BAB II
LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonomi yang saling berhubungan antara satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program, penggunaan bersama perangkat printer, harddisk dan sebagainya [6].

2.1.1 Pengelompokan Jaringan Komputer

Berdasarkan jarak dan area kerjanya jaringan komputer dibedakan menjadi tiga kelompok, yaitu :

1. LAN (*Local Area Network*)

   LAN adalah sejumlah komputer yang saling dihubungkan bersama di dalam satu area tertentu yang tidak begitu luas, seperti di dalam satu kantor atau gedung.

2. MAN (*Metropolitan Area Network*)

   MAN adalah jaringan yang banyak digunakan untuk menghubungkan simpul yang berada pada jarak 20 – 50 Km, jaringan ini biasa digunakan untuk antar kota dengan menggunakan poket radio atau fasilitas perusahaan telekomunikasi.

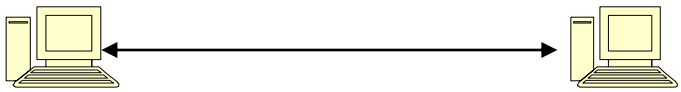
3. WAN (*Wide Area Network*)

   Wide Area Network (WAN) merupakan jaringan dari sistem komunikasi data yang masing-masing *node* berlokasi jauh (*Remote Location*) satu dengan yang lainnya. WAN disebut juga dengan nama *Remote Network* / *Long Distance network*. *Node* adalah titik yang dapat menerima *input* data ke dalam *network* atau menghasilkan *output* informasi atau kedua-duanya.

2.1.2 Tipe Jaringan Komputer

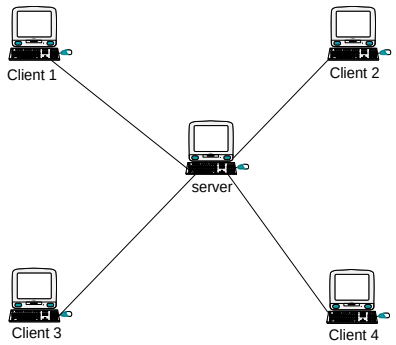
Tipe jaringan komputer menurut fungsinya dapat dibedakan menjadi 2 tipe, yaitu jaringan *peer to peer* dan *client server*.

Pada jaringan *peer to peer* setiap komputer yang terhubung dapat berkomunikasi dengan komputer–komputer yang lain secara langsung tanpa melalui komputer perantara. Gambar 2.1 dibawah ini menunjukkan skema logika sebuah jaringan *peer to peer*.



Gambar 2.1 Tipe Jaringan *Peer to Peer*

Berbeda dengan jaringan *peer to peer*, pada jaringan *client–server* terdapat sebuah komputer yang berfungsi sebagai *server* sedangkan komputer–komputer yang lain berfungsi sebagai *client*. Sesuai dengan namanya maka komputer *server* berfungsi dan bertugas melayani seluruh komputer yang terdapat dalam jaringan tersebut. Komputer–komputer ini sering disebut juga dengan *workstation*, yaitu komputer dimana pengguna jaringan dapat mengakses dan memanfaatkan pelayanan yang diberikan oleh komputer *server*. Gambar 2.2 menunjukkan tipe jaringan *client – server*.



Gambar 2.2 Tipe Jaringan *Client – Server*

2.1.3 Jenis Topologi Jaringan Komputer

Apabila dilihat dari jenis hubungannya, maka topologi dasar jaringan dapat dibagi menjadi tiga, yaitu :

1. Topologi Cincin (*Ring Topology*)
2. Topologi Bus (*Bus Topology*)
3. Topologi Bintang (*Star Topology*)

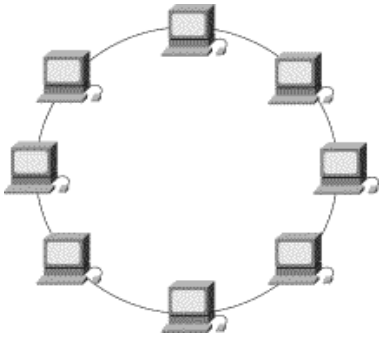
2.1.3.1 Topologi Ring (*Ring Topology*)

Topologi jenis *ring* menghubungkan satu komputer di dalam suatu *loop* tertutup. Pada topologi ini data atau *message* berjalan mengelilingi jaringan dengan satu arah pengiriman ke komputer selanjutnya terus hingga mencapai komputer yang dituju. Waktu yang di butuhkan untuk mencapai terminal tujuan disebut waktu transmisi.

Ada dua hal yang dilakukan oleh suatu terminal ketika menerima data dari komputer sebelumnya, yaitu :

1. Memeriksa alamat yang dituju dari data tersebut dan menerimanya jika terminal ini merupakan tujuan data tersebut.
2. Terminal akan meneruskan data ke komputer selanjutnya dengan memberikan tanda negatif ke komputer pengirim.

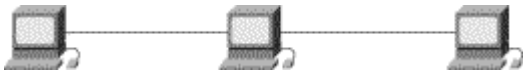
Apabila ada komputer yang tidak berfungsi maka hal tersebut tidak akan mengganggu jalannya jaringan, tapi apabila satu kabel putus akan mengakibatkan jaringan tidak berfungsi. Bentuk topologi *ring* digambarkan pada gambar 2.3.



Gambar 2.3 Topologi Cincin (*Ring Topology*)

2.1.3.2 Topologi Bus (*Bus Topology*)

Topologi jaringan jenis ini menggunakan sebuah kabel pusat yang merupakan media utama dari jaringan. Terminal-terminal yang akan membangun jaringan dihubungkan dengan kabel utama yang merupakan inti dari jaringan. Gambar 2.4 menunjukan bentuk topologi bus.



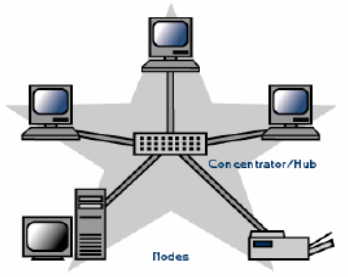
Gambar 2.4 Topologi Bus (*Bus topology*)

Data yang dikirimkan akan langsung menuju terminal yang dituju tanpa harus melewati terminal-terminal dalam jaringan, atau akan di *routing*kan ke *head end controller*. Tidak bekerjanya sebuah komputer tidak akan menghentikan kerja dari jaringan, jaringan akan tak bekerja apabila kabel utamanya dipotong atau putus.

