BAB II

LANDASAN TEORI

2.1. Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom yang saling berhubungan satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, aplikasi, dan perangkat keras secara bersama-sama.

Jaringan komputer dibangun untuk membawa informasi secara tepat tanpa adanya kesalahan dari sisi pengirim (*transmitter*)maupun sisi penerima (*receiver*) melalui media komunikasi.

Jaringan komputer mempunyai beberapa manfaat yang lebih dibandingkan dengan komputer yang berdiri sendiri karena memungkinkan manajemen sumber daya lebih efisien. (Sukmaaji & Rianto, 2008)

2.1.1. Protokol

Protokol adalah sebuah aturan yang mendefinisikan beberapa fungsi yang ada dalam sebuah jaringan komputer, misalnya mengirim pesan, data, informasi, dan fungsi lain yang harus dipenuhi oleh pengirim dan penerima agar komunikasi dapat berlangsung dengan benar. Selain itu, protokol juga berfungsi agar komputer yang berada dalam jaringan berkomunikasi dengan bahasa yang sama.

Secara umum, fungsi protokol adalah menghubungkan pengirim dan penerima dalam berkomunikasi serta dalam bertukar informasi agar dapat berjalan dengan baik dan akurat. Fungsi protokol secara detail adalah sebagai berikut (Sukmaaji & Rianto, 2008):

1. *Fragmentasi* dan *Reassembly*

   Fragmentasi adalah membagi informasi yang dikirim menjadi beberapa paket data. Proses ini terjadi di sisi pengirim informasi. *Reassembly* adalah proses menggabungkan lagi paket-paket tersebut menjadi satu paket lengkap. Proses ini terjadi di sisi penerima informasi.

2. *Encapsulation*

   Fungsi dari *encapsulation* adalah melengkapi berita yang dikirimkan dengan *address*, kode-kode koreksi, dan lain-lain.

3. *Connection Control*

   Fungsi dari *connection control* adalah membangun hubungan komunikasi dari *transmitter* ke *receiver* termasuk di dalam pengiriman data dan mengakhiri hubungan.

4. *Flow Control*

   Flow control berfungsi mengatur perjalanan data dari *transmitter* ke *receiver*.

5. *Error Control*

   Pengiriman data tidak terlepas dari kesalahan, baik dalam proses pengiriman maupun penerimaan. Fungsi *error control* adalah mengontrol terjadinya kesalahan yang terjadi pada waktu data dikirimkan

6. *Transmission Service*

   Fungsi *transmission service* adalah memberi pelayanan komunikasi data khususnya yang berkaitan dengan prioritas dan keamanan serta perlindungan data.

2.1.2. TCP/IP

Jaringan komputer merupakan kumpulan dari beberapa komputer yang terhubung antara satu dengan yang lainnya dan memungkinkan untuk dapat saling berbagi *resource* (sumber daya). Ketika akan membuat sebuah jaringan komputer, anda membutuhkan sebuah protokol jaringan sebagai penghubung komunikasi data dari satu *client* ke *client* lainnya ketika bertukar data, informasi, dan pesan oleh *transmitter* (pengirim) dan *receiver* (penerima). Selain itu, protokol juga dijadikan sebagai acuan standar komunikasi antara dua sistem dengan platform berbeda yang memungkinkan adanya komunikasi. Protokol yang sering digunakan adalah protokol TCP/IP.

TCP/IP adalah salah satu perangkat lunak jaringan komputer yang terdapat dalam sistem dan dipergunakan dalam komunikasi data dalam *Local Area Network* (LAN) maupun internet. TCP singkatan dari *Transmission Control Protocol* dan IP singkatan dari *Internet Protocol*. TCP/IP menjadi satu nama karena fungsinya selalu bergandengan satu sama lain dalam komunikasi data.

TCP/IP saat ini dipergunakan dalam banyak jaringan komputer lokal (LAN) yang terhubung ke internet, karena memiliki sifat :

1. Merupakan protokol standar yang terbuka, gratis dan dikembangkan terpisah dari perangkat keras komputer tertentu. Karena protokol ini banyak didukung oleh vendor perangkat keras, sehingga TCP/IP merupakan pemersatu perangkat keras komputer yang beragam merk begitu juga sebagai pemersatu berbagai perangkat lunak yang beragam merk sehingga walau memakai

perangkat keras dan perangkat lunak komputer yang berlainan, komputer dan komputer lainnya dapat berkomunikasi data melalui internet

2. Berdiri sendiri dari perangkat keras jaringan apapun. Sifat ini memungkinkan TCP/IP bergabung dengan banyak jaringan komputer. TCP/IP bisa beroperasi melalui sebuah *Ethernet*, sebuah saluran *dial-up*, dan secara *virtual* melalui berbagai media fisik transmisi data.
3. Bisa dijadikan alamat umum sehingga tiap perangkat yang memakai TCP/IP akan memiliki sebuah alamat unik dalam sebuah jaringan komputer lokal, atau dalam jaringan komputer *global* seperti internet.

