

**SISTEM MONITORING RUANGAN
DENGAN MENGGUNAKAN WEBCAM
YANG DAPAT DIAKSES MELAULI WIRELESS LAN**

SKRIPSI

**Diajukan untuk memenuhi sebagian prasyarat
memperoleh gelar Sarjana Teknik**



**Disusun oleh:
ERIEK FATHURROZI
NIM. 0210633035**

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2009**

**SISTEM MONITORING RUANGAN
DENGAN MENGGUNAKAN WEBCAM
YANG DAPAT DIAKSES MELAULI WIRELESS LAN**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

ERIEK FATHURROZI
NIM. 0210633035

Disahkan dan disetujui oleh dosen pembimbing :

Raden Arief Setiawan, ST., MT.
NIP. 19750819 199903 1 001

Suprpto, ST, MT.
NIP. 19710727 199603 1 001

SISTEM MONITORING RUANGAN DENGAN MENGGUNAKAN WEBCAM YANG DAPAT DIAKSES MELALUI WIRELESS LAN

Disusun oleh :

ERIEK FATHURROZI
NIM. 0210633035

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 10 Agustus 2009

Majelis Penguji :

Heru Nurwarsito, Ir., M.Kom.
NIP. 19650402 199002 1 001

Muhammad Aswin, Ir., MT.
NIP. 19640626 199002 1 001

Himawat Aryadita, ST., M.Sc.
NIP. 19801018 200801 1 003

Mengetahui :

Ketua Jurusan Teknik Elektro

Heru Nurwarsito, Ir., M.Kom.
NIP. 19650402 199002 1 001

PENGANTAR

Assalamualaikum Wr. Wb

Alhamdulillah Robbil ‘Alamin, segala puja dan puji syukur kehadiran Allah SWT atas Rahmat dan Karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi dengan judul “MONITORING RUANGAN DENGAN WEBCAM YANG DAPAT DIAKSES MELALUI WIRELESS LAN” ini dengan baik.

Dalam penyusunan laporan ini penulis banyak mendapatkan masukan dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

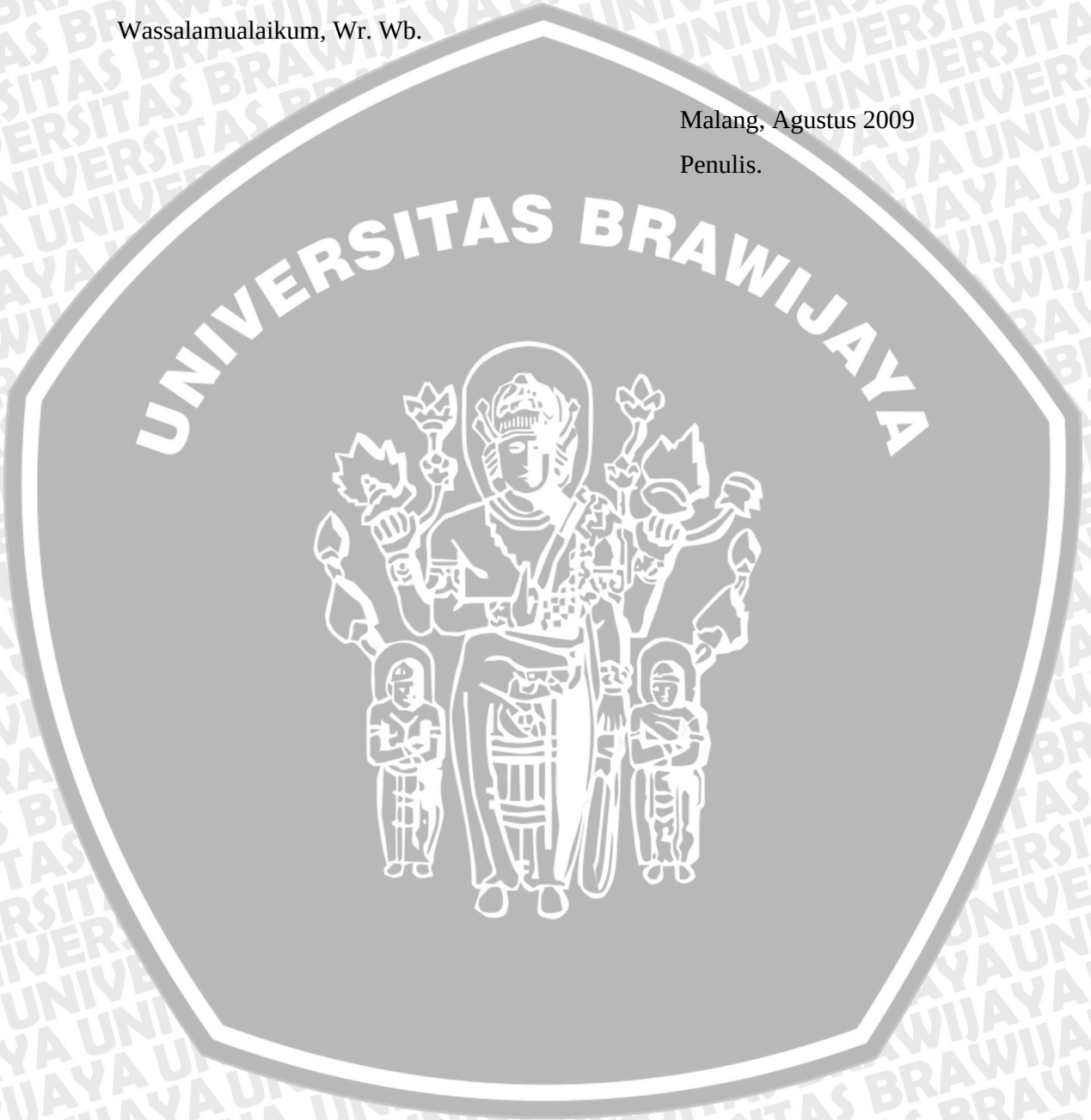
1. Allah SWT yang telah memberikan rahmat, kemudahan serta kesabaran sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak, Ibu, dan adik-adik tercinta yang memberi do’a, kasih sayang, perhatian dan dukungan moral.
3. Ir. Heru Nurwarsito M.kom., selaku Ketua Jurusan Teknik Elektro.
4. Rudi Yuwono, ST., M.Sc., selaku Sekretaris Jurusan Teknik Elektro.
5. Raden Arief Setiawan, ST., MT. selaku pembimbing I yang telah membantu dalam bimbingan skripsi ini.
6. Suprpto, ST., MT. selaku dosen pembimbing II yang telah membantu dalam bimbingan skripsi ini.
7. Seluruh jajaran staff Dosen dan Karyawan Teknik Elektro Universitas Brawijaya Malang.
8. Sahabat yang selalu memberikan motivasi, do’a dan berbagi ilmu yang bermanfaat, Swasti Komala, ST. dan Maya Indah, ST.
9. Kawan, sahabat, teman seperjuangan di Candi Mendut V / 21 (Anggas, Ancau, Bograh, Epeb, Mbah Edy, Bang Pit, Rosyid, Udin), terimakasih atas bantuannya selama 7 tahun terakhir. Serta teman sharing Galih Kusumo dan Abdul Azis, ST.
10. Teman-teman B-213 Dan Teman-teman Candi Mendut serta seluruh Elektro UB 2002 (Gamers dan precboth) atas dukungan, bantuan dan do’a nya.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan Skripsi ini masih banyak kekurangan dan kekeliruan, untuk itu penulis mengharapkan masukan, kritik dan saran demi kelengkapan dan kesempurnaan laporan ini. Semoga skripsi ini memberikan manfaat bagi kita semua. Amin.

Wassalamualaikum, Wr. Wb.

Malang, Agustus 2009

Penulis.



ERIEK FATHURROZI. Agustus 2009. Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. **Monitoring Ruangan dengan Menggunakan Webcam yang Dapat Diakses Melalui Wireless LAN.**

Dosen Pembimbing : Raden Arief Setiawan,ST., MT. dan Suprpto,ST., MT.

Penggunaan jaringan *wireless* sebagai media penyaluran sinyal kontrol dan monitoring telah berkembang pesat, selaras dengan perkembangan karakteristik masyarakat modern yang memiliki mobilitas tinggi, mencari layanan yang fleksibel dan efisien disegala aspek, serba mudah dan memuaskan dan pada akhirnya pertambahan jumlah pemakaian komputerpun tidak dapat dihindarkan lagi, diantaranya untuk sistem keamanan ruangan, pengaturan peralatan di rumah, kontrol robot dan mesin produksi di Industri.

Dengan semakin meningkatnya tindak kejahatan dan pencurian, maka semakin besar pula kebutuhan akan media yang dapat memonitor suatu ruangan. Untuk itu, dibutuhkanlah suatu sistem monitoring ruangan yang mampu memonitor suatu ruangan yang dapat diakses melalui *web* sehingga dapat dimonitor dari jarak jauh melalui jaringan lokal (LAN). Untuk memonitor ruangan dalam jarak yang jauh, dibutuhkan sebuah IP Cam yang berfungsi menampilkan gambar (*video streaming*) dalam suatu ruangan. Tetapi kendala yang dirasakan dalam menggunakan IP Cam tersebut harganya masih dirasa sangat mahal.

Dengan latar belakang dan pertimbangan tersebut, maka penulis mencoba menawarkan suatu solusi dimana gambar (*video streaming*) yang ditampilkan tersebut digantikan dengan menggunakan *webcam* sebagai alternatif.

Dari pengujian didapat nilai *packet loss* rata rata sebesar 0,031%. Nilai packet loss ini masih jauh dibawah nilai packet loss yang diperbolehkan yaitu 1%. Hal ini menunjukkan bahwa sistem jaringan yang telah dibangun *reliable* dalam proses *packet delivery*.

DAFTAR ISI

Kata Pengantar	i
Abstrak	ii
Daftar Isi	iii
Daftar Gambar	v
Daftar Tabel	vii
 BAB I. Pendahuluan	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Batasan masalah	1
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Sistematika Penulisan.....	3
 BAB II. Dasar Teori	
2.1. Umum	5
2.2. Konsep Jaringan Komputer	5
2.2.1. <i>Local Area Network (LAN)</i>	6
2.2.2. <i>Metropolitan Area Network</i>	7
2.2.3. <i>Wide Area Network</i>	7
2.2.4. <i>Jaringan Client-Server</i>	7
2.3. <i>Open Protocol Interconnection (OSI) Layer</i>	9
2.4. <i>Transport Layer</i>	10
2.4.1. <i>Real-Time Transport Protocol</i>	11
2.4.1.1. <i>Aktifitas RTP pada Pengirim</i>	12
2.4.1.2. <i>Aktifitas RTP pada Penerima</i>	13
2.4.2. <i>User Data Protocol</i>	15
2.5. <i>Streaming Protocol</i>	16
2.6. <i>Multimedia Streaming</i>	18

2.7. Kompresi <i>Multimedia</i>	19
2.7.1 Kompresi <i>Audio</i>	19
2.7.1 Kompresi <i>Video</i>	23
2.7.3 Kompresi <i>MPEG</i>	24
2.7.3.1. <i>MPEG Audio</i>	24
2.7.3.2. <i>MPEG Video</i>	25
2.7.3.3. <i>MPEG Transport System</i>	26
2.8. Aplikasi <i>Web</i>	27
2.8.1 <i>HTML (HyperText Markup Language)</i>	27
2.8.2 <i>PHP (Perypheral Hypertext Protocol)</i>	27
2.9. Performansi <i>Sistem</i>	28
2.9.1 <i>Bandwidth Multimedia Streaming</i>	29
2.9.2 <i>Troughput</i>	39
2.9.3 <i>Kecepatan Transmisi</i>	30
2.9.4 <i>Packet Loss</i>	30
2.10. <i>Webcam</i>	31
2.11. <i>USB (Universal Serial Bus)</i>	33
2.11.1 <i>Konfigurasi Pin USB</i>	34
2.12. <i>Karakteristik Video Digital</i>	35
 BAB III. Metodologi Penelitian	
3.1. <i>Observasi</i>	38
3.2. <i>Studi Literatur</i>	38
3.3. <i>Perancangan</i>	39
3.4. <i>Implementasi</i>	39
3.5. <i>Pengujian dan Analisis</i>	39
3.6. <i>Pengambilan Kesimpulan</i>	40
 BAB IV. Perancangan	
4.1. <i>Analisis Sistem</i>	41
4.1.1. <i>Analisis Kebutuhan Perangkat Lunak</i>	41
4.1.2. <i>Spesifikasi Kebutuhan Perankat Lunak</i>	42

