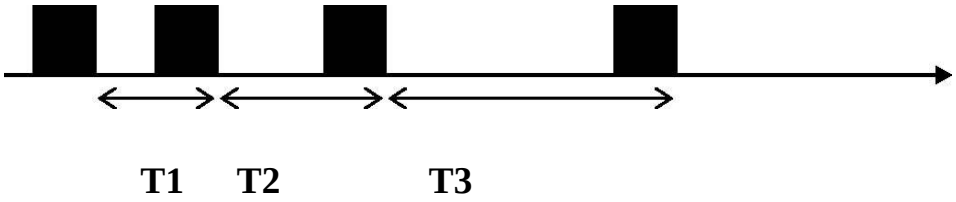
Interval waktu antara dua paket yang berurutan dapat dideskripsikan dengan fungsi periodik.



Gambar 2.3 Weakly Periodic *Stream*

c. Aperiodic Stream

Interval waktu tidak beraturan.



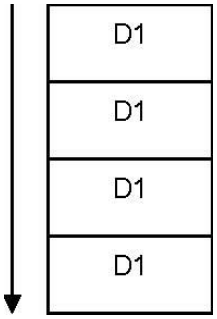
Gambar 2.4 Aperiodic *Stream*

3. Berdasarkan ukuran paket

a. Strongly Regular Stream

Ukuran paket konstan.

Contoh : *uncompressed* audio/video *stream.*

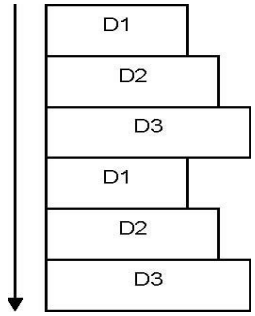


Gambar 2.5 Strongly Regular *Stream*

b. Weakly Regular Stream

Ukuran packet data berubah secara periodic.

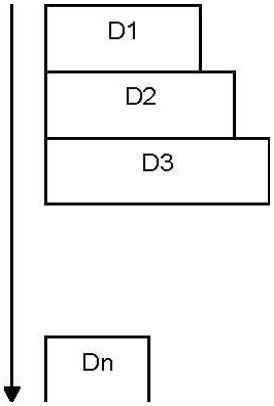
Contoh : MPEG



Gambar 2.6 Weakly Regular *Stream*

c. Irregular Data Stream

Ukuran packet data tidak tentu.



Gambar 2.7 Irregular Data *Stream*

2.2 Pengertian Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. [3]

Adapun pengertian lain, aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi *software* yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.3 Pengertian Data dan Informasi

Data dan Informasi mempunyai keterkaitan antara satu dengan yang lainnya. Data adalah fakta - fakta atau pesan - pesan yang belum dievaluasi atau bahan mentah dari informasi yang melalui pengolahan tertentu dibentuk menjadi suatu informasi. Data merupakan kenyataan yang menggambarkan suatu kejadian - kejadian nyata.

Suatu sistem yang kurang mendapatkan suatu informasi akan luruh, kerdil dan akhirnya berakhir. Definisi informasi sendiri seperti di bawah ini :

"*Informasi adalah data yang diolah menjadi bentuk yang lebih berharga dan lebih berguna serta lebih berarti bagi yang menerimannya*". [3]

Sedangkan sumber dari informasi sendiri adalah data yang berarti kenyataan untuk mengambarkan suatu kejadian – kejadian dan kesatuan yang nyata. Maksud dari kejadian – kejadian di sini adalah sesuatu yang telah terjadi pada saat tertentu. Sumber dari informasi adalah data. Data terbentuk dari karakter - karakter yang dapat berupa alfabet, angka maupun simbol khusus. Data disusun untuk diolah dalam bentuk struktur data, struktur file dan *database*. Terdapat perbedaan antara data dan informasi yaitu jika data merupakan bahan baku yang diolah untuk memberikan informasi, sedangkan informasi digunakan dalam pengambilan keputusan, karena itu informasi memiliki tingkat lebih tinggi dari data.

2.4 Video

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. [4]

Aplikasi video pada multimedia mencakup banyak aplikasi, antara lain:

1. *Entertainment : VCR/DVD recording*
2. *Interpersonal* : video *telephony, video conferencing*
3. *Interactive : windows*

Digital video adalah jenis sistem video recording yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi videonya. Biasanya digital video direkam dalam tape, kemudian didistribusikan melalui optical disc, misalnya VCD dan DVD. Salah satu alat yang dapat digunakan untuk menghasilkan video digital adalah camcorder, yang digunakan untuk merekam gambar-gambar video dan audio, sehingga sebuah camcorder akan terdiri dari kamera dan recorder. Macam-macam camcorder: miniDV, DVD camcorder, dan digital8. Camcorder terdiri dari 3 komponen:

1. Lensa, untuk mengatur banyak cahaya, *zoom*, dan kecepatan shutter.
2. Imager, untuk melakukan konversi cahaya ke sinyal electronic video.
3. Recorder, untuk menulis sinyal video ke media penyimpanan (seperti *magnetic videotape*)



mini-DV Camcorder

Gambar 2.8 Camcorder

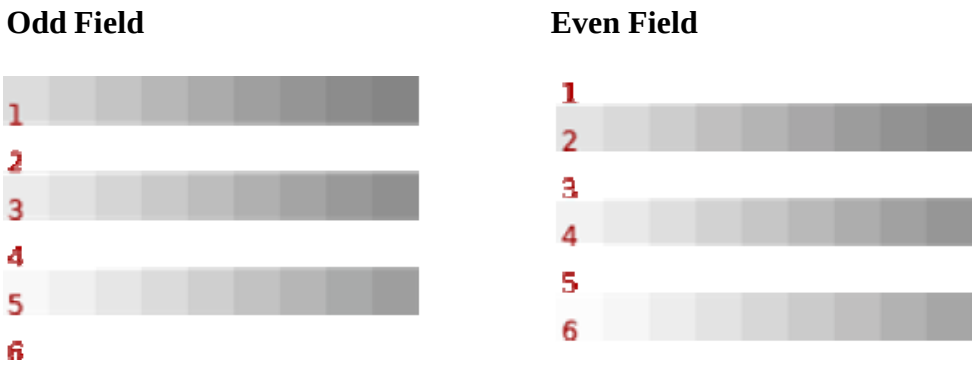
Video kamera menggunakan 2 teknik :

1. Interlaced

   Suatu metode untuk menampilkan image/gambar dalam *raster-scanned display device* seperti CRT televisi analog, yang ditampilkan

bergantian antara garis ganjil dan genap secara cepat untuk setiap frame. *Refresh rate* yang disarankan untuk metode *interlaced* adalah antara 50-80Hz. *Interlace* digunakan di sistem televisi analog yaitu :

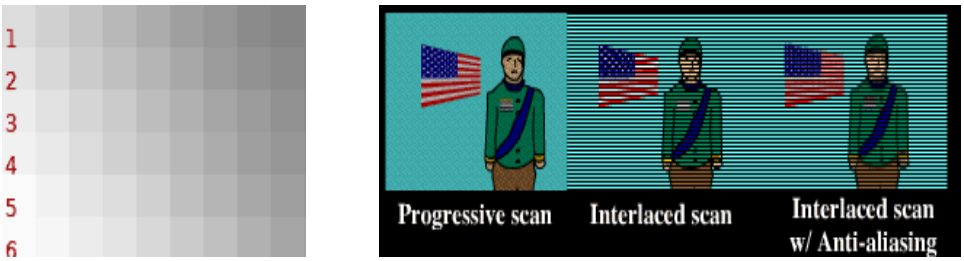
a. PAL (50 fields per second, 625 lines, even field drawn first)
b. SECAM (50 fields per second, 625 lines)
c. NTSC (59.94 fields per second, 525 lines, even field drawn first)



Gambar 2.9 Interlaced

2. Progressive scan

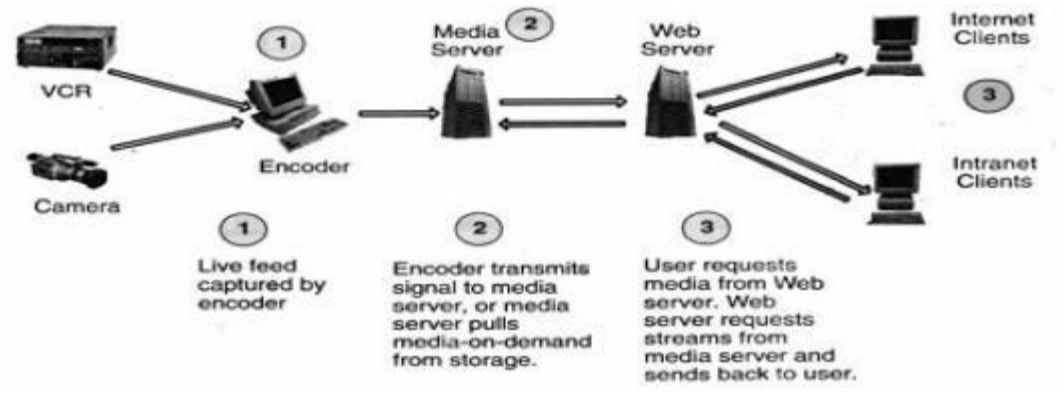
Suatu metode untuk menampilkan, menyimpan, dan memancarkan gambar dimana setiap baris untuk setiap *frame* digambar secara berurutan. Biasa digunakan pada CRT monitor komputer.



Gambar 2.10 Progressive Scan

2.5 Streaming

Streaming adalah suatu teknologi untuk memainkan *file* audio atau video secara langsung maupun dengan *prerecord* dari sebuah mesin *server* (*web server*). *File* audio atau video yang terletak pada sebuah *server* dapat secara langsung dijalankan pada komputer *client* sesaat setelah ada permintaan dari pengguna sehingga proses *download file* tersebut yang menghabiskan waktu cukup lama dapat dihindari. Saat *file* tersebut di-*stream* maka akan terbentuk sebuah *buffer* di komputer *client* dan data audio atau video tersebut akan mulai di-*download* ke dalam *buffer* yang telah terbentuk pada mesin *client*. Setelah buffer terisi dalam waktu hitungan detik, maka secara otomatis *file* video ataupun audio akan di jalankan oleh sistem. Sistem akan membaca informasi dari *buffer* sambil tetap melakukan proses *download file* sehingga proses *streaming* tetap berlangsung ke mesin *client*. [1]



Gambar 2.11 Konsep Video *Streaming*

Konsep dasar dari video *streaming* adalah membagi paket video ke dalam beberapa bagian, mentransmisikan pake tersebut, kemudian pada pihak penerima (*client*) dapat men-decode dan memainkan potongan paket *file* video tanpa harus menunggu seluruh *file* terkirim ke mesin penerima. Video atau audio dapat *diencode* untuk keperluan komunikasi secara *real time* atau dapat juga di *pre-encoded* dan disimpan dalam format CD-DVD untuk dijalankan pada saat dibutuhkan. Salah satu aplikasi yang membutuhkan real time *encoding* adalah *videophone* dan *video conferencing.* Sedangkan aplikasi yang membutuhkan *pre-encoded* antara lain DVD, VCD, yang dikenal dengan penyimpanan secara local atau *Video on Demand* (VoD), yang penyimpanannya dilakukan secara remote di *server* yang dikenal dengan *video streaming.*

Sebuah file video yang akan ditampilkan di user dapat menggunakan dua metode transfer file. Pertama, dengan mendownload file video tersebut dan yang kedua dengan melakukan proses *streaming*. Kedua metode ini memiliki keunggulan dan kekurangannya masing-masing. Sebuah file video yang diambil dari *server* dengan cara download tidak dapaat ditampilkan dalam sebuah file video tersebut selesai tersalin ke *buffer*. Metode ini memerlukan media penyimpanan yang cukup besar dan waktu yang diperlukan untuk proses *download* cukup lama karena *file video* biasanya berukuran besar. Metode kedua yang dapat digunakan adalah proses *streaming.* Metode ini berusaha untuk mengatasi masalah yang terdapat dalam metode *download*. Ide dasar dari *video streaming* ini adalah membagi paket video menjadi beberapa bagian, mentransmisikan paket *data* tersebut, kemudian penerima *(receiver)* dapat

mendecode dan memainkan potongan paket video tersebut tanpa harus menunggu keseluruhan file selesai terkirim ke mesin penerima.

