BAB II
TINJAUAN PUSTAKA

2.1 Umum

Kemajuan teknologi informasi telah mengubah cara kita belajar, menikmati hiburan dan lain sebagainya. Salah satunya adalah naiknya tren penggunaan *streaming* VoD. Namun kemajuan teknologi ini tidak hanya berdampak positif namun juga bisa berdampak negatif salah satunya adalah adanya pencurian data pribadi pengguna dengan memanfaatkan celah keamanan di jaringan Internet.

Salah satu cara untuk meminimalisir tindak kejahatan yang memanfaatkan jaringan Internet adalah dengan melakukan *filtering port* Internet. Menurut *Cox Communications* perusahaan telekomunikasi asal Amerika Serikat, ada beberapa alasan kenapa perlu dilakukannya *filtering port* yakni :

1. Melindungi pelanggan, Perusahaan dapat melindungi dari adanya *worm* dan layanan berbahaya lainnya pada komputer pelanggan mereka.
2. Melindungi *bandwidth upstream*, jika komputer pelanggan terinfeksi *virus* atau *worm* dapat menurunkan kualitas layanan pelanggan lain yang ada pada *node* yang sama.
3. Melindungi keseluruhan jaringan Internet, beberapa filter mencegah terjadinya serangan pada komputer lain melalui jaringan Internet.

Penyedia layanan VoD tentu tidak ingin kehilangan pelanggan karena pelanggan mereka tidak bisa mengakses konten *video* akibat adanya pemblokiran nomor *port* Internet. Untuk mengatasi hal tersebut penyedia layanan harus dapat menyajikan konten mereka melalui *port well-known* yang jarang terkena *filtering* seperti *port* untuk HTTP dan HTTPS yakni *port* 80 dan 443.

Namun belum ada hasil penelitian untuk melewatkan konten VoD menggunakan *port non-standar* IANA. Oleh karena itu diperlukan penelitian mengenai melewatkan VoD menggunakan *port non-standar* IANA. Kemudian dilaksanakan pengujian kinerja VoD berdasarkan parameter QoS jaringan dan perbandingan dengan rekomendasi ITU-T agar menjadi bahan acuan bagi penelitian dan pengembangan berikutnya untuk pengadaan layanan VoD.

2.2 *Cloud Computing*

Teknologi informasi telah menjadi bagian penting dalam sebuah perusahaan untuk menunjang keberhasilan bisnis. *Cloud computing* sebagai salah satu teknologi informasi telah dianggap mendukung kelancaran bisnis perusahaan dan menjadikan aktivitas kerja menjadi lebih efisien. *Cloud computing* adalah suatu bentuk evolusi lanjutan dari Internet yang menggabungkan pemanfaatan teknologi komputer dan pengembangan berbasis Internet.

National Institute of Standards and Technology (NIST) menyatakan bahwa *cloud computing* adalah suatu model yang memungkinkan pemberian akses darimana saja berdasarkan permintaan akses jaringan ke media tertentu yang dapat digunakan secara bersama-sama untuk dimanfaatkan sumber daya komputasinya. Sumber daya yang dimaksud meliputi aplikasi, tempat penyimpanan, jaringan, komputasi dan pengembangan (Hurwitz et al,2012).

Cloud computing erat kaitannya dengan komputasi paralel dan terdistribusi. *Cloud computing* adalah pengembangan dari *client server* yang terhubung dengan ratusan komputer bahkan lebih dimana pengguna menjalankan program melalui *remote desktop* pada komputer pengguna dan proses komputasi dilakukan menggunakan *server* yang ada di *cloud* (Marinescu, 2013).

2.2.1 Jenis Layanan *Cloud Computing*

Cloud computing dibagi menjadi tiga berdasarkan jenis layanannya, yakni :

1. *Software as a Service* (SaaS) – lapisan teratas dari arsitektur *cloud computing* dimana pengguna dapat secara langsung menggunakan *software* secara instan tanpa mengelola atau mengendalikan infrastruktur *cloud* secara langsung, termasuk jaringan, *server*, sistem operasi, media penyimpanan,. Contoh dari layanan ini adalah layanan *email* publik (Gmail, Ymail, dsb), *social network*, (Facebook, Twitter, dsb), *instant messsaging* (Line, Whatsapp, dsb) dan *productivity software* (Office 365, Adobe Creative *Cloud*, dsb).
2. *Platform as a Service* (Paas) – layanan *middleware* dimana pengguna dapat membangun, menguji dan menyebarkan aplikasi yang sedang dalam tahapan pengembangan. Para pengguna tidak mengelola infrastruktur, jaringan, *server*,

sistem operasi, ataupun media penyimpanan, tetapi pengguna memiliki hak pengendalian secara penuh terhadap aplikasi yang sedang digunakan. Contoh penyedia layanan PaaS ini adalah Amazon Web Service, Windows Azure, Google AppEngine, dsb.

3. *Infrastructure as a Service* (Iaas) - lapisan terbawah dari *cloud delivery service* dimana pengguna dapat memilih dan mengelola infrastruktur IT seperti kemampuan komputasi, kapasitas penyimpanan data, RAM, dan jaringan serta sumber daya lain seperti sistem operasi dan aplikasi. Contoh penyedia layanan IaaS ini adalah Amazon EC2, Windows Azure , Google Compute Engine, dsb.

2.2.2 Karakteristik *Cloud Computing*

Pada umumnya *cloud computing* memiliki tiga karakteristik yang dimiliki teknologi sebelumnya (jubilee, 2010), yakni :

1. Metode komputasi yang disebut *Grid Computing*. Pada metode ini sebuah komputer raksasa *virtual* yang terdiri dari beberapa komputer menjalankan sejumlah besar kegiatan sekaligus.
2. Konsep komputasi yang disebut *Utility Computing*. Pada konsep ini pengguna hanya membayar layanan tertentu yang digunakannya.
3. Konsep komputasi yang disebut *Automatic Computing*. Pada konsep ini komputer beserta sistem infrastruktur komputasi memiliki kemampuan memperbaiki diri sendiri (*self-healing*). Misalnya saat *harddisk server* mendekati penggunaan 90% maka secara otomatis *server* akan mengaktifkan *harddisk* cadangan agar *server* tidak mengalami kesulitan penyimpanan.

2.2.3 Keuntungan *Cloud Computing*

Penggunaan *cloud computing* memberikan sejumlah keuntungan bagi sebuah perusahaan, antara lain :

1. Murah, Setiap tahun sebuah perusahaan biasanya selalu menghadapi masa-masa mengeluarkan sejumlah dana untuk melakukan pembelian *software* dan *hardware*. Dengan penggunaan teknologi *Cloud Computing* ini, maka sebuah perusahaan bisa menghemat biaya dengan penggunaan satu terminal yang sederhana dan tahan lama. Selain itu, perusahaan juga tidak perlu membeli lisensi untuk aplikasi yang

digunakan dan tidak perlu mempekerjakan sumber daya manusia yang secara khusus mengawasi penyimpanan data, *server*, jaringan, dan penggunaan aplikasi bisnis yang ada.