4.2. Perancangan Perangkat Lunak	42
4.2.1. Perancangan Halaman Utama	43
4.2.2. Perancangan Metode <i>Multimedia Streaming</i>	47
4.2.3. Perancangan <i>Multicast Routing</i> pada Jaringan FTUB	48
4.2.4. Perancangan Perangkat Keras Jaringan	49
4.2.5. Perancangan Perangkat Lunak Jaringan	49
4.3. Perancangan Sistem	50
4.3.1. Diagram Blok Sistem	51
4.3.2. Diagram Konteks	51
4.3.3. Cara Kerja Sistem	52
 BAB V. Implementasi Perangkat Lunak	
5.1. Lingkungan Implementasi	54
5.2. Algoritma Pemrograman	55
5.2.1. Proses Sistem <i>Video Streaming</i>	55
5.2.2. Aplikasi <i>Video Streaming</i>	56
5.2.2.1. Proses Menampilkan <i>Video</i>	57
5.2.2.2. Proses <i>Play</i>	60
5.3. Implementasi Antar Muka	61
5.3.1. Algoritma <i>Log In</i>	61
5.4. Implementasi <i>Web Server</i>	64
 BAB VI. Pengujian dan Analisis	
6.1. Pengujian	65
6.1.1. Halaman Web	65
6.1.2. Pengujian Sistem Monitoring Ruangan	67
6.1.3. Pengujian Performansi Sistem aplikasi	68
6.1.4. Pengujian Performansi Jaringan	69
6.2. Analisis	74
6.2.1. Delay	74
6.2.1.1. Analisis <i>Delay</i> Enkapsulasi Dan Dekapsulasi	74
6.2.1.2. Analisis <i>Delay</i> Transmisi	75

6.2.1.3. Analisis <i>Delay</i> Propagasi	76
6.2.1.4. Analisis <i>Delay</i> Antrian	77
6.2.1.2. Analisis <i>Delay</i> End-to-end	78
6.2.2. Pengujian <i>Delay</i> Manual.....	79
6.2.3. Analisa <i>Packet Loss</i>	79
6.3. Pengujian <i>Mean Of Square (MOS)</i>	80
6.3.1. Pengujian Kualitas Video	81
6.3.2. Pengujian Pengaruh <i>Delay</i>	82
BAB VI. Kesimpulan dan Saran	
7.1. Kesimpulan	85
7.2. Saran	85

DAFTAR PUSTAKA
LAMPIRAN



DAFTAR GAMBAR

Gambar 2.1 <i>Network Topologies</i>	6
Gambar 2.2 <i>Computer Network Scheme</i>	9
Gambar 2.3 <i>RTP Sender</i>	12
Gambar 2.4 Standar IETF dan ITU untuk <i>protocol transport Audio/Video</i>	12
Gambar 2.5 Diagram <i>RTP sender</i>	13
Gambar 2.6 Diagram <i>RTP receiver</i>	14
Gambar 2.7 <i>Format paket UDP</i>	15
Gambar 2.8 Komunikasi <i>Streaming</i>	16
Gambar 2.9 Proses <i>Handshaking</i> dalam protokol RTMP	17
Gambar 2.10 Proses Konversi dan Kompresi Data.....	19
Gambar 2.11 Proses <i>Sampling</i>	20
Gambar 2.12 Proses <i>Quantizing</i> dan Coding.....	21
Gambar 2.13 Paket data <i>multimedia streaming</i> dengan enkapsulasi MPEGTS	27
Gambar 2.14 <i>PC Web Camera Sturdy</i>	31
Gambar 2.17 <i>Flowchart</i> tahap-tahap penampilan <i>video</i> pada PC server	33
Gambar 2.18 Konektor USB.....	34
Gambar 2.19 <i>Pin USB</i>	24
Gambar 4.1 Diagram Alir Analisis Sistem	41
Gambar 4.2 Diagram Alir Analisis Sistem Use case.....	43
Gambar 4.3 Perancangan <i>Web Monitoring</i> Ruangan.....	46
Gambar 4.4 Blok Diagram proses <i>streaming</i>	48
Gambar 4.5 Diagram Pohon Perancangan Sistem	50
Gambar 4.6 Koneksi <i>server</i> dan <i>client</i> melalui <i>WirelessLAN</i>	51
Gambar 4.7 <i>Data Flow Diagram</i> Sistem.....	52
Gambar 4.8 Diagram alir sistem	53
Gambar 5.1 Algoritma Sistem <i>Video Streaming</i>	56
Gambar 5.2 <i>VLC Application</i>	57
Gambar 5.3 <i>VLC Application > File</i>	57
Gambar 5.4 <i>VLC Application > File > Open Capture Device</i>	58
Gambar 5.5 <i>VLC Application > File > Open Capture Device: Setting</i>	59
Gambar 5.6 Algoritma Menampilkan <i>Video</i>	60
Gambar 5.7 Algoritma <i>Play Video</i>	61

Gambar 5.8 Algoritma <i>Log In</i>	62
Gambar 5.9 Antarmuka <i>Web</i>	62
Gambar 5.10 <i>HTML Code</i>	64
Gambar 5.11 Aplikasi <i>XAMPP</i>	65
Gambar 6.1 Pengujian Halaman system Monitoring Ruangan.....	68
Gambar 6.2 Pengujian Sistem Monitoring Ruangan.....	70



DAFTAR TABEL

Tabel 2.1 <i>Open System Interconnection</i>	10
Tabel 2.2 Standar Kompresi <i>Audio</i>	22
Tabel 2.3 Standar Kompresi <i>Video</i>	23
Tabel 2.4 Spesifikasi kompresi <i>MPEG Audio</i>	25
Tabel 2.5 Standar <i>Codec</i> untuk <i>video</i>	26
Tabel 2.6 Spesifikasi <i>Webcam Sturdy</i>	32
Tabel 2.7 Fungsi masing-masing <i>pin</i> saluran <i>USB</i>	35
Tabel 6.1 Aliran Proses Log In.....	67
Tabel 6.2 Performansi Video Streaming.....	70
Tabel 6.3 Hasil capture data dari kelas 2-1.....	71
Tabel 6.4 Hasil capture data dari kelas 2-2.....	71
Tabel 6.5 Hasil capture data dari kelas 2-3.....	71
Tabel 6.6 Hasil capture data dari kelas 2-4.....	72
Tabel 6.7 Hasil capture data dari kelas 2-5.....	72
Tabel 6.8 Hasil capture data dari kelas 2-6.....	72
Tabel 6.9 Hasil capture data dari kelas 2-7.....	73
Tabel 6.10 Hasil capture data dari kelas 2-8.....	73
Tabel 6.11 Hasil capture data dari kelas 1-1.....	73
Tabel 6.12 Hasil capture data dari kelas 1-2.....	74
Tabel 6.13 Hasil capture data dari kelas 1-3.....	74
Tabel 6.14 Hasil capture data dari kelas 1-4.....	74
Tabel 6.15 Hasil capture data dari kelas 1-5.....	75
Tabel 6.16 Hasil capture data dari kelas 1-6.....	75
Tabel 6.17 Tabulasi hasil pengujian	75
Tabel 6.18 Pengujian delay seara manual.....	81
Tabel 6.19 Hasil capture dari kelas 1-6	81
Table 6.20 Indeks kualitas video	82
Tabel 6.21 Indeks pengaruh delay	82

Tabel 6.22 Hasil analisa kualitas video menggunakan MOS83

Tabel 6.23 Hasil analisa pengaruh delay menggunakan MOS83



BAB I

PENDAHULUAN

1.1. Latar Belakang

Penggunaan jaringan *wireless* sebagai media penyaluran sinyal kontrol dan monitoring telah berkembang pesat, selaras dengan perkembangan karakteristik masyarakat modern yang memiliki mobilitas tinggi, mencari layanan yang fleksibel dan efisien disegala aspek, serba mudah dan memuaskan dan pada akhirnya pertambahan jumlah pemakaian komputerpun tidak dapat dihindarkan lagi, diantaranya untuk sistem keamanan ruangan, pengaturan peralatan di rumah, kontrol robot dan mesin produksi di Industri. Dengan semakin meningkatnya tindak kejahatan dan pencurian, maka semakin besar pula kebutuhan akan media yang dapat memonitor suatu ruangan. Untuk itu, dibutuhkanlah suatu sistem monitoring ruangan yang mampu memonitor suatu ruangan yang dapat diakses melalui *web* sehingga dapat dimonitor dari jarak jauh melalui jaringan lokal (LAN). Untuk memonitor ruangan dalam jarak yang jauh, dibutuhkan sebuah IP Cam yang berfungsi menampilkan gambar (*video streaming*) dalam suatu ruangan. Tetapi kendala yang dirasakan dalam menggunakan IP Cam tersebut harganya masih dirasa sangat mahal. Dengan latar belakang dan pertimbangan tersebut, maka penulis mencoba menawarkan suatu solusi dimana gambar (*video streaming*) yang ditampilkan tersebut digantikan dengan menggunakan *webcam* sebagai alternatif, sedangkan untuk mengakses sistem monitoring diperlukan jaringan lokal dengan spesifikasi *IP Address* satu kelas dan satu segmen. Dengan adanya teknologi ini, biaya untuk membangun sistem monitoring ruangan dapat ditekan secara signifikan. [ANY-08]

Ada tempat-tempat tertentu yang memerlukan pengawasan ketat dan pemantauan rutin dari petugas, seperti bangunan gedung. Pada kasus tersebut petugas jaga biasanya hanya terdiri atas satu atau dua orang, sehingga akan sangat merepotkan dan melelahkan bila petugas tersebut harus berkeliling untuk memeriksa bangunan yang luas tersebut. Oleh karena itu diperlukan suatu alat yang dapat menggantikan tugas penjagaan berkeliling tersebut.

Hal tersebut mendasari pembuatan alat dengan memanfaatkan perkembangan teknologi yang ada. Pesatnya perkembangan dunia internet akhir-akhir ini telah memicu berkembangnya teknologi baru yang memanfaatkan teknologi internet tersebut sebagai media untuk mewujudkan impian manusia akan sebuah aplikasi pengoperasian peralatan jarak jauh. Peralatan yang dikontrol dapat berupa sebuah robot yang disertai dengan kamera yang bisa digerakkan. Sehingga robot tersebut dapat dipakai sebagai pengganti petugas dalam pemeriksaan keliling suatu bangunan. [ARF-05]

1.2. Rumusan Masalah

Mengacu pada kondisi yang ada tersebut di atas, rumusan masalah yang diangkat dan ditekankan pada tugas akhir ini adalah pembangunan sistem monitoring ruangan yang terintegrasi dan dinamis yang memberikan informasi mengenai keadaan disekitar ruangan melalui camera yang dapat diakses melalui jaringan *wireless*.

1.3. Batasan Masalah

Ruang lingkup penulisan dibatasi pada :

1. Ruangan dengan intensitas cahaya yang relatif terang dan dimensi ruangan yang tidak terlalu besar.
2. Perancangan dan penerapan aplikasi *streaming* pada *server* menggunakan *open source software* yaitu VLC.
3. Perancangan dan penerapan sistem dilakukan pada *LAN (Local Area Network)*
4. Menggunakan *Hyper Text Transfer Protocol (HTTP)* sebagai protokol level kontrol untuk *multimedia streaming*.
5. Menggunakan teknologi HTML, CSS dan PHP sebagai *client interface* pada server.

1.4. Tujuan

Tujuan akhir dari penelitian ini adalah untuk mendapatkan suatu aplikasi yang dinamis dan bersifat ekstensibel, mudah dikembangkan, serta dapat memudahkan penggunaanya untuk memantau beberapa ruangan dengan menggunakan kamera web yang dapat diakses melalui jaringan *wireless LAN*.

1.5. Manfaat

Adapun manfaat dari sistem monitoring ruangan ini antara lain :

1. Meningkatkan keamanan ruangan.
2. Untuk memudahkan dalam memantau beberapa ruangan karena dapat diakses secara *real time* melalui jaringan yang tanpa batas.

1.6. Sistematika Penulisan

Pembahasan Tugas Akhir ini terdiri dari 7 bab yang disusun dengan sistematika penulisan sebagai berikut :

Bab I Pendahuluan

Bab I mendeskripsikan latarbelakang masalah yang melandasi perlunya sistem monitoring ruangan, kemudian rumusan masalah yang menjelaskan tentang masalah yang akan dibahas, dilanjutkan dengan batasan masalah yang memberikan batasan dan arah penulisan, tujuan penelitian, manfaat, serta sistematika penulisan.

Bab II Landasan Teori

Bab II membahas teori-teori yang menunjang dalam penulisan tugas akhir ini, diantaranya adalah Wireless LAN, konsep Client-Server, Video Streaming Serta bahasa pemrograman yang akan dipakai.

Bab III Metodologi

Bab ini membahas metodologi yang digunakan dalam penulisan tugas akhir.

Bab IV Perancangan

Bab ini membahas analisis kebutuhan perangkat lunak dan perancangan sistem. Perancangan sistem terdiri dari : diagram arus data dan desain struktur menu serta rancangan *user interface* yang akan dibuat.