Jaringan ini merupakan jaringan yang banyak digunakan karena hanya dalam beberapa meter kabel dapat dihubungkan ke banyak terminal *client*. Jaringan ini biasanya menggunakan kabel *coaxial* sebagai media transmisinya. Kabel *coaxial* dilihat dari bentuk fisiknya mirip dengan kabel antena. Kabel ini mempunyai kapasitas *bandwidth* 2MB.

2.1.3.3 Topologi Bintang (*Star Topology*)

Jenis topologi jaringan ini menggunakan satu terminal sebagai terminal sentral yang mengubungkan ke semua terminal *client*. Terminal sentral ini yang mengarahkan setiap data yang dikirimkan ke komputer yang dituju. Jenis jaringan ini apabila ada salah satu terminal *client* tidak berfungsi atau media transmisi putus atau terganggu maka tidak akan mempengaruhi kerja dari jaringan, karena gangguan tersebut hanya mempengaruhi terminal yang bersangkutan. Bentuk topologi *star* digambarkan pada gambar 2.5 berikut:



Gambar 2.5 Topologi Bintang (*Star Topology*)

Kelemahan dari jenis topologi jaringan ini adalah ketergantungan terhadap suatu terminal sentral. Hal tersebut merupakan suatu gangguan yang sangat berarti apabila terminal sentral tersebut mendapatkan gangguan, sehingga dicari suatu solusi yang dapat mengatasi masalah tersebut. Salah satu solusi yang banyak dilakukan adalah dengan menggunakan dua buah terminal sebagai *server*, sehingga apabila satu *server* dalam keadaan *down* dapat dialihkan ke *server* yang kedua dan begitu seterusnya. Keuntungan dan kerugian masing-masing topologi dapat dilihat pada tabel 2.1 berikut:

Tabel 2.1. Keuntungan dan Kerugian Topologi Jaringan

Topologi	Keuntungan	Kerugian
BUS	- Hemat kabel - Layout kabel sederhana - Mudah dikembangkan - Tidak butuh kendali pusat	- Dapat memungkinkan terjadinya tabrakan data - Kepadatan lalu lintas tinggi - Kecepatan akan menurun bila jumlah pemakai bertambah - Diperlukan repeater untuk jarak jauh
RING	- Hemat kabel - Tidak perlu penanganan pembundelan kabel - Kabel khusus - Dapat melayani lalu lintas data yang padat	- Peka kesalahan - Pengembangan jaringan lebih kaku - Kerusakan pada media pengirim /terminal dapat melumpuhkan kerja seluruh jaringan lambat, karena pengiriman menunggu giliran token.
STAR	- Paling fleksibel karena pemasangan kabel mudah - Penambahan atau pengurangan stasiun sangat mudah dan tidak menggangu bagian jaringan yang lain	- Boros kabel - Kontrol terpusat (HUB) jadi elemen kritis

	- Kontrol terpusat karena memudahkan dalam deteksi dan isolasi kesalahan/kerusakan - Memudahkan penggelolaan jaringan	

2.2 Protokol Jaringan

Protokol jaringan adalah aturan-aturan atau tatacara yang digunakan dalam melaksanakan pertukaran data dalam sebuah jaringan. Protokol mengurusi segala hal dalam komunikasi data, mulai dari kemungkinan perbedaan format data yang dipertukarkan hingga ke masalah koneksi listrik dalam jaringan. Dalam suatu jaringan komputer, terjadi sebuah proses komunikasi antar entitas atau perangkat yang berlainan sistemnya. Entitas atau perangkat ini adalah segala sesuatu yang mampu menerima dan mengirim. Untuk berkomunikasi mengirim dan menerima antara dua entitas dibutuhkan saling-pengertian di antara kedua belah pihak. Pengertian inilah yang dikatakan sebagai protokol. Jadi protokol adalah himpunan aturan-aturan main yang mengatur komunikasi data [8].

2.2.1 Model OSI

Model OSI atau *Open System Interconnection* menggambarkan bagaimana informasi dari suatu perangkat lunak aplikasi di sebuah komputer berpindah melewati sebuah media jaringan ke suatu perangkat lunak aplikasi di komputer lain. Model OSI dibuat untuk mengatasi masalah *internetworking* akibat perbedaan arsitektur dan protokol jaringan [6]. Model OSI secara konseptual terbagi ke dalam 7 lapisan dimana masing-masing lapisan memiliki fungsi jaringan yang spesifik seperti diperlihatkan pada gambar 2.6.



Gambar 2.6 Model OSI

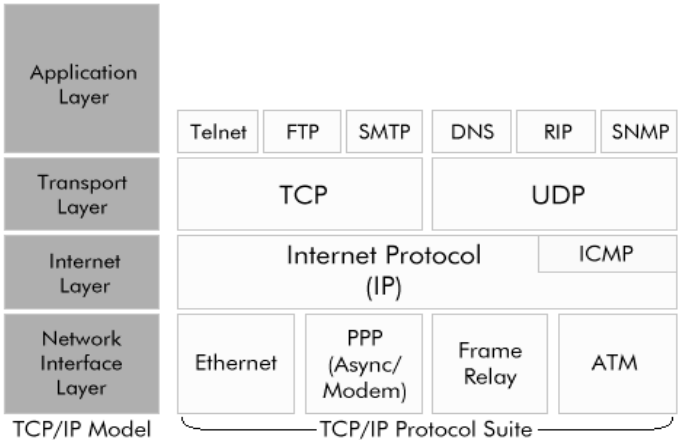
Masing-masing *layer* pada gambar 2.6 dapat dijelaskan sebagai berikut:

1. *Layer Phsical* berfungsi untuk mendefinisikan media transmisi jaringan, sinkronisasi *bit*, arsitektur jaringan. Selain itu, *layer* ini juga mendefinisikan bagaimana *network interface card* berinteraksi dengan media *wire* atau *wireless.*
2. *Layer Data Link* berfungsi untuk menentukan bagaimana *bit-bit* dikelompokan menjadi format yang disebut *frame*. Pada *layer* ini juga terjadi pengalamatan perangkat keras dan menentukan bagaimana perangkat-perangkat jaringan beroprasi. Contoh protokol pada *layer* ini, Ethernet, Token bus, Token ring.
3. *Layer Network* berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket data, dan melakukan *routing*. Contoh protokol pada *layer* ini, IP, IPX.
4. *Layer Transport* berfungsi untuk memecah data menjadi paket-paket data serta untuk memberikan nomor urut setiap paket sehingga dapat disusun kembali setelah diterima. Contoh protokol yang digunakan pada *layer* ini TCP dan UDP.
5. *Layer Session* berfungsi untuk mendefinisikan bagaimana koneksi dimulai, dipelihara, dan diakhiri. Contoh protokol pada *layer* ini, NetBIOS, ADSP, PAP.