2. Ramah lingkungan, dengan penggunaan *Cloud Computing*, maka jumlah komputer bisa dikurangi sebab tidak lagi dibutuhkan. Selain aktivitas mencetak yang rnembutuhkan kertas dan tinta juga bisa ditekan karena aktivitas berbagi informasi bisa dilakukan melalui Internet. Nah, hal-hal seperti inilah yang membuat penggunaan *Cloud Computing* lebih ramah lingkungan.
3. Operasional dan manajemen lebih mudah, *Cloud Computing* membuat operasional dan manajemen lebih mudah karena sistem pribadi atau perusahaan yang terkoneksi dalam sebuah *Cloud* dapat dimonitor dan diatur dengan mudah.
4. Kolaborasi yang dilakukan lebih terjamin dan efisien.

2.2.4 Kerugian *Cloud Computing*

Meskipun memiliki sejumlah keuntungan yang menggiurkan *Cloud Computing* juga memiliki beberapa kerugian atau kendala, antara lain :

1. Level layanan, dalam penggunaan *Cloud Computing* ada kemungkinan *provider* tidak konsisten dengan performa aplikasi atau transaksi. Selain itu anda diharuskan untuk memahami *level* layanan yang meliputi waktu respon transaksi, perlindungan data dan kecepatan data *recovery*.
2. Privasi dan keamanan, Oleh karena banyak orang atau perusahaan lain juga menggunakan hosting, maka ada kemungkinan data anda dapat dibaca oleh orang lain tanpa sepengetahuan dan izin anda. Untuk itu, para pengguna layanan *Cloud Computing* harus memahami *software* dan *hardware* keamanan sistem informasi seperti firewall, VPN (*Virtual Private Networks*), *encryption* untuk data penting dan komponen sejenis lainnya.

2.3 *Video*

Video adalah informasi yang berisi gabungan gambar yang ditampilkan secara berurutan dengan kecepatan tertentu sehingga membentuk gambar bergerak. *Video* biasanya memiliki komponen *audio* yang sesuai dengan gambar-gambar yang ditampilkan pada layar (businessdictionary.com, 2015).

2.3.1 *Video Streaming*

Video streaming adalah informasi yang berisi gabungan gambar yang ditampilkan secara berurutan dengan kecepatan tertentu yang kemudian dikompresi menjadi *file* yang lebih kecil kemudian dikirimkan melalui jaringan Internet dan ditampilkan oleh *media player*. *Media player* adalah aplikasi yang dapat melakukan proses *coding* dan *decoding video* agar *video* dapat ditampilkan ke layar monitor (Austerberry, 2005).

Ada dua jenis tipe *video streaming*, yaitu:

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

2.3.2 *Video on Demand*

Video on demand adalah suatu sistem yang mampu memberikan pengguna keleluasaan untuk memilih dan melihat *file* atau konten *video* sesuai dengan permintaan (Pathan et al, 2014).

Gambar 2.1 Proses *Streaming*
Sumber : (howstuffwork.com, 2007)

Pada gambar 2.1 ditunjukkan proses yang terjadi dalam pembuatan dan pendistribusian VoD. Pada bagian pertama terdapat proses *recording* yakni merekam *video* menggunakan *film* atau perekam digital. Kemudian dilakukan proses konversi menjadi *file digital* dengan

cara melakukan *importing file* ke dalam komputer dan dilakukan proses *encoding* menggunakan format yang diinginkan. Tahapan ketiga adalah melakukan *upload* ke *server* dimana *video* akan disimpan didalam tempat penyimpanan dan nantinya akan dilakukan penyiaran melalui jaringan Internet. Tahapan ke empat menggambarkan pengguna Internet melakukan *streaming video*. Agar *video* dapat dinikmati oleh pengguna (menampilkan ke layar *monitor*) maka diperlukan tahapan kelima yakni proses *decoding* menggunakan media player yang sesuai dengan codec *video*.

2.3.3 *Video Coding*

Data video memuat berbagai informasi yang merepresentasikan gambar pada periode waktu tertentu. Isi dalam data *video* tidak hanya mencakup obyek bergerak namun juga scenes yang memiliki tekstur yang kompleks dengan informasi warna tertentu. Tanpa adanya kompresi, besarnya data *video* akan menghabiskan seluruh kapasitas suatu *devices* dengan cepat. Sebagai contoh, *video* dengan format yang digunakan untuk High-Defintion Television (HDTV) memiliki resolusi 1920 *pixel* secara *horizontal* dan 1080 pixel secara *vertical*, *frame rate* yang digunakan sebesar 30fps, dengan kedalaman bit 24 bits (8 bit untuk setiap komponen warna RGB) dan *video* berdurasi 90 menit maka dibutuhkan kapasitas penyimpanan sebesar 1008 GB (Gao et al,2014).

Hampir semua *video* yang dimiliki oleh pengguna telah mengalami proses kompresi agar *video* dapat disimpan dalam komputer dengan kapasitas kecil. *Video* yang telah dikompres telah mengalami penurunan kualitas dibanding *file* aslinya (*raw file*).

Pada data *video* terdapat *container* yang terdiri dari *video codec* dan *audio codec* bahkan dapat berisi subtitle. *Container* digunakan untuk mengemas *video* dan komponennya (*audio/metadata*) dan teridentifikasi dengan ekstensi file, seperti .AVI, .MKV, .MP4 atau .MOV.

Tabel 2.1 Daftar *Container*

Jenis Container	Keterangan
AVI (*Audio Video Interleave*)	Standar *container* untuk *file multimedia* dengan sistem operasi Windows
MP4 (dikenal sebagai .mp4)	Bentuk *container* standar untuk MPEG-4
FLV (*Flash Video*)	Format yang digunakan untuk mengirimkan *video* MPEG melewati *Flash Player*
MOV	Format *container* untuk program Quicktime milik Apple
OGG, OGM dan OGV	*Container* dengan standar *opensource*
MKV (Mastroska)	*Container* lain dengan standar *opensource* yang biasa digunakan untuk anime
VOB (DVD *Video Object*)	*Container* standar untuk DVD
ASF	Format dari Microsoft untuk WMV dan WMA dengan akhiran .wmv atau .asf

Sumber: shallowsky.com/linux/*video*formats.html, 2014

Codec (*coder/decoder*) merupakan suatu *program* atau alat yang dapat melakukan proses *encoding* atau *decoding* atau kompresi dan dekompresi data *video* atau suara. *Codec* bertugas menerjemahkan *file video* dan menentukan bagaimana *video* tampil di layar monitor. Pada umumnya komputer telah dilengkapi dengan *codec* secara *pre-installed*. Namun pengguna komputer juga dapat melengkapi *codec* dengan cara mengunduh *codec-pack* seperti *program* VLC, K-lite Codec Pack, dsb.