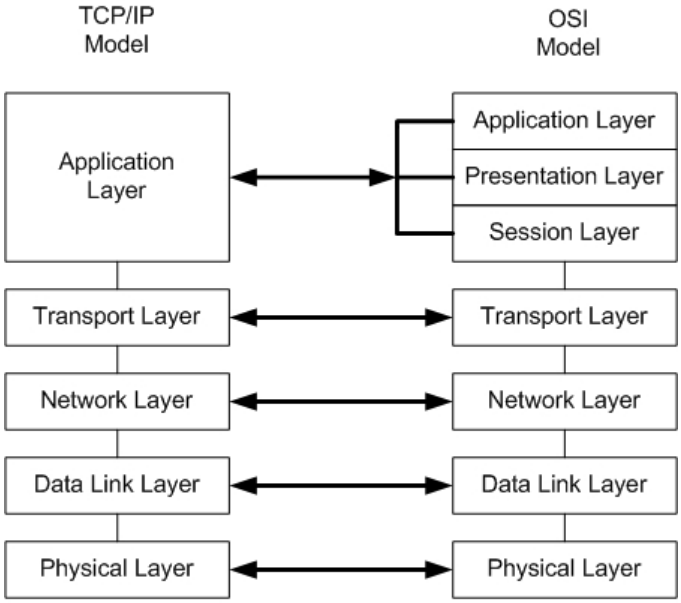
2.1.3. *Layer* Protokol TCP/IP

Berbicara mengenai protokol dan jaringan, tidak akan terlepas dari topik mengenai TCP/IP. TCP/IP yang merupakan komunikasi data yang dipakai oleh seluruh pengguna internet saat melakukan komunikasi data. TCP/IP juga biasa diartikan protokol yang dirancang secara standar untuk dapat digunakan pada berbagai jenis jaringan.

TCP/IP terbagi ke dalam beberapa lapisan (*layer*), antara lain :

1. Lapis pertama : *Physical Layer*
2. Lapis kedua : *Network Interface Layer*
3. Lapis ketiga : *Internet Layer*
4. Lapis keempat : *Transport Layer*
5. Lapis kelima : *Application Layer*

Hal ini sedikit berbeda dengan spesifikasi OSI *Layer* yang terdiri dari 7 *layer*. Perbedaannya dapat dilihat pada Gambar 2.1.



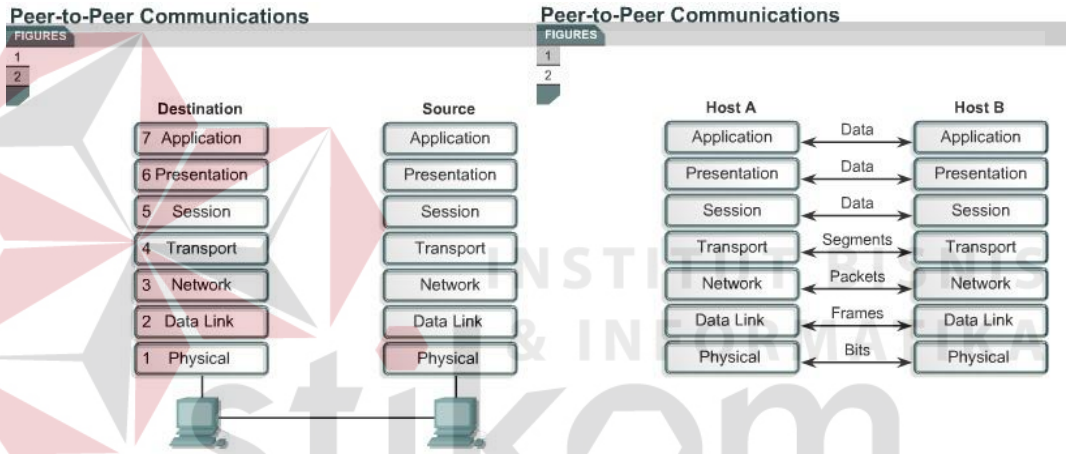
Gambar 2.1. Perbandingan TCP/IP dan OSI Model.

2.1.3.1. Model Referensi OSI

OSI memberikan pandangan yang "abstrak" dari arsitektur jaringan yang dibagi dalam 7 lapisan. Model ini diciptakan berdasarkan sebuah proposal yang dibuat oleh *International Standard Organization* (ISO) sebagai langkah awal menuju standarisasi protokol internasional yang digunakan pada berbagai *layer*. Model ini disebut *OSI Reference Model*, karena model ini ditujukan untuk interkoneksi *Open System*. *Open System* diartikan sebagai suatu sistem yang terbuka untuk berkomunikasi dengan sistem-sistem lain yang berbeda arsitektur maupun Sistem Operasi. Prinsip-prinsip yang digunakan bagi ketujuh *layer* tersebut adalah :

1. Sebuah *layer* harus dibuat bila diperlukan tingkat abstraksi yang berbeda.
2. Setiap *layer* harus memiliki fungsi tertentu.
3. Fungsi *layer* di bawah adalah mendukung fungsi *layer* di atasnya.