6. *Layer Presentation* berfungsi untuk mentranslasikan data yang hendak ditransmisikan kedalam format yang dimengerti dalam jaringan. Contoh format kompresi, GIF, TIF, JPG.
7. *Layer Aplication* berfungsi sebagai antarmuka dengan *user*. Mengatur bagaimana aplikasi dapat mengakses jaringan. Contoh protokol yang berada pada *layer* ini, FTP, telnet, HTTP, SMTP, POP3, dan NFS.

2.2.2 TCP/IP

TCP/IP merupakan protokol jaringan komputer terbuka dan bisa terhubung dengan berbagai jenis perangkat keras dan lunak. Pada mulanya TCP/IP digunakan pada jaringan bernama ARPANET, namun saat ini telah menjadi protokol standar bagi jaringan yang lebih umum yang disebut internet. Seperti halnya OSI, pengiriman data pada model TCP/IP pun menggunakan prinsip enkapsulasi [6]. TCP terdiri beberapa layer atau lapisan yang memiliki fungsi tertentu dalam komunikasi data. Setiap fungsi dari *layer* selain dapat bekerja sama dengan *layer* pada tingkat lebih rendah atau lebih tinggi, juga bisa berkomunikasi dengan *layer* sejenis pada *remote host (peering)*. TCP/IP protokol terdiri dari 4 layers yaitu *Application, Transport*, *Internetwork*, dan *Network Interface*. Layer tersebut dapat dilihat sebagai hirarki seperti pada gambar 2.7.



Gambar 2.7 TCP/IP Layer

Berikut adalah penjelasan masing-masing *layer* pada gambar 2.7:

1. *Layer Application* adalah sebuah aplikasi yang mengirimkan data ke *transport layer*. Misalnya FTP, email program dan *web browser*.
2. *Layer Transport* bertanggung jawab untuk komunikasi antara aplikasi. *Layer* ini mengatur aluran informasi dan mungkin menyediakan pemeriksaan kesalahan. Data dibagi kedalam beberapa paket yang dikirim ke internet *layer* dengan sebuah *header*. *Header* mengandung alamat tujuan, alamat sumber dan *checksum*. *Checksum* diperiksa oleh mesin penerima untuk melihat apakah paket tersebut ada yang hilang pada rute.
3. *Layer Internetwork* bertanggung jawab untuk komunikasi antara mesin. *Layer* ini meng-*engcapsulation* paket dari *transport layer* ke dalam IP datagrams dan menggunakan algoritma *routing* untuk menentukan kemana datagaram harus dikirim. Masuknya datagram diproses dan diperiksa kesalahannya sebelum melewatinya pada *Transport layer*.
4. *Layer Networks Interface* adalah level yang paling bawah dari susunan TCP/IP. Layer ini adalah *device driver* yang memungkinkan datagaram IP dikirim ke atau dari jaringan fisik. Jaringan dapat berupa sebuah kabel, Ethernet, Frame relay, Token ring, ISDN, ATM jaringan, radio, satelit atau alat lain yang dapat mentransfer data dari sistem ke sistem. Layer *network interface* adalah abstraksi yang memudahkan komunikasi antara *multitude arsitektur network.*

Streaming sebagai dasar dari layanan VoD memerlukan protokol *transport* yang sesuai. *Streaming* dapat dilakukan pada protokol biasa seperti HTTP, tetapi dapat pula dilakukan menggunakan protokol khusus *streaming* multimedia.

2.2.3 IP Address

IP (*Internet Protocol) Address* adalah deretan angka yang dipakai sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan Internet [6]. Setiap IP itu sendiri memiliki beberapa bagian serta tipe-tipe kelas yang berbeda-beda. IP *Address*

dapat dipisahkan menjadi 2 bagian, yakni bagian *network* (*net* ID) dan bagian *host* (*host* ID). *Net* ID berperan dalam identifikasi suatu *network* dari *network* yang lain, sedangkan *host* ID berperan untuk identifikasi *host* dalam suatu *network*. Jadi, seluruh *host* yang tersambung dalam jaringan yang sama memiliki net ID yang sama. Sebagian dari bit-bit bagian awal dari IP *Address* merupakan *network* bit/*network number*, sedangkan sisanya untuk *host*. Garis pemisah antara bagian *network* dan *host* tidak tetap, bergantung kepada kelas *network*.

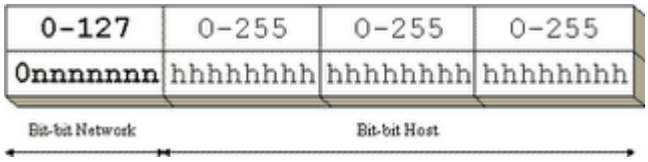
Perbedaan tiap kelas adalah pada ukuran dan jumlahnya. Contohnya IP kelas A dipakai oleh sedikit jaringan namun jumlah *host* yang dapat ditampung oleh tiap jaringan sangat besar. Kelas D dan E tidak digunakan secara umum, kelas D digunakan bagi jaringan *multicast* dan kelas E untuk keprluan eksperimental. Perangkat lunak *Internet Protokol* menentukan pembagian jenis kelas ini dengan menguji beberapa bit pertama dari IP *Address*.