Secara garis besar, konsep video *streaming* di bagi ke dalam tiga tahap, antara lain:

1. Mempartisi atau membagi data video yang telah terkompresi ke dalam paket - paket data.
2. Pengiriman paket - paket data video.
3. Pihak penerima (*client*) mulai men-decode dan menjalankan video walaupun paket data yang lain masih dalam proses pengiriman ke PC *client*.

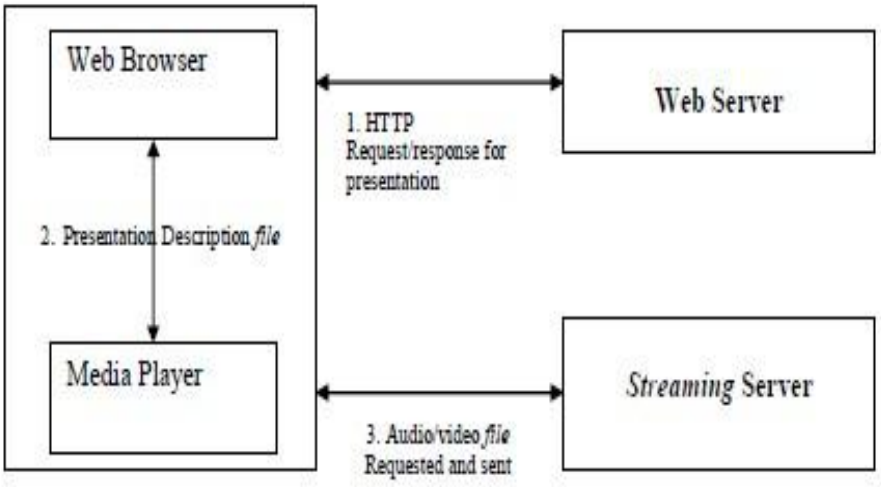
2.5.1 Format Streaming

Hingga kini setidaknya ada tiga jenis format *streaming* yang banyak digunakan diantaranya adalah untuk *Real Media* (**.rm / .ra / .ram**), *Windows Media* (**.asf / .wmf / .asx**), QuickTime (**.mov**) dan *Flash Video* (.**flv**). Masing-masing format tersebut memiliki kekurangan dan kelebihan sendiri-sendiri. Format *Real Media* dan *Windows Media* sangat handal di proses *streaming*, tetapi tidak terlalu handal untuk proses editing dan playback lokal sedangkan format QuickTime rata-rata cukup handal untuk proses *streaming*, editing dan playback lokal. Ketiga format tersebut membutuhkan sejenis player atau plug-in yang terinstal di komputer *client* agar dapat menikmati *streaming* yang ditawarkan suatu situs. Tentu saja, ketiga format tersebut tidak saling kompatibel dengan *player* yang bukan peruntukkannya. Biasanya di sebuah komputer akan terinstal

tiga *player* sekaligus, karena setiap situs di Internet belum tentu memilih format *streaming* yang digunakannya.

2.5.2 Arsitektur Streaming

Sistem *streaming* tersusun dari kombinasi *server , player, transmisi* dan metode *encoding* yang digunakan. Berikut ini bagian hubungan setiap komponen penyususun sistem *streaming*. Untuk *framework* aplikasi video *streaming* protokol pada layer network akan menggunakan model IP (Internet Protokol), sedangkan unutk layer transpot protokol yang dipakai merupakan gabungan antara TCP *(Transport Control Protokol)* dan UDP *(User Datagram Protocol).*



Gambar 2.12 Arsitektur *Streaming*

2.5.3 Metode Streaming

Dalam proses *streaming* terdapat beberapa metode, antara lain :

a. **Broadcast**

Broadcast adalah bentuk komunikasi *one to many* (yang pada dasarnya adalah *one to all*) yang paling dikenal, salah satu contohnya

adalah siaran TV. Dengan menggunakan *broadcast* maka informasi yang akan dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang tergabung dalam jaringan. Pada *broadcast* semua penerima mau tidak mau akan menerima informasi ini. [1]

b. **Multicast**

Multicast juga 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 diduplikasi dan dikirimkan ke setiap *clien*. Setiap *client* akan menerima *stream* yang sama dengan *client* lainnya. Salah satu contoh penggunaan *multicast* adalah *live video*. [1]

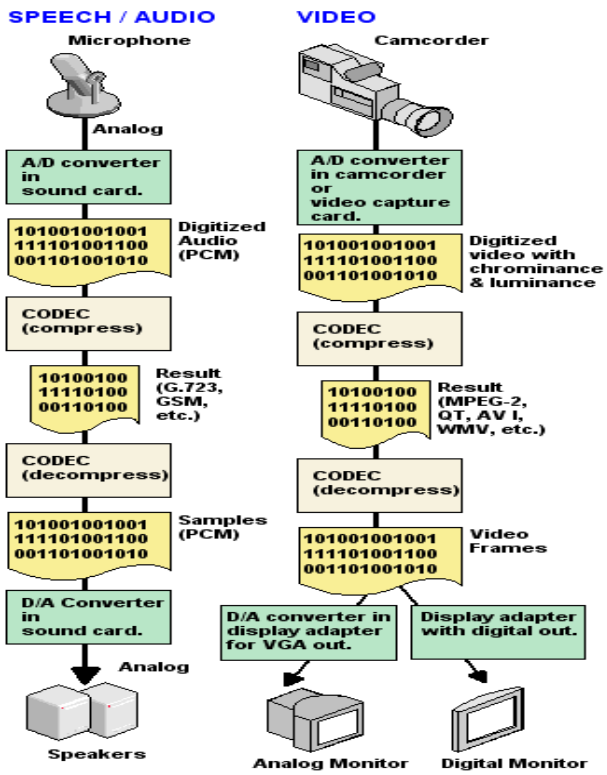
c. **Unicast**

Unicast merupakan bentuk komunikasi *one to one* atau *point to point*. Pada unicast informasi yang dikirimkan berasal dari satu sumber atau titik ke satu titik lainnya. *Server* akan mengirimkan file *streaming* ke komputer *client* berkali-kali bergantung pada banyaknya jumlah permintaan. Setiap *client* akan menerima file *streaming* yang terpisah dari *client* lainnya. Contoh penggunaan *unicast* adalah *video on demand*. [1]

2.6 Codec

Codec adalah kependekan dari *compression/decompression*, dalam komunikasi dan pemrosesan informasi, pengkodean atau penyandian (*encoding*) adalah proses konversi informasi dari suatu sumber (objek) menjadi data, yang

selanjutnya dikirimkan ke penerima atau pengamat, seperti pada sistem pemrosesan data. Pengawakodean atau pengawasandian (*decoding*) adalah proses kebalikannya, yaitu konversi data yang telah dikirimkan oleh sumber menjadi informasi yang dimengerti oleh penerima. Dalam konteks *streaming*, *codec* merupakan metode atau algoritma yang terdapat pada sebuah *streaming* player yang fungsinya adalah untuk melakukan proses pengkompresan dan pendekompresan *file* media *streaming*. Codec mengkompresi *file* media baik video maupun audio agar ukurannya dapat diperkecil, lalu *file* tersebut di*streaming* dan dibroadcast melalui Internet atau intranet. Setelah sampai ke komputer *client*, *file* tersebut kemudian didekompres ke ukuran asal untuk dapat didengarkan atau ditonton. Proses ini memungkinkan kita untuk dapat menikmati media *streaming* dengan lebih cepat.



Gambar 2.13 Proses Pengkodean

2.7 Webcam

Webcam (singkatan dari *web camera*) adalah sebutan bagi kamera *real-time* (keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui *World Wide Web*, program *instant messaging*, atau aplikasi *video call*. Istilah webcam merujuk pada teknologi secara umumnya, sehingga kata web kadang-kadang diganti dengan kata lain yang mendeskripsikan pemandangan yang ditampilkan di kamera, misalnya *StreetCam* yang memperlihatkan pemandangan jalan. Ada juga *MetroCam* yang memperlihatkan pemandangan panorama kota dan pedesaan, *TraffiCam* yang digunakan untuk memonitor keadaan jalan raya, cuaca dengan *WeatherCam*, bahkan keadaan gunung berapi dengan *VolcanoCam*. *Webcam* atau web kamera adalah sebuah kamera video digital kecil yang dihubungkan ke komputer melalui (biasanya) *port* USB ataupun *port* COM.



Gambar 2.14 Webcam

2.8 Real-Time Transport Protocol (RTP)

Real-Time Transport Protokol merupakan standar format paket untuk mengirimkan *file audio* dan *video* pada internet. RTP dikembangkan oleh *Audio-Video Transport Working Group* pada IETF dan pertama kali keluar pada tahun 1996 sebagai RFC 1889 dan diperbaharui lagi pada tahun 2003 sebagai RFC 3550. RTP berfungsi untuk transmisi *real-time* seperti *audio, video*, multimedia secara *end-to-end*. RTP mendukung transmisi *unicast, broadcast* dan *multicast*. RTP hanyalah sebagai *protocol transport* dan hal ini tidak menjamin *QoS* untuk layanannya. Biasanya RTP memiliki port tertentu untuk melakukan pengiriman *payload* data. Walaupun tidak ada standar yang menyatakan secara pasti, RTP biasanya dikenali pada port 16384-32767. [6]

Menurut RFC 1889, layanan RTP termasuk:

1. *Payload-tipe Identification* – Indikasi yang menyatakan jenis *content* / isi yang dibawa.
2. Nomor sekuensial – Nomor Sekuensial *PDU* (*Protocol Data Unit*).
3. *Time stamp* – Waktu yang digunakan *content* yang dibawa oleh *PDU*.
4. *Delivery monitoring*.

2.8.1 Karakteristik

1. Standar untuk transportasi real-time melalui jaringan IP yaitu untuk *streaming* video dan *Voice over IP*.
2. *Streaming server* memotong media data untuk masing-masing track menjadi paket RTP yang kemudian ditransmisikan melalui UDP.

3. Kemampuan berinteraksi dengan aplikasi lainnya melalui suatu protokol yang disetujui bersama lewat bermacam-macam jalur komunikasi (interoperabilitas), biasanya lewat jaringan TCP/IP dan protokol HTTP.
4. RTP tidak menyediakan mekanisme untuk menjamin pengiriman tepat waktu atau memberikan jaminan pada QoS (*Quality of Service*).
5. RTP mendukung transfer data ke beberapa tujuan dengan menggunakan distribusi *multicast* jika disediakan oleh jaringan yang mendasarinya., tapi bergantung pada layanan pada lapisan lebih rendah untuk melakukannya.
6. RTP tidak menjamin pengiriman atau mencegah kegagalan pada proses pengiriman, tidak menganggap bahwa jaringan yang mendasari dapat diandalkan dan memberikan paket secara berurutan.

2.8.2 Komponen Protokol

1. Payload Format

a. Memberikan lapisan adaptasi antara *codec* tertentu.

b. Dioptimalkan untuk *packet loss.*

c. Banyak format payload, antara lain H.261, H.263, M-JPEG, MPEG-2, MPEG-4, BT.656, dan SMPTE-292.

