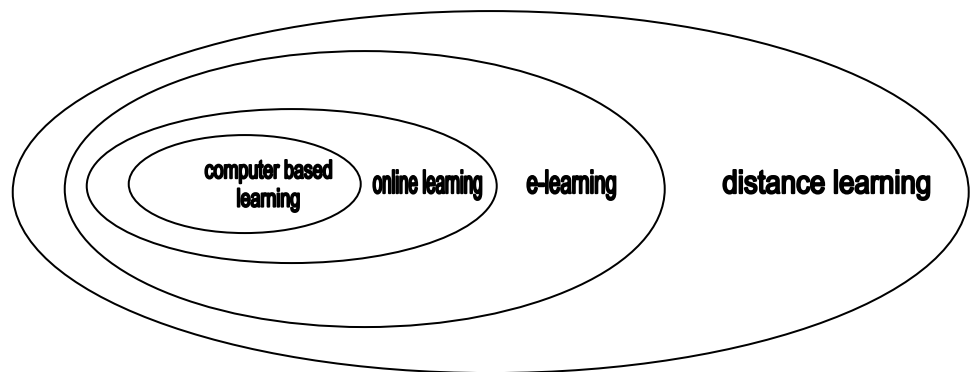


BAB II

TINJAUAN PUSTAKA

2.1 Terminologi *Distance Learning*

Saat ini banyak sekali terminologi yang muncul sehubungan dengan *distance learning*. Namun beberapa terminologi tersebut sebenarnya bermuara pada definisi yang sama (Eileen, 2001). Adapun terminologi yang bisa mewakili definisi *distance learning* dapat dilihat pada gambar 2.1



Gambar 2.1 Terminologi *Distance Learning*

- *Distance Learning*; merupakan seluruh bentuk pembelajaran (pendidikan dan pelatihan) jarak jauh, baik yang berbasis

korespondensi (modul tercetak) maupun yang berbasis teknologi.

- *E-learning*; merupakan bentuk pembelajaran jarak jauh yang menggunakan teknologi (*synchronous* dan *asynchronous*).
- *Online Learning*; memanfaatkan teknologi internet/intranet yang dikenal dengan *world wide web (web based learning)*.
- *Computer Based Learning*; memanfaatkan komputer sebagai terminal akses ke proses belajar (*CBT-Computer Based Learning, CD-ROM learning*).

Pendidikan jarak jauh adalah sekumpulan metode pengajaran dimana aktivitas pengajaran dilaksanakan secara terpisah dari aktivitas belajar. Pemisah kedua kegiatan tersebut dapat berupa jarak fisik, misalnya karena peserta ajar bertempat tinggal jauh dari lokasi institusi atau pelatihan. Pemisah dapat pula jarak non-fisik yaitu berupa keadaan yang memaksa seseorang yang tempat tinggalnya dekat dari lokasi institusi pendidikan atau pelatihan namun tidak dapat mengikuti kegiatan pembelajaran di institusi tersebut. Keterpisahan kegiatan pengajaran dari kegiatan belajar adalah ciri yang khas dari pembelajaran jarak jauh. Pada sistem pendidikan pelatihan ini tenaga pengajar dan peserta didik tidak harus berada dalam lingkungan geografi yang sama. Sarana penunjang dari pendidikan jarak jauh ini adalah teknologi informasi.

Kemunculan teknologi informasi dan komunikasi pada pendidikan jarak jauh ini sangat membantu sekali. Seperti dapat dilihat, dengan munculnya berbagai pendidikan secara *online*, baik pendidikan *formal* atau *non-formal*, dengan menggunakan fasilitas internet. Pendekatan sistem pengajaran yang dapat dilakukan adalah dengan melakukan pengajaran secara langsung (*real time*) ataupun dengan cara menggunakan sistem sebagai tempat pemusatan pengetahuan (*knowledge*) (Rebecca Green, 2001).