Bab V Implementasi

Bab V membahas implementasi hasil dari perancangan. Dalam tugas akhir ini program dibuat dengan menggunakan teknologi HTML, CSS dan PHP sebagai *client interface* pada server.

Bab VI Pengujian dan Analisis

Bab VI membahas pengujian dan analisa program yang dibuat.

Bab VII Penutup

Sebagai penutup bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program, serta saran-saran untuk pengembangan lebih lanjut.



BAB II

DASAR TEORI

2.1. Umum

Perkembangan teknologi jaringan komputer dan komunikasi data memicu tumbuhnya inovasi dalam sistem komunikasi. Salah satu inovasi tersebut adalah *video streaming*, yaitu sebuah teknologi untuk memainkan *file video* secara langsung ataupun dengan *pre-recorded* dari sebuah mesin server. Dengan metode *streaming* maka *live video* dapat diakses dimanapun dan kapan saja yaitu melalui sebuah jaringan komputer.

Pada bab ini dijelaskan kajian pustaka dan dasar teori yang digunakan untuk menunjang penyelesaian skripsi ini. Tinjauan pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan skripsi ini. Pada bab ini dipaparkan dasar teori tentang konsep jaringan komputer, UDP/IP, RTP, IP Multicast, *multimedia streaming*.

Sistem telekomunikasi adalah proses penyaluran informasi dari suatu titik ke titik yang lain melalui media transmisi tertentu. Dalam teknologi komunikasi data informasi yang berbentuk berupa teks, *audio* maupun *video* dikirimkan sebagai data digital. Perkembangan teknologi terutama jaringan komputer dan telekomunikasi yang berbasis komunikasi data yang pesat memicu munculnya inovasi-inovasi baru dalam sistem telekomunikasi. Salah satunya adalah jaringan komputer yang memungkinkan terjadinya komunikasi data antara dua komputer atau lebih, yang terhubung dengan protokol dan aturan-aturan tertentu yang bersifat universal. Dengan memanfaatkan jaringan komputer pertukaran data digital menjadi lebih efektif dan efisien.

2.2. Konsep Jaringan Komputer

Jaringan komputer adalah kumpulan sekelompok komputer otonom yang saling berhubungan antar satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi sumberdaya yang ada. Berbagai perangkat yang terhubung dalam jaringan komputer terkoneksi dengan topologi jaringan tertentu. Topologi jaringan adalah, hal yang menjelaskan

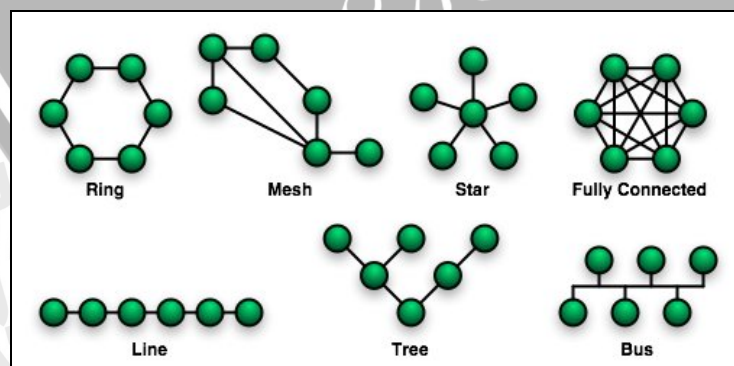
hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu *node*, *link*, dan *station* (Wikipedia, 2008).

Node adalah perangkat-perangkat yang menjadi titik penghubung dalam jaringan komputer. *Link* adalah media transmisi yang digunakan, sedangkan *station* adalah *client* atau *server* dalam jaringan. Perangkat yang berfungsi sebagai *node* antara lain gateway, *router*, dan *switch*. *Router* merupakan perangkat untuk menghubungkan dua atau lebih jaringan dan bertugas sebagai perantara dalam menyampaikan data antar jaringan komputer. *Router* berfungsi untuk mengatur kemana paket data akan diteruskan dalam jaringan.

Dalam pembahasan konsep ini jarak menjadi hal yang penting dalam merancang sebuah jaringan komputer, karena untuk setiap jarak tertentu diperlukan teknik yang berbeda pula. Berdasarkan jarak dan area jangkauannya jaringan komputer di bagi menjadi tiga kelompok yaitu : *Local Area Network (LAN)*, *Metropolitan Area Network (MAN)*, dan *Wide Area Network (WAN)* .

2.2.1. *Local Area Network (LAN)*

LAN adalah jaringan yang mencakup area atau jangkauan yang sempit/tidak begitu luas, seperti jaringan dalam rumah, kampus, atau kantor di gunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* menggunakan peralatan secara bersama-sama dan saling bertukar informasi. LAN dimiliki oleh suatu instansi tanpa menggunakan fasilitas yang telah di sediakan oleh penyedia *line* dari perusahaan telekomunikasi umum. LAN dibangun menggunakan topologi tertentu. Topologi LAN yang di gunakan ada lima macam yaitu : *Linier Bus*, *Ring* , *Tree*, *Star*, dan *Mesh*



Gambar 2.1 Network Topologies

Sumber : http://en.wikipedia.org/Network_topology

2.2.2. **Metropolitan Area Network (MAN)**

MAN merupakan versi LAN yang berukuran lebih besar dan memakai teknologi yang sama dengan LAN. MAN merupakan pilihan yang tepat untuk membangun infrastruktur jaringan komputer antar kantor dalam satu kota. MAN dapat mencakup perusahaan yang memiliki kantor-kantor yang letaknya sangat berdekatan. Jaringan ini mampu menjangkau cakupan area sampai dengan radius 10-50 km.

2.2.3. **Wide Area Network (WAN)**

Wide Area Network adalah sebuah jaringan yang memiliki jangkauan radius yang sangat luas, karena jangkauannya tidak lagi mencakup suatu kota, bahkan lebih yaitu bisa mencakup lintas negara atau lintas benua. Pada sebagian besar WAN, komponen yang dipakai dalam berkomunikasi terdiri dari dua komponen yaitu media transmisi dan perangkat *switching*. Media transmisi digunakan untuk menyalurkan sinyal dan perangkat *switching* digunakan untuk meneruskan sinyal.

2.2.4. **Jaringan Client-Server**

Server adalah komputer yang menyediakan fasilitas bagi komputer-komputer lain didalam jaringan dan *client* adalah komputer-komputer yang menerima atau menggunakan fasilitas yang disediakan oleh *server*. *Server* di jaringan tipe *client-server* disebut dengan *Dedicated Server* karena murni berperan sebagai *server* yang menyediakan fasilitas kepada *workstation* dan *server* tersebut tidak dapat berperan sebagai *workstation*.

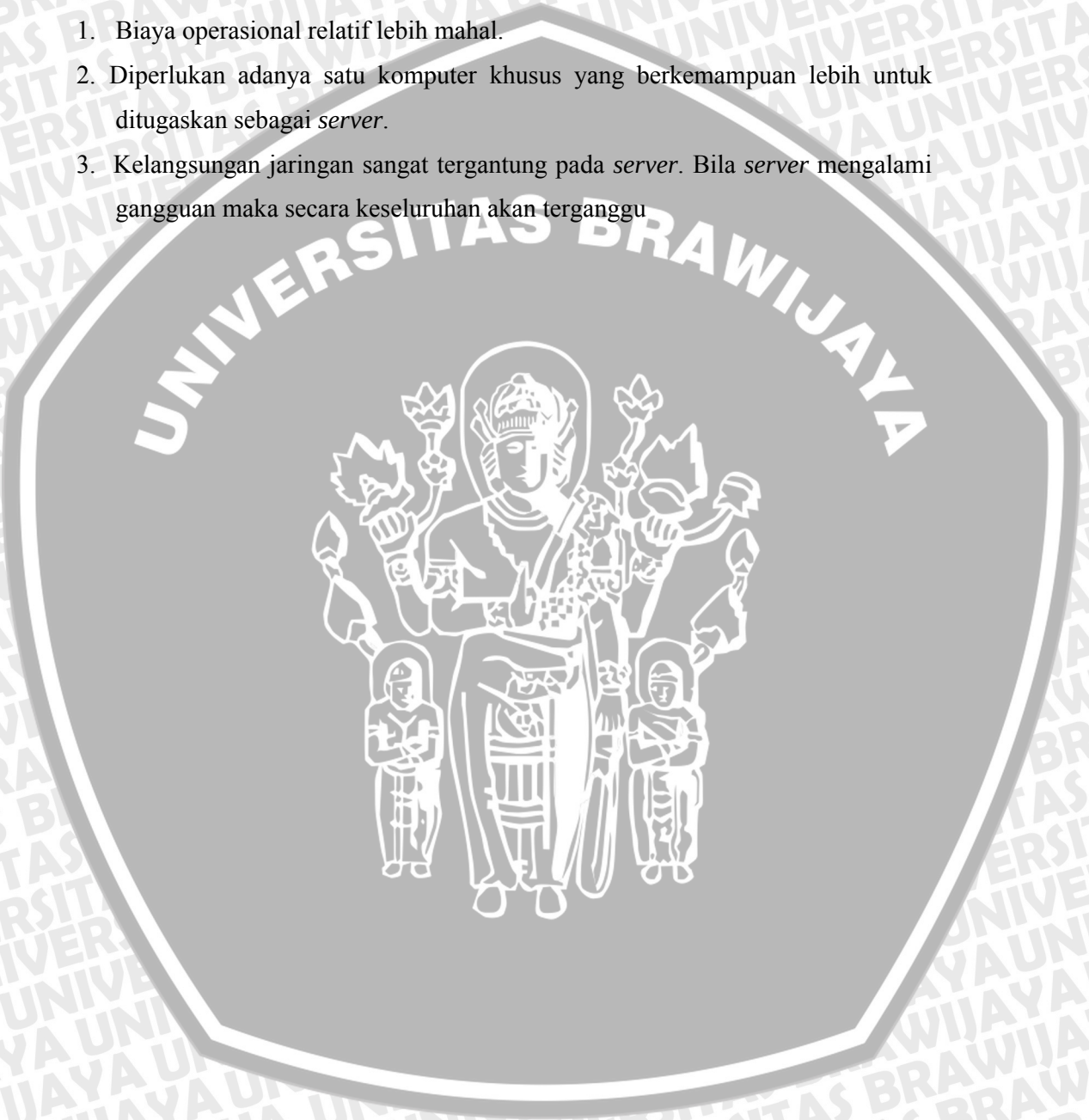
Keunggulan

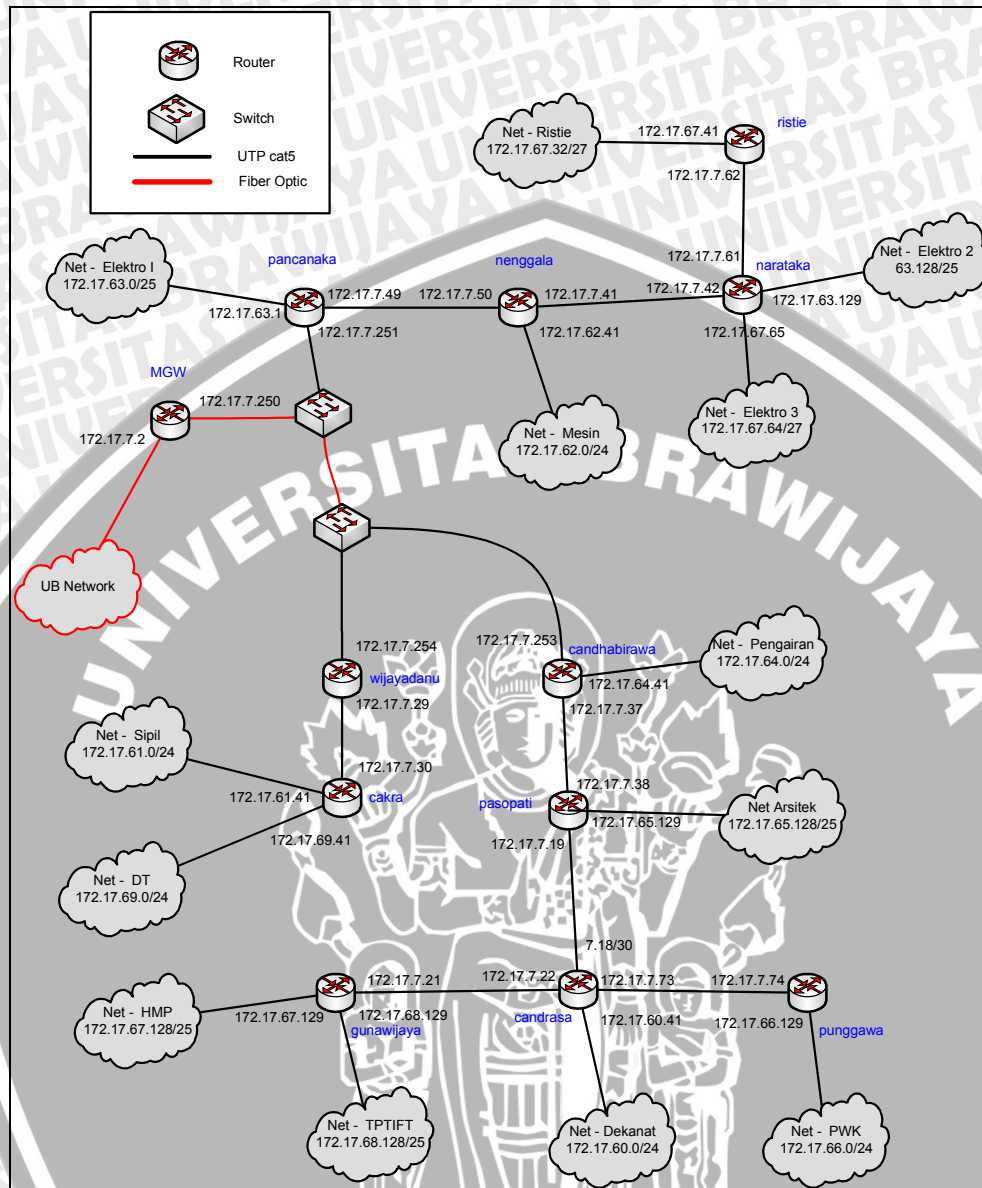
1. Kecepatan akses lebih tinggi karena penyediaan fasilitas jaringan dan untuk sistem pengelolaannya dilakukan secara khusus oleh satu komputer (*server*) yang tidak dibebani dengan tugas lain sebagai *workstation*.
2. Sistem keamanan dan administrasi jaringan lebih baik, karena terdapat seorang pemakai yang bertugas sebagai administrator jaringan, yang mengelola administrasi dan sistem keamanan jaringan.