2. RTP Profile

a. *Broadcast* dari satu sumber.

b. Operasi channel satu arah.

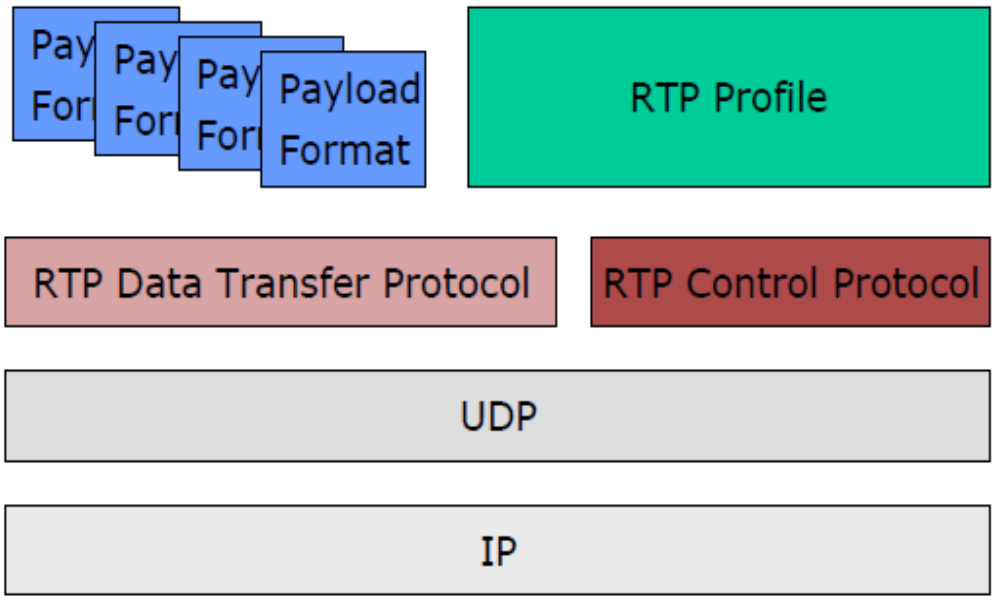
c. Menyediakan nama untuk format *payload.*

3. RTP Data Transfer Protocol

a. Angkutan unit data pada aplikasi berupa audio dan video.

b. Satu RTP stream mengangkut setiap jenis media.

c. Memberikan aliran media tunggal dari pengirim ke satu atau lebih penerima.

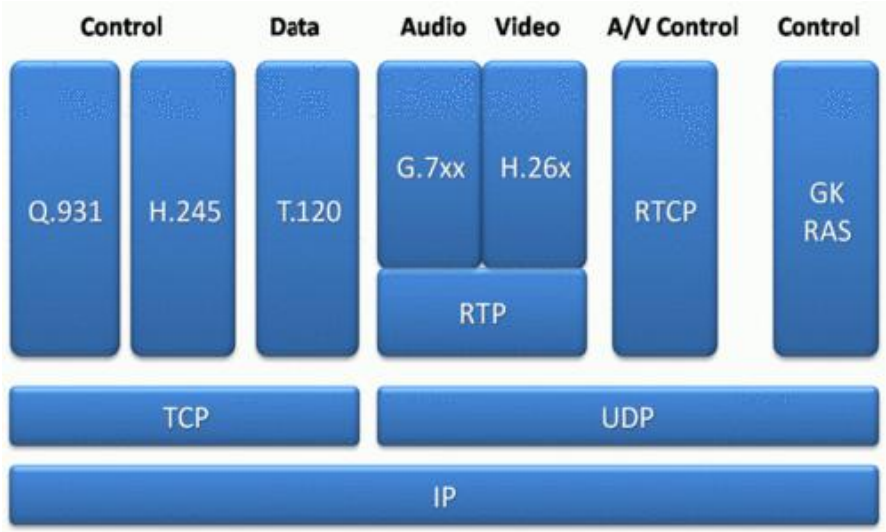


Gambar 2.15 Komponen Protokol RTP

*Real-time Transport Protocol (*RTP*)* mendefinisikan format paket standar untuk mengirimkan audio atau video melalui jaringan IP (internet). RTP digunakan secara luas dalam sistem komunikasi dan hiburan yang melibatkan media *streaming*, seperti telepon, aplikasi video dan *teleconference* berbasis web. RTP digunakan bersama dengan RTP *Control Protocol (*RTCP*),* Sementara RTP membawa media stream (misalnya audio atau video), RTCP digunakan untuk memantau statistik transmisi dan kualitas layanan *(QoS)* dan alat bantu sinkronisasi beberapa aliran.

RTP digunakan bersama dengan protokol lain seperti H.323 dan RTSP. Standar RTP mendefinisikan sepasang protokol, RTP dan RTCP. RTP digunakan untuk transfer data multimedia, dan RTCP digunakan untuk secara periodik mengirim informasi kontrol dan parameter QoS. RTP didesain untuk *end-to-end*, *real-time*, dan *transfer data stream*. RTP dianggap sebagai standar utama untuk audio / video dalam jaringan *transport* IP dan digunakan dengan profil yang terkait dan format *payload*.

Aplikasi multimedia *real-time streaming* memerlukan pengiriman informasi secara tepat waktu dan dapat mentolerir hilangnya beberapa paket untuk mencapai tujuan ini. Protokol *Transmission Control Protocol (TCP)*, meskipun standar untuk digunakan RTP, tidak biasanya digunakan dalam aplikasi RTP karena kehandalan TCP lebih kepada ketepatan waktu, sehingga mayoritas implementasi RTP dibangun pada *User Datagram Protocol (*UDP*)*.



Gambar 2.16 Susunan Protokol

RTP dimaksudkan untuk memberikan informasi yang diperlukan oleh aplikasi tertentu dan sering akan diintegrasikan ke dalam proses aplikasi bukan

diimplementasikan sebagai lapisan yang terpisah. RTP adalah suatu kerangka kerja protokol yang sengaja tidak lengkap. Tidak seperti protokol konvensional dimana fungsi tambahan mungkin akan diakomodasi dengan membuat protokol yang lebih umum atau dengan menambahkan mekanisme pilihan yang akan memerlukan parsing. RTP dimaksudkan untuk disesuaikan melalui modifikasi dan / atau penambahan pada header yang diperlukan. Sejauh ini, kita telah mengasumsikan bahwa semua situs ingin menerima data media dalam format yang sama. Namun, hal ini tidak selalu tepat. Pertimbangkan kasus dimana pengguna dalam satu area yang terhubung melalui koneksi dengan kecepatan rendah dan sebagian besar pengguna yang menikmati akses jaringan berkecepatan tinggi, memaksa semua orang untuk menggunakan bandwidth rendah, sehingga kualitas video menjadi berkurang.

Untuk pengolahan protokol yang efisien, jumlah titik *multiplexing* harus diminimalkan, seperti dijelaskan dalam prinsip pengolahan lapisan desain terpadu. Dalam RTP, *multiplexing* disediakan oleh alamat transportasi tujuan (jaringan alamat dan nomor port) yang berbeda untuk setiap sesi RTP. Misalnya, dalam teleconference terdiri dari media audio dan video dikodekan secara terpisah, setiap media harus dilakukan pemisahan alamat tranportasi tujuan dalam sesi RTP. Pemisahan audio dan video *stream* tidak harus dilakukan dalam sesi RTP tunggal dan *demultiplexed* berdasarkan jenis atau bidang.

2.9 FFMPEG (Fast Forward Motion Picture Expert Group)

FFMPEG adalah konverter audio dan video yang sangat cepat dan juga bisa diambil dari sumber audio / video secara langsung. Hal ini juga dapat mengkonversi video dan mengubah ukuran dengan cepat menggunakan filter berkualitas tinggi. Antarmuka baris perintah ini dirancang untuk menjadi intuitif, dalam arti bahwa *ffmpeg* mencoba untuk mengetahui semua parameter yang mungkin dapat diturunkan secara otomatis. Biasanya hanya perlu menentukan target *bitrate* yang inginkan. Sebagai aturan umum, pilihan yang diterapkan ke file yang ditentukan berikutnya. Oleh karena itu, ketertiban adalah penting, dan dapat memiliki pilihan yang sama pada baris perintah beberapa kali. Setiap kejadian ini kemudian diterapkan pada masukan berikutnya atau *output* file. [7]

2.10 Internet

Internet (kependekan dari *interconnected-networking*) ialah sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *Internet Protocol Suite* (TCP/IP) untuk melayani miliaran pengguna di seluruh dunia. Manakala Internet ialah sistem komputer umum, yang berhubung secara global dan menggunakan TCP/IP sebagai protokol pertukaran paket (*packet switching communication protocol*). Rangkaian internet yang terbesar dinamakan Internet. Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internetworking*. Internet dijaga oleh perjanjian bilateral atau multilateral dan spesifikasi teknikal (protokol yang menerangkan tentang perpindahan data antara rangkaian). Protokol-protokol ini dibentuk berdasarkan perbincangan *Internet Engineering Task Force (*IETF*)* yang terbuka kepada umum.

Jumlah pengguna Internet yang besar dan semakin berkembang, telah mewujudkan budaya Internet. Internet juga mempunyai pengaruh yang besar atas ilmu, dan pandangan dunia. Dengan hanya berpandukan mesin pencari seperti Google, pengguna di seluruh dunia mempunyai akses Internet yang mudah atas bermacam-macam informasi. Dibanding dengan buku dan perpustakaan, Internet melambangkan penyebaran (*decentralization*) atau pengetahuan (*knowledge*) informasi dan data secara ekstrim.

2.11 Jaringan Komputer

2.11.1 Pengertian

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer, software dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan yang sama. [5]

Tujuan dari jaringan komputer adalah:

1. Membagi sumber daya: contohnya berbagi pemakaian printer, CPU, memori, harddisk
2. Komunikasi: contohnya surat elektronik, *instant messaging*, *chatting*
3. Akses informasi contohnya *web browsing*

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/mengirim layanan disebut pelayan (*server*). Arsitektur ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

2.11.2 Klasifikasi Jaringan

Berdasarkan skala jaringan komputer dapat diklasifikasikan antara lain :

1. Local Area Network (LAN): suatu jaringan komputer yang menghubungkan suatu komputer dengan komputer lain dengan jarak yang terbatas.
2. Metropolitant Area Network (MAN): prinsip sama dengan LAN, hanya saja jaraknya lebih luas, yaitu 10-50 km.
3. Wide Area Network (WAN): jaraknya antar kota, negara, dan benua. ini sama dengan internet.

Pada dasarnya setiap jaringan komputer ada yang berfungsi sebagai *client* dan juga *server*. Tetapi ada jaringan yang memiliki komputer yang khusus didedikasikan sebagai *server* sedangkan yang lain sebagai *client*. Ada juga yang tidak memiliki komputer yang khusus berfungsi sebagai *server* saja. Karena itu berdasarkan fungsinya maka ada dua jenis jaringan komputer yaitu :

a. Client-server

Yaitu jaringan komputer dengan komputer yang didedikasikan khusus sebagai *server*. Sebuah service/layanan bisa diberikan oleh sebuah komputer atau lebih. Contohnya adalah sebuah domain seperti *www.youtube.com* yang dilayani oleh banyak komputer *web server*. Atau bisa juga banyak service/layanan yang diberikan oleh satu komputer.

b. Peer-to-peer

Yaitu jaringan komputer dimana setiap host dapat menjadi *server* dan juga menjadi *client* secara bersamaan. Contohnya dalam file sharing antar komputer di Jaringan Windows Network Neighbourhood ada 5 komputer (kita beri nama A,B,C,D dan E) yang memberi hak akses

terhadap file yang dimilikinya. Pada satu saat A mengakses file share dari B bernama data_nilai.xls dan juga memberi akses file soal_uas.doc kepada C. Saat A mengakses file dari B maka A berfungsi sebagai *client* dan saat A memberi akses file kepada C maka A berfungsi sebagai *server*. Kedua fungsi itu dilakukan oleh A secara bersamaan maka jaringan seperti ini dinamakan peer to peer.