2.2 Perkembangan *Distance Learning*

Sejak dahulu *distance learning* atau belajar jarak jauh sudah ada, menggunakan media yang berubah-ubah dari surat menyurat dengan pensil dan kertas hingga yang sekarang dengan memanfaatkan perkembangan teknologi telekomunikasi dan informasi secara *real time*. tahun 1973 Moore memperkenalkan suatu teori mengenai cara belajar mandiri yang merupakan dasar penting dalam distance education, disebutkan bahwa pengajaran yang sukses dapat terlaksana meskipun pengajar dan siswa terpisah secara fisik selama proses belajar (Eileen, 2001).

2.3 IP Triple Play

IP *Triple Play* merupakan suatu proses menggabungkan komunikasi suara dan *video* pada jaringan data. Jaringan seperti ini dapat juga disebut sebagai jaringan konvergen. Jaringan konvergen telah banyak digunakan oleh organisasi perusahaan besar karena persyaratan jaringan infrastruktur dan manajemen kompleks yang terlibat untuk membuat para karyawan dapat bekerja dengan cepat dan lancar. Diperlukan biaya operasional jaringan yang tinggi terkait dengan konvergensi karena perangkat keras yang digunakan lebih mahal untuk mendukung kebutuhan *bandwidth* yang besar. Jaringan ini juga dibutuhkan manajemen bandwidth yang efisien dalam kaitannya dengan *Quality of Service* (Qos) karena lalu lintas suara dan *video* harus diklasifikasikan dan diprioritaskan pada jaringan (Vanessa Alvarez, 2006).

Konvergensi suara, *video* dan jaringan data telah menjadi populer dikalangan perusahaan kecil dan menengah karena kemajuan teknologi. Salah satu keuntungan dari jaringan yang konvergensi adalah hanya ada satu jaringan yang mengelola suara, *video* dan data. Manfaat lain dalam penggunaan jaringan yang konvergen adalah biaya pelaksanaan dan manajemen yang relatif lebih murah. Jika pada jaringan tradisional, paket suara, *video* dan data terpisah dalam pengelolaannya. Dengan jaringan konvergen

paket suara, *video* dan data dikelola dalam 1 kelompok pengelolaan.

2.3.1 Video conference

Video conference yang juga dikenal dengan *video teleconference* adalah suatu teknologi interaktif yang memungkinkan dua lokasi atau lebih untuk berinteraksi lewat *video* dan *audio* secara simultan. Teknologi utama yang digunakan dalam sistem *video conference* adalah kompresi *digital* suara dan *video stream* yang *real time*.

Teknologi *video conference* tidak lepas dari kemajuan teknologi kompresi *audio* dan *video*. Dengan banyaknya teknik kompresi yang ada saat ini memungkinkan *audio* dan *video* dapat dikirimkan secara bersamaan dalam jaringan dengan *bandwidth* yang seefisien mungkin dan dengan kualitas yang dapat diterima. *Hardware* atau *software* yang melakukan kompresi yang disebut dengan codec (*coder* atau *decoder*). *Codec* merupakan singkatan dari *compresi-decompresi* yang merupakan proses pembungkusan suara ataupun *video* analog menjadi data *digital* dengan metoda tertentu sehingga pengiriman suara atau *video* dapat dilakukan dalam bentuk paket-paket data. *Codec* dapat

melewatkan suara atau *video* dalam jaringan IP dengan bandwidth yang kecil dan kualitas yang masih dapat diterima. Komponen-komponen yang dibutuhkan untuk sebuah sistem *video conference* diantaranya:

- a. *Video input* : kamera video atau *webcam*
- b. *Video output* : monitor komputer atau proyektor
- c. *Audio input* : *microphones*
- d. *Audio output* : *speaker* atau *headphones*
- e. *Media transfer data* : LAN atau internet

Untuk *video conference*, digunakan *webcam* sebagai data sumber yang akan dikirimkan. Webcam memiliki resolusi pengambilan gambar dan resolusi antar satu webcam dengan webcam yang lain dapat bervariasi. Semakin besar ukuran resolusi , semakin besar ukuran resolusi semakin besar pula ukuran data yang dikirimkan, sehingga bandwidth yang diperlukan juga semakin besar . umumnya ukuran resolusi yang digunakan untuk *video conference* adalah 320x240. Selain itu, hal yang berpengaruh pada ukuran data adalah *frame rate*. *Frame rate* adalah jumlah gambar yang dikirimkan tiap detik (Michael Gough, 2006).