3. Sistem backup data lebih baik, karena pada jaringan *client-server* backup dilakukan terpusat di *server*, yang akan membackup seluruh data yang digunakan di dalam jaringan.

Kelemahan

1. Biaya operasional relatif lebih mahal.
2. Diperlukan adanya satu komputer khusus yang berkemampuan lebih untuk ditugaskan sebagai *server*.
3. Kelangsungan jaringan sangat tergantung pada *server*. Bila *server* mengalami gangguan maka secara keseluruhan akan terganggu





Gambar 2.2 Computer Network Scheme

Sumber : Data Inventaris Jaringan komputer FTUB 2008, TPTIFT-UB

2.3. *Open System Interconnection (OSI) Layer*

Protokol adalah aturan dalam komunikasi data yang berguna untuk menjaga agar pengiriman data dapat berlangsung. Dalam komunikasi data diperlukan protokol yang bersifat universal agar komunikasi bisa dilakukan oleh berbagai macam pengguna dengan platform yang berbeda. Tujuan dari penggunaan protokol adalah tercapainya pengiriman dan penerimaan data yang benar dan efektif.

Agar komputer dapat berhubungan dengan komputer yang lain dalam jaringan komputer maka diperlukan suatu protokol yang sama. Fungsi protokol mirip dengan bahasa. Untuk mempermudah pengertian, penggunaan, desain, dan keseragaman diantara pembuat perangkat jaringan maka diperlukan suatu protokol standar yang bisa mengakomodasi kebutuhan tersebut. Oleh karena itu *International Standard Organization* (ISO) mengeluarkan suatu model lapisan jaringan yang disebut *Open System Interconnection* (OSI).

OSI terdiri dari 7 layer. Dimana bagian atas dari layer-nya (layer 7,6,dan 5) difokuskan untuk bentuk pelayanan dari suatu aplikasi. Sedangkan untuk layer bagian bawahnya (layer 4,3, 2 dan 1) berorientasikan tentang aliran data dari ujung satu ke ujung yang lainnya.

Tabel 2.1 *Open System Interconnection*

Nama layer	Fungsi	Contoh
Aplikasi (layer 7)	Aplikasi yang saling berkomunikasi antar komputer. Aplikasi layer mengacu pada pelayanan komunikasi pada suatu aplikasi.	Telnet, HTTP, FTP, WWW Browser, NFS, SMTP, SNMP
Presentasi (Layer 6)	Pada layer bertujuan untuk mendefinisikan format data, seperti ASCII text, binary dan JPEG.	JPEG, ASCII, TIFF, GIF, MPEG, MIDI
Sesi (Layer 5)	Sesi layer mendefinisikan bagaimana memulai, mengontrol dan mengakhiri suatu percakapan (biasa disebut session)	RPC, SQL, NFS, SCP
Transport (Layer 4)	Pada layer 4 ini bisa dipilih apakah menggunakan protokol yang mendukung error-recovery atau tidak. Melakukan multiplexing terhadap data yang datang, mengurutkan data yang datang apabila datangnya tidak berurutan.	TCP, UDP, SPX
Network (Layer 3)	Layer ini mendefinisikan pengiriman data dari ujung ke ujung. Untuk melakukan pengiriman pada layer ini juga melakukan pengamatan. Mendefinisikan pengiriman jalur (routing).	IP, IPX, Appletalk DDP
Data Link (layer 2)	Layer ini mengatur pengiriman data dari interface yang berbeda. Semisal pengiriman data dari ethernet 802.3 menuju ke High-level Data Link Control (HDLC), pengiriman data WAN.	IEEE 802.2/802.3, HDLC, Frame relay, PPP, FDDI, ATM
Physical (Layer 1)	Layer ini mengatur tentang bentuk interface yang berbeda-beda dari sebuah media transmisi. Spesifikasi yang berbeda misal konektor, pin, penggunaan pin, arus listrik yang lewat, encoding, sumber cahaya dll	EIA/TIA-232, V35, EIA/TIA- 449, V.24, RJ45, Ethernet, NRZI, NRZ, B8ZS

Sumber : Jaringan komputer, Anonim

2.4. *Transport Layer*

Transport Layer adalah layer yang dirancang agar host sumber dan host tujuan memungkinkan melakukan percakapan. *Transport Layer* merupakan layer komunikasi data yang mengatur aliran data antara dua host yang saling berkomunikasi, untuk keperluan aplikasi yang ada di atasnya . Ada dua buah protokol pada layer ini yaitu TCP dan UDP.

2.4.1. Real Time Transport Protocol (RTP)

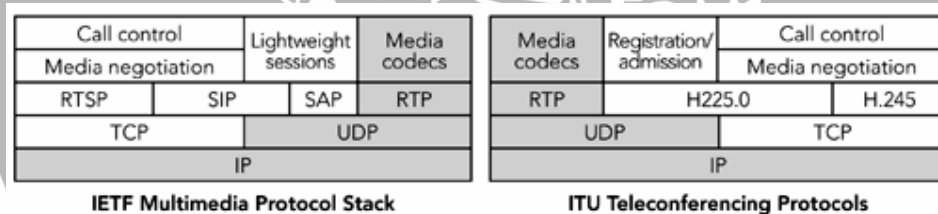
Menggunakan internet sebagai media untuk menghantarkan content *audio* dan *video* bukanlah hal baru. Banyak eksperimen dan penemuan yang telah dilakukan dalam mencari metode terbaik dalam proses *audio/video transport*. Salah satu yang terakhir adalah *Realtime Transport Protocol*. Metode standar untuk proses transmisi *audio/video* dalam jaringan berbasis IP saat ini adalah RTP. RTP bertujuan untuk menyediakan servis yang berguna untuk pertukaran media yang *real-time* seperti *audio* dan *video*, di atas jaringan IP. Servis tersebut meliputi perbaikan waktu, deteksi kesalahan dan koreksi, pengidentifikasian *payload* dan *source*, *reception quality feedback*, sinkronisasi media, dan manajemen keanggotaan. RTP asalnya didesain untuk digunakan pada konferensi *multicast*, menggunakan *session model* yang ringan. Sejak saat itu, RTP terbukti berguna untuk sebuah *range* aplikasi yang lain seperti pada H.323, *webcasting*, dan distribusi TV, baik pada media kabel dan telepon selular. Protokol ini telah diterapkan mulai dari skala *point-to-point* hingga skala *multicast session* dengan ribuan pengguna, serta diterapkan pada jalur telepon selular dengan *bandwidth* kecil hingga aplikasi yang mengantarkan *HDTV* dengan *bandwidth* yang ber-orde *gigabit*.

RTP dikembangkan oleh *Audio/Video Transport working group* oleh IETF dan telah diadopsi oleh ITU sebagai bagian dari seri rekomendasi dari H.323, serta badan standar Internasional lainnya. Disamping RTP, sebuah sistem yang lengkap biasanya membutuhkan protokol yang lain dan standar sesi pengumuman, inisiasi, dan kontrol; kompresi media dan *network transport*. Gambar 2.3 menunjukkan format *header* dari paket RTP

+ Bits	0-1	2	3	4-7	8	9-15	16-31
0	Ver.	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	... CSRC identifiers ...						
96+(CC×32)	Extension header (optional).						
96+(CC×32) + (X×((EHL+1)×32))	Data						

Gambar 2.3 RTP sender
Sumber: Collins Perkins, 2003

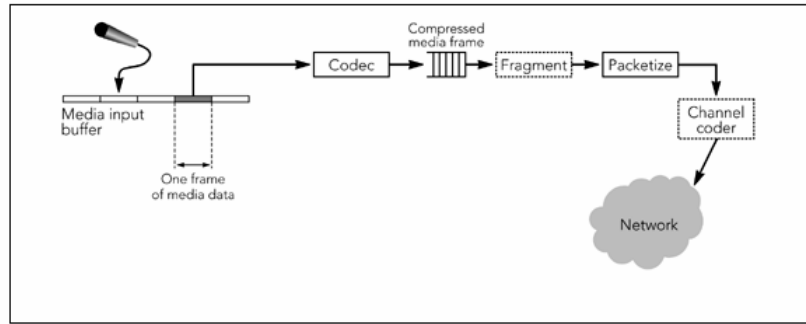
Gambar 2.3 menunjukkan negosiasi dan protokol *call control*, media *transport layer* (disediakan oleh RTP), algoritma kompresi dan dekompresi (codec) dan dasar jaringan keduanya sangat berhubungan, baik yang menurut *framework* dari ITU ataupun IETF. Kedua set dari *call control* yang paralel dan standar negosiasi media menggunakan *framework* dari media *transport* yang sama. Media *codec* umumnya tidak mempermasalahkan bagaimana *session* bernegosiasi, dan terlepas dari dasar *transport* jaringan.



Gambar 2.4 Standar IETF dan ITU untuk *protocol transport Audio/Video* pada jaringan
Sumber: Collins Perkins, 2003

2.4.1.1. Aktifitas RTP pada Pengirim

Pengirim bertanggung jawab untuk menangkap dan mengubah *audiovisual* menjadi data untuk dikirimkan, sebaik dalam membuat paket RTP. RTP membolehkan juga berpartisipasi pada koreksi kesalahan dan pengendalian kongesti dengan mengadaptasi media *stream* yang dikirim untuk merespon *receiver feedback*.



Gambar 2.5 Diagram RTP sender
Sumber: Collins Perkins 2003

Media data yang tak terkompres *audio/video* ditangkap ke dalam *buffer*, dari *frame* dikompres yang diproduksi. *Frame* mungkin diekodekan pada beberapa jalan tergantung dari algoritma pemrosesan yang digunakan, dan *frame* yang terencode boleh di gantungkan pada awal atau akhir data.

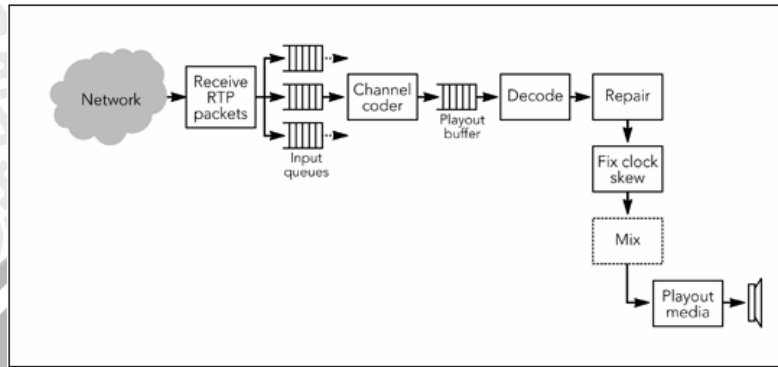
Frame yang terkompres dimasukkan dalam paket RTP, siap untuk dikirimkan. Jika *frame* besar, mungkin akan difragmentasi ke dalam beberapa paket. Jika terlalu kecil beberapa *frame* akan di satukan menjadi paket RTP tunggal. Tergantung pada prediksi koreksi kesalahan, sebuah kanal *coder* mungkin digunakan untuk menghasilkan paket *error correction* atau meminta paket sebelum dikirim. Setelah paket RTP dikirim, *buffer media* mencocokkan paket tersebut dan akhirnya dibebaskan. Pengirim tidak boleh menghapus data yang mungkin diperlukan untuk koreksi kesalahan atau untuk proses *encoding*. Keperluan ini mungkin berarti bahwa pengirim harus membuffer data untuk beberapa waktu setelah itu mencocokkan paket yang telah dikirim, tergantung pada *codec* dan prediksi kesalahan yang digunakan.