4. Fungsi setiap *layer* harus dipilih dengan teliti sesuai dengan ketentuan standar protokol internasional.
5. Batas-batas setiap *layer* diusahakan untuk meminimalkan aliran informasi yang melewati antarmuka.
6. Jumlah *layer* harus cukup banyak, sehingga fungsi-fungsi yang berbeda tidak perlu disatukan dalam satu *layer* di luar keperluannya. Akan tetapi jumlah *layer* juga harus diusahakan sesedikit mungkin sehingga arsitektur jaringan tidak menjadi sulit dipakai.



Gambar 2.2. *Peer to Peer Communication OSI Layer*.

Berikut adalah penjelasan dari masing-masing *layer* pada *OSI Layer* pada Gambar 2.2.

Layer-1 (Physical Layer)

Physical Layer atau lapisan fisik melakukan fungsi pengiriman dan penerimaan *bit stream* dalam medium fisik. Dalam lapisan ini kita akan mengetahui spesifikasi mekanikal dan elektrikal dari media transmisi serta antar mukanya. Hal-hal penting yang dapat dibahas lebih jauh dalam lapisan fisik ini adalah :

1. Karakteristik fisik dari media dan antarmuka.

2. Representasi bit-bit. Dalam hal ini lapisan fisik harus mampu menerjemahkan bit 0 atau 1, termasuk pengkodean dan bagaimana mengganti sinyal 0 ke 1 atau sebaliknya.
3. *Data rate* (laju data).
4. Sinkronisasi bit.
5. Line configuration (konfigurasi saluran). Misalnya : *point-to-point* atau p*oint-to-multipoint configuration*.
6. Topologi fisik. Misalnya : *mesh topology*, *star topology*, *ring topology*, *bus topolog*y.
7. Mode transmisi. Misalnya : *half-duplex* mode, *full-duplex* (*simplex*) mode.

Layer-2 (Data Link)

Pada Layer-2 (*Data Link Layer*) komunikasi data dilakukan dengan menggunakan identitas berupa alamat simpul fisik yang disebut sebagai alamat *hardware* atau *hardware address*. Proses komunikasi antara komputer atau simpul jaringan hanya mungkin terjadi, bila kedua belah pihak mengetahui identitas masing-masing melalui alamat fisik (*physical address*).

Tugas utama lapisan utama data link dalam proses komunikasi data adalah :

1. *Framing* : membagi *bit stream* yang diterima dari lapisan *network* menjadi unit-unit data yang disebut *frame*.
2. *Physical Addressing*, definisi identitas pengirim dan/atau penerima yang ditambahkan dalam *header*.
3. *Flow Control* : melakukan tindakan untuk membuat stabil laju *bit* jika *rate* atau laju *bit stream* berlebih atau berkurang.

4. *Error Control* : penambahan mekanisme deteksi dan retransmisi *frame-frame* yang gagal terkirim.
5. *Communication control* : menentukan *device* yang harus dikendalikan pada saat tertentu jika ada dua koneksi yang sama.

Layer-3 (Network Layer)

Pada lapisan ini terjadi proses pendefinisian alamat logis (*logical addressing*), kemudian mengkombinasikan multiple data link menjadi satu *network*. Lapisan *network* bertanggung jawab untuk membawa paket dari satu simpul ke simpul lainnya dengan mengacu pada *logical address*. Fungsi lain adalah sebagai *packet forwarder* (penerus). Lapisan *Network* sebagai *packet forwarder* mengantarkan paket dari sumber (*source*) ke tujuan (*destination*) yang disebut dengan istilah *routing*.

Ada dua tugas pokok lapisan network yaitu :

1. *Logical addressing* : pengalamatan secara logis yang ditambahkan pada *header* lapisan network. Pada jaringan TCP/IP pengalamatan logis ini populer dengan sebutan *IP address*.
2. *Routing*. Hubungan antarjaringan yang membentuk internetwork membutuhkan metode jalur alamat agar paket dapat dikirim dari satu *device* yang berasal dari jaringan satu menuju *device* lain pada jaringan yang lain. Fungsi *routing* didukung oleh *routing protocol* yaitu protokol yang bertujuan mencari jalan terbaik menuju tujuan dan tukar-menukar informasi tentang topologi jaringan dengan *router* yang lainnya.