Tabel 2.2 Daftar Jenis *Codec Video*

Format	ISO/IEC *number* *Issue date*	Target *bandwidth* bit/s	Resolusi	Aplikasi
H.261	1988-1990	384 k-2 M	176 x 144 or 352 x 288	*Video Conference, low delay*
H.263	1992	28,8 k-768 k	128 x 96 to 720 x 480	*Video Conference*
MPEG-1	11172 1993	400 k-1.5 M	352 x 288	CD-ROM
MPEG-2	13818 1994	1.5 M-15 M	720 x 480	Siaran Televisi, DVD
MPEG-4	14496 1998	28.8 k-500 k	176 x 144 or 352 x 288	Fixed dan *mobile web*
AVC, H.264	14496-10 2002			Keperluan umum

Sumber: Austerberry, 2005

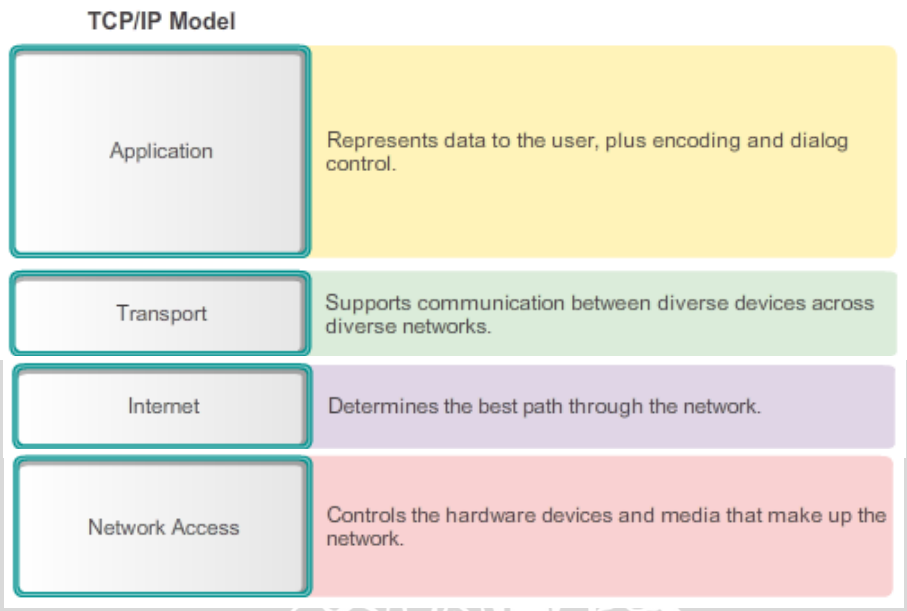
2.3.4 Parameter *Video*

1. *Frames per Second* (FPS) – standar untuk FPS adalah 29,97, meningkatkan nilai FPS menyebabkan semakin banyak gambar per detik sehingga menghasilkan gambar yang lebih halus. Menurunkan nilai FPS akan membuat *video* menjadi sedikit kaku dan kurang halus.
2. *Video Bitrate – Bitrate* adalah nilai pengukuran dari bit yang dikirimkan sejauh waktu tertentu. Nilai *bitrate* akhir/total adalah kombinasi dari *bitrate stream video, audio* stream, dan *metadata* pada *file* dengan nilai *bitrate stream video* yang mendominasi. Semakin besar nilai *bitrate* maka semakin baik kualitas *video* dan semakin besar *file* tersebut.
3. Resolusi – resolusi menunjukkan jumlah *pixel* yang ditampilkan pada gambar suatu *video*. Hal ini menentukan apakah *video* termasuk *standard definition* atau *high*

definition. Semakin tinggi resolusi semakin jernih gambar dan semakin besar file tersebut (rice.edu, 2013).

2.4 Protokol-Protokol pada Layanan *Multimedia*

Protokol merupakan suatu aturan yang dapat mengatur komunikasi data. TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di Internet. Karena menggunakan bahasa yang sama, yaitu protocol TCP/IP, perbedaan jenis komputer dan sistem operasi tidak menjadi masalah.



Gambar 2.2 TCP/IP Model
Sumber : (CCNA, 2014)

Seperti dapat dilihat pada gambar 2.2 diatas, TCP/IP terdiri dari empat layer model. Layer paling bawah atau *Network Access Layer* berfungsi mengubah IP *datagram* ke *frame* yang ditransmisikan oleh *network*, dan memetakan IP *address* ke *physical address* yang digunakan dalam jaringan. IP *address* ini harus diubah ke alamat apapun yang diperlukan untuk *physical layer* agar dapat mentransmisikan *datagram*. *Internet Layer* berfungsi menentukan jalur terbaik pada jaringan untuk pengiriman data. *Transport Layer* bertanggungjawab terjadinya koneksi antar perangkat yang berbeda dengan jaringan yang berbeda. Pada *Transport Layer* terdapat dua protokol yakni *Transfer Control Protocol* (TCP) dan *User Datagram Protocol* (UDP). Dan layer paling atas atau Application Layer berfungsi menyajikan data ke pengguna serta melakukan *encoding* dan *dialog control* (CCNA, 2014).

Layanan multimedia *streaming* melibatkan beberapa protokol salah satunya adalah protokol pada *transport layer*. Seperti yang telah dijelaskan diatas bahwa pada *transport layer* terdapat dua protokol yakni TCP dan UDP. Hampir seluruh aktivitas di Internet menggunakan protokol TCP. TCP didesain untuk menyediakan pengiriman data yang dapat diandalkan (*reliable*).

Namun proses *Streaming* tidak membutuhkan sebuah protokol transmisi yang melakukan retransmisi data. Karena jika terjadi error dan retransmisi data dilakukan maka dapat menyebabkan proses *streaming* memiliki *delay* yang relatif besar. Oleh karena itu pada *streaming* digunakan protokol UDP. Namun protokol yang digunakan pada *streaming* tidak hanya protokol UDP, tetapi membutuhkan protokol Real Time Trans*port* Protocol (RTP).

2.4.1 *Real Time Transport Protocol*

RTP adalah suatu protokol berbasis IP yang mampu melakukan pengiriman data secara *real-time* seperti *streaming audio* dan *video.* Layanan yang diberikan oleh RTP antara lain : *time reconstruction, loss detection, security* dan *content identification*. RTP pada awalnya didesain untuk layanan *multicast* dengan data real-time, namun dapat juga digunakan untuk layanan *unicast*. RTP didesain untuk bekerja berdampingan dengan protokol tambahan yakni protokol RTCP. Hal ini bertujuan agar didapatkan *feedback* mengenai kualitas transmisi data dan informasi pada sesi yang sedang berjalan di pihak pengguna.