Pengirim bertanggung jawab untuk menghasilkan status *rept* secara periodik untuk media *stream* yang dihasilkan, termasuk yang dibutuhkan untuk sinkronisasi. Dan juga menerima *reception quality feedback*. Dari partisipan yang lain dan mungkin menggunakan informasi untuk beradaptasi pada pengiriman.

2.4.1.2. Aktifitas RTP pada Penerima

Sebuah penerima bertanggung jawab untuk mengumpulkan paket RTP dari jaringan, memeriksa semua kesalahan, memperbaiki waktu, mendekompres

media dan menampilkannya pada user. Dan juga mengirimkan reception quality feedback, mengijinkan pengirim beradaptasi saat pengiriman ke penerima, dan menjaga database partisipan pada session. Ditunjukkan pada gambar 2.6.



Gambar 2.6 Diagram RTP receiver

Sumber: Collins Perkins 2003

Langkah pertama dari proses penerimaan adalah mengumpulkan paket dari jaringan, memvalidasi untuk dikoreksi, dan memasukkannya kedalam input yang spesifik. Paket dikumpulkan dari *input queue* dan dilewatkan ke *channel-coding* untuk dikoreksi kesalahannya. Mengikuti *channel coder*, paket dimasukkan ke dalam *playout buffer*. *Playout coder* diatur berdasarkan *timestamp*, dan proses pemasukan paket kedalam *buffer* membenarkan setiap permintaan yang terhalangi selama *transport*. Paket masih berada di layout *buffer* sampai *frame* telah selesai diterima, dan menambahkan *buffer* untuk menghilangkan setiap variasi pada *timing* interpaket yang disebabkan oleh jaringan. Perhitungan sejumlah *delay* untuk menambah salah satu aspek yang kritis pada desain dari implementasi RTP. Masing-masing paket dilabeli dengan waktu *playout* yang diinginkan untuk *frame* yang cocok.

Setelah waktu *playout* tercapai, paket dikumpulkan untuk dibentuk *frame* yang sempurna, dan setiap kerusakan atau kehilangan paket diperbaiki. Mengikuti setiap perbaikan yang perlu, *frame* dikodekan (tergantung dari *codec* yang digunakan, dan mungkin media perlu dikodekan sebelum paket yang hilang dapat diperbaiki). Pada poin ini mungkin ada perbedaan yang dapat di observasi pada *clock-rate* nominalnya pada pengirim maupun penerima. Beberapa perbedaan sebagai perwujudan sendiri sebagai penyimpangan pada harga dari *clock media*

RTP relatif terhadap *playout clock*-nya. Penerima harus mengkompensasi untuk *clock* ini untuk menghindari gap pada *playout*

Akhirnya, media data dimainkan user. Tergantung pada format dari media tersebut dan *output device*, hal itu memungkinkan untuk memainkan setiap *stream* sendiri-sendiri sebagai contoh, menampilkan beberapa *video stream*, pada masing-masing *window*. Alternatifnya, mungkin perlu untuk mencampur media dari beberapa sumber jadi satu *stream* tunggal untuk memainkan via sebuah satu set speaker.

Seperti yang dijelaskan pada pendahuluan, operasional dari penerima RTP kompleks, dan sedikit mempengaruhi operasi dari pengirim. Kompleksitas ini bertambah besar karena kemajemukan dari jaringan IP.

2.4.2. User Datagram protocol (UDP/IP)

UDP merupakan protokol yang berada dilapisan selain TCP. Protokol ini bersifat *connectionless* dan *unreliable*. Dalam pengiriman data, *connectionless* berarti tidak diperlukan pembentukan suatu hubungan terlebih dahulu untuk pengiriman data. Sedangkan *unreliable*, berarti protokol ini, tidak menjamin data akan sampai pada tujuan dalam kondisi benar. Aplikasi yang memanfaatkan UDP sebagai protokol *transport*, dapat mengirimkan data tanpa melalui proses pembentukan koneksi terlebih dahulu. Lebih sering diimplementasikan untuk aplikasi seperti *multimedia streaming* dimana rugi-rugi paket data kecil masih bisa ditoleransi dari pada nilai *delay* yang lebih besar.

+	Bits 0 - 7	8 - 15	16 - 23	24 - 31
0	Source address			
32				
64				
96				
128	Destination address			
160				
192				
256				
288	UDP length			
320	Zeros			Next Header
352	Source Port		Destination Port	
384	Length		Checksum	
416	Data			

Gambar 2.7 Format paket UDP
Sumber: RFC 768

2.5. *Streaming Protocol*

Protokol *streaming* terdiri dari :

a. IP (*Internet Protocol*)

Protokol terbawah yang digunakan sinyal informasi pada jaringan internet.

b. TCP (*Transport Control Protocol*)

Protokol ini berada di atas lapisan internet yang berfungsi untuk mengatasi koneksi dan bersifat *reliable*.

c. RTP (*Real-time Transport Protocol*)

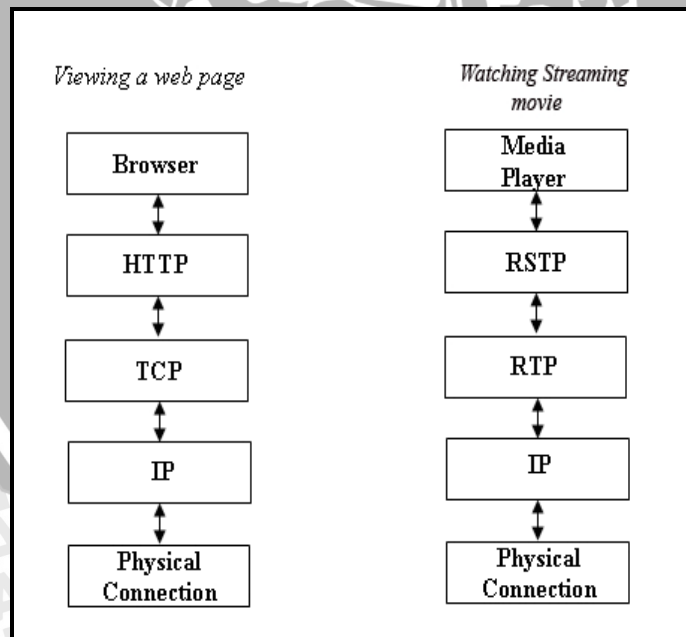
Lapisan (*layer*) yang berada di atas IP (*Internet Protocol*) dan mendukung pengiriman data secara *real-time* (waktu nyata).

d. HTTP (*HyperText Transfer Protocol*)

Protokol ini digunakan untuk transmisi melalui *web page*. Protokol ini berada di atas *layer* TCP.

e. RSTP (*Real-time Transport Streaming Protocol*)

Protokol ini berada diatas lapisan RTP yang digunakan untuk media *Streaming*. Berikut hubungan tiap lapisan yang digunakan pada sistem *Streaming*, baik untuk pengaksesan dari sebuah *web page* maupun dengan menggunakan media *player*.

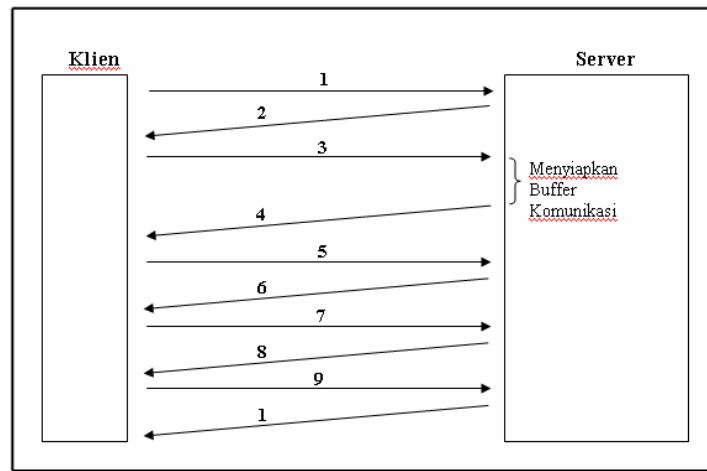


Gambar 2.8 Komunikasi Streaming

Sumber : Kadir A, 2002:11

f. RTMP (*Real-Time Messaging Protocol*)

RTMP (*Real-Time Messaging Protocol*) yaitu protokol ini digunakan oleh *flash movie* untuk menyalurkan objek *real-time*, video dan audio ke klien menggunakan TCP *connection* atau *polling http tunnel*. Protokol ini dapat berisikan paket data mentah dari audio maupun video seperti misalnya *flv*.



Gambar 2.9 Proses Handshaking dalam protokol RTMP

Sumber : <http://www.osflash.org/rtmp>

Keterangan gambar 2.9, yaitu proses *handshaking* dalam protokol RTMP :

1. Klien ke Server = mengirimkan permintaan *handshake* dalam *single byte*.
2. Server ke Klien = mengirimkan tanggapan / response *handshaking* (*acknowledge*).
3. Klien ke Server = mengirimkan id dalam bentuk *single byte*.
4. Server ke Klien = menanggapi dengan menyiapkan *buffer* data komunikasi, kemudian mengirimkan ACK (*acknowledge*).
5. Klien ke Server = memulai mengirimkan paket data koneksi RTMP (misalkan kirim data suara).
6. Server ke klien = menerima paket data dan mengirmkan tanggapan ke klien (*acknowledge*) bahwa data telah diterima.
7. Klien ke Server = mengirimkan data ke-2, akan tetapi terdapat gangguan sehingga data yang diterima server tidak lengkap.
8. Server ke klien = memberi tanggapan bahwa data yang dikirimkan tidak lengkap (*not acknowledge*).

9. Klien ke *server* = setelah menerima NACK (*not acknowledge*), klien mengirimkan kembali paket data ke-2, proses seperti ini akan dilakukan sampai data yang diterima *server* lengkap sepenuhnya sesuai dengan data yang dikirimkan klien.
10. *Server* ke klien = setelah menerima data sesuai yang dikirimkan *server*, maka *server* akan merespon dengan mengirimkan ACK(*acknowledge*). Dan seterusnya dimulai dari klien mengirimkan paket data.

2.6. **Multimedia Streaming**

Multimedia adalah kombinasi dari dua atau lebih jenis informasi. Jenis informasi itu antara lain suara, teks, grafik, *image*, *audio*, dan *video*. Suara, *audio*, dan *video* merupakan jenis informasi yang berupa sinyal analog. Teks, *image*, grafik merupakan data digital.

Konsep dasar pada *multimedia streaming* adalah membagi data *multimedia* yang di-download ke dalam beberapa paket, mentransmisikan paket tersebut, kemudian *client* dapat men-decode dan memainkan paket tersebut tanpa harus menunggu seluruh data *multimedia* terkirim ke komputer *client*. Permasalahan yang ada pada *streaming* adalah keterbatasan kecepatan transmisi, *delay jitter* dan *packet loss*.

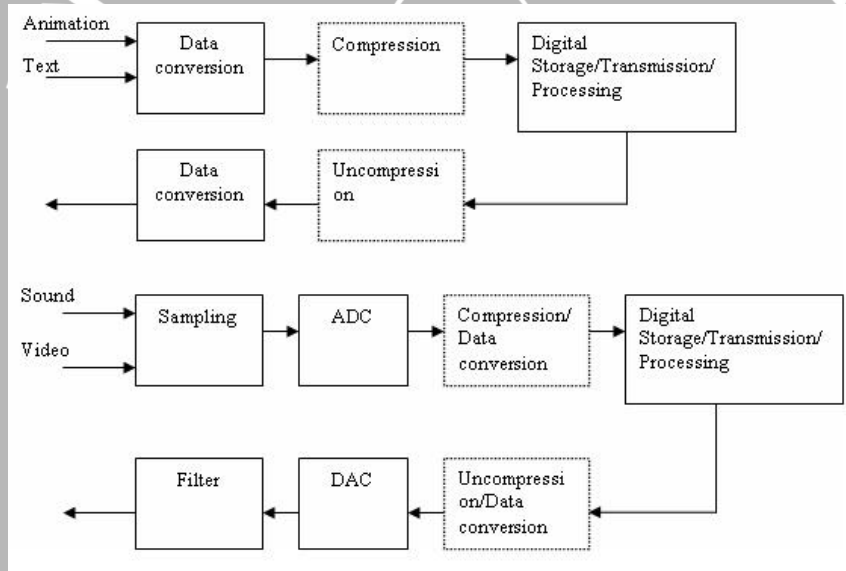
Kecepatan transmisi antara dua titik di internet secara pasti tidak dapat diketahui. Jika sebuah *server* mengirimkan aplikasi dengan *bandwidth* yang lebih besar daripada kecepatan transmisi pada internet maka akan terjadi kongesti, paket hilang, kualitas data *multimedia* buruk. Sebaliknya, jika *server* mengirimkan aplikasi dengan *bandwidth* yang lebih kecil daripada kecepatan transmisi pada internet maka kualitas data *multimedia* tidak optimal. Salah satu ide untuk mengatasi hal itu adalah dengan mengestimasi *bandwidth* aplikasi kemudian mencocokkannya dengan kecepatan transmisi di internet.