2.2.3.1 Pembagian Kelas IP Address

Secara umum IP *Address* itu sendiri dikelompokan dalam beberapa kelas, antara lain :

1. Kelas A

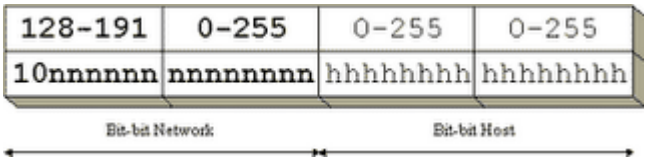
   Bit pertama IP *address* kelas A adalah 0, dengan panjang net ID 8 bit dan panjang *host* ID 24 bit. Jadi byte pertama IP *address* kelas A mempunyai range dari 0-127. Jadi pada kelas A terdapat 127 *network* dengan tiap *network* dapat menampung sekitar 16 juta *host*. IP *address* kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar, IP kelas ini dapat dilukiskan pada gambar 2.8 berikut:



Gambar 2.8 IP address kelas A

2. Kelas B

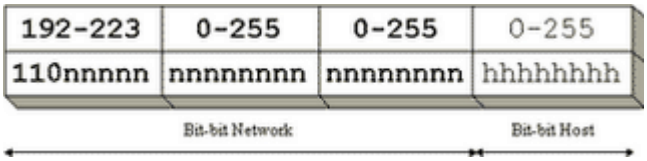
Dua bit IP *address* kelas B selalu di*set* 10 sehingga byte pertamanya selalu bernilai antara 128-191. *Network* ID adalah 16 bit pertama dan 16 bit sisanya adalah *host* ID sehingga kalau ada komputer mempunyai IP *address* 167.205.26.161, *network* ID = 167.205 dan *host* ID = 26.161. Pada IP *address* kelas B ini mempunyai *range* IP dari 128.0.xxx.xxx sampai 191.155.xxx.xxx, yakni berjumlah 65.255 *network* dengan jumlah *host* tiap *network* 2^{16} *host* atau sekitar 65 ribu *host*. Penjelasan pembagian *network* ID dan *host* ID dijelaskan pada gambar 2.9 berikut:



Gambar 2.9 IP address kelas B

3. Kelas C

Alamat IP kelas C mulanya digunakan untuk jaringan berukuran kecil seperti LAN. Tiga bit pertama IP *address* kelas C selalu di*set* 111. *Network* ID terdiri dari 24 bit dan *host* ID 8 bit sisanya sehingga dapat terbentuk sekitar 2 juta *network* dengan masing-masing *network* memiliki 2^8 *host* atau sekitar 256 *host*. IP kelas ini dapat dilukiskan pada gambar 2.10.



Gambar 2.10 IP address kelas C

4. Kelas D

   Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu di*set* ke bilangan biner 1110. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

5. Kelas E

   Alamat IP kelas E tidak diperuntukan untuk keperluan umum. Empat bit pertama selalu di*set* kepada bilangan biner 1111. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*. Secara garis besar pembagian kelas IP *address* dapat dilihat pada table 2.2 dibawah ini:

Tabel 2.2 Pembagian Kelas IP *address*

Kelas Alamat IP	Oktet Pertama (Desimal)	Oktet Pertama (Biner)	Digunakan Oleh
Kelas A	1-126	0xxx xxxx	Alamat *unicast* untuk jaringan skala besar
Kelas B	128-191	10xx xxxx	Alamat *unicast* untuk jaringan skala menengah hingga skala besar
Kelas C	192-223	110x xxxx	Alamat *unicast* untuk jaringan skala kecil
Kelas D	224-239	1110 xxxx	Alamat *multicast* (bukan alamat *unicast*)
Kelas E	240-255	1111 xxxx	Tidak digunakan untuk

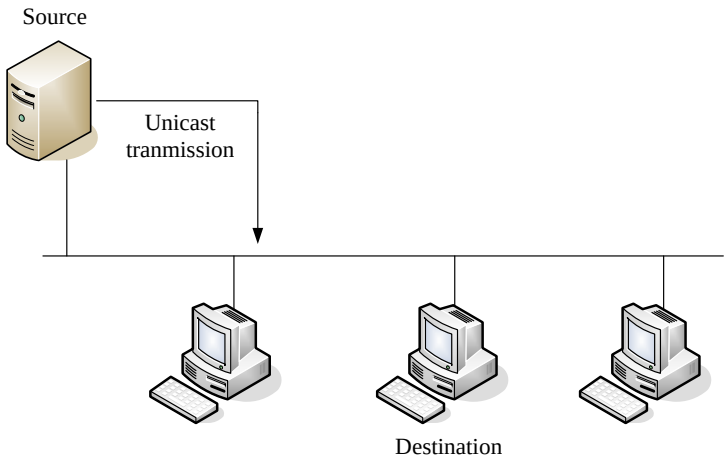
			kepentingan umum

2.2.3.2 Alamat *Unicast*

Unicast adalah jenis transmisi di mana informasi dikirim hanya dari satu pengirim ke satu penerima. Dalam kata lain, *Unicast* transmisi adalah antara satu-ke-satu *node*. Contoh dari transmisi *Unicast* yaitu, HTTP, SMTP, Telnet, SSH, POP3 di mana permintaan informasi diarahkan dari satu pengirim ke satu penerima di ujung lainnya [1].

Alamat *unicast* inilah yang harus digunakan oleh semua *host* TCP/IP agar dapat saling terhubung. Komponen alamat ini terbagi menjadi dua jenis, yakni alamat *host* (*host identifier*) dan alamat jaringan (*network identifier*).

Alamat *unicast* menggunakan kelas A, B, dan C dari kelas-kelas alamat IP yang telah disebutkan sebelumnya, sehingga ruang alamatnya adalah dari `1.x.y.z` hingga `223.x.y.z`. Sebuah alamat *unicast* dibedakan dengan alamat lainnya dengan menggunakan skema subnet mask. Penggambaran aliran data pada *unicast* dapat dilihat pada gambar 2.11 berikut:



Gambar 2.11 *Unicast Single-Stream*

2.3 VoD *(Video On Demand)*

VoD (*Video On Demand)* adalah sebuah istilah penyajian video yang bisa diakses secara *online* melalui jaringan, dimana pemirsa bisa melihat tayangan kapan saja. Video bisa disajikan langsung secara *streaming.* Fungsi VoD seperti layaknya video rental, dimana pelanggan dapat memilih program atau tontonan yang ingin ditayangkan [2]. Salah satu hal yang ingin dicapai adalah memberikan kontrol yang penuh terhadap para penggunanya. Selain itu juga karena semakin berkembangnya teknologi berbasis IP.