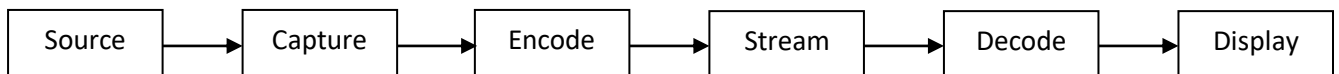
2.3.2 Audio

Video conference, digunakan sebuah *microphone* untuk *input audio*. Sama halnya dengan data video, terdapat faktor yang dapat mempengaruhi ukuran data yang dikirimkan, misalnya *sampling rate* (dalam satuan kHz) dan jumlah channel. Pada umumnya, ukuran data audio yang dikirimkan melalui streaming ini lebih kecil dibandingkan dengan data video. Sebuah data audio yang tidak dikompres menghasilkan data sebesar 5 megabyte per channel per menit. Tetapi, masih dimungkinkan jika *input* dari device ingin dikompres sehingga menghemat bandwidth yang ada (Hardi W.C, 2003).

2.4 Teknologi Streaming

Streaming adalah sebuah teknik yang digunakan untuk melakukan transfer data sehingga dapat diproses secara tetap dan kontinyu (Austerberry, 2005). Teknologi *streaming* berkembang sesuai dengan perkembangan internet, di mana kebanyakan pengguna internet masih belum memiliki koneksi *broadband* untuk mengunduh file multimedia yang berukuran besar dengan cepat. Namun, sumber yang digunakan untuk streaming tidak selalu dalam file multimedia seperti *webcam*, kamera, televisi dan lain

sebagainya. *Streaming* biasanya diidentikkan dengan realtime. Namun tidak dapat dipungkiri bahwa setiap media yang digunakan memiliki *latency*. Televisi memiliki latency dalam hitungan milidetik. Dengan menggunakan *high compression codec*, *latency* yang ditimbulkan bisa dalam hitungan detik. Faktor utama *streaming* yang menyebabkan *real time* adalah tidak adanya media penyimpanan yang digunakan untuk menyimpan paket data. Paket data akan disimpan pada sebuah *buffer* dan kemudian ditampilkan ke layar. Setelah selesai, data pada *buffer* akan dibuang dan *buffer* digunakan untuk menyimpan data yang baru. Walaupun demikian, masih tetap dimungkinkan jika data *streaming* ingin disimpan pada media penyimpanan seperti *hardisk*. Selain itu, data *streaming* juga dapat dikirim melalui jaringan, seperti pada aplikasi VoIP dan *video conference*. Proses *streaming* melalui jaringan dapat digambarkan seperti terlihat pada gambar 2.2:



Gambar 2.2 Proses *Streaming* melalui jaringan

Pada awalnya, data dari *source* (bisa berupa *audio* maupun *video*) akan di-*capture* dan disimpan pada sebuah *buffer* yang berada pada memori komputer (bukan media penyimpanan seperti *harddisk*) dan kemudian di-*encode* sesuai dengan format yang

diinginkan. Dalam proses encode ini, pengguna dapat mengompresi data sehingga ukurannya tidak terlalu besar (bersifat optional). Namun pada aplikasi *streaming* menggunakan jaringan, biasanya data akan dikompresi terlebih dahulu sebelum dilakukan *streaming*, karena keterbatasan *bandwidth* jaringan. Setelah di-*encode*, data akan di-*stream* ke pengguna yang lain. Pengguna akan melakukan *decode* data dan menampilkan hasilnya ke layar pengguna. Waktu yang dibutuhkan agar sebuah data sampai mulai dari pemancar sampai penerima disebut dengan *latency* (Michael Gough, 2006).

2.5 Quality of Service

Quality of Service (QoS) merupakan mekanisme jaringan yang memungkinkan aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan yang diharapkan. Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *bandwidth*, *latency* dan *jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Sebagai contoh, komunikasi suara (seperti VoIP atau IP Telephony) serta *video streaming* dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *bandwidth* yang tidak cukup, dengan *latency* yang tidak dapat diprediksi, atau *jitter* yang berlebih.