Paket-paket yang dikirim di Internet memiliki *delay* dan *jitter* yang tidak dapat diprediksi. Namun aplikasi multimedia membutuhkan ketepatan waktu untuk transmisi data. RTP menyediakan *timestamping, sequence numbering* dan mekanisme lain untuk mengatasi masalah waktu. Melalui mekanisme ini, RTP memberikan layanan pengiriman *data end-to-end* secara *real-time.*

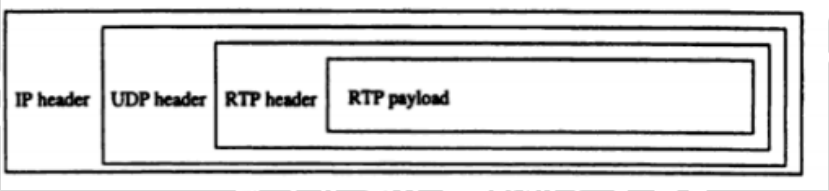
Timestamping merupakan informasi yang sangat penting bagi aplikasi *real-time*. Pengirim menetapkan *timestamp* berdasarkan oktet pertama pada paket yang disampling. *Timestamp* meningkat seiring dengan berjalannya waktu. Setelah penerima menerima paket data, penerima menggunakan *timestamp* untuk merekonstruksi paket sesuai dengan waktu pengirim agar data dapat dimainkan pada *rate* yang benar. *Timestamping* juga digunakan untuk melakukan sinkronisasi *different stream* menggunakan *timing properties.* seperti data

audio dan *video* dalam MPEG. Namun RTP sendiri tidak bertanggung jawab terhadap proses sinkronisasi karena proses ini dilakukan di level aplikasi.

UDP tidak mengirimkan paket secara urut, sehingga digunakanlah *sequence number* untuk menyusun paket data sesuai dengan urutan yang benar. *Sequence number* juga digunakan untuk mendeteksi terjadinya *packet loss*.

Payload type identifier mengidentifikasikan jenis payload sesuai dengan teknik kompresi yang digunakan. Melalui *payload type identifier* ini aplikasi penerima dapat menentukan cara menerjemahkan dan memproses *payload data*.

Fungsi penting lainnya adalah *source identification. Source identification* memberikan informasi darimana data berasal. Contoh : ketika pada *audio conference* pengguna dapat mengetahui siapa yang berbicara dari informasi *source identification*.



Gambar 2.3 RTP data pada paket UDP/IP
Sumber : (osso, 2000)

Seluruh menakisme diatas diimplementasikan melalui RTP header. Pada gambar 2.3 terlihat paket RTP dienkapsulasi pada paket UDP/IP (Osso, 2000).

2.4.1.1 *Real Time Control Protocol*

RTCP merupakan protokol pelengkap dari RTP yang memiliki fungsi utama untuk memberikan *feedback* kualitas pengiriman data dan informasi. Setiap sesi pada RTP diidentifikasi melalui *network address* dan sepasang *ports* dimana *port* satu digunakan untuk data RTP dan *port* lainnya digunakan untuk data RTCP. Contoh : jika data media dikirim lewat UDP *port* 5004 maka *control channel* akan dikirimkan ke *network address* yang sama namun pada *port* 5005 (perkins, 2003).

Seluruh partisipan pada sesi mengirimkan paket RTCP dan sebaliknya akan menerima paket RTCP yang dikirim oleh seluruh partisipan yang lain. Fungsi dari pengiriman paket ini adalah setiap partisipan pada sebuah sesi mengetahui mengenai kehadiran partisipan lain pada sesi dan mendapatkan informasi kualitas penerimaan *(packet loss* dan *jitter*).

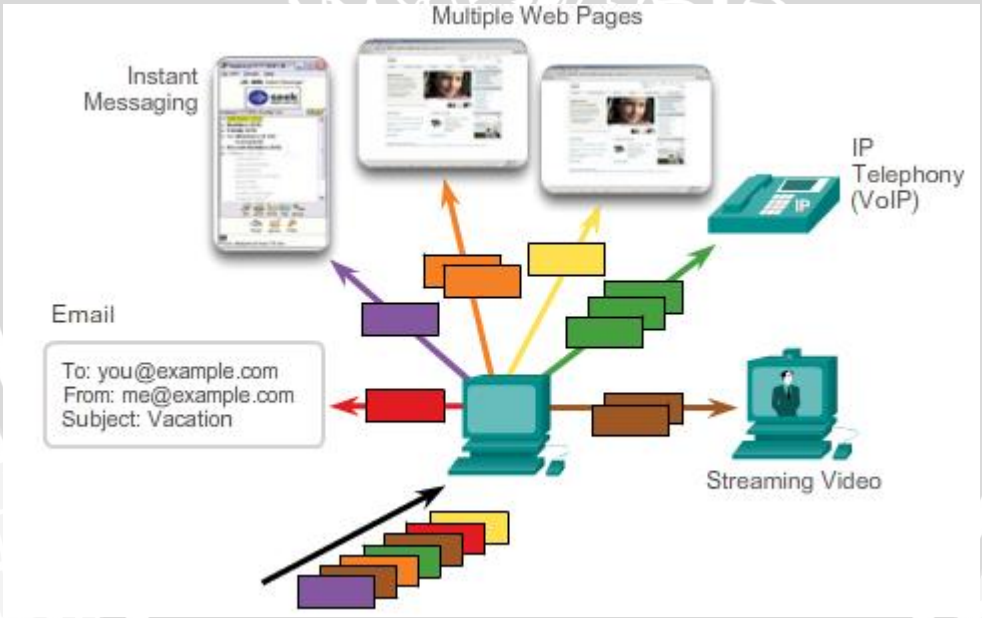
2.4.2 *Real Time Streaming Protocol*

RTSP merupakan sebuah protokol presentasi *client-server multimedia* yang dapat melakukan kontrol pengiriman *streaming data multimedia* melalui jaringan IP. RTSP menyediakan "VCR-style" fungsi *remote kontrol* untuk *streaming audio* dan *video* seperti fungsi *pause, fast forward, reverse*, dan *absolute positioning*.

RTSP adalah protokol pada *level application* yang didesain dapat bekerja dengan protokol yang berada dibawahnya seperti RTP untuk menyediakan layanan *streaming* secara utuh melalui jaringan Internet.

RTSP menyediakan mengontrol aliran *audio* dan *video* secara terus menerus antara *media server* dan klien. RTSP adalah sebuah *network remote control* antara *server* dengan klien. Protokol ini bertujuan memberikan pelayanan pada *audio* dan *video streaming* seperti HTTP memberikan layanan pada *text* dan gambar.