Layer-4 (Transport Layer)

Lapisan *Transport* bertanggung jawab terhadap pengiriman *source-to-destination* (*end-to-end*) yang dapat dijelaskan sebagai berikut :

1. *Service-point Addressing*. *Transport Layer* tidak hanya menangani pengiriman *source-to-destination*, namun lebih spesifik kepada pengiriman jenis *message* untuk aplikasi yang berlainan. Setiap *message* yang berlainan aplikasi harus memiliki alamat tersendiri yang disebut *service point address* atau yang lebih umum disebut *port address*
2. *Segmentation dan Reassembly*. Sebuah *message* dibagi dalam segmen-segmen yang terkirim. Setiap segmen memiliki *sequence number* yang berguna bagi lapisan *transport* untuk merakit (*reassembly*) segmen-segmen yang terpecah menjadi *message* yang utuh.
3. *Connection Control*. Pada lapisan *transport* terdapat dua kondisi yakni *connectionless* atau *connection-oriented*. Fungsi dari *connection control* adalah mengendalikan kondisi tersebut.
4. *Flow Control*. Seperi halnya lapisan *data link*, lapisan *transport* bertanggung jawab untuk melakukan kontrol aliran (*flow kontrol*). Bedanya dengan *flow control* di lapisan *data link* adalah dilakukna untuk *end-to-end*.
5. *Error Control*. Fungsi tugas ini sama dengan tugas *error control* di lapisan *data link*, namun berorientasi e*nd-to-end*.

Dalam jaringan berbasis TCP/IP protokol yang terdapat pada lapisan ini adalah *Transmission Control Protocol* (*TCP*) dan *User Datagram Protocol* (*UDP*).

Layer-5 (Session Layer)

Lapisan sesi membuka, merawat, mengendalikan dan melakukan terminasi hubungan antarsimpul. Lapisan aplikasi dan presentasi melakukan *request* dan menunggu *response* yang dikoordinasikan oleh lapisan di atasnya, misalnya :

1. RPC (*Remote Procedure Call*) : Protokol yang mengeksekusi program pada komputer *remote* dan memberikan nilai balik kepada komputer lokal sebagai hasil eksekusi tersebut.
2. Netbios API : *Session Layer Application Programming Interface*.
3. NFS (*Network File System*)
4. SQL (*Structured Query Language*)

Layer-6 (Presentation Layer)

Berfungsi untuk mentranslasikan data yang akan ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. Lapisan presentasi melakukan *coding* dan konversi data.

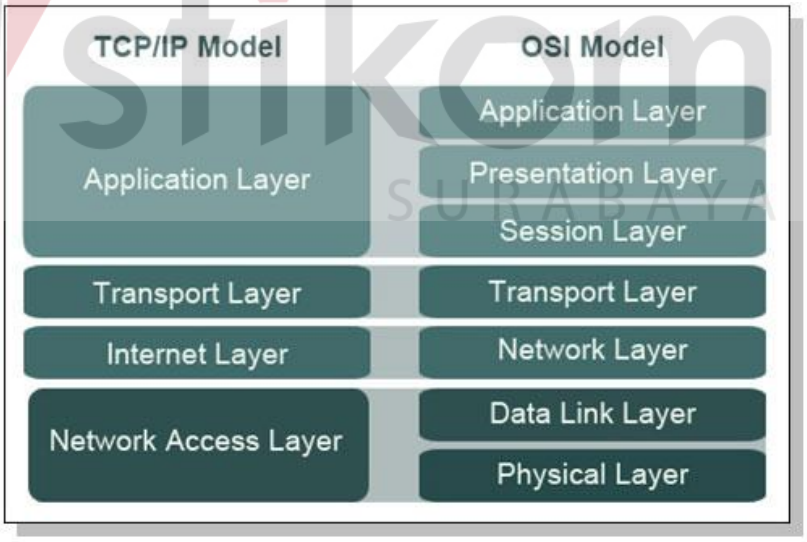
Layer-7 (Application Layer)

Aplikasi adalah layanan/*service* yang mengimplementasikan komunikasi antarsimpul. *Application Layer* berfungsi sebagai antamuka antara aplikasi dengan fungsionalitas jaringan yang mengatur bagaimana aplikasi dapat mengakses jaringan dan membuat pesan-pesan kesalahan. Beberapa hal yang dilakukan oleh lapisan aplikasi adalah mengidentifikasi mitra komunikasi, aplikasi transfer data, *resource availability*, dan lapisan aplikasi terkait dengan aplikasi *end-user*. (Sukmaaji & Rianto 2008)

2.1.3.2. Model Referensi TCP/IP

TCP bertugas menerima pesan elektronik dengan panjang sembarang dan membaginya ke dalam bagian-bagian berukuran 64 kb (kilo *bit*). Dengan membagi pesan menjadi bagian-bagian, perangkat lunak yang mengontrol komunikasi jaringan dapat mengirim tiap bagian dan menyerahkan prosedur pemeriksaan bagian demi bagian. Apabila suatu bagian mengalami kerusakan selama transmisi, maka program pengirim hanya perlu mengulang transmisi bagian itu dan tidak perlu mengulang dari awal.