Fitur *Quality of Service* (QoS) ini dapat menjadikan *bandwidth*, *latency*, dan *jitter* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan di dalam jaringan tersebut yang ada (Giordano S, 2003).

Menurut Cisco ada 2 faktor yang memengaruhi pengiriman paket, yaitu dapat berupa internal atau eksternal.

- Faktor Eksternal

Faktor eksternal yang dapat memengaruhi komunikasi jaringan yang kompleks adalah jumlah perangkat yang harus dilewati untuk mencapai tujuan akhir.

Faktor eksternal yang memengaruhi keberhasilan komunikasi meliputi:

- Kualitas jalur antara pengirim dan penerima
- Banyaknya proses encapsulasi
- Banyaknya proses *redirect* dan *readdress* dari paket tersebut
- Jumlah paket lainnya yang ditransmisikan secara simultan pada jaringan komunikasi
- Jumlah waktu yang dialokasikan sehingga komunikasi berhasil.

- Faktor Internal

Berbagai jenis paket sangat bervariasi dalam kompleksitas dan tingkat kepentingannya. Paket yang singkat biasanya lebih mudah dipahami daripada paket yang kompleks.

Faktor internal yang memengaruhi keberhasilan dari komunikasi jaringan meliputi:

- Ukuran paket
- Kompleksitas dari paket
- Tingkat kepentingan dari paket tersebut

Sebuah paket dengan tingkat kepentingan yang rendah bisa hilang atau dapat di *drop* jika jaringan yang dilewatinya padat.

Kedua faktor internal dan eksternal yang memengaruhi penerimaan paket harus diantisipasi dan dikendalikan agar dapat terjadi komunikasi jaringan yang sukses dan handal. Inovasi baru dalam perangkat keras jaringan dan perangkat lunak dapat digunakan untuk menjamin kualitas dan kehandalan jaringan dalam berkomunikasi.

Tabel 2.1 QoS untuk kebutuhan bandwidth

Traffic type	Bandwidth	Packet loss (max)	Delay (max)	Jitter (max)
Interaktif voice (G.711)	12-106 kbit/s	1%	150ms	30ms
Streamed video (MPEG-4)	0.005-10Mbit/s	2%	5000ms	Insensitive
Streamed Audio (MP3)	32-320kbits/s	2%	5000ms	Insensitive
Data	Variable	Sensitive	Insensitive	Insensitive

2.6 *Bandwidth*

Seperti telah kita ketahui, *bandwidth* paling banyak digunakan sebagai ukuran kecepatan aliran data. *Bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. *Bandwidth* dapat dipakai untuk mengukur baik aliran data analog maupun aliran data digital. Sekarang telah menjadi umum jika kata *bandwidth* lebih banyak dipakaikan untuk mengukur aliran data *digital*. Satuan yang dipakai untuk *bandwidth* adalah *bits per second* atau sering disingkat sebagai bps. Seperti diketahui bahwa atau *binary digit* adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat lain dalam setiap detiknya melalui suatu media.

Bandwidth adalah konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara mengirimkan informasi maupun media yang dipakai dalam proses penghantaran informasi. Hal ini karena adanya hukum fisika maupun batasan teknologi. Ini akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, maupun perlakuan khusus terhadap media yang dipakai. Berikut adalah contoh tabel batasan panjang medium dan kecepatan maksimum aliran data (Wiley, 2006).

Tabel 2.2 Spesifikasi media transmisi data

Media	Panjang Maksimum	Kecepatan Maksimum
Kabel Coaxial 50 ohm (Ethernet 10Base2, ThinNet)	200m	10-100Mbps
Kabel Coaxial 75 ohm (Ethernet 10Base5, ThinNet)	500m	10-100Mbps
UTP kategori 5 (Ethernet 10BaseT, 10Base-Tx)	100m	10Mbps
UTP kategori 5 (Ethernet 10BaseT, FastEthernet)	100m	100Mbps
Multimode (62.5/12um) Serat Optik 100Base-	2km	100Mbps

FX		
Singlemode (10um core) Serat Optik 1000Base-LX	3km	1000Mbps
Wireless	100m	2Mbps
Inframerah	1m	4Mbps

Secara umum, koneksi dengan *bandwidth* yang besar/tinggi memungkinkan pengiriman informasi yang besar seperti pengiriman gambar/images dalam *video presentation*.