2.4.3 *Port Number*



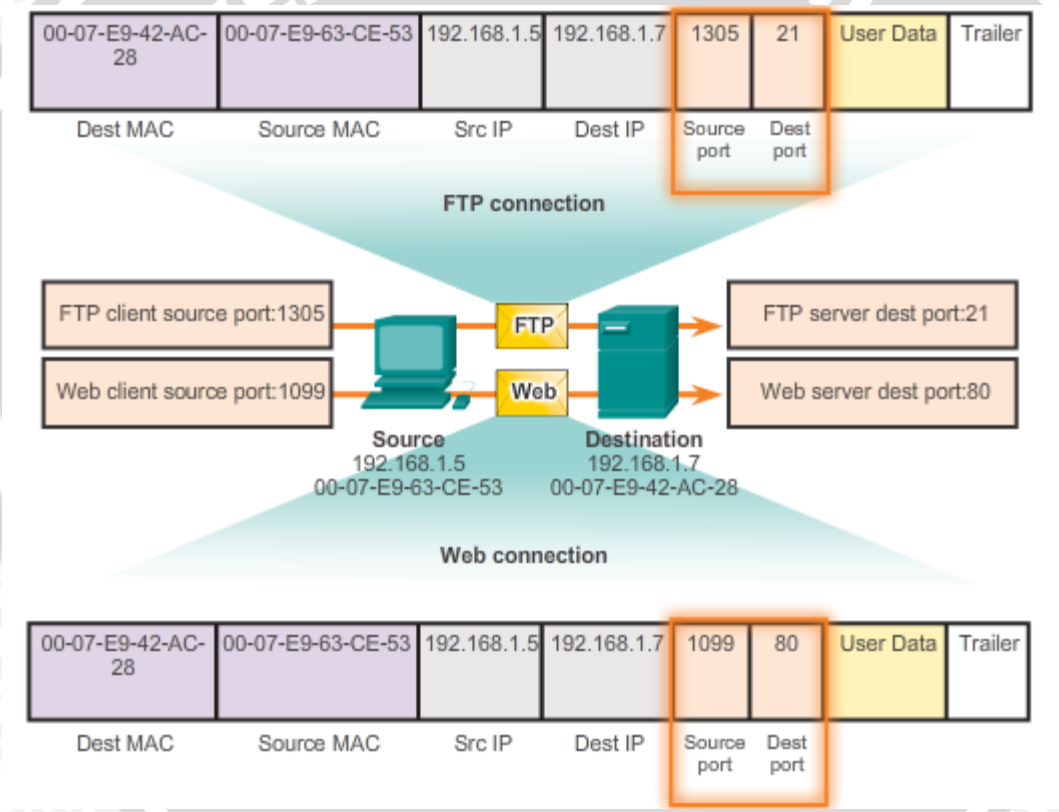
Gambar 2.4 Pengiriman *Data* ke *Server* Berdasarkan *Port Number*
Sumber : (CCNA, 2014)

Pada jaringan terdapat kemungkinan *server* menjalankan beberapa layanan/aplikasi. Agar dapat menyalurkan data ke aplikasi yang tepat, *Transport Layer* harus dapat mengidentifikasi aplikasi yang dituju seperti pada gambar 2.4. *Transport Layer* memberikan setiap aplikasi sebuah *identifier* yakni sebuah *port*. Sehingga dapat ditarik kesimpulan bahwa

Port number merupakan teknik untuk mengidentifikasi pesan yang diterima oleh *server* dari jaringan Internet atau jaringan lain agar dapat diproses pada jalur yang tepat.

Pada bagian *header* disetiap *datagram*, terdapat *port* sumber dan tujuan. *Source port number* merupakan nomor yang digunakan untuk berkomunikasi yang diasosiasikan dengan aplikasi asal dari klien. *Destination port number* merupakan nomor yang digunakan untuk berkomunikasi yang diasosiasikan dengan aplikasi tujuan di *server*.

Source port number dihasilkan secara acak oleh sistem klien. Hal ini agar klien dapat melakukan komunikasi secara bersamaan. Dengan kata lain, perangkat dapat mengirim beberapa *request* layanan HTTP menuju *web server* pada saat bersamaan.



Gambar 2.5 Penggunaan *Source* dan *Destination Port*
Sumber : (CCNA, 2014)

Sebagai contoh pada gambar 2.5, terdapat *request* pada klien menuju *server* melalui jaringan Internet dimana klien melakukan *request* pada sebuah *file* yang terdapat pada sisi *server File Transfer Protocol* (FTP) dan *request* pada layanan *Hypertext Transfer Protocol* (HTTP). Agar dapat mengirimkan *request* pada layanan FTP dan HTTP pada *remote server*,

layer transport protocol pada komputer mengidentifikasi adanya *port number* 21 dan 80 (dimana *port number* 21 ini diasosiasikan dengan request FTP sedangkan *port* number 80 diasosiakan dengan request HTTP) pada pesan yang dikirimkan. Pada sisi *server layer transport protocol* membaca adanya *port number* 21 dan 80 pada *request* dan *request* akan langsung diteruskan menuju program yang tepat pada *server*.

Internet Assigned Numbers Authority (IANA) membagi jenis nomor *port* menjadi tiga, yakni :

1. *Well-known Ports* (Nomor 0 sampai 1023) – Nomor-nomor ini telah dipesan untuk beberapa layanan dan aplikasi. Seperti HTTP (*web server*), *Internet Message Access Protocol* (IMAP)/*Simple Mail Transfer Protocol* (SMTP) (*email server*) dan Telnet. Dengan menetapkan *port well-known* ini, aplikasi milik klien dapat diprogram untuk melakukan *request* menggunakan *port* yang sesuai dengan layanan yang digunakan.
2. *Registered Ports* (Nomor 1024 sampai 49151) – Nomor *port* ini dapat digunakan oleh layanan dan aplikasi apa saja. *Developer* dapat mendaftarkan *port* melalui IANA. Setelah *port* didaftarkan, *developer* dapat menggunakan *port* ini pada program dan mengaksesnya via Internet. Namun hal ini tidak dapat mencegah orang lain untuk menggunakan *port* tersebut (Syngress, 2000). Ketika *registered port* tidak digunakan untuk layanan apapun, *port* ini dapat dipilih secara acak oleh sistem klien untuk menjadi *source port* (CCNA, 2014).
3. *Dynamic atau Private Ports* (Nomor 49152 sampai 65535) – Atau dikenal juga sebagai *ephemeral ports*, *port-port* ini digunakan secara acak oleh aplikasi klien ketika klien melakukan koneksi ke sebuah layanan

2.5 Parameter Kinerja Jaringan

Quality of Service (QoS) adalah parameter yang menyatakan kinerja suatu jaringan berdasarkan beberapa parameter. Sedangkan menurut ITU-T E.800, QoS adalah sekumpulan efek performansi yang menentukan derajat kepuasan pengguna terhadap *service* yang diberikan oleh jaringan. Parameter-parameter yang yang digunakan untuk mengukur kinerja QoS adalah probabilitas *packet loss*, *delay*, dan *throughput*.