IP mengambil bagian-bagian, memeriksa ketepatan bagian-bagian, pengalamatan ke sasaran yang dituju, dan memastikan apakah bagian-bagian tersebut sudah dikirim sesuai dengan urutan yang benar. IP memiliki informasi tentang berbagai skema pengalamatan yang berbeda-beda. Dari Gambar 2.3. dapat dilihat perbedaan *layer* antara model TCP/IP dengan model OSI.



Gambar 2.3. TCP/IP dan *OSI Layer*

Internet Layer

Internet layer menentukan format paket dan protokol resmi yang disebut IP. Tugas *internet layer* adalah mengirimkan paket-paket IP yang berisi informasi

tujuan paket tersebut. Di sini diperlukan *routing packet*, sebab adanya *routing packet* dapat menghindarkan terjadinya kemacetan pada waktu transmisi data. Secara tidak langsung, kita bisa melihat bahwa *internet layer* fungsinya hampir sama dengan *network layer* pada model OSI.

Transport Layer

Layer yang berada di atas *internet layer* pada model TCP/IP adalah *transport layer*. Ada dua jenis *transport layer*, yaitu *Transmission Control Protocol* yang mempunyai fungsi untuk memecah data menjadi paket-paket dan meneruskannya ke *internet layer* dan *User Datagram Protocol* yang merupakan protokol yang tidak bisa diandalkan bagi aplikasi-aplikasi yang tidak memerlukan pengurutan TCP. (Sukmaaji & Rianto, 2008)

1. TCP

TCP berfungsi untuk mengubah suatu blok data yang besar menjadi segmen-segmen yang diberi nomor dan disusun secara berurutan agar penerima dapat menyusun kembali segmen-segmen tersebut seperti pada waktu pengiriman. TCP ini adalah jenis protokol *connection-oriented*, dengan kata lain protokol memberikan layanan yang bergaransi.

2. UDP

UDP adalah jenis protokol *connectionless-oriented*. UDP bergantung pada lapisan atasnya untuk mengontrol kebutuhan data. UDP banyak digunakan untuk aplikasi-aplikasi yang tidak peka terhadap gangguan yang terjadi pada jaringan.

Application Layer

Model TCP/IP tidak memiki *session layer* dan *presentation layer*. *Application layer* terdapat di puncak model TCP/IP. Layer ini berisi bermacam-macam

protokol tingkat tinggi, yaitu *telnet*, ftp, smtp, dns, http, dan www. (Sukmaaji & Rianto, 2008)

2.2. Video *Streaming*

Streaming merupakan sebuah teknologi yang digunakan untuk memainkan sebuah file *audio* atau *video* secara langsung maupun dengan *prerecord* yang berada di *web server*. Ada 3 jenis cara data multimedia dapat ditransmisikan dalam internet. (Prasetiya, 2008)

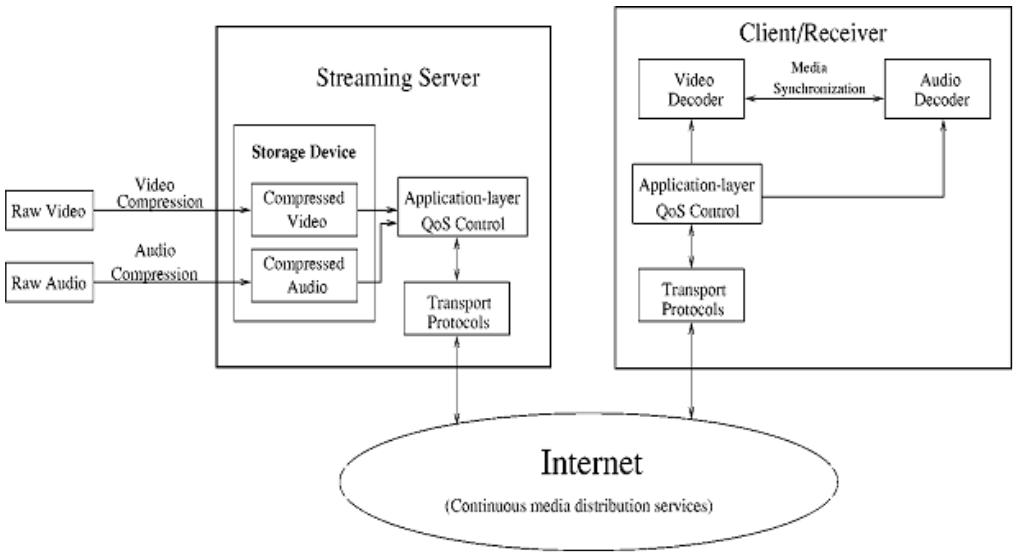
1. *Download mode*, klien dapat memainkan media setelah semua file media telah dilakukan proses *download* dari *server* ke komputernya
2. *Streaming mode*, klien dapat memainkan media secara langsung tanpa melakukan proses *download*. Bagian media yang diterima melalui proses transmisi dapat langsung dimainkan seketika itu juga.
3. *Progressive download*, media dapat dimainkan beberapa detik setelah proses *download* dimulai atau klien dapat melihat media selagi media itu dalam proses *download*. Secara langsung terlihat seperti *streaming* tetapi kenyataannya adalah melakukan *download*.