Paket-paket yang ditransmisikan memiliki *delay end to end* yang fluktuatif. Variasi dari *delay end to end* ini menyebabkan *delay jitter*. *Delay* ini menjadi masalah karena *client* harus men-decode dan menampilkan paket-paket pada kecepatan yang konstan dan keterlambatan paket akan menyulitkan dalam rekonstruksi data *multimedia* yang diterima.

Packet loss adalah jumlah paket yang hilang selama pen-download-an data *multimedia* dari *server*. *Packet loss* ini menimbulkan penurunan kualitas data *multimedia* hasil rekonstruksi.

2.7. Kompresi multimedia

Aplikasi *multimedia streaming* membutuhkan *bandwidth* yang besar. Agar aplikasi *multimedia* dapat ditransmisikan pada sistem transmisi yang mempunyai kecepatan transmisi yang terbatas maka diperlukan kompresi pada data *multimedia* tersebut. Kompresi data hanya dapat dilakukan pada data digital. Setelah sinyal *audio* maupun *video* mengalami konversi dari data analog menjadi digital, data *audio* dan *video* dapat dikompresi menjadi data digital yang lebih kecil.



Gambar 2.10 Proses Konversi dan Kompresi Data
Sumber : W. Buchanan, 1997 : 11

2.7.1. Kompresi Audio

Metode yang umum dipakai untuk konversi sinyal suara dan *audio* analog menjadi data digital adalah Pulse Code Modulation (PCM). PCM terdiri atas tiga proses yaitu :

1. *Sampling* (pencuplikan)

Sinyal analog di-*sampling* dengan frekuensi *sampling* tertentu yang mengacu pada teorema Nyquist seperti yang terlihat dalam persamaan (2-1)

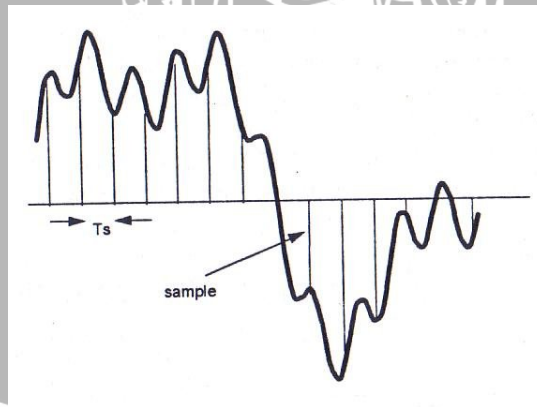
$$f_s = 2f(t)_{maks} \quad (2-1)$$

Dengan :

f_s = frekuensi *sampling* (Hz)

$f(t)_{maks}$ = frekuensi maksimum sinyal analog (Hz)

Untuk suara, frekuensi maksimumnya 4 kHz sehingga harus di-*sampling* dengan frekuensi *sampling* 8 kHz. Untuk *audio* dengan kualitas Hi-Fi, frekuensi maksimumnya 20 kHz sehingga harus di-*sampling* dengan frekuensi *sampling* 40 kHz (untuk professional Hi-Fi menggunakan 44,1 kHz). Sedangkan untuk *video*, frekuensi maksimumnya 6 MHz sehingga harus di-*sampling* dengan frekuensi *sampling* 12 MHz. Hasil dari proses *sampling* ini berupa *Pulse Amplitude Modulation* (PAM) dimana level amplitudonya masih berupa analog, seperti yang terlihat dalam Gambar 2.16.



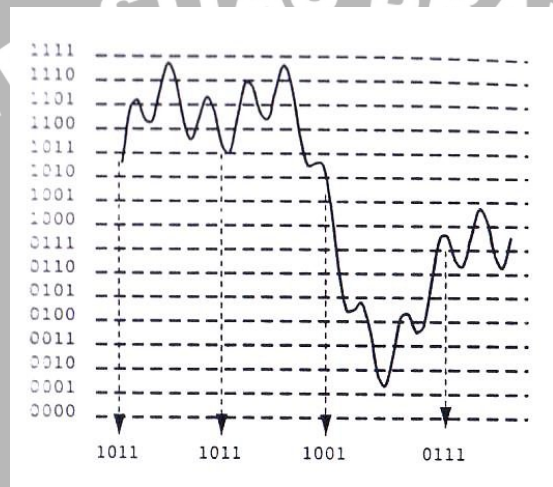
Gambar 2.11 Proses *Sampling*
Sumber : W. Buchanan, 1997 : 12

2. *Quantizing* (kuantisasi)

Quantizing merupakan proses mengubah level amplitudo analog hasil *sampling* menjadi level amplitudo diskrit dengan level kuantisasi tertentu.

3. *Coding* (pengkodean)

Coding merupakan proses mengubah level amplitudo diskrit hasil kuantisasi menjadi data digital yang mengacu pada level kuantisasi. Jika level kuantisasinya $16 = 2^4$ maka setiap *sample* akan dikodekan dengan 4 bit, seperti yang terlihat dalam Gambar 2.17.



Gambar 2.12 Proses *Quantizing* dan *Coding*
Sumber : W. Buchanan, 1997 : 13

Data sering kali mengalami redundansi dalam penyimpanan datanya. Kompresi ditujukan untuk mengurangi kuantitas data dengan tetap mempertahankan kualitas hasil rekonstruksinya. Proses pengurangan kuantitas data tersebut dilakukan dengan membuang informasi-informasi yang tidak berguna sehingga yang dikirimkan atau disimpan hanya informasi yang penting saja.

Setelah mengalami konversi dari sinyal analog menjadi data digital, proses selanjutnya adalah kompresi. Berikut ini beberapa contoh implementasi kompresi terhadap data *audio* digital.

1. Suara
Lebar spektrum sinyal suara untuk telepon dibatasi pada 4 kHz. Sehingga frekuensi *sampling*-nya 8 kHz. Jika setiap *sample* dikodekan dengan 8 bit maka kecepatan *encoding*-nya adalah :

$$\text{Kecepatan encoding} = 8 \times 8 \text{ kbps} = 64 \text{ kbps}$$

2. Audio
Lebar spektrum sinyal *audio* untuk high-quality monochannel dibatasi pada 20 kHz sehingga frekuensi *sampling*-nya 40 kHz (tetapi biasanya 44,1 kHz). Jika setiap *sample* dikodekan dengan 16 bit maka kecepatan *encoding*-nya adalah

$$\text{Kecepatan encoding} = 44,1 \times 16 \text{ kbps} = 705,6 \text{ kbps.}$$

Sehingga untuk kualitas stereo diperoleh 1,4112 Mbps. Data digital tersebut masih dapat mengalami kompresi sehingga dapat terjadi penurunan *bitrate*. Standar kompresi yang digunakan *audio* ditunjukkan dalam Tabel 2.3.

Tabel 2.2 Standar Kompresi Audio

Kompresi Audio	Kecepatan Encoding	Rasio Kompresi
Uncompressed audio (mono)	705.6 kbps	1:1
Uncompressed audio (stereo)	1.4112 Mbps	1:1
AC-1	512 kbps	3:1
AC-2	256 kbps	6:1
AC-3	512 kbps	6:1
MPEG (mono)	64 kbps	11:1
MUSICAM (stereo)	192 kbps	7:1
AAC (mono)	24 kbps	30:1
AAC-LD (mono)	32 kbps	22:1

Sumber : W. Buchanan, 1997 : 141 dan ISO/IEC JTC1/SC29/WG11

2.7.2. Kompresi Video

Secara garis besar tipe sinyal *video* dibagi menjadi tiga yaitu, *National Television Standards Committe* (NTSC) untuk USA, *Phase Alternation Line* (PAL) untuk UK dan beberapa daerah Asia serta *Sequential Couleur a Memoire* (SECAM) untuk Perancis. Ketiga sinyal di atas merupakan sinyal *video* gabungan dari kombinasi *Brightness* (*Luminance*), Warna (*Hue*), dan Saturasi warna (*Saturation*).

Sinyal *video* menggunakan tiga warna dasar yaitu merah, hijau dan biru. Sinyal *video* seperti halnya sinyal *audio*, merupakan sinyal analog. Supaya dapat dikompres maka sinyal tersebut harus diubah menjadi data digital.

Standar kompresi yang digunakan *video* ditunjukkan dalam Tabel 2.6.

Tabel 2.3 Standar Kompresi Video

Kompresi Video	Kecepatan <i>Encoding</i>	Rasio Kompresi
Uncompressed TV	162 Mbps	1 : 1
MPEG-1	4 Mbps	40 : 1
MPEG-2	1.2 Mbps	130 : 1
MPEG-4	64 kbps	2530 : 1

Sumber : W. Buchanan, 1997 : 94 dan ISO/IEC JTC1/SC29/WG11

Karakteristik suatu *video* digital ditentukan oleh resolusi gambar (*frame*), kedalaman *pixel*, laju *frame*. Karakteristik-karakteristik ini akan menentukan kualitas *video* dan jumlah bit yang dibutuhkan untuk menyimpan atau mentransmisikannya.

1. Resolusi Gambar (resolution)

Resolusi gambar/*frame* adalah ukuran sebuah gambar/*frame*. Resolusi dinyatakan dengan perkalian antara panjang dan lebar gambar/*frame*. Semakin tinggi resolusi, semakin baik kualitas *video* tersebut. Namun, resolusi tinggi akan mengakibatkan jumlah bit yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

2. Kedalaman Pixel

Kedalaman *pixel* menentukan jumlah bit yang digunakan untuk merepresentasikan tiap *pixel* pada sebuah *frame*. Kedalaman bit dinyatakan dalam bit/*pixel*. Semakin banyak jumlah bit yang digunakan

untuk merepresentasikan sebuah pixel, semakin tinggi kedalaman pixel-nya yang berarti semakin tinggi pula kualitas *videonya*. Namun, hal ini akan mengakibatkan jumlah bit yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

3. Laju *Frame* (*frame rate*)

Laju *frame* menunjukkan jumlah *frame* yang ditampilkan secara bergantian danurut setiap detik-nya. Semakin tinggi laju *frame*, semakin tinggi pula kualitas *videonya*. Namun, hal ini akan mengakibatkan jumlah bit yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

2.7.3. Kompresi MPEG

Sejak tahun 1988, MPEG *project* telah dikembangkan dibawah ISO dan IEC (*International Electrotechnical Committee*). MPEG merupakan singkatan dari *Moving Pictures Experts Group*. MPEG merupakan sebuah metode kompresi dan sinkronisasi data digital, baik gambar maupun suara. Terdapat beberapa standar MPEG yang telah dikeluarkan. Standar MPEG-1 ditujukan untuk data dengan trafik menengah dengan *data rate* berkisar 1,5 Mbit/s. MPEG-2 ditujukan untuk data dengan rate tinggi berkisar 10 Mbit/s. Sedangkan MPEG-4 ditujukan untuk *data rate* yang sangat rendah hingga 64 Kbit/s.

2.7.3.1. MPEG Audio

Salah satu standar kompresi yang dikeluarkan oleh MPEG adalah MPEG-1 *Audio*. Terdapat 3 jenis *encoding* yang tercakup dalam MPEG-1 *Audio* yaitu MPEG *Layer 1*, *Layer 2* dan *Layer 3*. Standar MPEG-1 *Audio Encoding Layer 3* (MP3) merupakan metode kompresi yang paling banyak digunakan saat ini dengan keunggulan pada kualitas *audio* yang bagus namun dengan *bitrate* yang rendah. Keunggulan standar kompresi MPEG yang lain adalah skalabilitas pada *bitrate* dan kualitas hasil kompresi yang diinginkan.

MPEG mengeluarkan beberapa standar dalam *audio encoding* untuk penggunaan yang luas dalam aplikasi. Tabel 2.7 menunjukkan spesifikasi standar kompresi MPEG *Audio*.