2.5.1 Delay

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

Delay total pada sistem VoD dihitung dengan menggunakan persamaan:

$$t_{total} = t_{end\ to\ end} + t_{codec} \quad (2.1)$$

$$t_{end\ to\ end} = t_{proses} + t_t + t_p + t_w \quad (2.2)$$

Keterangan:

$t_{end\ to\ end}$ = *delay end-to-end* pada jaringan (s)

t_{proses} = *delay* proses, yakni *delay* enkapsulasi dan dekapsulasi (s)

t_t = *delay* transmisi (s)

t_p = *delay* propagasi (s)

t_w = *delay* antrian (s)

t_{codec} = *delay* codec (s)

t_{total} = *delay* total pada sistem VoD (s)

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

1. *Delay* Propagasi

Delay propagasi adalah waktu yang dibutuhkan oleh paket data untuk merambat dari *server* menuju *client* melalui media transmisi. Kecepatan propagasi berbagai jenis kabel (media) ditunjukkan pada Tabel 2.3. Persamaan yang digunakan untuk mencari nilai *delay* propagasi ditunjukkan pada persamaan 2.3 (Forouzan, 2007).

$$t_p = \frac{d_{max}}{v},\ c = 3 \times 10^8 \quad (2.3)$$

Keterangan :

t_p = *Delay* propagasi (s)

d_{max} = Panjang maksimum kabel (m)

v = Kecepatan Propagasi (m/s)

Tabel 2.3 Kecepatan Propagasi

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

Sumber : Messer, 2014

2. *Delay* Enkapsulasi

Delay enkapsulasi adalah waktu yang dibutuhkan untuk proses pemaketan data dengan menambahkan *header* pada paket data, sehingga paket data tersebut dapat dikirimkan ke tujuan. Persamaan yang digunakan untuk mencari nilai *delay* enkapsulasi ditunjukkan pada persamaan 2.4 (Purbo *et al*. 2001).

$$t_{enc} = \frac{L_{Header\ Ipv4} + L_{Header\ NALU} + L_{Header\ UDP} + L_{Header\ RTP} + L_{Header\ Ethernet} + L_{Header\ FCS}}{C_{proses}} \quad (2.4)$$

Dengan,

$$C_{proses} = \frac{N_k}{T_k} \quad (2.5)$$

Keterangan:

t_{enc} = *Delay* enkapsulasi (s)

L = Panjang paket data (*byte*)

C_{proses} = Kecepatan pemrosesan data pada *node* (bps)

N_k = Jumlah total data (bit)

T_k = Waktu pengiriman (s)

3. *Delay* Dekapsulasi

Delay Dekapsulasi adalah waktu yang dibutuhkan untuk proses pembacaan keseluruhan *header* dari sebuah paket yang diterima suatu *node*. Persamaan yang digunakan untuk mencari nilai *delay* dekapsulasi ditunjukkan pada persamaan 2.6 dan 2.7 (Purbo *et al*. 2001).

$$t_{dec} = \frac{L_{Header\ Ipv4} + L_{Header\ NALU} + L_{Header\ UDP} + L_{Header\ RTP} + L_{Header\ Ethernet} + L_{Header\ FCS}}{C_{proses}} \quad (2.6)$$

Dengan,

$$C_{proses} = \frac{N_k}{T_k} \quad (2.7)$$

Keterangan:

t_{dec} = *Delay* dekapsulasi (s)

L = Panjang paket data (*byte*)

C_{proses} = Kecepatan pemrosesan data pada *node* (bps)

N_k = Jumlah total data (bit)

T_k = Waktu pengiriman (s)

4. *Delay* antrian

Delay antrian adalah lamanya waktu yang diperlukan maupun keterlambatan yang dialami oleh paket data dalam kurun waktu tertentu sebagai akibat terjadinya pemberlakuan antrian untuk pemrosesan di dalam jaringan komputer. Persamaan yang digunakan untuk mencari nilai *delay* dekapsulasi ditunjukkan pada persamaan 2.8, persamaan 2.9 dan persamaan 2.10 (Schwartz, 1987).

$$E(T) = \frac{1}{\mu - \lambda_p} \quad (2.8)$$

Dengan,

$$\lambda_p = \frac{N}{T} \quad (2.9)$$

Dan,

$$\mu = \frac{C}{L_t} \quad (2.10)$$

Keterangan :

λp = kecepatan kedatangan paket pada *server* (paket/s)

N = Total paket yang dikirim (paket)

T = Waktu pengiriman paket total (s)

μ = Kecepatan pelayanan *server* (paket/s)

C = Kapasitas kanal (bps)

Lt = Panjang paket data (bit/paket)

5. *Delay* codec

Video yang digunakan pada penelitian ini menggunakan *video* dengan codec H.264 dan *audio* dengan codec AAC. Spesifikasi codec *audio* dan *video* yang digunakan pada penelitian dapat dilihat pada Tabel 2.4.

Tabel 2.4 Spesifikasi *Audio* dan *Video* Codec

CODEC	***Delay* CODEC (ms)**
H.264	10 - 50
AAC	20 - 31

Sumber: Texas Instrument, 2007

Besarnya nilai *delay* codec pada *video* yang digunakan pada penelitian ditunjukkan dengan persamaan 2.11.

$$t_{codec} = t_{video} + t_{audio} \quad (2.11)$$

Keterangan:

t_{codec} = *Delay codec* keseluruhan (s)

t_{video} = *Delay codec video* (s)

t_{audio} = *Delay codec audio* (s)

6. *Delay* Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket multimedia ke media transmisi. Persamaan yang digunakan untuk mencari nilai *delay* dekapsulasi ditunjukkan pada persamaan 2.12 (Forouzan, 2007).

$$t_t = \frac{(L+L')}{C} \quad (2.12)$$

Keterangan:

L = panjang paket data (*bit*)

L' = panjang *header* paket data (*bit*)

C = Kapasitas kanal (bps)

2.5.2 Throughput

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

Berikut adalah persamaan untuk menghitung *bitrate* atau *throughput* tanpa *losses* secara matematis (Amerasinghe, 2009) :

$$\lambda_{FL} = \lambda_{FLVideo} + \lambda_{FLAudio} \quad (2.13)$$

$$\lambda_{FLVideo} = Width * Height * Frame\ Rate * 0{,}07 * Motion\ Rank \quad (2.14)$$

dengan:

λ_{FL} = *Throughput* tanpa *losses* (bps)

$\lambda_{FLVideo}$ = *Throughput video* (bps)

$\lambda_{FLAudio}$ = *Throughput audio* (bps)

Width = Lebar *video*

Height = Tinggi *video*

Frame Rate = Jumlah *frame* dalam *video* setiap detiknya

Motion Rank = Bernilai 1 hingga 4 tergantung gerakan yang ada pada *video*

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

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

- *High motion* adalah *video* yang memiliki gerakan yang sangat dinamis seperti *action movies* dengan pergerakan dan perubahan *scene* yang sangat cepat. Nilai dari *high motion* = 4.

2.5.3 Probabilitas *Packet Loss*

Packet loss merupakan parameter yang menunjukkan banyaknya jumlah paket yang hilang atau tidak sampai ke tujuan ketika melakukan pengiriman data dari sumber ke tujuan. Pada beberapa situasi jika *packet loss* terjadi terdapat penurunan performansi secara signifikan sehingga dibutuhkan upaya untuk meminimalkan terjadinya *packet loss* ini.

Pada protokol TCP paket data yang hilang akan terdeteksi dan akan segera ditransmisikan ulang. Sedangkan pada protokol UDP paket data yang hilang tidak akan ditransmisikan ulang. Hampir semua jaringan mengalami terjadinya kehilangan paket data. Salah satu penyebab terjadinya hilangnya paket data adalah *router* menerima data yang lebih banyak dibandingkan kemampuannya untuk mentransmisikan data.