4.1 VoD *Server*

Perancangan sistem VoD pada jaringan ini merupakan salah satu bentuk aplikasi pemanfaatan teknologi *broadcast*. Sistem ini menggunakan VoD *server*. VoD *server* mempunyai kemampuan menyimpan dan memberikan *output* film dalam jumlah besar secara simultan. Sebagai gambaran, tiap film rata-rata memerlukan 800 MB bila dikompresi ke dalam AVI. Selain itu VoD juga mengatur permintan pengiriman video yang sama kepada beberapa *client.*

VoD *server* memiliki beberapa ketentuan, antara lain. Harus dapat menangani beberapa *user* sekaligus, dimana masing - masing *user* dapat mengakses video yang sama ataupun yang bebeda secara bersamaan. VoD *server* melakukan pengiriman video dan suara secara simultan oleh karena itu dibutuhkan *bandwidth* yang cukup besar. Karena merupakan *database* konten video maka VoD *server* juga memerlukan *storage* yang besar.

2.4 Set Top Box

Set-top box (STB) adalah sebuah perangkat yang memungkinkan sebuah pesawat televisi menjadi pengguna internet dan juga memungkinkan sebuah pesawat televisi untuk menerima dan membaca siaran televisi digital. STB yang kadang-kadang disebut *receivers*, diperlukan untuk pemirsa televisi yang saat ini

menggunakan *set* televisi analog untuk menerima siaran digital. Dalam dunia internet, STB merupakan sebuah komputer yang dibuat khusus dapat berkomunikasi dengan internet dimana di dalamnya berisi sebuah *web browser*.

STB yang ada saat ini berisi satu atau lebih mikroprosesor untuk menjalankan sistem operasi Linux atau Windows, dan untuk mengirimkan sebuah MPEG *streaming*. Sebuah STB yang memiliki RAM, yaitu sebuah *chip decoder* MPEG, maka dapat melakukan *decoding* dan pemrosesan data. Jenis - jenis STB bergantung pada standar *digital television* (DTV) pada tiap - tiap negara. Gambar 2.12 dibawah merupakan contoh STB [9].



Gambar 2.12 Set Top Box

2.5 PC Router

Pengertian PC *router* secara umum yaitu sebuah komputer yang difungsikan sebagai *router*. Fungsi dari *Router* adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. *Routing* sendiri merupakan proses pemindahan data dari satu *network* ke *network* yang lain dengan cara meneruskan paket data melalui *gateway* [6].

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan *router* jenis itu disebut juga dengan IP *Router*. Selain IP *Router*, ada lagi *AppleTalk Router*, dan masih ada beberapa jenis *router* lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router* IP.

2.6 Sistem Operasi

Secara umum, sistem operasi adalah perangkat lunak pada lapisan pertama yang ditempatkan pada memori komputer pada saat komputer dinyalakan. Sedangkan perangkat lunak lainnya dijalankan setelah *Sistem operasi* berjalan, dan *Sistem operasi* akan melakukan layanan inti umum untuk perangkat lunak tersebut. Layanan inti umum tersebut seperti akses ke disk, manajemen memori, *Scheduling task*, dan antar-muka *user*. Sehingga masing-masing perangkat lunak tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan dilakukan oleh sistem operasi. Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan kernel suatu sistem operasi [10].

Kalau sistem komputer terbagi dalam lapisan-lapisan, maka *Sistem operasi* adalah *penghubung* antara lapisan perangkat keras dan lapisan perangkat lunak. Lebih jauh daripada itu, *Sistem operasi* melakukan semua tugas-tugas penting dalam komputer, dan menjamin aplikasi-aplikasi yang berbeda dapat berjalan secara bersamaan dengan lancar. *Sistem operasi* menjamin aplikasi perangkat lunak lainnya dapat menggunakan memori, melakukan *input* dan *output* terhadap peralatan lain, dan memiliki akses kepada sistem file. Apabila beberapa aplikasi berjalan secara bersamaan, maka *Sistem operasi* mengatur penjadwalan yang tepat, sehingga sedapat mungkin semua proses yang berjalan mendapatkan waktu yang cukup untuk menggunakan prosesor serta tidak saling mengganggu.

2.6.1 Linux

Linux adalah sistem operasi yang bersifat *open source* dan bebas di bawah lisensi GNU (GNU *is not Unix*) GPL (*General Public License*). Arti kata *open source* adalah pada Linux kode sumber yang menjadi inti diikutsertakan dalam program sehingga dapat dilihat oleh siapa saja tanpa harus menandatangani suatu perjanjian khusus. Arti kata bebas atau *free* mengacu pada kebebasan bukan berarti bebas biaya.

Linux bebas untuk diduplikasikan, dimodifikasi, dan disebarluaskan baik untuk tujuan tak-komersial ataupun tujuan komersial [4,7].

Linux lahir pada tahun 1991. Pada waktu itu, Linus Torvald, seorang mahasiswa dari Suoen Tasavalta, Republik Finlandia memulai sebuah proyek pembuatan sistem operasi yang stabil dan bisa dijalankan pada mesin Intel x86. Pada awalnya, Linus Torvald menggunakan salah satu sistem operasi berbasis *Unix* yaitu *Minix*. Namun akhirnya dia merasa sistem operasi ini memiliki banyak kekurangan, dia merasa dapat membuat sistem operasi yang lebih baik dari *Minix*. Pada bulan Oktober 1991, Linus Torvald mempublikasikan sistem operasi yang diberi nama Linux (Linus' *Unix*). Dia mengajak pengembang perangkat lunak di seluruh dunia untuk bersama-sama mengembangkan sistem operasi Linux. Era *open source* mulai menunjukan kejayaannya.