Tabel 2.4 Spesifikasi kompresi MPEG Audio

	<i>Bit rate range (kbit/s)</i>	<i>Target bit rate (kbit/s)</i>	<i>Typical compression ratio</i>
Layer 1	32 to 448	192 (mono)	1:4
Layer 2	32 to 384	128 (mono)	1:6 to 1:8
Layer 3 (MP3)	32 to 320	64 (mono)	1:10 to 1:12
		128 (stereo)	
AAC	32 to 128	96 (stereo)	1:16
HE AAC	32 to 128	64 (stereo)	1:24
HQ Parametric		24 (stereo)	1:64

Sumber : Austerberry,2005

2.7.3.2. MPEG Video

Teknologi kompresi *video* yang dikeluarkan oleh MPEG untuk *video* antara lain :

1. MPEG-1 : Diselesaikan pada tahun 1991. Merupakan standar untuk menyimpan dan memutar *video* dan *audio* pada media penyimpanan. Standar ini digunakan untuk Video CD dan format MP3 yang populer. *Bitrate* optimum berada pada kisaran 1,5Mbit/s namun dapat digunakan dengan kisaran lebih dari itu.
2. MPEG-2 : Diselesaikan pada tahun 1994. Standar MPEG-2 ditujukan untuk digital television *broadcasting*. Banyak digunakan untuk siaran televisi digital seperti Digital Video Broadcasting (DVB) atau televisi satelit dan tv kabel.
3. MPEG 4 : Diperkenalkan pada tahun 1998. MPEG-4 merupakan standar kompresi untuk media pada web, *streaming* media, CD Distribution, percakapan, *videophone*, dan *broadcast* television. Semua aplikasi tersebut memiliki kebutuhan *bandwidth* yang beragam yang dapat diakomodasi oleh MPEG-4. MPEG-4 banyak mengadopsi fitur pada standar sebelumnya dan menambahkan fitur fitur seperti 3D *rendering* dan *Digital Rights Management* (DRM). Beberapa standar kompresi dengan efisiensi lebih tinggi dari MPEG-2 juga disertakan seperti *Advanced Simple Profile*, *Advanced Video Coding* dan *Advanced Audio*

Coding untuk *audio*. Kelebihan MPEG-4 adalah skalabilitas dan kualitas kompresi yang lebih baik daripada standar sebelumnya. Selain itu, standar MPEG-4 juga dapat diterapkan pada berbagai jenis media telekomunikasi. Kekurangan MPEG-4 adalah pada proses kompresi yang memakan waktu lebih lama karena algoritma kompresi yang kompleks. Pada MPEG-4 dengan *Simple Profile* proses *encoding* dan *decoding* berkisar 100ms.

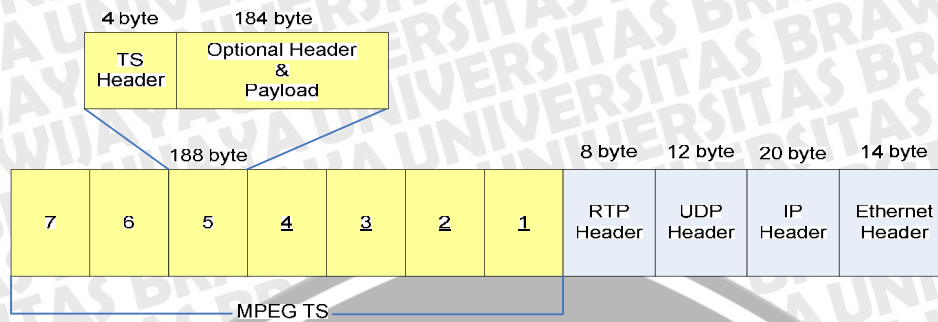
Tabel 2.5 Standar Codec untuk *video*

Compression format	ISO/IEC number Issue date	Target bandwidth bit/s	Typical resolution pixels	Application
H.261	1988–1990	384 k–2 M	176 × 144 or 352 × 288	Video conferencing, low delay
H.263	1992	28.8 k–768 k	128 × 96 to 720 × 480	Video conferencing
MPEG-1	11172 1993	400 k–1.5 M	352 × 288	CD-ROM
MPEG-2, MP@ML	13818 1994	1.5 M–15 M	720 × 480	Broadcast television, DVD
MPEG-4	14496 1998	28.8 k–500 k	176 × 144 or 352 × 288	Fixed and mobile web
AVC, H.264	14496–10 2002			General purpose

Sumber : Austerberry, 2005

2.7.3.3. MPEG Transport Stream

Menurut standar MPEG-2, dalam proses *delivery* paket data *audio* dan *video* yang di*encoding* dengan MPEG codec dienkapsulasi menjadi sebuah paket *multimedia* yang terdiri dari data *audio* dan *video* yang sudah termultipleks. Paket data ini disebut MPEG Transport Stream (MPEG TS). Sebuah TS terdiri dari sebuah paket data dengan panjang tetap yaitu 188 byte. Setiap paket TS terdiri dari 4 byte TS Header dan 184 bytes sisanya terdiri dari *optional header* dan *payload* data *multimedia*.



Gambar 2.13 Paket data multimedia streaming dengan enkapsulasi MPEG TS
Sumber : Vigato, 2005

2.8. Aplikasi Web

Aplikasi web atau dalam bahas inggris disebut *web application* adalah sebuah aplikasi yang dapat diakses melalui *web browser* melalui sebuah jaringan komputer atau internet. Aplikasi web merupakan sebuah perangkat lunak yang dikembangkan menggunakan bahasa pemrograman yang dapat dieksekusi dan dikenali oleh *browser* seperti HTML (HyperText Markup Language), PHP, Javascript, Java, dan lainnya. Keunggulan aplikasi web adalah kemudahan bagi *client* dalam menggunakan aplikasi karena tidak perlu melakukan proses instalasi pada komputernya.

2.8.1. HTML (HyperText Markup Language)

HyperText Markup Language (HTML) adalah bahasa pemrograman yang digunakan untuk membuat *web page*. Sebuah *web page* tersusun atas *head* dan *body*. *Head* merupakan bagian pertama dari *web page*. *Head* berisi judul dan parameter lain yang digunakan oleh *web browser*. Sedangkan *body* merupakan isi dari *web page* yang terdiri dari teks dan *tag*. *Tag* mendefinisikan tampilan dari dokumen.

2.8.2. PHP (Perypheral Hypertext Preprocessor)

PHP (*Perypheral Hypertext Preprocessor*) pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. PHP adalah bahasa pemrograman yang

memungkinkan para *web developer* untuk membuat aplikasi web yang dinamis dengan cepat.

PHP merupakan bahasa pemrograman yang tergolong ke dalam jenis Interpreter. Untuk menjalankan *script*nya, program ini harus diletakkan pada *directory root* pada *web server*.

Script PHP sangat mudah digunakan, salah satunya adalah bahwa *script* PHP dapat berjalan secara *embedded* di dalam *tag-tag* XML. Untuk membuat program WAP berbasis PHP, maka pada setiap halaman harus dikenalkan bahwa program tersebut adalah *script* WAP.

Kelebihan dari PHP dibandingkan dengan bahasa pemrograman yang lain adalah sebagai berikut :

- Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
- Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.
- Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.

PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.9. Performansi Sistem

Performansi sistem digunakan untuk mengetahui baik buruknya sebuah system yang telah dibuat. Sistem yang dimaksud adalah sistem *multimedia streaming* pada jaringan *multicast* . Dari sistem yang telah ada akan diuji beberapa parameter parameter sehingga sistem tersebut dapat dikatakan baik atau buruk. Berikut ini akan dijelaskan parameter parameter yang digunakan untuk sebagai tolok ukur dalam performansi sistem.

2.9.1. Bandwidth Multimedia Streaming

Pengukuran *bandwidth multimedia streaming* ini ditujukan untuk mengetahui *bandwidth* yang dibutuhkan pada proses *streaming*.

Untuk mengukur *bandwidth multimedia streaming* digunakan persamaan (2-2) :

$$BW_{multimedia} = N_{paket\ multimedia} \times (M_{paket\ multimedia} + Header_{(RTP+UDP+IP+Ethernet)}) \times 8 \quad (2-2)$$

Dengan :

$BW_{multimedia}$	= <i>bandwidth multimedia</i> (bps)
$N_{paket\ multimedia}$	= jumlah paket <i>multimedia</i> setiap detik
$M_{paket\ multimedia}$	= panjang paket <i>multimedia</i> (byte)
$Header_{(RTP+UDP+IP+Ethernet)}$	= <i>Header</i> paket <i>multimedia</i> (byte)
<i>Header RTP</i>	= 8 byte
<i>Header UDP</i>	= 12 byte
<i>Header IP</i>	= 20 byte
<i>Header Ethernet II</i>	= 14 byte

2.9.2. Throughput

Throughput didefinisikan sebagai jumlah paket yang diterima di sisi penerima dengan benar setiap detik. Karena sistem yang dirancang menggunakan protokol Go Back N maka *throughput* ditentukan dengan Persamaan (2-3) (Mischa Schwartz, 1987 : 129)

$$\lambda = \frac{1}{t_v} = \frac{(1-p)}{t_i[1+(a-1)p]} \quad (2-3)$$

Dengan :

λ	= <i>throughput</i> (paket/s)
t_v	= waktu rata-rata untuk mentransmisikan 1 paket yang benar (s)

- t_i = waktu yang dibutuhkan untuk mentransmisikan 1 paket (s)
 α = konstanta transmisi ternormalisasi
 p = probabilitas kesalahan paket

2.9.3. Kecepatan Transmisi

Kecepatan transmisi rata-rata yang diterima merupakan jumlah bit yang dapat diterima di *client* setiap detik. Kecepatan transmisi rata-rata ditentukan dengan Persamaan (2-4) (Mischa Schwartz, 1987 : 132)

$$D = \lambda l \times 8 = \frac{(1 - p) \times 8}{t_i [1 + (\alpha - 1)p]} \quad (2-4)$$

Dengan :

- D = kecepatan transmisi rata-rata yang diterima (bps)
 λ = *throughput* (paket/s)
 l = panjang paket data (byte/paket)
 t_i = waktu yang dibutuhkan untuk mentransmisikan 1 paket (s)
 p = probabilitas kesalahan paket
 α = konstanta transmisi ternormalisasi

2.9.4. Packet Loss

Packet loss adalah banyaknya paket *multimedia* yang hilang saat di-download oleh *client*. Pada *streaming*, ketika *bandwidth multimedia* lebih besar daripada kecepatan transmisi rata-rata yang diterima di internet maka akan terjadi kongesti sehingga mengakibatkan beberapa paket hilang. Prosentase packet loss maksimum yang diperbolehkan oleh ISO untuk aplikasi *multimedia* adalah 5 %. Prosentase packet loss ditentukan dengan Persamaan (2-5)

$$N_{\text{packet loss}} (\%) = \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100 \% \quad (2-5)$$

Dengan :

$N_{packet\ loss}$ (%) = prosentase packet loss

$N_{packet\ loss}$ = jumlah packet loss

N_{paket} = jumlah paket *multimedia* yang diterima

2.10. Webcam

Webcam (*web camera*) adalah sebutan bagi kamera *real-time* yang gambarnya bisa diakses atau dilihat melalui *World Wide Web*, program *instant messaging*, atau aplikasi *video call*. Dengan webcam tidak hanya memungkinkan kita untuk mengabadikan sebuah *image*, namun juga memungkinkan kita *men-transfer image* via komputer dengan menggunakan sistem IP (*Internet Protocol*). *Webcam* merupakan komponen penting dalam perancangan sistem monitoring tempat parkir ini, karena *webcam* dapat menangkap gambar lingkungan sekitar secara *real-time*.

Webcam sendiri terdiri dari sebuah lensa, *sensor image* dan *support electronic*. Lensa digunakan untuk mengatur *focus* dari kamera. *Sensor image* berfungsi untuk mengolah gambar yang telah di-*capture* baik resolusi atau jumlah *frame per second*-nya. Sedangkan *support electronic* digunakan untuk membaca gambar dari *sensor* dan mentransmisikannya kepada komputer melalui *USB* (*Universal Serial Bus*).



Gambar 2.14 PC Web Camera Sturdy

Sumber : <http://www.ecrater.com>

Salah satu jenis *webcam* yang digunakan adalah *USB webcams*. Webcam ini merupakan jenis yang paling umum digunakan, biasanya difungsikan untuk merekam lingkungan sekitar. Salah satu perusahaan yang memproduksi jenis

webcam tersebut adalah *Sturdy*. *Webcam* yang diproduksi oleh *Sturdy* ini mempunyai spesifikasi sebagai berikut.