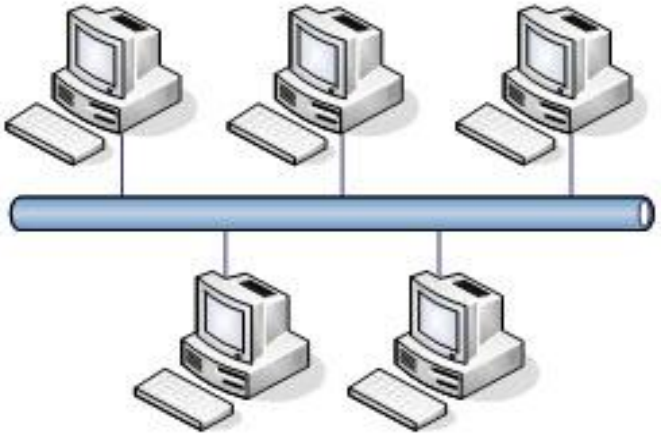
2.11.3 Topologi Jaringan

Arsitektur topologi merupakan bentuk koneksi fisik untuk menghubungkan setiap node pada sebuah jaringan. Pada sistem LAN terdapat tiga topologi utama yang paling sering digunakan: bus, star, dan ring. Topologi jaringan ini kemudian berkembang menjadi topologi tree dan mesh yang merupakan kombinasi dari star, mesh, dan bus. Dengan populernya teknologi nirkabel dewasa ini maka lahir pula satu topologi baru yaitu topologi wireless. Berikut topologi-topologi yang dimaksud : [8]

1. Topologi Bus

Topologi bus ini sering juga disebut sebagai topologi backbone, dimana ada sebuah kabel coaxial yang dibentang kemudian beberapa komputer dihubungkan pada kabel tersebut. Ciri-ciri topologi bus antara lain :

a. Secara sederhana pada topologi bus, satu kabel media transmisi dibentang dari ujung ke ujung, kemudian kedua ujung ditutup dengan “terminator” atau terminating-resistance (biasanya berupa tahanan listrik sekitar 60 ohm).



Gambar 2.17 Prinsip Topologi Bus

b. Pada titik tertentu diadakan sambungan (tap) untuk setiap terminal.

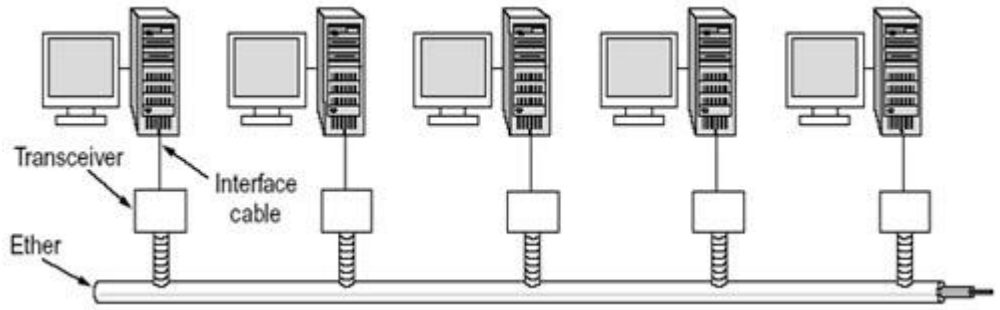
c. Wujud dari tap ini bisa berupa "kabel *transceiver*" bila digunakan "*thick coax*" sebagai media transmisi.

d. Atau berupa "BNC T-connector" bila digunakan "thin coax" sebagai media transmisi.

e. Atau berupa konektor "RJ-45" dan "hub" bila digunakan kabel UTP.

f. Transmisi data dalam kabel bersifat "*full duplex*", dan sifatnya "*broadcast*", semua terminal bisa menerima transmisi data.

g. Suatu protokol akan mengatur transmisi dan penerimaan data, yaitu Protokol Ethernet atau *CSMA/CD*.



Gambar 2.18 Koneksi kabel-transceiver pada topologi Bus

h. Pemakaian kabel coax (10Base5 dan 10Base2) telah distandarisasi dalam IEEE 802.3, yaitu sbb:

Tabel 2.1 Karakteritik Kabel Coaxial

	10Base5	**10Base2**
Rate Data	10 Mbps	10 Mbps
Panjang / segmen	500 m	185 m
Rentang Max	2500 m	1000 m
Tap / segmen	100	30
Jarak per Tap	2.5 m	0.5 m
Diameter kabel	1 cm	0.5 cm

i. Melihat bahwa pada setiap segmen (bentang) kabel ada batasnya maka diperlukan “Repeater” untuk menyambungkan segmen-segmen kabel.

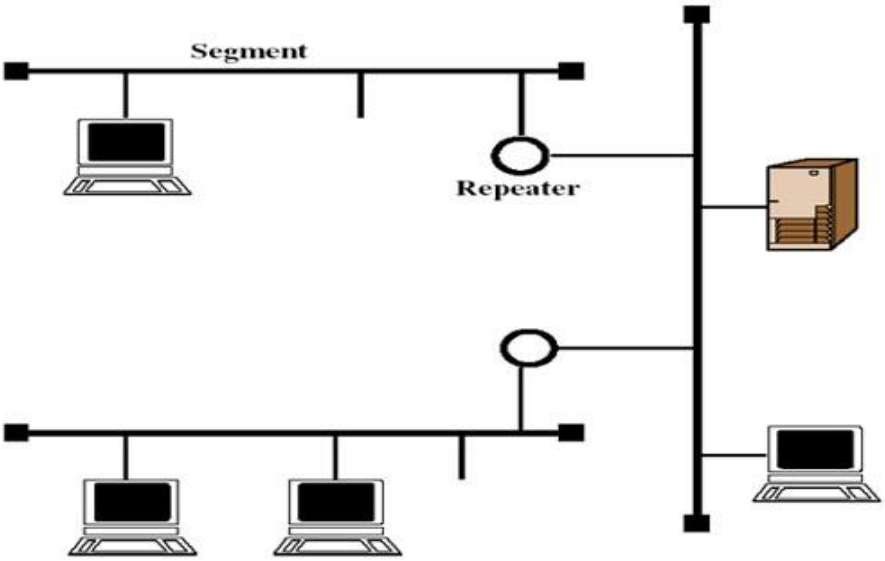
j. Kelebihan topologi Bus yaitu :

1. Instalasi relatif lebih murah
2. Kerusakan satu komputer *client* tidak akan mempengaruhi komunikasi antar *client* lainnya
3. Biaya relatif lebih murah

k. Kelemahan topologi Bus yaitu :

1. Jika kabel utama (bus) atau backbone putus maka komunikasi gagal.

2. Bila kabel utama sangat panjang maka pencarian gangguan menjadi sulit.
3. Kemungkinan akan terjadi tabrakan data (data collision) apabila banyak *client* yang mengirim pesan dan ini akan menurunkan kecepatan komunikasi.



Gambar 2.19 Perluasan topologi Bus menggunakan *repeater*

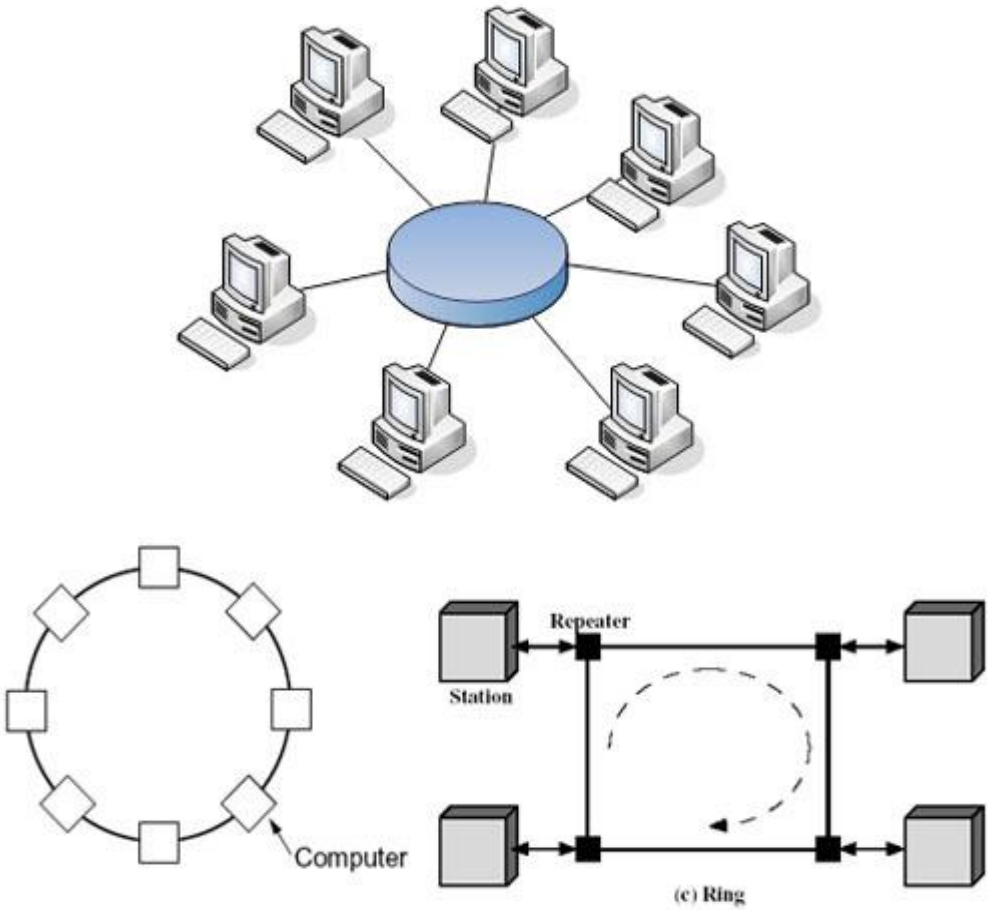
2. Topologi Ring (Cincin)

Topologi ring biasa juga disebut sebagai topologi cincin karena bentuknya seperti cincing yang melingkar. Semua komputer dalam jaringan akan di hubungkan pada sebuah cincin. Cincin ini hampir sama fungsinya dengan *concenrator* pada topologi star yang menjadi pusat berkumpulnya ujung kabel dari setiap komputer yang terhubung. Secara lebih sederhana lagi topologi cincin merupakan untaian media transmisi dari satu terminal ke terminal lainnya hingga membentuk suatu lingkaran, dimana jalur transmisi hanya “satu arah”.

Tiga fungsi yang diperlukan dalam topologi cincin : penyelipan data, penerimaan data, dan pemindahan data.

a. Penyelipan data adalah proses dimana data dimasukkan kedalam saluran transmisi oleh terminal pengirim setelah diberi alamat dan bit-bit tambahan lainnya.
b. Penerimaan data adalah proses ketika terminal yang dituju telah mengambil data dari saluran, yaitu dengan cara membandingkan alamat yang ada pada paket data dengan alamat terminal itu sendiri. Apabila alamat tersebut sama maka data kiriman disalin.
c. Pemindahan data adalah proses dimana kiriman data diambil kembali oleh terminal pengirim karena tidak ada terminal yang menerimanya (mungkin akibat salah alamat). Jika data tidak diambil kembali maka data ini akan berputar-putar dalama saluran. Pada jaringan bus hal ini tidak akan terjadi karena kiriman akan diserap oleh “terminator”.
d. Pada hakekatnya setiap terminal dalam jaringan cincin adalah “repeater”, dan mampu melakukan ketiga fungsi dari topologi cincin.
e. Sistem yang mengatur bagaimana komunikasi data berlangsung pada jaringan cincin sering disebut *token-ring*.
f. Kemungkinan permasalahan yang bisa timbul dalam jaringan cincin adalah:
    1. Kegagalan satu terminal / repeater akan memutuskan komunikasi ke semua terminal.