Data dalam jumlah besar akan menempuh saluran yang memiliki *bandwidth* kecil lebih lama dibandingkan melewati saluran yang memiliki *bandwidth* yang besar. Kecepatan transmisi tersebut sangat dibutuhkan untuk aplikasi Komputer yang memerlukan jaringan terutama aplikasi *real time*, seperti *video conference*. Penggunaan *bandwidth* untuk LAN bergantung pada tipe alat atau medium yang digunakan, umumnya semakin tinggi *bandwidth* yang ditawarkan oleh sebuah alat atau medium, semakin tinggi pula nilai jualnya. Sedangkan penggunaan *bandwidth* untuk WAN bergantung dari kapasitas yang ditawarkan dari pihak ISP, perusahaan harus membeli *bandwidth* dari ISP, dan semakin tinggi *bandwidth* yang diinginkan, semakin tinggi pula harganya. Sebuah teknologi jaringan baru dikembangkan dan infrastruktur jaringan yang ada diperbaharui, aplikasi yang akan digunakan umumnya juga akan

mengalami peningkatan dalam hal konsumsi *bandwidth*. *Video streaming* dan *Voice over IP* (VoIP) adalah beberapa contoh penggunaan teknologi baru yang turut mengkonsumsi *bandwidth* dalam jumlah besar.

2.7 **Packet Loss**

Menurut Francisco J Hens dan Jose M Caballero sebuah paket dikatakan hilang jika tidak tiba di tempat tujuan. Hal ini dapat dianggap bahwa paket tersebut berisi kesalahan atau datang terlambat. "Terlambat" dapat memiliki arti yang berbeda untuk aplikasi yang berbeda. Dalam aplikasi real time *audio* dan *video*, paket yang diterima terlambat akan menjadi paket yang tidak berguna. Alamat IPPM paket yang hilang telah dievaluasi pada RFC 2640. Definisi yang diberikan berdasarkan pada *application-dependent timeout*. IPPM hanya melibatkan satu paket oleh karena itu paket hanya terdiri dari 2 nilai yaitu "0" bila berhasil diterima sebelum batas waktu dan "1" ketika hilang.

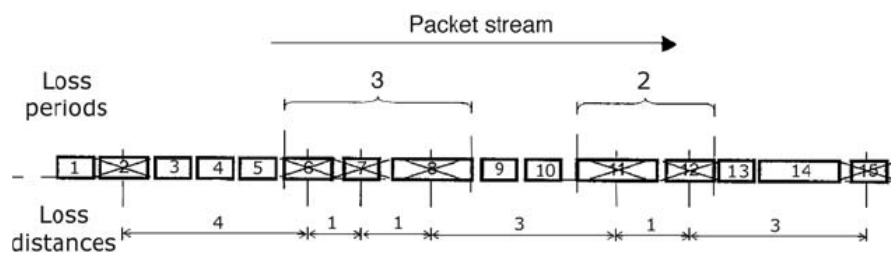
Terjadinya yang paket hilang tergantung pada batas waktu dan dapat menimbulkan masalah dalam prakteknya karena sebagian besar waktu itu membutuhkan paket *time-stamping* dan sinkronisasi antara analisa dan generator. Dalam hal ini *timestamping* dan sinkronisasi dihindari dan hanya paket *sequencing* yang dibutuhkan.