Streaming video, juga dikenal sebagai "media *streaming*" atau "*Video online*," adalah proses penyampaian *video* dan *audio* klip di atas protokol (IP) jaringan internet. Ada berbagai metode teknis untuk melakukannya, tapi poin terpenting adalah bahwa aliran *video* berasal dari *server* pusat, atau beberapa *server*, dan dikirimkan ke beberapa pengguna yang melihatnya pada komputer mereka, perangkat *mobile* atau televisi. Ada 3 tipe *video streaming* menurut bentuk layanannya. (Prasetiya, 2008)

1. *Video-on-Demand* (VoD), suatu bentuk *streaming* pada permintaan data yang sudah ada atau tersimpan dalam *server*. VoD mengijinkan pengguna untuk dapat melakukan proses *pause*, *rewind*, *fast forward* atau melakukan indeks isi multimedia.
2. *Live streaming*, aplikasi *live streaming* dapat dijumpai dalam teknologi *broadcast* radio dan televisi. Aplikasi ini mengijinkan pengguna untuk menerima siaran radio dan televisi secara langsung (*live*). Dalam *live streaming* tidak ada data *video* yang disimpan di dalam *server* sehingga klien tidak dapat melakukan proses *fast forward* dalam media yang diakses. Proeses *capture* dan *encoding* secara langsung dilakukan sesuai dengan format *video*-nya sebelum *video* itu ditransmisikan kepada klien.
3. *Real-time streaming*, aplikasi ini mengijinkan pengguna untuk berkomunikasi dengan *audio* atau *video* dalam waktu yang riil. Contohnya adalah komunikasi tatap muka langsung melalui internet atau sering disebut dengan komunikasi *video conference*.

Konsep dasar dari *video streaming* yaitu membagi paket *video* ke dalam beberapa bagian (dipecah), dan mentransmisikan paket-paket tersebut, kemudian dari sisi penerima (*client*) dapat men-*decode* dan memainkan potongan paket file *video* tanpa harus menunggu seluruh file terkirim ke mesin penerima. (Siadari, 2010)

Sistem *streaming* tersusun dari kombinasi *server*, *player*, transmisi dan metode *encoding* yang digunakan. Berikut ini bagian hubungan setiap komponen penyususun sistem *streaming* yang dijelaskan pada Gambar 2.4.



Gambar 2.4. Arsitektur *Video Streaming*.

Data mentah *video* dan *audio* sebelum terkompresi oleh algoritma kompresi *video* dan *audio*, kemudian disimpan dalam perangkat penyimpanan. Berdasarkan permintaan klien, *streaming* server mengambil kompresi *video* dan *audio* dari perangkat penyimpanan dan kemudian modul *application layer* kontrol QoS menyesuaikan *video* / *audio bit-stream* sesuai dengan status jaringan dan persyaratan QoS. Setelah adaptasi, transport protokol memaketkan *bit-stream* yang dikompresi dan mengirimkan paket *video*/*audio* ke Internet. Paket mungkin dapat di-*drop* atau mengalami *delay* berlebihan karena kemacetan dalam jaringan dan karenanya untuk meningkatkan kualitas media pengiriman, layanan distribusi media terus-menerus dikerahkan dalam jaringan. Pada sisi klien, berbagai *bit-stream* yang diterima dalam bentuk paket perlu disinkronkan antara satu sama lain. (Fajrien, 2012)

2.3. *Network Protocol Analyzer*

Jaringan protokol analisis adalah proses untuk sebuah program atau perangkat untuk memecahkan kode *header* protokol jaringan dan *trailer* untuk memahami data dan informasi di dalam paket dienkapsulasi oleh protokol. Untuk melakukan analisis protokol, paket harus ditangkap pada *real time* untuk analisis jalur kecepatan atau analisis nanti. Program atau perangkat disebut *Analyzer Protocol*.

Dalam arsitektur jaringan yang khas, pendekatan berlapis yang digunakan untuk merancang protokol jaringan dan komunikasi. Jaringan yang paling populer arsitektur model referensi disebut model OSI. Protokol pada satu lapisan harus berkomunikasi dengan protokol pada lapisan yang sama. Fungsi utama dari *Network Protocol Analyzer* adalah untuk *decode* protokol di setiap lapisan. Protokol Informasi dari beberapa lapisan dapat digunakan oleh *Network Protocol Analyzer* untuk mengidentifikasi kemungkinan masalah dalam komunikasi jaringan, yang disebut Ahli Analisis. Analisa protokol dapat men-*decode* protokol lapisan ganda dan paket untuk kembali membangun paket tingkat yang lebih rendah (seperti tingkat *Link*, IP atau TCP) ke tingkat yang lebih tinggi (seperti tingkat aplikasi) pesan untuk pemahaman mendalam tentang lalu lintas jaringan dan aktivitas pengguna. Teknik ini digunakan dalam analisa protokol ketika lalu lintas jaringan pemantauan dan pengawasan pengguna adalah tujuan utama.