2. Pemasangan terminal baru menyebabkan gangguan terhadap jaringan, terminal baru harus mengenal dan dihubungkan dengan kedua terminal tetangganya.

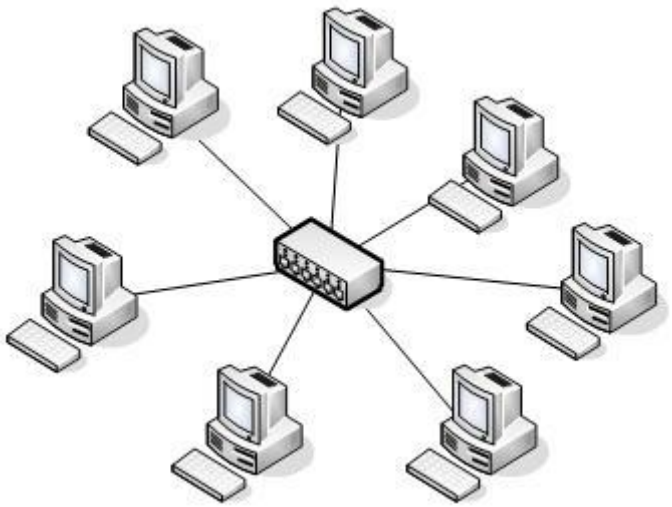


Gambar 2.20 Prinsip Koneksi Topologi Ring

3. Topologi Star (Bintang)

Disebut topologi star karena bentuknya seperti bintang, sebuah alat yang disebut *concentrator* bisa berupa hub atau switch menjadi pusat, dimana semua komputer dalam jaringan dihubungkan ke *concentrator* ini. Ciri-ciri topologi Star antara lain :

a. Pada topologi Bintang (Star) sebuah terminal pusat bertindak sebagai pengatur dan pengendali semua komunikasi yang terjadi. Terminal-terminal lainnya melalukan komunikasi melalui terminal pusat ini.
b. Terminal kontrol pusat bisa berupa sebuah komputer yang difungsikan sebagai pengendali tetapi bisa juga berupa "HUB" atau "MAU" (Multi Accsess Unit).



Gambar 2.21 Prinsip Koneksi Topologi Star

c. Terdapat dua alternatif untuk operasi simpul pusat.
    a. Simpul pusat beroperasi secara "broadcast" yang menyalurkan data ke seluruh arah. Pada operasi ini walaupun secara fisik kelihatan sebagai bintang namun secara logik sebenarnya beroperasi seperti bus. Alternatif ini menggunakan HUB.
    b. Simpul pusat beroperasi sebagai "switch", data kiriman diterima oleh simpul kemudian dikirim hanya ke terminal tujuan (bersifat point-to-point), akternatif ini menggunakan mau sebagai pengendali.

d. Bila menggunakan HUB maka secara fisik sebenarnya jaringan berbentuk topologi Bintang namun secara logis bertopologi Bus. Bila menggunakan maka baik fisik maupun logis bertopologi Bintang.

e. Kelebihan topologi bintang :

    1. Karena setiap komponen dihubungkan langsung ke simpul pusat maka pengelolaan menjadi mudah, kegagalan komunikasi mudah ditelusuri.
    2. Kegagalan pada satu komponen/terminal tidak mempengaruhi komunikasi terminal lain.

f. Kelemahan topologi bintang :

    1. Kegagalan pusat kontrol (simpul pusat) memutuskan semua komunikasi
    2. Bila yang digunakan sebagai pusat kontrol adalah HUB maka kecepatan akan berkurang sesuai dengan penambahan komputer, semakin banyak semakin lambat.

4. Topologi Tree (Pohon)

Karakteristik topologi Tree antara lain :

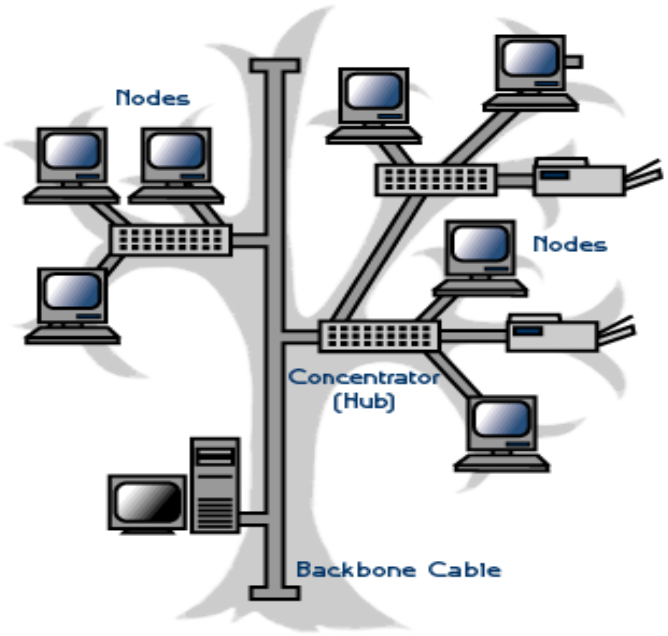
a. Topologi pohon adalah pengembangan atau generalisasi topologi bus. Media transmisi merupakan satu kabel yang bercabang namun loop tidak tertutup.

b. Topologi pohon dimulai dari suatu titik yang disebut “headend”. Dari headend beberapa kabel ditarik menjadi cabang, dan pada setiap

cabang terhubung beberapa terminal dalam bentuk bus, atau dicabang lagi hingga menjadi rumit.

c. Ada dua kesulitan pada topologi ini :

1. Karena bercabang maka diperlukan cara untuk menunjukkan kemana data dikirim, atau kepada siapa transmisi data ditujukan.
2. Perlu suatu mekanisme untuk mengatur transmisi dari terminal terminal dalam jaringan.



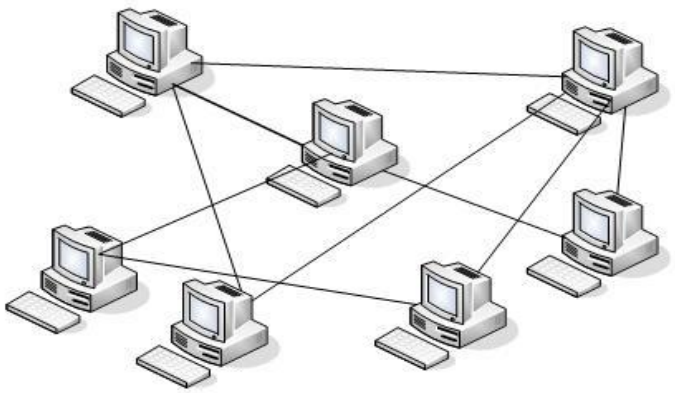
Gambar 2.22 Prinsip Koneksi Topologi Tree

5. Topologi Mesh (Tak beraturan)

Karakteristik topologi Mesh antara lain :

a. Topologi Mesh adalah topologi yang tidak memiliki aturan dalam koneksi. Topologi ini biasanya timbul akibat tidak adanya perencanaan awal ketika membangun suatu jaringan.

b. Karena tidak teratur maka kegagalan komunikasi menjadi sulit dideteksi, dan ada kemungkinan boros dalam pemakaian media transmisi.



Gambar 2.23 Prinsip Koneksi Topologi Mesh

6. Topologi Wireless (Nirkabel)

Karakteristik topologi Wireless antara lain :

a. Jaringan nirkabel menjadi trend sebagai alternatif dari jaringan kabel, terutama untuk pengembangan LAN tradisional karena bisa mengurangi biaya pemasangan kabel dan mengurangi tugas-tugas relokasi kabel apabila terjadi perubahan dalam arsitektur bangunan dsb. Topologi ini dikenal dengan berbagai nama, misalnya WLAN, WaveLAN, HotSpot, dsb.

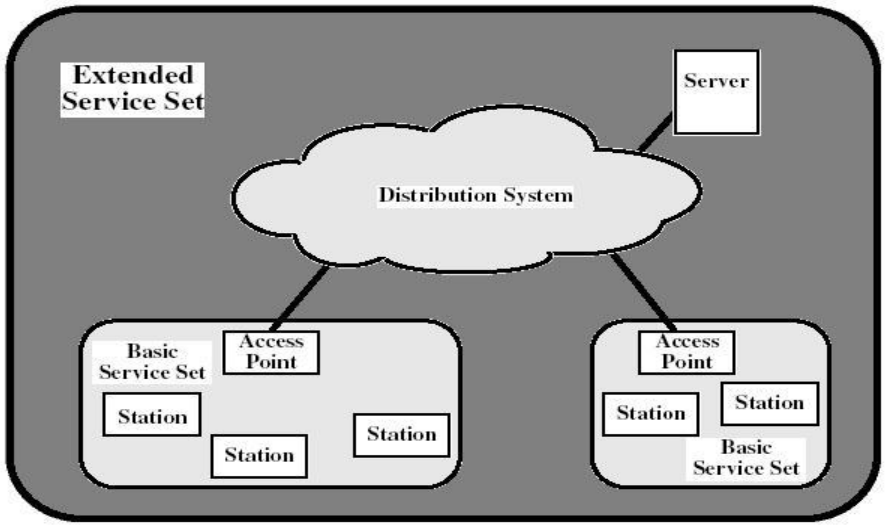
b. Blok terkecil dari LAN Nirkabel disebut ***Basic Service Set (BSS)***, yang terdiri atas sejumlah station / terminal yang menjalankan protokol yang sama dan berlomba dalam hal akses menuju media bersama yang sama.

c. Suatu BSS bisa terhubung langsung atau terpisah dari suatu sistem distribusi backbone melalui titik akses (Access Point).

d. Protokol MAC bisa terdistribusikan secara penuh atau terkontrol melalui suatu fungsi kordinasi sentral yang berada dalam titik akses.

e. Suatu ***Extended Service Set (ESS)*** terdiri dari dua atau lebih BSS yang dihubungkan melalui suatu sistem distribusi.