Untuk menghitung probabilitas *packet loss* digunakan persamaan 2.15 (Tibor *et al*. 2012).

$$P_{Loss}(N,A) = \frac{\frac{A^N}{N!}}{\sum_{i=0}^{N}\frac{A^i}{i!}} \times 100\% \quad (2.15)$$

Dengan,

$$A = \frac{\lambda}{\mu} \quad (2.16)$$

Keterangan :

N = Jumlah *server*

A = Intensitas traffic

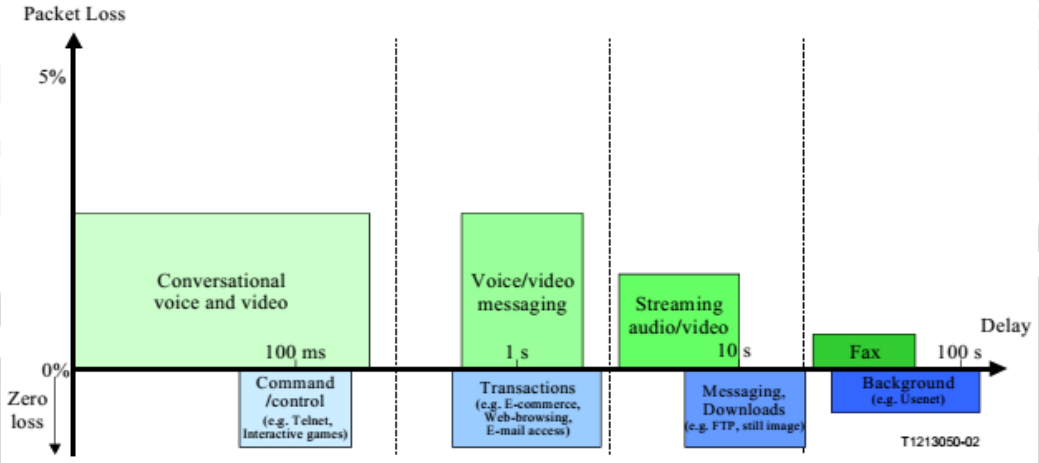
λ = Kedatangan paket (pps)

μ = Rata-rata kedatangan paket yang dapat dilayani (pps)

2.6 Rekomendasi dan Standar Perangkat

2.6.1 Rekomendasi ITU-T G.1010

Berdasarkan ITU-T G.1010, Aplikasi *streaming* memiliki persyaratan *packet loss* dan *delay* yang berbeda-beda berdasarkan media yang dikirimkan. Persyaratan untuk QoS parameter *delay* dan *packet loss* juga ditunjukkan pada gambar 2.6.



Gambar 2.6 *Mapping of User-Centric QoS Requirements*
Sumber: ITU-T G.1010, 2001

Dan berikut adalah tabel yang menunjukkan syarat *packet loss* dan *delay* berdasarkan ITU-T G.1010.

Tabel 2.5 QoS ITU-T G.1010

Medium	**Application**	**Degree of Symmetry**	**One Way *Delay***	**Information *Loss***
Audio	*Conversational Voice*	*Two-way*	<150 ms	*< 3% Packet Loss Ratio (PLR)*
Audio	*Voice Messaging*	*One-way*	*<1 s*	*< 3% PLR*
Audio	*High Quality Audio Streaming*	*One-way*	*<1 s*	*< 1% PLR*
Video	*Videophone*	*Two-way*	*<150 ms*	*< 1% PLR*
Video	*Streaming*	*One-way*	*<10 s*	*< 1% PLR*

2.6.2 Perangkat Keras

Perangkat keras yang dibahas dalam sub bab ini merupakan perangkat keras yang berada pada sisi penyedia layanan *video on demand*, dan pada sisi klien/pengguna. Komponen perangkat keras yang digunakan pada penelitian ini antara lain : *server*, laptop, switch, kabel *Unshielded Twisted Pair* (UTP), kabel *single mode fiber* dan *media converter*.

2.6.2.1 *Server*

Server adalah sebuah komputer yang dapat memberikan layanan kepada komputer lain. *Server* memberikan fasilitas kepada klien untuk berbagi data, informasi, maupun sumber daya komputasi. Secara umum klien terhubung oleh *server* melalui jaringan.

Gambar 2.7 *Data center* milik *Google*
Sumber: Google, 2014

2.6.2.2 Laptop

Laptop adalah sebuah komputer *port*abel yang biasanya memiliki berat dua hingga empat kilogram, tergantung dari ukuran layar, *hardware* lainnya. Laptop dipasarkan sebagai komputer untuk bisnis yang dapat digunakan untuk *coding, spreadsheet, presentation*, dan *financial* karena laptop memiliki kemampuan komputasi seperti PC desktop pada umumnya tergantung spesifikasi yang digunakan.

Gambar 2.8 Laptop
Sumber: lenovo.com

2.6.2.3 *Switch*

Switch adalah perangkat jaringan komputer yang dapat menghubungkan antar perangkat pada jaringan komputer. Dengan *switch* antar komputer dapat melakukan pertukaran data dengan metode *packet switching* yang dimiliki oleh *switch. Switch* hanya meneruskan data hanya ke satu atau beberapa perangkat yang membutuhkan. Berbeda dengan hub yang hanya memiliki kecepatan transfer data hingga 100mbps, *switch* mampu melakukan transfer data dengan kecepatan hingga 10 Gbps.

Gambar 2.9 Switch
Sumber: tp-link.com

Switch yang digunakan pada perancangan ini merupakan *Switch Manageable*. *Switch Manageable* ini memiliki beberapa fungsi *router* yang dapat dikonfigurasikan, seperti *port* VPN, *bandwitdh control*, *port trunking*, *port-based* VLAN, dan lain-lain.

2.6.2.4 Kabel *Unshielded Twisted Pair*

Kabel *Unshielded Twisted Pair* (UTP) dan *Shielded Twisted Pair* (STP) adalah kabel jaringan yang umum dan digunakan di dalam jaringan komputer terutama kabel UTP. Berbeda dengan STP, kabel UTP tidak terdapat pelindung pada bagian inti konduktornya. Jenis kabel ini berbahan tembaga dengan harga yang jauh lebih rendah dibandingkan dengan kabel *Coaxial*. Kabel jaringan UTP (*Unshielded Twisted Pair*) ini memiliki setidaknya delapan buah kategori, namun yang umum digunakan di dalam kehidupan sehari-hari adalah Cat5e (*Enhanced Category* 5) dan Cat5 (*Category* 5).

Gambar 2.10 Perbandingan kabel UTP dan STP
Sumber: lanshack.com

2.6.2.5 *Kabel Single Mode Fiber*

Serat optik atau *Fiber Optic* (*optical fiber*) merupakan media transmisi pada jaringan komputer yang menggunakan teknologi serat optik (serat gelas murni) berbentuk kabel, dengan kehalusan melebihi rambut manusia, namun fiber optik memiliki kemampuan pengiriman data yang sangat cepat yakni hingga tera bit per detik (ntt.co.jp, 2010). Dengan kemampuannya ini, Serat Optik banyak digunakan sebagai media transmisi untuk *backbone* saluran telekomunikasi.