Ubuntu

Ubuntu adalah salah satu distribusi Linux yang berbasiskan pada Debian. Ubuntu berasal dari bahasa kuno Afrika, yang berarti rasa perikemanusian terhadap sesama manusia. Ubuntu juga bisa berarti aku adalah aku karena keberadaan kita semua. Tujuan dari distribusi Linux Ubuntu adalah membawa semangat yang terkandung di dalam Ubuntu ke dalam dunia perangkat lunak. Ubuntu adalah sistem operasi lengkap berbasis Linux, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional [11].

Ubuntu dibentuk berdasarkan gagasan yang terdapat di dalam filosofi Ubuntu bahwa perangkat lunak harus tersedia dengan bebas biaya, bahwa aplikasi perangkat lunak tersebut harus dapat digunakan dalam bahasa lokal masing-masing dan untuk orang-orang yang mempunyai keterbatasan fisik, dan bahwa pengguna harus mempunyai kebebasan untuk mengubah perangkat lunak sesuai dengan apa yang mereka butuhkan.

Ubuntu cocok digunakan baik untuk *desktop* maupun *server*. Ubuntu saat ini mendukung berbagai arsitektur komputer seperti PC Intel x86, PC 64-bit AMD64, PowerPC, Sun UltraSPARC dan T1. Ubuntu menyertakan lebih dari 16.000 buah perangkat lunak, dan untuk instalasi *desktop* dapat dilakukan dengan menggunakan satu CD saja. Ubuntu menyertakan semua aplikasi standar untuk *desktop* mulai dari pengolah kata, aplikasi *spreadsheet* hingga aplikasi untuk mengakses internet, perangkat lunak untuk *server web*, peralatan untuk bahasa pemrograman dan beragam permainan.

2.6.2 Unix

Unix adalah nama sebuah sistem operasi yang asal mulanya dikembangkan pada laboratorium Bell, AT&T. UNIX terdiri atas sejumlah program atau instruksi untuk memperoleh hasil tertentu. Yang dirancang untuk mengontrol interaksi antara fungsi – fungsi tugas dari sistem operasi, diantaranya :

1. Melakukan fungsi manajemen sistem berkas
2. Mengendalikan berbagai sumber pada sistem, seperti disk dan printer.
3. Mengantar sejumlah pemakai yang menggunakan sistem bersamaan
4. Membentuk penjadwalan proses didalam sistem.

BSD adalah salah satu turunan Unix yang dikembangkan oleh Universitas California, Berkeley. Sistem operasi Unix digunakan secara luas baik sebagai *server* atau *workstation*. Arsitektur Unix dan model *client*/*server* merupakan elemen yang paling penting dalam perkembangan internet dan mengubah proses komputasi secara terpusat dalam jaringan dari pada proses tunggal di komputer.

OpenBSD

OpenBSD merupakan sistem operasi yang mengutamakan keamanan sistem. Para pengembang OpenBSD sangat menekankan pengamanan yang kuat dalam mengembangkan sistem operasi ini. Mereka ingin menjadikan OpenBSD sebagai

sistem operasi nomor satu dalam hal keamanan. OpenBSD dikembangkan dengan model *Open Software Development* sehingga celah-celah keamanan sekecil apapun dapat segera diketahui dan diperbaiki. OpenBSD menggunakan pendekatan *Secure By Default*, yaitu sistem operasi sudah di*set* dalam keadaan *secure* tanpa perlu melakukan konfigurasi lagi. Hal ini merupakan salah satu kelebihan OpenBSD dalam bidang keamanan dibandingkan sistem operasi lain, dimana *user* harus melakukan konfigurasi ulang jika menginginkan sistem yang benar-benar aman.

OpenBSD juga menerapkan *full disclosure* pada *bug-bug* dan masalah keamanan yang ditemuinya. Hal ini tentu sangat berbeda dengan vendor lain yang biasanya menyembunyikan *bug* dan masalah keamanan tersebut dari *user*. Pengembang OpenBSD juga sangat proaktif dalam proses *auditing* keamanan. OpenBSD telah menggunakan berbagai algoritma *kriptografi* untuk menambah keamanan sistemnya.

2.6.3 Windows

Microsoft Windows atau lebih dikenal dengan sebutan Windows adalah sistem operasi komputer pribadi yang dikembangkan oleh Microsoft yang menggunakan antarmuka dengan pengguna berbasis grafik GUI (*graphical user interface*). Sistem operasi Windows telah berevolusi dari MS-DOS, sebuah sistem operasi yang berbasis modus teks dan command-line. Microsoft Windows adalah sistem operasi yang paling banyak digunakan oleh masyarakat.

2.7 Bahasa Pemrograman

Bahasa Pemrograman merupakan notasi yang dipergunakan untuk mendeskripsikan proses komputasi dalam format yang dapat dibaca oleh komputer dan manusia. Komputer adalah mesin yang dapat melaksanakan seperangkat perintah dasar atau *instruction set*. Komputer hanya dapat diberi perintah yang terdiri dari perintah-perintah dasar tersebut. Perintah-perintah yang lebih rumit harus

diterjemahkan menjadi serangkaian perintah-perintah dasar yang dapat dimengerti komputer yaitu perintah-perintah yang termasuk dalam *instruction set* komputer tersebut yang pada akhirnya dapat menyelesaikan tugas yang diinginkan, meskipun dijalankan dengan beberapa operasi dasar, bukan satu operasi rumit.

8.1 HTML

Bahasa pemrograman *Hyper-text Markup Language* (HTML) merupakan bahasa pemrograman yang populer seiring dengan popularitas internet. HTML digunakan untuk objek-objek statis dalam suatu halaman *website*. Keuntungan HTML adalah:

1. Populer
2. Sederhana
3. Dapat bekerja dengan semua web *browser* dan web *server*

8.2 Javascript

Bahasa pemrograman javascript digunakan untuk menangani *field-field* dinamis pada *client-side. Field* yang mempergunakan javascript merupakan *field* yang *source-code* dapat dilihat langsung oleh *client.* Javascript akan digunakan untuk mempermudah pada pembuatan portal VoD, seperti pada pembuatan *frame* yang akan menayangkan film.