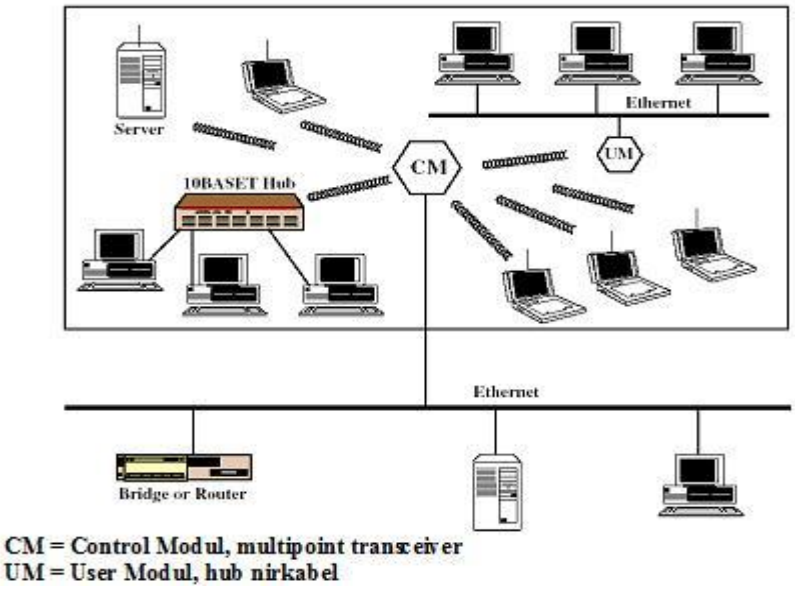
f. Model dasar dari LAN nirkabel adalah sebagai berikut :



Gambar 2.24 Prinsip LAN Nirkabel

g. Pada suatu jaringan LAN bisa terdapat LAN berkabel backbone, seperti "Ethernet" yang mendukung *server*, workstation, dan satu atau lebih bridge / router untuk dihubungkan dengan jaringan lain. Selain itu terdapat modul kontrol (CM) yang bertindak sebagai interface untuk jaringan LAN nirkabel. CM meliputi baik fungsi bridge ataupun fungsi router untuk menghubungkan LAN nirkabel dengan jaringan induk. Selain itu terdapat Hub dan juga modul pemakai yang mengontrol sejumlah stasiun LAN berkabel.

h. Interaksi antara LAN nirkabel dengan jenis LAN lainnya digambarkan sebagai berikut :



Gambar 2.25 Koneksi Jaringan Nirkabel

i. Penggunaan teknologi LAN nirkabel lainnya adalah untuk menghubungkan LAN pada bangunan yang berdekatan.

j. Syarat-syarat LAN nirkabel :

1. Laju penyelesaian: protokol medium access control harus bisa digunakan se-efisien mungkin oleh media nirkabel untuk memaksimalkan kapasitas.
2. Jumlah simpul: LAN nirkabel perlu mendukung ratusan simpul pada sel-sel multipel.
3. Koneksi ke LAN backbone: modul kontrol (CM) harus mampu menghubungkan suatu jaringan LAN ke jaringan LAN lainnya atau suatu jaringan ad-hoc nirkabel.
4. Daerah layanan: daerah jangkauan untuk LAN nirkabel biasanya memiliki diameter 100 hingga 300 meter.

5. Kekokohan dan keamanan transmisi: sistem LAN nirkabel harus handal dan mampu menyediakan sistem pengamanan terutama penyadapan.

k. Teknologi LAN nirkabel :

1. LAN infrared (IR) : terbatas dalam sebuah ruangan karena IR tidak mampu menembus dinding yang tidak tembus cahaya.
2. LAN gelombang radio : terbatas dalam sebuah kompleks gedung, seperti *bluetooth*, *WiFi*, dan *HomeRF*.
3. LAN spektrum penyebaran: beroperasi pada band-band ISM (industrial, scientific, medical) yang tidak memerlukan lisensi.
4. Gelombang mikro narrowband : beroperasi pada frekuensi gelombang mikro yang tidak termasuk dalam spektrum penyebaran.

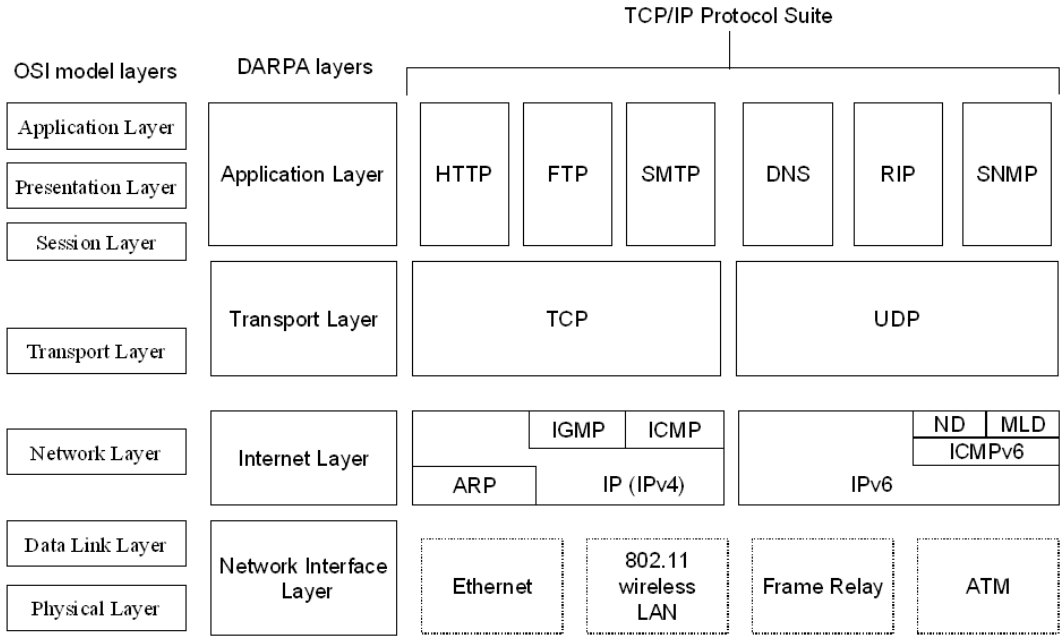
2.12 TCP/IP (Transmission Control Protocol)

TCP/IP adalah sekumpulan protokol (aturan) yang didesain untuk melakukan fungsi-fungsi komunikasi data pada WAN. Karena penggunaannya meluas, TCP/IP juga digunakan sebagai protocol standar *de facto.* Jika model OSI terdiri dari tujuh layer, model TCP/IP hanya terdiri dari 4 layer. Kesamaan fungsi antara referensi OSI dan TCP/IP dapat anda lihat pada gambar berikut. [1]

2.12.1 Application Layer

Layer ini berfungsi sebagai antarmuka protocol TCP/IP dengan user. Protokol-protokol pada layer ini adalah Telnet (*Telecommunication Network*),

FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), TFTP (*Trivial File Transfer Protocol*), NFS *(Network File Server*), RPC (*Remote Procedure Calls*), dan SNMP (*Simple Network Management Protocol*).



Gambar 2.26 Layer TCP/IP

2.12.2 Transport Layer

Layer ini bertanggung jawab untuk mengadakan komunikasi antara dua computer. Protokol-protokol yang berperan pada lapisan ini adalah :

a. Transmision Control Protocol (TCP)

Hubungan yang dibangun TCP mempunyai sifat *connection-oriented, reliable* dan *byte stream service*. *Connection-oriented* berarti sebelum melakukan pertukaran data,dua aplikasi pengguna TCP harus melakukan proses *handshaking* terlebih dahulu. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan transmisi ulang jika terjadi kesalahan saat pertukaran data. *Byte stream service* berarti paket dikirimkan dan

sampai ke tujuan secara terurut. Kelemahan dari hubungan ini adalah kurang efektif dalam penggunaan bandwidth. Selain itu, komunikasi akan terputus jika terjadi gangguan pada jalur komunikasi. Hal ini disebabkan oleh jalur yang digunakan, sehingga harus dilakukan dengan jalur yang telah ditentukan.

b. User Datagram Protocol (UDP)

Hubungan yang dibangun UDP bersifat *connectionless.* Pada UDP tidak ada pengurutan kembali (*sequencing*) paket yang datang. Selain itu, UDP tidak menjamin apakah data sampai ke tujuan atau tidak.

2.12.3 Internet Layer

Protokol yang terdapat pada lapisan ini adalah IP (Internet Protocol) yang merupakan inti dari TCP/IP dan merupakan protocol terpenting dalam internet layer. IP menyediakan pelayanan pengiriman paket elementer dimana jaringan TCP?IP dibangun. Fungsi internet protocol adalah :

a. Mendefinisikan paket yang merupakan unit transmisi elementer di internet.
b. Mendefinisikan skema pengalamatan jaringan.
c. Pemecahan (fragmentatiton) dan penyatuan dari paket-paket.
d. Routing, adalah sebuah proses untuk meneruskan paket-paket jaringan dari satu jaringan ke jaringan lainnya melalui sebuah *internetwork*.

ARP (*Address Resolution Protocol*) adalah protocol yang berfungsi untuk translasi dari IP address yang diketahui menjadi *MAC address*. RARP (*Reverse Address Resolution Protocol*) adalah protocol yang fungsinya berkebalikan dengan ARP, yakni mengadakan translasi dari *MAC address* yang diketahui

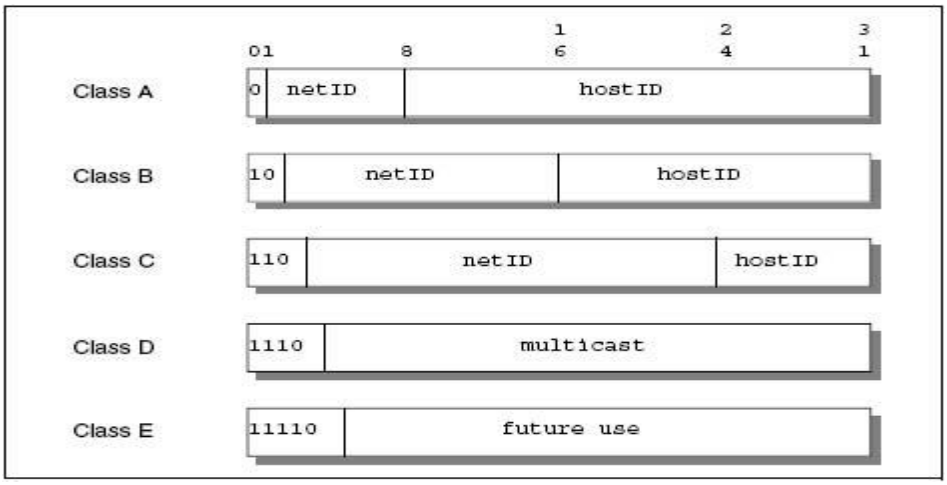
menjadi IP address. ICMP (*Internet control Massage Protocol*) berfungsi untuk mengirimkan pesan bila terjadi masalah dalam pengiriman data. Selain itu ICMP juga berperan dalam penstabilan kondisi jaringan.

2.12.4 Network Access Layer

Lapisan ini adalah lapisan terendah dari hirarki / protokol TCP IP, berfungsi untuk mengirim dan menerima data ke dan dari media fisik. Protokol dalam lapisan ini menyediakan sarana untuk sistem untuk memberikan data ke perangkat lain pada jaringan yang langsung terpasang. Hal ini mendefinisikan cara menggunakan jaringan untuk mengirimkan sebuah datagram IP. Tidak seperti protokol yang lebih tinggi tingkatnya, Network Access Layer harus mengetahui rincian jaringan yang mendasarinya (struktur paket dan alamat) dengan benar format data yang dikirim untuk memenuhi kendala jaringan. Lapisan ini dapat mencakup fungsi dari ketiga lapisan bawah acuan OSI Model (Network, Data Link, dan Fisik).

2.12.5 Internet Protocol (IP)

Internet Protocol Address atau sering disingkat IP adalah deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer host dalam jaringan Internet. Panjang dari angka ini adalah 32-bit (untuk IPv4 atau IP versi 4), dan 128-bit (untuk IPv6 atau IP versi 6) yang menunjukkan alamat dari komputer tersebut pada jaringan Internet berbasis TCP/IP. Jumlah IP address yang tersedia secara teoritis adalah 255x255x255x255 atau sekitar 4 milyar lebih yang harus dibagikan ke seluruh pengguna jaringan internet di seluruh dunia.