Network Protocol Analyzer dapat digunakan baik untuk manajemen jaringan yang sah atau untuk mencuri informasi dari jaringan. Jaringan operasi dan personil pemeliharaan menggunakan *Network Protocol Analyzer* untuk

memonitor lalu lintas jaringan, menganalisis paket, menonton pemanfaatan sumber daya jaringan, melakukan analisis forensik dari pelanggaran keamanan jaringan dan memecahkan masalah jaringan. Analisa protokol yang tidak sah bisa sangat berbahaya bagi keamanan jaringan karena mereka hampir mustahil untuk mendeteksi dan dapat dimasukkan hampir di mana saja.

Network Protocol Analyzer yang biasa digunakan adalah *TCPdump* dan *Wireshark*, tetapi pada penelitian kali ini yang digunakan adalah *Wireshark*. *Wireshark* digunakan untuk penelitian ini karena lebih *user friendly* atau GUI (*Graphic User Interface*), dan juga *Wireshark* juga bisa digunakan di Sistem Operasi *Linux* dan *Windows*, sedangkan *TCPdump* hanya bisa digunakan di Sistem Operasi *Linux*.

2.3.1. Wireshark

Wireshark merupakan *Network Protocol Analyzer*, juga termasuk salat satu *network analysis tool* atau *packet sniffer*. *Wireshark* memungkinkan pengguna mengamati data dari jaringan yang sedang beroperasi atau dari data yang ada di *disk*, dan langsung melihat dan mensortir data yang tertangkap, mulai dari informasi singkat dan detail bagi masing-masing paket termasuk *full header* dan porsi data, dapat diperoleh. *Wireshark* memiliki beberapa fitur termasuk *display* filter *language* yang banyak dan kemampuan me-reka ulang sebuah aliran pada sesi TCP.

Packet sniffer sendiri diartikan sebuah *tool* yang berkemampuan menahan dan melakukan pencatatan terhadap *traffic* data dalam jaringan. Selama terjadi aliran data dalam jaringgan, *packet sniffer* dapat menangkap *Protocol Data Unit* (PDU), melakukan *decoding* serta analisis terhadap isi paket. *Wireshark* sebagai

salah satu *packet sniffer* yang diprogram demikian agar mengenali berbagai macam protokol jaringan. *Wireshark* juga mampu menampilkan hasil enkapsulasi dan *field* yang ada di dalam PDU.

2.3.2. *TCPdump*

TCPdump adalah *tools* yang berfungsi meng-*capture*, membaca atau men-*dumping* paket yang sedang ditransmisikan melalui jalur TCP.

TCPdump diciptakan untuk menolong *programer* ataupun *administrator* dalam menganalisa dan *troubleshooting* aplikasi *networking*. Seperti pisau yang bermata dua (hal ini sering kali disebut-sebut), *TCPdump* bisa digunakan untuk bertahan dan juga bisa digunakan untuk menyerang.

Utility ini juga seringkali digunakan oleh para *cracker* untuk melaksanakan perkerjaannya, karena *TCPdump* bisa meng-*capture* atau men-*sniff* semua paket yang diterima oleh *network interface*, Sama halnya dengan tujuan diciptakannya *TCPdump*, dalam artikel ini saya akan coba membahas bagaimana *TCPdump* digunakan untuk menganalisa koneksi yang terjadi antara dua sistem.

Langkah pertama yang harus anda lakukan adalah menginstal *TCPdump* pada *box* anda. Anda bisa mendapatkan *TCPdump* di http://www.tcpdump.org. *TCPdump* membutuhkan *libpcap* (*packet capture library*) yang juga tersedia di situs *TCPdump*. *Libpcap* harus diinstal terlebih dahulu dalam mesin anda. Tentunya anda juga membutuhkan *software* lain untuk mengkompile *TCPdump* dan *libpcap* seperti *gcc*.