8.3 PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan untuk digunakan untuk pemakaian lain. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (*Form Interpreted*). Pada saat tersebut PHP adalah sekumpulan *script* yang digunakan untuk mengolah data form dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan

dari PHP/FI adalah *Personal Home Page/Form Interpreter*. Dengan pelepasan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini *interpreter* sudah diimplementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan

Pada tahun 1997, sebuah perusahaan bernama *Zend*, menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0. Pada pertengahan tahun 1999, *Zend* merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi. Pata Juni 2004, *Zend* merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

2.8 Video Streaming

Video dapat juga disebut sebagai gambar-gambar yang bergerak. Dalam video menampilkan sejumlah gambar atau *frame* dengan kecepatan tertentu yang disebut dengan istilah *frame rate*, yang dihitung dalam skala *frame per second* (fps). Seperti jenis data yang lain, data video juga dapat disimpan, disunting, ataupun dikirim melalui jaringan. Video *streaming* dapat diartikan sebagai suatu metode yang memanfaatkan *streaming server* untuk mentransmisikan video digital melalui suatu jaringan data sehingga memungkinkan video *playback* dapat langsung dilakukan tanpa perlu menunggu sampai proses *download* selesai ataupun menyimpannya terlebih dahulu di komputer *client*. Sistem video *streaming* melibatkan proses

encoding terhadap isi dari data video, dan kemudian mentransmisikan video *streaming* melalui suatu jaringan *wired* ataupun jaringan *wireless*, sehingga *client* tujuan dapat mengakses, melakukan *decoding*, dan memunculkan video tersebut secara *real-time*.

Video *streaming* dapat digunakan di banyak aplikasi, seperti *video conference*, e-*learning*/*distance learning*, video pengawas atau *video surveillance*, *remote monitoring*, dan lain-lain. Kehadiran jaringan yang memungkinkan *server* sebagai video *source* dapat terpisah jarak secara fisik dari *client* merupakan faktor pembeda video *streaming* dari *pre-recorded* video yang sudah umum digunakan oleh konsumen elektronik, seperti CD/DVD. Oleh sebab itu video *streaming* memiliki karakteristik utama. Teknologi *streaming* memungkinkan akses *real-time* ataupun *on-demand*. Data *streaming* ditransmisikan dari sisi *server*, *playback* segera dilakukan di sisi *client* setelah diterima dan diproses terlebih dahulu, dan tidak meninggalkan data residu di sisi *client* setelah proses *streaming* selesai.

Teknologi *streaming* cenderung bersifat *bandwidth-dependant*, sehingga sangat bergantung pada kondisi jaringan. Agar data *stream* dapat di-*playback* secara baik, perlu diperhatikan beberapa pertimbangan supaya data *stream* memiliki *bit rate/data transfer rate* yang cukup, karena dengan mengurangi *bit rate* berarti sama saja dengan mengirimkan lebih sedikit data. Mengurangi *bit rate* dapat dilakukan dengan cara membuat dimensi *frame* video menjadi lebih kecil, membuat jumlah *frame per second* (fps) video menjadi lebih rendah, ataupun dengan cara mengurangi jumlah informasi yang ada di setiap *frame* video melalui proses kompresi.

2.9 Web Sever

Menggunakan Apache *Web Server*, yang merupakan *web server open source* yang dikembangkan oleh Apache Group dan mendukung berbagai jenis bahasa pemrograman yang populer di internet, diantaranya HTML, Javascript dan PHP.

Web server digunakan untuk melayani data teks, gambar dan *hyperlink*. Metode yang digunakan ialah *streaming* dengan protokol HTTP/TCP. Secara *default* Apache menggunakan port 80 untuk berkomunikasi.

Keuntungan menggunakan Apache *Web Serv*er adalah:

1. Sederhana dan mudah digunakan
2. Dapat digunakan dengan jaringan yang ada
3. Tutorial penjelasan Apache sangat banyak di internet

2.10 CoDeC

CoDeC (*Encoder Decoder*) merupakan proses memampatkan atau *compress* dan menguraikan kembali atau *decompress*, dalam hal ini sinyal video dan audio digital. Prinsip utama kompresi adalah mengurangi ukuran dengan membuang informasi tertentu yang dianggap kurang signifikan sehingga lebih efisien saat dikirimkan melalui kanal komunikasi atau disimpan dalam media penyimpanan. Sedangkan *decompress* berfungsi melakukan rekonstruksi sinyal audio video agar sinyal tersebut dapat digunakan kembali [5]. Jenis format video hasil kompresi yang digunakan ialah H.264

H.264 adalah standar terbaru CoDeC yang berorientasi pada gerak-kompensasi. Standar codec ini dikembangkan oleh ITU-T Video Coding bersama ISO / IEC Moving Picture Expert Group (MPEG). Standar H.264 digunakan dalam aplikasi seperti Blu-ray Disc, video dari YouTube dan iTunes Store, DVB broadcast, siaran langsung layanan televisi satelit, layanan televisi kabel, dan real-time video conference.

2.11 Quality of Service (QoS)

QoS merupakan kemampuan suatu jaringan IP untuk menyediakan suatu layanan tertentu pada trafik data tertentu pada berbagai jenis *platform* teknologi. QoS

tidak diperoleh secara langsung dari infrastruktur yang dipakai, tetapi diperoleh dengan mengimplementasikannya pada jaringan yang bersangkutan.

Pada umumnya, ada beberapa faktor yang mempengaruhi QoS *streaming* multimedia, yaitu *delay, bandwith, packet loss, troughtput, jitter* dan *latency*. *Bandwith* merupakan daya tampung dari kabel jaringan untuk dapat dilewati oleh paket data. *Bandwith* juga bisa diartikan jumlah penggunaan paket data per satuan waktu, dinyatakan dalalm *bit per second* (bps). *Packet loss* merupakan jumlah paket yang hilang pada saat pengiriman, semakin baik suatu jaringan jika jumlah *packet loss* paling kecil. *throughput* adalah ukuran kecepatan transmisi data, dalam satuan kbps atau Mbps. *Jitter* adalah ukuran delay penerimaan paket. *Latency* merupakan waktu yang dibutuhkan dari transmisi ke penerima yang diukur dalam satuan milidetik [13].

MDI (*Media Delivery Index*) (RFC 4445) merupakan suatu perhitungan yang memantau pengiriman video pada jaringan, MDI memberikan indikator relatif terhadap kebutuhan *buffer* pada sisi *client* sesuai dengan *packet jitter* dan *packet loss*. Hal ini dilakukan dengan mengukur jaringan IP yang digunakan dan memberikan suatu beban tertentu terhadap sistem. MDI memerukan dua parameter utama, yaitu *Delay Factor* (DF) dan *Media Loss Rate* (MLR).