Gambar 2.27 Kelas IP

Pembagian kelas-kelas ini ditujukan untuk mempermudah alokasi IP Address, baik untuk host/jaringan tertentu atau untuk keperluan tertentu. Karakteristik dari kelas-kelas IP antara lain sebagai berikut :

a. Kelas A

Format	:	0nnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh
Bit pertama	:	0
Panjang NetID	:	8 Bit
Panjang HostID	:	24 Bit
Byte Pertama	:	0 - 127
Jumlah	:	126 KelasA (0 dan127 dicadangan)
Range IP	:	1.xxx.xxx.xxx sampai126.xxx.xxx.xxx
Jumlah IP	:	16.777.214 IP Address padatiapkelasA
Contoh IP	:	113.46.5.6
Network ID	:	113
Host ID	:	46.5.6

b. Kelas B

Format	:	10nnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh
Bit pertama	:	10
Panjang NetID	:	16 Bit
Panjang HostID	:	16 Bit
Byte Pertama	:	128 - 191
Jumlah	:	16.184 Kelas B
Range IP	:	128.0.xxx.xxx sampai 191.255.xxx.xxx
Jumlah IP	:	65.532 IP Address pada tiap kelasB
Contoh IP	:	132.92.121.1
Network ID	:	132.92
Host ID	:	121.1

c. Kelas C

Format	:	110nnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh
Bit pertama	:	110
Panjang NetID	:	24 Bit
Panjang HostID	:	8 Bit
Byte Pertama	:	192 - 223
Jumlah	:	2.097.152 Kelas C
Range IP	:	192.0.0.xxx sampai 223.255.255.xxx
Jumlah IP	:	245 IP Address pada tiap kelas C
Contoh IP	:	222.124.203.53

Network ID : 222.124.203

Host ID : 53

d. Kelas D

4 Bit Pertama : 1110

Bit Multicast : 28 Bit

Byte Inisial : 224 - 247

Bit cadangan : -

Jumlah : 268.435.455 Kelas D

Deskripsi : Digunakan untuk *multicast*

e. Kelas E

4 Bit Pertama : 1111

Bit Multicast : -

Byte Inisial : 248 - 255

Bit cadangan : 28 Bit

Jumlah : 268.435.455 Kelas E

Deskripsi : Dicadangkan untuk keperluan eksperimental

2.13 Quality of Service pada Jaringan IP

Quality of Service (QoS) didefinisikan sebagai suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. Pada jaringan berbasis *IP*, *IP QoS* mengacu pada performansi dari paket -paket *IP* yang lewat melalui satu atau lebih jaringan. QoS didesain untuk membantu end user menjadi lebih produktif dengan memastikan bahwa end user mendapatkan performansi yang handal dari aplikasiaplikasi berbasis jaringan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. Parameter-parameter performansi dari jaringan *IP* adalah :

1. *Delay*, didefinisikan sebagai total waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. *Delay* di dalam jaringan dapat digolongkan sebagai berikut *delay processing*, *delay packetization*, *delay serialization*, *delay jitter buffer* dan *delay network*.
2. *Jitter*, didefinisikan sebagai variasi dari *delay* atau variasi waktu kedatangan paket. Banyak hal yang dapat menyebabkan *jitter*, diantaranya adalah peningkatan trafik secara tiba-tiba sehingga menyebabkan penyempitan *bandwith* dan menimbulkan antrian. Selain itu, kecepatan terima dan kirim paket dari setiap node juga dapat menyebabkan *jitter*.
3. *Packet loss*, adalah perbandingan seluruh paket *IP* yang hilang dengan seluruh paket IP yang dikirimkan antara pada *source* dan *destination*.

Salah satu penyebab paket loss adalah antrian yang melebihi kapasitas *buffer* pada setiap *node*. Beberapa penyebab terjadinya paket loss yaitu:

a. *Congestion*, disebabkan terjadinya antrian yang berlebihan dalam jaringan.
b. *Node* yang bekerja melebihi kapasitas *buffer*.
c. *Memory* yang terbatas pada node.
d. *Policing* atau *control* terhadap jaringan untuk memastikan bahwa jumlah trafik yang mengalir sesuai dengan besarnya *bandwidth*. Jika besarnya trafik yang mengalir didalam jaringan melebihi dari kapasitas bandwidth yang ada maka policing control akan membuang kelebihan trafik yang ada.

4. *Bandwidth*, adalah banyaknya ukuran suatu data atau informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam sebuah network di waktu tertentu. Bandwidth dapat dipakai untuk mengukur baik aliran data analog maupun data digital. Sekarang sudah menjadi umum jika kata bandwith lebih banyak dipakai untuk mengukur aliran data digital. Satuan yang dipakai untuk bandwidth adalah Bits Per Second atau sering disingkat Bps. Seperti diketahui bahwa bit atau binary digit adalah basis angka yang terdiri dari 0 dan 1. Satuan ini menggambarkan berapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lainnya dalam setiap detiknya melalui suatu media.

2.14 Aplikasi Pendukung

2.14.1 Web Site

Sering disingkat sebagai WWW atau *web* saja, yakni sebuah sistem dimana informasi dalam bentuk teks gambar,suara, dan lain-lain dipresentasikan dalam bentuk hypertext dan dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam format HTML. Informasi lainnya disajikan dalam bentukgrafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World).

WWW dijalankan dalam *server* yang disebut HTTPD. WWW adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen *Web* disebut *Web* Page dan link dalam *Web* menyebabkan user bisa pindah dari satu page ke page lain (hyper text), baik antar page yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Pages diakses dandibaca melalui *Web Browser*.

Web menjadi pusat kegiatan internet karena *Web Pages* yang berisi text dan grafik mudah diakses melalui *Web Browser*, *Web* menyediakan *interface* bagi jaringan informasi online terbesar di dunia, dan jumlah informasi ini terus bertambah dengan pesat. *Web* telah banyak diadopsi sebagai bagian dari strategi teknologi informasi dikarenakan akses informasi mudah, setup *server* lebih mudah, informasi mudah didistribusikan, bebas platform. WWW bekerja berdasarkan pada tiga mekanisme berikut :

1. *Protocol standard* aturan yang di gunakan untuk berkomunikasi pada komputer *networking, Hypertext Transfer Protocol* (HTTP) adalah *protocol* untuk WWW.
2. *Address* WWW memiliki aturan penamaan alamat *web* yaitu URL(*Uniform Resource Locator)* yang di gunakan sebagai standard alamat internet.
3. HTML digunakan untuk membuat dokumen yang bisa di akses melalui *web*.

2.14.2 Hypertext Transfer Protocol (HTTP)

HTTP *(Hypertext Transfer Protocol)* adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia. Penggunaannya banyak pada pengambilan sumber daya yang saling terhubung dengan tautan, yang disebut dengan dokumen hiperteks, yang kemudian membentuk *World Wide Web* pada tahun 1990 oleh fisikawan Inggris, Tim Berners-Lee. Hingga kini, ada dua versi mayor dari protokol HTTP, yakni HTTP/1.0 yang menggunakan koneksi terpisah untuk setiap dokumen, dan HTTP/1.1 yang dapat menggunakan koneksi yang sama untuk melakukan transaksi. Dengan demikian, HTTP/1.1 bisa lebih cepat karena memang tidak usah membuang waktu untuk pembuatan koneksi berulang-ulang.Pengembangan standar HTTP telah dilaksanakan oleh Konsorsium *World Wide Web* (*World Wide Web Consortium/W3C*) dan juga *Internet Engineering Task Force (*IETF*),* yang berujung pada publikasi beberapa dokumen *Request for*

*Comments (*RFC*)*, dan yang paling banyak dirujuk adalah RFC 2616 (yang dipublikasikan pada bulan Juni 1999), yang mendefinisikan HTTP/1.1.

2.14.3 Hypertext Markup Language (HTML)

HTML *(Hypertext Markup Language)* merupakan standard bahasa yang digunakan untuk menampilkan document *web*, yang bisa dilakukan dengan HTML yaitu:

1. Mengontrol tampilan dari *web* page dan contentnya.
2. Mempublikasikan document secara online sehingga bisa di akses dari seluruh dunia.
3. Membuat online form yang bisa di gunakan untuk menangani pendaftaran, transaksi secara online.
4. Menambahkan object-object seperti image, audio, video dan juga java applet dalam dokumen HTML.

2.14.4 Web Server

Web server adalah *software* yang menjadi tulang belakang dari *world wide web* (www). *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti *Internet Explorer, Mozilla Firefox, Google Chrome,* dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut. Contohnya, bila data yang dikirim berupa gambar,

browser yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja. *Web server*, untuk berkomunikasi dengan *client*-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*).

Dengan protokol ini, komunikasi antar *web server* dengan *client*-nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Kata *HyperText* mempunyai arti bahwa seorang pengguna internet dengan *web browser*nya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (*web*) itu. Proses yang dimulai dari permintaan *web-client* (*browser*), diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya*web server* hanya memproses semua masukan yang diperolehnya dari *web-client* nya.

2.14.5 Web Browser

Web Browser merupakan perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *server* web. Web sendiri adalah kumpulan jaringan berisi dokumen dan tersambung satu dengan yang lain, yang dikenal sebagai *World Wide Web*. *Software* ini juga telah dikembangkan dengan menggunakan user *interface* gratis, sehingga pengguna dapat melakukan point dan click untuk pindah antar dokumen. Banyak sekali *software web browser* yang dapat dipakai secara gratis, seperti *internet explorer*, *mozilla firefox*, *google chrome*, dan safari.

2.14.6 Uniform Resource Locators (URL)

URL singkatan dari *Uniform Resource Locator*, adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di Internet. URL dapat diibaratkan sebagai suatu alamat, dimana alamat tersebut terdiri dari :

1. Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi,
2. Nama dari komputer dimana informasi tersebuut berada, dan
3. Jalur serta nama file dari suatu informasi.

Format umum dari suatu URL adalah sebagai berikut:

Protokol_transfer://domain_host[:port]/path/nama_file

2.15 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan

struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.

2.15.1 Diagram

Diagram menampilkan representasi sebagian dari sistem, dan semantiks dengan *view* yang lain. Dalam UML, ada dua tipe diagram, yaitu :

a. Static Diagram : use case diagram, class diagram, object diagram,component diagram dan deployment diagram.
b. Dinamic Diagram : sequence diagram, collaboration diagram, activity diagram dan statechart diagram.

2.15.2 Usecase Diagram

Usecase diagram menjelaskan sistem secara fungsional yang terlihat user. Biasanya dibuat pada awal pengembangan. Tujuan *usecase* adalah untuk:

a. Menspesifikasikan *context* sistem
b. Menggambarkan kebutuhan sistem
c. Memvalidasikan arsitektur sistem
d. Menjalankan impelementasi dan menggenerate test case

1. Include

Keterhubungan secara *include* antar use case menunjukkan bahwa usecase asal secara eksplisit memasukkan perilaku dari use case lain yang ditunjuk oleh use case tersebut. *Included usecase* tidak pernah berdiri sendiri, tetapi hanya merupakan bagian dari beberapa use case yang lebih besar yang diikutinya.

Keterhubungan use case secara include pada dasarnya merupakan sebuah contoh dari pendelegasian-sekumpulan dari tanggung jawab sebuah system diambil dan ditangkap di dalam satu tempat (*included use case*), kemudian bagian lainnya dari sistem (use case yang lain) memasukkan kumpulan tanggung jawab yang baru setiap saat mereka memerlukan fungsifungsi use case tersebut.

2. Extend

Keterhubungan use case secara *extend* menunjukkan bahwa usecase yang ditunjuk merupakan perluasan perilaku dari use case asal. Use case asal dapat berdiri sendiri, tetapi untuk kondisi tertentu perilaku use case tersebut dapat diperluas oleh perilaku dari use case lain. Hubungan perluasan digunakan untuk memodelkan bagian dari use case yang dapat dilihat oleh user sebagai perilaku yang dapat dipilih dari sistem. Hubungan perluasan juga dapat digunakan untuk memodelkan sub aliran yang terpisah-pisah yang hanya dapat dijalankan dalam kondisi tertentu.