2.4. Pengukuran QoS (Quality of Service)

Suatu jaringan dapat disebut ideal apabila mampu mengirimkan informasi apapun, tidak terbatas jumlah dan ukuran, serta tanpa menimbulkan *delay* ataupun *loss*. Akan tetapi dalam prakteknya akan sangat sulit untuk menciptakan jaringan dengan karakteristik seperti itu, karena *bit error*, bit *loss*, *delay*, *latency*, dan terbatasnya *bandwidth* merupakan hal-hal yang bersifat temporal. Faktor performansi dari sistem *video streaming* dalam hubungannya dengan jaringan dapat dijelaskan sebagai berikut:

Delay adalah waktu yang dibutuhkan oleh sebuah paket data untuk menempuh jarak dari asal ke tujuan, terhitung dari saat pengiriman oleh *transmitter* sampai saat diterima oleh *receiver*. (Rifiani dkk, 2011) Semakin kecil waktu *delay*, maka akan semakin baik kualitas *streaming*. *Delay* tidak boleh lebih dari 4 atau 5 detik.

$$\text{Waktu tunda (t)} = (T_r - T_s) \text{ detik} \quad (1)$$

$$0 \leq t \leq T$$

dimana : T_r = Waktu penerimaan paket (detik)

T_s = Waktu pengiriman paket (detik)

T = Waktu simulasi (detik)

t = Waktu pengambilan sampel (detik)

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 *bandwidth* dan menimbulkan antrian. Selain itu kecepatan terima dan kirim paket dari setiap *node* juga dapat menyebabkan *jitter*.

Jitter adalah variasi *delay*, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan. *Jitter* dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter* -nya akan semakin besar. (Clark, 2003)

$$\boldsymbol{\sigma^2 = \frac{\sum(X_{i+1} - X_i)}{N}} \quad (2)$$

dimana : X_i = jumlah *delay* sesi ke-i

N = banyaknya *jitter* yang terjadi

***Packet Loss*.** Hilangnya sebagian dari data *video* yang dikirimkan melalui jaringan dapat disebabkan oleh banyak hal, seperti *congestion*, penolakan oleh sistem karena *delay* yang terlalu lama, ataupun kesalahan pada jaringan itu sendiri. Suatu sistem *video streaming* tidak dapat mengabaikan kemungkinan terjadinya data *error* ataupun data *loss* selama proses transmisi, karena akan mengakibatkan kualitas yang buruk dari *video* pada saat ditampilkan. *Loss* yang terjadi pada jaringan dapat mengakibatkan keadaan yang lebih parah di sisi *client*, misalkan paket data pertama dari *frame-frame video* yang ditransmisikan itu rusak atau hilang selama berada di jaringan, maka seluruh paket data sisanya tidak akan dapat ditampilkan meskipun berhasil dikirimkan dan diterima oleh *client*. *Packet loss ratio* (PLR) untuk standar *streaming* adalah seperti Gambar 2.5 berikut (Prasetiya, 2008):

$$\text{PLR} \begin{cases} \leq 5\% & streaming \text{ berada dalam batasan normal QoS} \\ > 5\% & streaming \text{ tidak berada dalam standar QoS} \end{cases}$$

Gambar 2.5. Perbandingan nilai PLR.

Packet loss adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Paket hilang terjadi ketika satu atau lebih paket data yang melewati

suatu jaringan gagal mencapai tujuannya. (Rifiani dkk, 2011)

$$Packet\ loss = \left(\frac{Pd}{Ps}\right) \text{x } 100\% \quad (3)$$

$$0 \leq t \leq T$$

dimana : Pd = Paket yang mengalami *drop* (paket)

Ps = Paket yang dikirim (paket)

T = Waktu simulasi (detik)

t = Waktu pengambilan sampel (detik)

Utilisasi merupakan parameter yang menunjukkan seberapa besar prosentase suatu sumber daya yang digunakan. Dalam hal ini sumber daya yang dimaksud adalah *bandwidth* suatu *link* yang menghubungkan antara kedua sisi yaitu sisi pelanggan dan *provider*.

Bandwidth, besarnya kapasitas yang dapat ditransmisikan dalam jaringan. *Bandwidth* sangat berpengaruh dalm pengiriman paket *video streaming*. *Bandwidth* berpengaruh untuk tipe format *video* dan *video bit-rate* yang ditransmisikan. Semakin besar *bandwidth*, maka semakin baik kualitas pengiriman *video*-nya.

Utilisasi *bandwidth* menunjukan rasio antara ukuran *bandwidth* total terpakai oleh pelanggan dengan *bandwidth* yang tersedia, sehingga bila dirumuskan akan terlihat seperti dibawah ini (Muslim, 2007) :

$$utility = \frac{rate\_bit}{bandwidth} \text{x } 100\% \quad (4)$$

dimana :

rate_bit = merupakan laju *bit* (*bandwidth*), total paket yang terpakai oleh pelanggan pada satu waktu (bps).

bandwidth = merupakan jumlah besaran *bandwidth* yang tersedia (bps).

Parameter-parameter tersebut di atas merupakan parameter utama QoS pada jaringan untuk *video streaming*. Hampir semua jaringan tidak memliliki mekanisme khusus (QoS *control*) untuk memprioritaskan sumber daya jaringan untuk memfasilitasi data *stream* sebagai prioritas utama yang *time-sensitive*. Karena ketidaktersediaan QoS *control*, sistem *video streaming* biasanya bersifat *end-based*, dalam artian bahwa jaringan memang tidak diharapkan untuk dapat menyediakan dukungan apapun demi terjaminnya proses transmisi.