2.11.1 Delay Factor

Untuk memahami *Delay Factor* (DF), diperlukan pembahasan mengenai hubungan antara *Jitter* dan *Buffer* yang diperlukan. *Jitter* merupakan perubahan *end-to-end latency* terhadap waktu. Saat *client* menerima data dalam laju konstan, maka *jitter* bernilai nol, sedangkan saat laju data berubah-ubah, maka nilai *jitter* tidak nol.

Buffer merupakan jumlah data yang diperlukan untuk disimpan di sisi *client* untuk mengatasi *jitter*. Semakin tinggi *jitter*, maka diperlukan *buffer* yang semakin besar. Sistem ideal yang diinginkan adalah sisi *client* mempergunakan jumlah data yang sama dengan jumlah data yang diterima pada suatu waktu tertentu. Jika jumlah

data yang datang lebih kecil dari yang dibutuhkan, hal ini bisa disebabkan oleh gangguan jaringan, maka *playback content* akan terganggu. Hal ini dikenal dengan istilah *underflow*. Sebaliknya, saat data yang diterima *client* lebih besar dari yang diperlukan, maka akan terjadi *overflow*. Kedua keadaan ini tidak diinginkan karena dapat menurunkan QoS.

DF sebagai salah satu komponen MDI merupakan suatu nilai waktu yang mengindikasikan jumlah waktu yang diperlukan untuk melakukan *buffering* data dalam mengeliminasi *jitter*. DF dihitung dari jumlah paket data yang datang dan ditampilkan dalam satuan waktu. Berikut cara perhitungannya :

1. Setiap kedatangan paket, hitung perbedaan antara jumlah data yang diterima (*bytes_receive*) dan jumlah data yang digunakan (*bytes_drained*). Nilai ini dikenal sebagai MDI *virtual buffer depth* (Δ)

$$\Delta = | bytes_receive - bytes_drained | \quad (2.1)$$

2. Dalam suatu interval waktu tertentu, hitung perbedaan nilai maksimal dan minimal dan bagi dengan *bitrate* (*media_rate*)

$$DF = (\max(\Delta)-\min(\Delta)) / media_rate \quad (2.2)$$

Nilai DF yang termasuk kategori dapat diterima menurut *Agilent Technologhies* adalah 9 - 50 ms.

2.11.2 Media Loss Rate

Media Loss Rate (MLR) didefinisikan jumlah paket data yang hilang tiap detik. Setiap paket yang hilang dapat menyebabkan gangguan pada tampilan konten yang terkirim. Nilai maksimum MLR yang diharapkan adalah 0. Berikut adalah nilai rata-rata MLR yang dapat diterima :

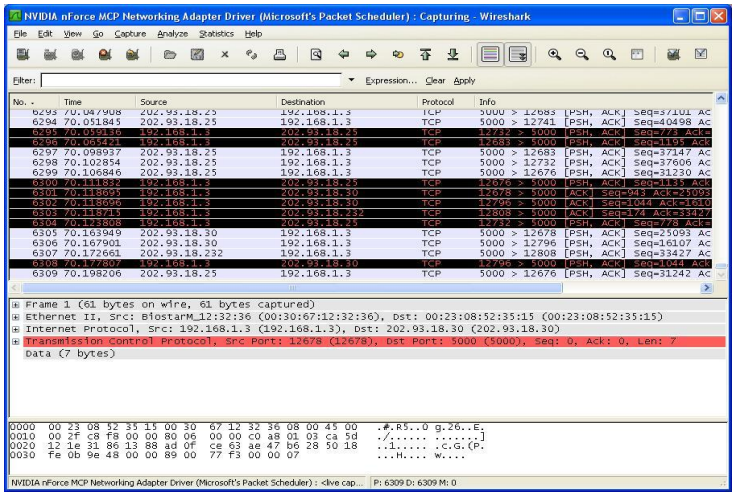
Tabel 2.3 Katagori *Media Loss Rate*

Jenis Layanan	MLR yang dapat diterima
SDTV	0,004
VoD	0,004
HDTV	0,0005

Selanjutnya gabungan data tersebut dikenal dengan *Media Delivery Index* (MDI) dan dipisahkan dengan tanda semicolon sesuai dengan RFC 4445.

2.12 Network Protocol Analyzer

Alat ukur yang digunakan dalam laporan ini ialah *network protocol analyzer* Wireshark. *Network protocol analyzer* adalah perangkat yang digunakan untuk mengetahui kondisi trafik yang ditransmisikan pada jaringan. Data paket trafik yang diperoleh dapat digunakan untuk menganalisis performa paket dan jaringan. Program yang digunakan adalah Wireshark. Wireshark merupakan perangkat lunak yang bersifat *open source* dapat di-*download* langsung dari internet. Wireshark yang digunakan untuk meng-*capture* paket dan tugas akhir ini adalah Wireshark versi 0.99.6a. Gambar 2.13 berikut adalah tampilan dari Wireshark:



Gambar 2.13 Tampilan Wireshark

Sebagai salah satu *network protocol analyzer*, tentu saja Wireshark memiliki beberapa fitur. Berikut merupakan fitur utama Wireshark

1. Multi *platform* bisa digunakan pada Unix dan Windows
2. *Open source* dan gratis
3. Dapat menampilkan dan menyimpan paket yang di-*capture*
4. Mendukung beberapa macam protokol jaringan. Protokol – protokol tersebut antara lain TCP, IP, RTP, UDP, RTCP, RTSP, dan lain lain.

Untuk mendapatkan data hasil pengukuran, dilakukan dengan meng-*capture* data yang diinginkan.

Hal – hal yang perlu diperhatikan dalam proses pengukuran ini adalah:

1. Jaringan yang digunakan LAN yang diimplementasikan untuk VoD.
2. Pengukuran dilakukan untuk mengetahui komponen nilai – nilai MDI, yaitu DF dan MLR.
3. Pengukuran dilakukan PC ke PC.
4. Alat ukur dipasang di setiap *client.*