Ping adalah suatu perintah yang dikenal untuk mengukur paket yang hilang dari proses *round-trip*. Namun demikian, ada alternatif lain untuk mensinkronkan paket hilang yang satu arah, jika paket merupakan bagian dari aliran yang diurutkan: paket akan dianggap hilang jika paket diterima dengan nomor urut yang lebih tinggi dari yang diharapkan. Sedikit paket yang hilang tidak berarti cukup untuk menjelaskan sifat dari jalur antara pemancar dan penerima. Ada statistik metrik yang memberikan gambaran yang lebih akurat. Sebagai contoh, paket yang hilang sering dinyatakan sebagai rasio dari paket yang hilang dan paket yang ditransmisikan. Paket yang hilang mungkin terjadi ketika kesalahan transmisi, tetapi alasan utama paket hilang adalah jalur jaringan yang padat sehingga terjadinya *collision*. *Node intermediate* bereaksi terhadap beban lalu lintas data yang tinggi dengan menjatuhkan paket sehingga dengan demikian menghasilkan paket yang hilang. *Congestion* atau kepadatan lalu lintas data ini merugikan dekoder suara dan *video* yang menyebabkan kerugian terdistribusi secara merata.

Loss distance dan *loss period* adalah metrik yang memberikan informasi tentang loss event (lihat gambar 2.2)

- *Loss distance* adalah perbedaan dalam nomor urut dua paket yang berturut-turut hilang, dipisahkan atau tidak dengan paket yang diterima.
- *Loss period* adalah jumlah paket dalam sebuah kelompok dimana semua paket telah hilang.

Parameter-parameter ini memungkinkan untuk mengumpulkan statistik yang menggambarkan suatu proses paket yang hilang antara pemancar dan penerima. Di mana standar minimal *delay* menurut ITU-T 1541 adalah sebesar 0-150ms.



Gambar 2.2 Variasi *delay* 1 arah : pengukuran dan dampak pada periodisitas data.

2.8 Jitter

Tipikal paket IP membawa data MPEG-2 *video-streaming* data yang terdiri atas 7 MPEG *transport stream packets* yang masing-masing berisi *payload* dengan ukuran 184 byte dan *header* yang berukuran 4 byte. Hasilnya akan menghasilkan 36 byte, ditambah

dengan paket *overhead-UDP header* yang berukuran 8 byte, 20 byte untuk *IP header*, untuk *Ethernet header* dan 10 byte untuk *ATM overhead* untuk ukuran total frame sebesar 1367 bytes. *Jitter* didefinisikan variasi dalam waktu kedatangan paket (*packet arrival time*), disebabkan oleh kemacetan jaringan atau server (Justin Paul, 2007). jitter menurut ITU-T 1541 adalah sebesar kurang dari 75ms.

2.9 Protokol

Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan komunikasi dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Protokol perlu diutamakan pada penggunaan standar teknis, untuk menspesifikasi bagaimana membangun komputer atau menghubungkan peralatan perangkat keras. Protokol secara umum digunakan pada komunikasi *real time* dimana standar digunakan untuk mengatur struktur dari informasi untuk penyimpanan jangka panjang. Sangat sulit untuk menggeneralisir protokol dikarenakan protokol memiliki banyak variasi di dalam tujuan penggunaanya.

Kebanyakan protokol memiliki salah satu atau beberapa dari hal berikut:

- Melakukan deteksi adanya koneksi fisik atau ada tidaknya komputer atau mesin lainnya.
- Melakukan metode "jabat-tangan" (*handshaking*).
- Negosiasi berbagai macam karakteristik hubungan.
- Bagaimana mengawali dan mengakhiri suatu pesan.
- Bagaimana format pesan yang digunakan.
- Yang harus dilakukan saat terjadi kerusakan pesan atau pesan yang tidak sempurna.
- Mendeteksi kerugian pada hubungan jaringan dan langkah-langkah yang dilakukan selanjutnya
- Mengakhiri suatu koneksi.

Jenis-jenis protocol yang akan digunakan dalam penelitian ini adalah RTMP (*Real Time Messaging Protocol*), SIP (*Session Initiation Protocol*) dan HTTP (*Hypertext Transfer Protocol*).