2.15.3 Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antara *object* sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

2.15.4 Class Diagram

Class Diagram menggambarkan kamus data yang berlaku dalam sistem. Tujuannya, untuk membuat konsep nama dan pemodelan dan menspesifikasikan setiap proses yang terjadi.

2.16 Bahasa Pemrograman Ruby on Rails

2.16.1 Pengertian Ruby on Rails

Bahasa pemprograman Ruby merupakan bahasa pemprograman *open source* berorientasi objek yang dinamis, mudah dimengerti, dan produktif. Tujuan dari Ruby adalah menggabungkan kelebihan dari semua bahasa-bahasa pemprograman *scripting* yang ada di dunia . Secara sintak, Ruby banyak diinspirasi oleh bahasa pemprograman Perl dengan fitur yang mirip dengan Smalltalk. Secara garis besar Ruby banyak dipengaruhi oleh Smalltalk, Perl, Lisp, Scheme, Python, CLU, Eiffel, Ada, Dylan.

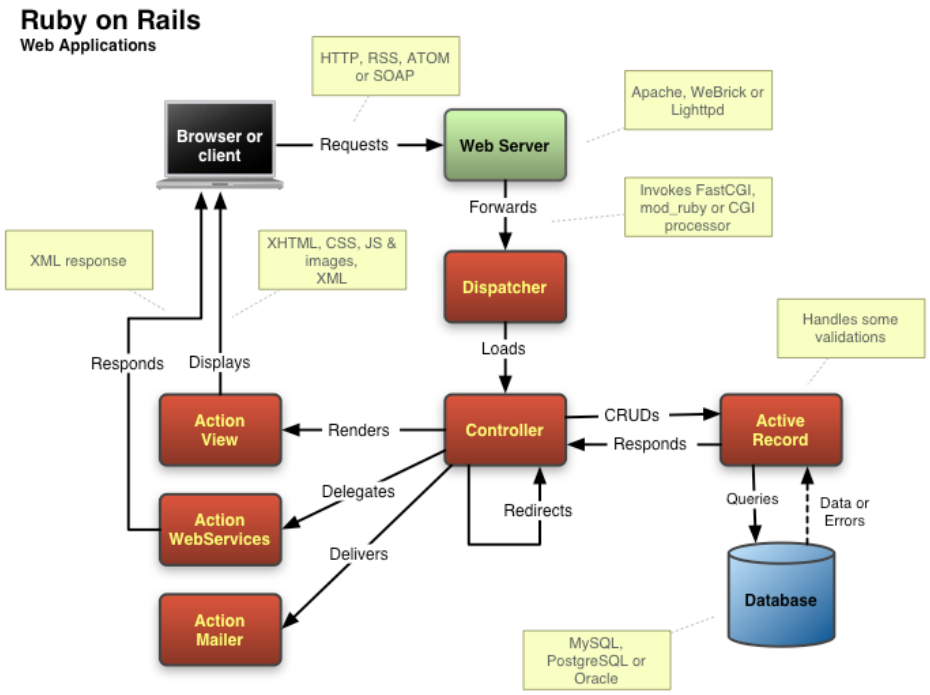
2.16.2 Sejarah Ruby on Rails

Ruby pertama kali dibuat oleh seorang programmer jepang bernama Yukihiro Maksumoto. Pada tahun 1993 Yukihiro Maksumoto ingin membuat sebuah bahasa scripting yang memiliki kemampuan orientasi obyek. Pada saat itu pemograman berorientasi obyek sedang berkembang tetapi belum ada bahasa pemrograman scripting yang mendukung pemograman obyek. Ruby dikatakan sebagai bahasa pemprograman yang mengikuti *principle of least astonishment* (POLA), yang berarti bahasa pemprograman Ruby harus memiliki perilaku yang dapat meminimalisir kebingungan bagi pengguna yang telah berpengalaman dengan bahasa-bahasa pemprograman lainnya.

Ruby on Rails atau seringkali disingkat *Rails* atau *ROR* adalah sebuah *framework* aplikasi WEB yang bersifat *open source*. Framework ini ditujukan untuk bahasa pemprograman Ruby. Ditujukan untuk pengembangan aplikasi

dengan metodologi *agile* yang biasa digunakan oleh para pengembang aplikasi web untuk proses pengembangan aplikasi secara cepat.

Berikut ini adalah arsitektur dari *framework* Rails :



Gambar 2.28 Arsitektur Framework Rails

Framework Rails dibangun untuk memenuhi beberapa kaidah yang memudahkan pihak developer dalam mengembangkan aplikasinya, diantaranya adalah :

Convention over configuration

Framework Rails lebih menekankan aturan penulisan dalam masa pengembangan dibandingkan dengan penekanan konfigurasi. Hal ini menyebabkan pengembang aplikasi memiliki pola yang sama, sehingga memudahkan pengembang aplikasi lainnya dalam memahami *source code* dalam aplikasi tersebut dengan tingkat kesulitan pemahaman yang kecil. Dengan

memahami aturan-aturan penulisan pada *framework* Rails, pengembang aplikasi juga dapat mengurangi kode-kode konfigurasi aplikasi yang biasanya terus menerus berulang-ulang dilakukan sehingga meningkatkan kompleksitas kode aplikasi yang dikembangkannya. Aturan-aturan dan kaidah-kaidah penulisan yang digunakan dalam *framework* Rails meliputi *case-sensitive* (huruf besar dan kecil) dan *pluralization* (kata jamak dan tunggal) dalam menentukan Object Relational Mapping (ORM).

Less Code

Dengan aturan-aturan yang ditetapkan dalam *framework* Rails, ini juga berarti bahwa *framework* Rails mengeliminasi kode-kode konfigurasi. Ini juga berarti bahwa *framework* Rails menutupi kode-kode detail, sehingga pengguna *framework* ini tidak perlu melakukan pengkodean hingga ke tahap *low-level*. Dengan demikian maka proses pengembangan aplikasi menjadi sangat cepat dengan bugs yang sangat minim, dan juga membuat kode program yang dikembangkan mudah untuk dimengerti, dikelola, dan ditingkatkan.

Generators

Framework Rails juga menyediakan *generator script* yang bertujuan untuk mempercepat proses pengembangan aplikasi, sehingga pihak pengembang aplikasi dapat difokuskan untuk berfikir lebih ke dalam kode aplikasi dan *business logic* tanpa harus menentukan struktur aplikasi yang dibuat.

Zero turnaround time

Secara umum siklus proses pengembangan aplikasi untuk melakukan pengetesan terhadap perubahan yang dilakukan memiliki tahapan seperti *configure*, *compile*, *deploy*, *reset*, dan *test*, tahapan-tahapan tersebut tentunya

sangat memakan waktu. Namun *framework* Rails tidak demikian, dengan sederhana pengembang aplikasi dapat melakukan perubahan dan melihat perubahannya langsung pada *browser* tanpa harus melakukan tahapan-tahapan tersebut.

Scaffolding

Framework Rails dapat secara otomatis membuatkan sekumpulan operasi CRUD (Create, Retrieve, Update, dan Delete) untuk setiap proses pada database, yang dapat secara langsung digunakan. Selain itu kode-kode hasil *Scaffolding* tersebut dapat dikostumisasi oleh pengembang aplikasi.

Dont Repeat Yourself

Framework Rails juga menyediakan kemampuan untuk meminimalisir pengulangan penulisan kode yang sama untuk operasi-operasi yang berbeda.

2.16.3 MVC (Model View Controller)

Berikut ini adalah konsep MVC (*Model View Controller*) yang digunakan dalam *framework Ruby on Rails* :

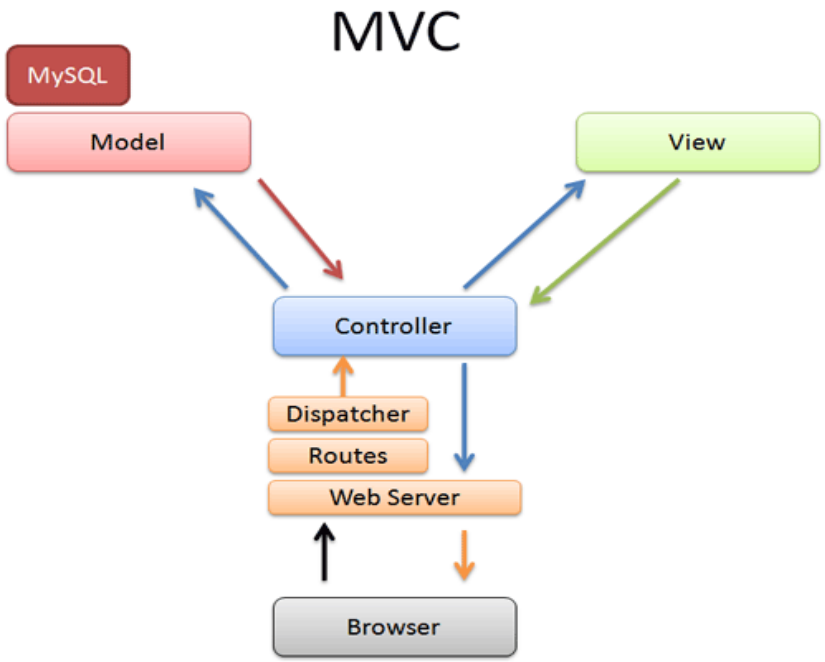
a. ***Browser*** membuat permintaan, contoh :“http://mysite.com/video/show/15”

b. ***Web Server*** (mongrel, WEBrick, dan lain-lain) menerima permintaan tersebut. Pembacaan permintaan tersebut diatur oleh kaidah pembacaan yang tetapannya ditentukan dalam ***routes***. *Routes* memiliki peran untuk menentukan *pattern* dari suatu permintaan, secara *default pattern*-nya adalah “/controller/action/id” yang didefinisikan pada file config/routes.rb. Pada kasus permintaan sebelumnya (“http://mysite.com/video/show/15”), maka *controller*-nya adalah ”video”, *method*-nya adalah “show”, dan *id*-

nya adalah "15". Lalu *Web Server* menggunakan ***dispatcher*** untuk membuat *controller* baru, memanggil *action*, dan mengirimkan *parameter*.

c. ***Controller*** melakukan pekerjaannya dengan mem-*parsing* permintaan-permintaan pengguna, submisi data, *cookies*, *sessions*, dan "perlengkapan *browser*". Dalam kasus sebelumnya telah diketahui bahwa *method "show"* pada *controller "video"* digunakan untuk melihat sebuah video. Dan meminta ***model*** untuk mengambil video dengan *id* 15, dan dengan segera akan ditampilkan kepada pengguna.

d. ***Model*** merupakan suatu class yang berinteraksi langsung dengan database, dengan kebutuhan seperti penyimpanan dan validasi data, melakukan *business logic*, dan mengangkat suatu data ke dalam aplikasi. Dalam kasus sebelumnya *model* mengangkat data video dengan *id* 15 dari database.



Gambar 2.29 Aristektur MVC

e. ***View*** merupakan sesuatu yang dilihat oleh *end-user* (pengguna aplikasi), dapat berupa : HTML, CSS, XML, Javascript, JSON. *View* memiliki tugas untuk membaca segala sesuatu yang diberikan oleh *controller*. Dalam kasus sebelumnya *controller* memberikan data video dengan *id* yang telah didapatkan dari database untuk kemudian ditampilkan pada *view* "*show*". Di dalam *view* "*show*" kemudian dibuatkan elemen-elemen HTML untuk mempermudah *end-user* dalam membaca data, seperti : *div*, *table*, *text*, *description*, *footer*, dan lain-lain.

Terakhir, *controller* kembali mengirimkan respon berupa (HTML, XML, metadata) kepada *server*, kemudian *server* menggabungkan data-data mentah kedalam bentuk respon HTTP yang layak dan mengirimkannya kepada *end-user*.