Gambar 2.11 Patchcord Fiber
Sumber: abblg.com, 2014

2.6.2.6 *Media Converter*

Media converter adalah sebuah perangkat yang umumnya berfungsi mengubah sinyal listrik yang digunakan oleh tembaga pada kabel *Unshielded Twisted Pair* (UTP) ke dalam gelombang cahaya yang digunakan dalam kabel serat optik.

Gambar 2.12 Media Converter
Sumber: tp-link.com

2.6.3 Perangkat Lunak

Perangkat lunak yang dibahas dalam sub bab ini merupakan perangkat lunak digunakan pada sisi penyedia layanan *video on demand*, dan pada sisi klien/pengguna. Komponen

perangkat keras yang digunakan pada penelitian ini antara lain : Operating System Windows *Server* 2008, Firewall, Wowza *Streaming* Engine, Wireshark, WinPcap dan VLC Media Player.

2.6.3.1 Windows *Server* 2008

Windows *Server* adalah sistem operasi keluaran Microsoft yang digunakan untuk *server*. Windows *Server* 2008 merupakan project Microsoft yang bernama Longhorn yang memiliki fitur seperti Windows Vista. Baik Windows *Server* 2008 dan Windows Vista keduanya memiliki beberapa kesamaan seperti sistem keamanan, manajemen dan fitur administratif. Namun Windows *Server* 2008 menawarkan fungsi yang sangat lebih baik, fitur-fitur yang dirancang khusus untuk *server* dan diatas segalanya dibanding apa yang ditawarkan oleh Windows Vista.

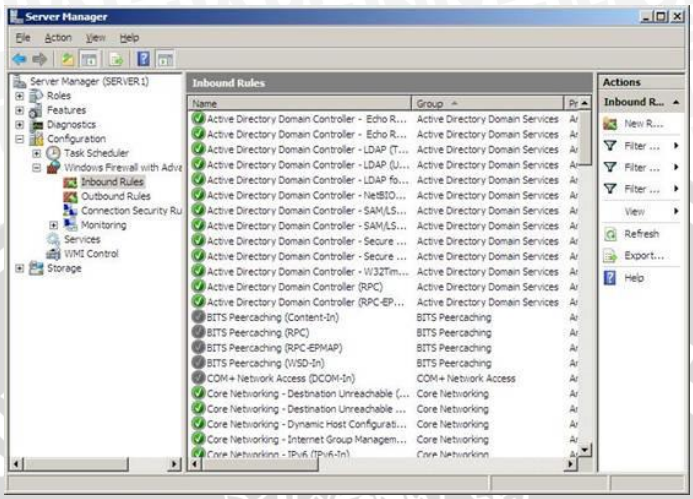
Gambar 2.13 Logo Windows *Server* 2008 R2
Sumber: microsoft.com

Large-scale deployment options, *self-diagnostic tools* yang lebih baik, *advanced reliability* dan *performance monitoring*, dan *enhanced security* adalah beberapa fitur yang ada pada Windows *Server* 2008 (tittel, 2008).

2.6.3.2 Firewall

Firewall adalah suatu piranti atau *software* yang dirancang spesifik untuk melindungi komputer dan jaringan dari berbagai aktivitas jaringan yang berbahaya. Firewall dapat membatasi jenis trafik tertentu yang masuk dan keluar pada jaringan dengan cara melakukan blok pada *port* tertentu atau protokol tertentu. Membantu mencegah hacker atau *software* perusak seperti worm untuk memperoleh akses komputer melalui jaringan. Dan firewall juga secara aktif mencatat aktivitas jaringan yang berguna untuk penelusuran, misalnya saat terjadi usaha penyusupan atau penyerangan terhadap komputer (rafiudin, 2008).

Dalam sistem operasi Windows sudah terdapat *software* firewall yang memberikan proteksi *packet-filtering*. Gambar 2.14 merupakan *screenshot* dari Windows Firewall with Advanced Security yang ada pada Windows *Server*.



Gambar 2.14 Tampilan Firewall pada Windows *Server*
Sumber: windowssecurity.com, 2009

Windows Firewall with Advanced Security menggabungkan host firewall dan Internet Protocol security (IPsec). Program ini didesain untuk digunakan oleh IT administrator agar dapat mengatur keamanan jaringan pada lingkungan *enterprise*. *Software* ini tidak ditujukan untuk penggunaan pada jaringan rumah. Namun pengguna biasa (Home *user*s) dapat menggunakan program ini melalui Control Panel (microsoft.com).

2.6.3.3 Wowza *Streaming* Engine

Wowza *Streaming* Engine atau yang lebih dikenal Wowza Media *Server* adalah *software streaming* media *server* yang dikembangkan oleh Wowza Media Systems. Wowza *Streaming* Engine dapat menggunakan berbagai format *video* dan mampu mentransmisikannya menggunakan kualitas tinggi untuk perangkat apa saja.

Software ini merupakan *software multi-threaded* yang dapat digunakan oleh penyedia jasa *video streaming* baik *live* maupun *on-demand* dengan menggunakan *server* sendiri atau melalui *cloud*.

Gambar 2.15 Logo Wowza
Sumber: wowza.com

Wowza *Streaming* Engine dapat menggunakan berbagai format VOD seperti Flash *Video* (.flv), MP4 (.mp4, .f4v, .mov, .m4a, .mp4a, .3gp, .3g2), PIFF(.isma, .ismv), konten MP3 (.mp3) dan konten H.264/AAC.

2.6.3.4 Wireshark

Wireshark adalah *software open source* dan bersifat gratis yang paling banyak digunakan orang-orang di dunia untuk menghitung, menganalisis paket data yang melewati suatu perangkat jaringan seperti ethernet, dan WiFi pada komputer atau laptop.

Gambar 2.16 Logo Wireshark
Sumber: wireshark.com

Kemampuan analisis pada *software* ini dikarenakan wireshark mampu melakukan capture seluruh paket data yang ditransmisikan melalui jaringan dan kemudian dilakukan decode untuk menganalisis paket data.

2.6.3.5 WinPcap

WinPcap adalah perangkat tambahan bagi perangkat *packet sniffer* seperti Wireshark. Adapun fungsi dari perangkat ini, antara lain:

- WinPcap mengijinkan aplikasi seperti Wireshark untuk menangkap dan mentransmisikan paket jaringan melewati protokol-protokol.
- *kernel packet filtering*,
- mesin statistik jaringan,
- dan mendukung *remote packet capture*.

2.6.3.6 VLC Media Player

VLC *Media Player* adalah *software* yang digunakan untuk memutar *audio* dan *video*. VLC dapat menampilkan berbagai jenis codec *video* yang ada seperti, .MPEG, .MP4, .MKV dan codec lainnya. VLC juga dapat digunakan untuk memutar *video streaming* yang ada pada *server* dengan memasukkan alamat url pada menu pilhan url jaringan.