2.9.1 Real-Time Messaging Protocol (RTMP)

Real Time Messaging Protocol adalah protokol komunikasi yang dikembangkan sebagai koneksi *real time messaging* dua arah antara *clients* dan *Server*, yang membantu dalam

pengiriman data, *audio*, *visual* ke dan dari klien. Para pengguna lainnya pun dapat menghubungkan kepada *server* yang sama sehingga dapat menerima pesan, mendapatkan data yang terbaru, *audio*, maupun *visual*.

2.9.2 Session Initiation Protocol (SIP)

SIP adalah *Session Initiation Protocol*, yaitu protokol yang digunakan untuk inisiasi, modifikasi dan terminasi sesi komunikasi VoIP. SIP adalah protokol Open Standard yang dipublikasikan oleh IETF, RFC 2543 dan RFC 3261. Selain digunakan untuk negosiasi sesi komunikasi suara, SIP juga dapat digunakan untuk negosiasi sesi komunikasi data media lain seperti *video* dan teks. Disebutkan sebagai hanya melakukan “negosiasi sesi komunikasi” adalah karena SIP merupakan *signaling protocol*, bukan *media transfer protocol*. Artinya SIP tidak menghantar data media (*voice*, *video* dan teks), melainkan hanya melakukan negosiasi sesi komunikasi saja dan memanfaatkan protokol lain seperti RTP sebagai *media transfer protocol*.

2.9.3 *Hypertext Transfer Protocol (HTTP)*

HTTP adalah sebuah protokol meminta/menjawab antara klien dan server. Sebuah klien HTTP (seperti *web browser* dan lain sebagainya), biasanya memulai permintaan dengan membuat hubungan ke *port* tertentu di sebuah server *Web hosting* tertentu (biasanya *port* 80). Klien yang mengirimkan permintaan HTTP juga dikenal dengan *pengguna agent*. Server yang meresponsnya, yang menyimpan sumber daya seperti berkas HTML dan gambar, dikenal juga sebagai *origin server*. Di antara pengguna agent dan juga *origin server*, bisa saja ada penghubung, seperti halnya *proxy*, *gateway*, dan juga *tunnel*.

HTTP tidaklah terbatas untuk penggunaan dengan TCP/IP, meskipun HTTP merupakan salah satu protokol aplikasi TCP/IP paling populer melalui Internet. Memang HTTP dapat diimplementasikan di atas protokol yang lain di atas Internet atau di atas jaringan lainnya. seperti disebutkan dalam "*implemented on top of any other protocol on the Internet, or on other networks.*", tapi HTTP membutuhkan sebuah protokol lapisan transport yang dapat diandalkan.

2.10 *Prototype* SDLC

Salah satu model pengembangan perangkat lunak pada suatu penelitian adalah SDLC yang merupakan singkatan dari *System Development Life Cycle*. SDLC merupakan keseluruhan proses dalam membangun sistem melalui beberapa langkah. Ada beberapa model SDLC. Model yang cukup populer dan banyak digunakan adalah waterfall. Beberapa model lain SDLC misalnya *fountain*, *spiral*, *rapid*, *prototype*, *incremental*, *build & fix*, dan *synchronize & stabilize*. Dengan siklus SDLC, proses membangun sistem dibagi menjadi beberapa langkah dan pada sistem yang besar, masing-masing langkah dikerjakan oleh tim yang berbeda. Dalam penelitian ini model SDLC yang digunakan adalah *prototype*.

Spesifikasi fungsi pada umumnya berhubungan dengan pandangan pengguna terhadap sistem, hanya dengan *prototype* tampilan bagi pengguna sudah cukup untuk memeriksa yang dibutuhkan. Menu-menu, bentuk tampilan *input*, tampilan keluaran, atau laporan yang dicetak, pertanyaan-pertanyaan, pesan-pesan merupakan calon yang ideal untuk *prototype*.

Di lain pihak, perhitungan yang rumit, kumpulan *update* data dan *realtime*, dan sistem yang bersifat *scientific* sangat sulit untuk

dijadikan model. Sistem yang paling sesuai untuk prototipe adalah satu dari banyak hal yang bergantung pada sistem *input* atau *output* dari *pengguna*.

Keuntungan dari prototype yaitu:

1. Menghasilkan syarat yang lebih baik dari produksi yang dihasilkan oleh metode 'spesifikasi tulisan'.
2. Pengguna dapat mempertimbangkan sedikit perubahan selama masih bentuk prototipe.
3. Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.
4. Pengguna merasa puas. Pertama, pengguna dapat mengenal melalui komputer. Dengan melakukan prototipe (dengan analisis yang sudah ada), pengguna belajar mengenai komputer dan aplikasi yang akan dibuatkan untuknya. Kedua, pengguna terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung. (Janner Simarmata, 2010)

Metode *prototype* dapat memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.

2.11.1 Bahasa Pemrograman yang Digunakan

Bahasa pemrograman atau sering diistilahkan juga dengan bahasa komputer adalah teknik komando/instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi. Bahasa pemrograman dapat berbasis *web* atau *desktop*. Suatu perangkat lunak berbasis web bersifat lebih dinamis karena dapat diterapkan pada *intranet* maupun *internet*. Oleh karena itu, pada tugas akhir ini akan digunakan bahasa pemrograman berbasis web terkait dengan pengkodean aplikasi *video conferencing*. Bahasa pemrograman dan perangkat lunak yang digunakan antara lain :

a. *Hyper Text Markup Language (HTML)*

HyperText Markup Language (HTML) adalah sebuah bahasa__*markup* yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah *Penjelajah_web Internet* dan format *hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. HTML menjadi alat utama dalam membangun halaman *web*. Namun, HTML bersifat statis sehingga menyulitkan dalam mengolah dan memperbaharui isi dari halaman *web*. Oleh karena itu, HTML dipadankan dengan bahasa pemrograman *web* yang bersifat dinamis seperti ASP, PHP dan sebagainya.

b. *Cascading Style Sheet (CSS)*

CSS merupakan salah satu bahasa *style sheet* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan suatu keharusan dalam membuat *web*, akan tetapi penggunaan CSS merupakan kelebihan tersendiri. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang

dibuat dengan bahasa HTML dan XHTML (Nurhasyim, 2008).

c. *Extensible Markup Language (XML)*

XML (*Extensible Markup Language*) dibuat untuk menjembatani *interoperabilitas* antar komponen dari *platform* yang berbeda. Misalnya komponen *flash* yang ingin mengakses komponen basis data atau dapat juga suatu komponen yang menyediakan informasi dalam format XML, dimanfaatkan oleh komponen lainnya.

Untuk format XML, digunakan *tool* yang bersifat *web service*. XML bekerja sama dengan *web service* menciptakan layanan *interoprabilitas* dengan cara menandai informasi secara terstruktur sehingga memudahkan aplikasi lain mengekstrak, dan menggunakannya.

d. Flex

Flex adalah software yang sekarang dikembangkan oleh Adobe yang bisa digunakan untuk membuat *rich internet application* (RIAs). Flex secara sederhana merupakan sebuah *framework* untuk

membuat RIAs yang berbasis *Flash Player*. Selain merupakan sebuah *framework*, Flex juga merupakan bahasa baru. Dan secara sengaja disebut MXML, *markup language based on Extensible Markup Language* (XML). Layaknya XML, Flex yang merupakan MXML hanyalah sebuah text, jadi MXML sangat mudah dikembangkan hanya dengan menggunakan *text editor*. Flex yang berbasis MXML pada dasarnya menggunakan script XML dan Javascript. Flex merupakan penggabungan dari beberapa keunggulan bahasa pemrograman desktop dengan standar modern dan kemudahan pemrograman web. Dan satu lagi yang membuat Air dan Flex ini memukau adalah tidak bergantung kepada sistem operasi yang digunakan.

e. Ubuntu Server

Ubuntu server merupakan sistem operasi yang dirancang untuk penggunaan sebagai server. Ubuntu Server ini sangat meringankan beban kinerja server sehingga server tidak repot-repot untuk mengolah tampilan desktop.

Kelebihan-kelebihan ubuntu server dibandingkan dengan sistem operasi yang lain adalah gratis, support terhadap berbagai macam aplikasi dan yang paling penting adalah dukungan penuh terhadap segala jenis komputer.