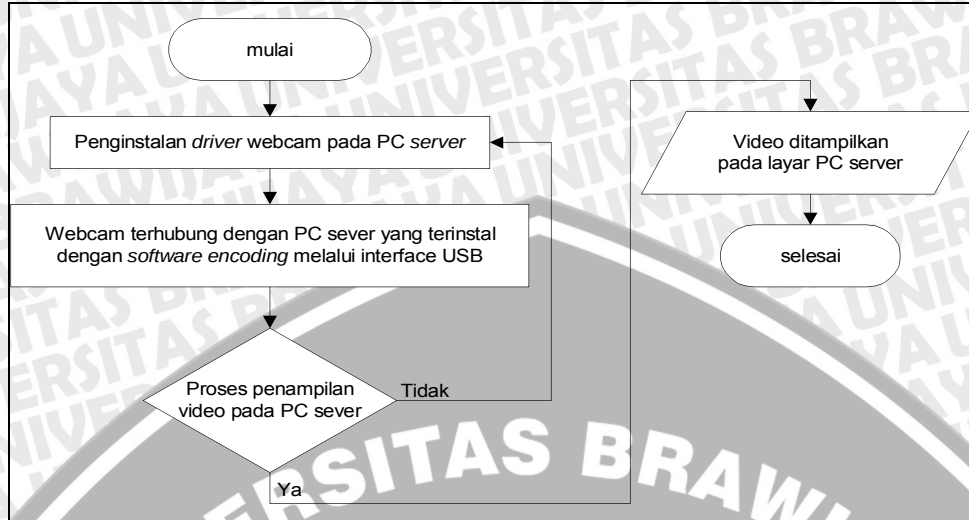
Tabel 2.6 Spesifikasi *Webcam Sturdy*

Parameter	Default
<i>Data format</i>	RGB 24
<i>Port</i>	USB 2.0
<i>Frame Rate</i>	10-15 fps VGA
Resolusi	640 x 480
<i>Bitrate</i> (USB)	480 Mbps

Sumber : www.mysturdy.com

Hasil tangkapan gambar dari *webcam* adalah *file* dengan format AVI (*Audio Video Interleaved*) memiliki *header* dengan panjang keseluruhan 56 *Byte* dimana *header* ini memiliki kode-kode tertentu.

Hasil tangkapan gambar dari *webcam* pada sistem *monitoring* ini akan terkirim ke PC dengan *interface* USB hingga gambar ditampilkan pada *software encoding*. Proses pengiriman gambar dari *webcam* hingga terlihat pada layar PC adalah sebagai berikut.



Gambar 2.15 Flowchart tahap-tahap penampilan video pada PC server
Sumber : RealNetworks

2.11. USB (Universal Serial Bus)

USB (*Universal Serial Bus*) pada dasarnya merupakan standar antarmuka eksternal untuk komunikasi komputer dengan berbagai perangkat lain. Namun kini USB dapat menghubungkan peralatan tambahan komputer seperti *mouse*, *keyboard*, pemindah gambar, kamera digital, *printer*, *hard disk*, dan komponen *networking*. USB kini telah menjadi standar bagi peralatan multimedia seperti pemindah gambar dan kamera digital.

Saat ini ada dua versi USB yang digunakan, yakni USB 1.1 dan USB 2.0. Karena USB selalu memiliki kompatibilitas ke belakang dengan versi-versi sebelumnya, sehingga USB 2.0 juga kompatibel dengan USB 1.1. Perangkat-perangkat USB baru biasanya kompatibel dengan USB 2.0. USB 2.0 memiliki kecepatan transmisi data maksimal sebesar 480 Mbps atau 60 MBps dan inilah yang menjadi perbedaan paling mendasar antara kedua standar ini. Kelebihan lain dari USB adalah, perangkat-perangkatnya *hot-pluggable*, yang berarti untuk menggunakan USB tidak perlu me-restart sistem.

Kecepatan *transfer* data USB dibagi menjadi tiga, antara lain:

- *Low speed data*, yaitu USB yang memiliki kecepatan transmisi data sebesar 1.50 Mbps. Contoh jenis *hardware* yang menggunakan tipe USB ini adalah *Keyboard* dan *Mouse*.

- *Full speed* data, yaitu USB yang memiliki kecepatan transmisi data sebesar 12 Mbps. Contoh jenis *hardware* yang menggunakan tipe USB ini adalah *broadband*, *audio* dan *microphone*.
- *High speed* data, yaitu USB yang memiliki kecepatan transmisi data sebesar 480 Mbps. Contoh jenis *hardware* yang menggunakan tipe USB ini adalah *webcam*, *harddisk*, *imaging*, dan *broadband*



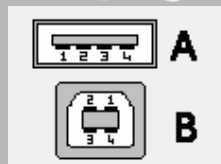
Gambar 2.16 Konektor USB
Sumber : www.wikipedia.com

Karena kabel USB dalam skripsi ini digunakan transfer data pada *webcam*, maka kecepatan data pada *transfer* data pada *webcam* adalah *transfer* data USB tipe *High speed* yang kecepatan datanya mencapai 480 Mbps.

Berdasarkan *datasheet* USB 2.0 memiliki besar tegangan 2 volt, dan menyediakan arus hingga 500 mA dengan parameter tersebut maka USB ini cukup untuk menyuplai *device* yang menggunakan antarmuka USB.

2.11.1 Konfigurasi pin USB

Konektor USB ada 2 macam, yakni konektor tipe A dan tipe B seperti terlihat pada gambar berikut.



Gambar 2.17 Pin USB
Sumber : www.pinouts.ru

Konektor tipe A digunakan untuk aliran data *upstream* (sebagai pengirim data ke *host/master*). Soket untuk konektor tipe A ini terdapat pada *host* yang

biasanya terpasang pada *motherboard* komputer dan juga terdapat *hub* USB yang digunakan untuk jaringan peralatan USB. Konektor tipe B digunakan untuk aliran data *downstream* (sebagai penerima data pada peralatan). Soket untuk konektor tipe B terdapat pada peralatan yang terhubung, misalnya pada *printer*, *scanner*, dan lain sebagainya. Fungsi masing-masing pin konektor USB bisa dilihat pada tabel 2.18 berikut.

Tabel 2.7 Fungsi masing-masing pin saluran USB

PIN	WARNA KABEL	FUNGSI
1	Merah	Saluran tegangan VUSB 5 volt
2	Putih	Saluran data negatif (D-)
3	Hijau	Saluran data positif (D+)
4	Hitam	Saluran <i>Ground</i>

Sumber : Peacock, 2002:5

2.12. Karakteristik Video Digital

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Sedangkan *video* digital sendiri berarti jenis sistem *video recording* yang bekerja menggunakan sistem digital dalam hal representasi *videonya*. *Video* digital pada dasarnya tersusun atas serangkaian *frame* yang ditampilkan dengan kecepatan tertentu (*frame / detik*). Jika laju *frame* cukup tinggi, maka mata manusia melihatnya sebagai rangkaian yang kontinyu. Seperti gambar yang ditangkap oleh *webcam* dengan format AVI (*Audio Video Interleaved*) memiliki karakteristik *video* digital, parameter-parameter yang membangun *video* digital adalah resolusi, kedalaman bit dan laju *frame*.

Pada *video*, *frame* merupakan gambar atau citra digital. Suatu citra digital direpresentasikan dengan sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas.

Karakteristik-karakteristik yang akan menentukan kualitas suatu video adalah sebagai berikut.

1. Resolusi/ dimensi *frame*

Resolusi adalah kerapatan (*density*) piksel-piksel dalam satu satuan luas persegi (umumnya dalam 1 inch=2.54 cm). Resolusi menggunakan satuan dpi (*dot per inch*) atau ppi (*pixel per inch*). Jadi bisa dikatakan bahwa *frame* dapat dinyatakan dengan parameter panjang dan lebar yang terlihat pada persamaan berikut.

$$\text{Ukuran } \textit{frame} = \text{panjang (piksel)} \times \text{lebar (piksel)} \dots\dots\dots(2-15)$$

Semakin tinggi resolusi maka semakin baik tampilan *video* tersebut. Namun, resolusi yang semakin tinggi membuat bit yang besar juga untuk mentransmisikannya.

2. Kedalaman bit

Kedalaman bit, yang disebut juga sebagai *bit depth*, adalah satuan yang digunakan untuk mengukur jumlah bit yang ditampilkan dalam tiap piksel gambar. Ukuran kedalaman bit yang sering digunakan adalah 1, 8, 16, 24, 32, dan 64. Kedalaman bit juga dapat dinyatakan pula dengan jumlah warna yang dapat ditampilkan dalam suatu gambar. Sehingga jika ada pernyataan gambar dengan warna 1 bit, maka gambar tersebut dapat ditampilkan dengan dua warna (2 pangkat 1) yaitu hitam dan putih.

Semakin banyak jumlah *bit* yang digunakan untuk merepresentasikan sebuah piksel, semakin tinggi kedalaman pikselnya yang berarti semakin tinggi pula kualitas *video*-nya. Sehingga mengakibatkan jumlah *bit* yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

3. Laju *frame*

Laju *frame* adalah jumlah *frame* yang ditampilkan secara bergantian dan urut setiap detiknya (*frame*/detik). Semakin tinggi laju *frame*, semakin tinggi pula kualitas *video*-nya. Sehingga mengakibatkan jumlah *bit* yang diperlukan untuk menyimpan atau mentransmisikannya meningkat. Sehingga laju *frame* dapat dinyatakan sesuai persamaan berikut.

$$\text{Laju frame (frame / detik)} = \frac{\text{Banyaknya frame}}{1 \text{ detik}} \dots\dots\dots (2-16)$$



BAB III

METODE PENELITIAN

Pada tahap ini dijelaskan mengenai langkah-langkah yang akan dilakukan untuk mengimplementasikan aplikasi perangkat lunak yang akan dibuat. Dalam mengimplementasikan perangkat lunak tersebut diperlukan data, informasi dan metode dalam penyelesaiannya. Langkah – langkah yang perlu dilakukan untuk dapat merealisasikan perangkat lunak yang akan dibuat adalah sebagai berikut :

- Observasi
- Studi literatur
- Perancangan
- Implementasi
- Pengujian dan Analisis
- Pengambilan Kesimpulan

3.1 Observasi

Observasi dilakukan dengan pengumpulan informasi mengenai aplikasi sejenis yang sudah pernah ada baik melalui internet maupun buku. Selain itu observasi dilakukan dengan pengamatan secara langsung pada obyek penelitian untuk mendapatkan informasi tambahan yang berguna.

3.2 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung dalam perancangan dan perealisasiian aplikasi, diantaranya adalah:

1. Hardware
2. Video Streaming.

3. Network/Wireless LAN.
4. Protokol.
5. HTML, CSS dan PHP sebagai *client interface* pada server
6. Teori Perancangan Perangkat Lunak

3.3 Perancangan

Perancangan perangkat lunak terdiri dari analisis kebutuhan perangkat lunak, perancangan sistem, perancangan basis data, perancangan proses, dan perancangan antarmuka. Perancangan didasarkan pada teori-teori yang nantinya akan diaplikasikan pada sistem. Perancangan perangkat lunak dilakukan pada tiap-tiap blok aplikasi sistem untuk mempermudah analisis terstruktur, perancangan, dan pemrograman.

3.4 Implementasi

Implementasi perangkat lunak terdiri dari implementasi basis data dan implementasi antarmuka perangkat lunak. Implementasi perangkat lunak dilakukan berdasarkan pada perancangan perangkat lunak. Melakukan implementasi dari hasil perancangan yang telah didefinisikan sebelumnya kedalam bentuk kode (*coding*) dengan bahasa pemrograman.

3.5 Pengujian dan Analisis

Melakukan pengujian perangkat lunak yang telah dibuat yang ditujukan untuk menemukan kesalahan, sehingga perangkat lunak akan terbebas dari segala kesalahan pada saat dieksekusi oleh pengguna. Pengujian meliputi pengujian basis data, pengujian algoritma dengan teknik *white box testing* dan *black box testing*, pengujian sistem secara *online* setelah dilakukan pengujian maka dilakukan analisis dari keseluruhan sistem.

3.6 Pengambilan Kesimpulan

Pada tahap ini dilakukan pengambilan kesimpulan dari aplikasi yang telah dibuat. Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktek. Kesimpulan ini merupakan informasi akhir dari perancangan aplikasi yang berisi mengenai berhasil atau tidaknya aplikasi tersebut dijalankan.

Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan.



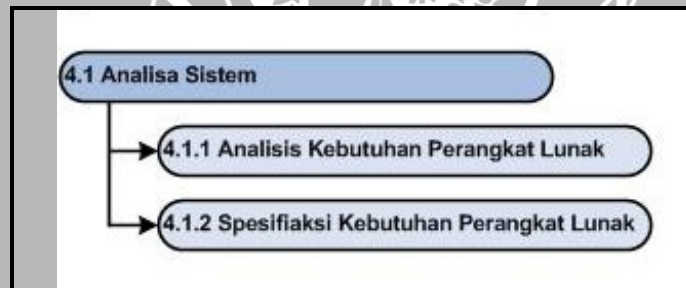
BAB IV PERANCANGAN

Bab ini menjelaskan perancangan Monitoring Ruangan dengan *Webcam* (*Video Streaming*). Perancangan yang dilakukan meliputi dua tahap, yaitu analisis sistem dan perancangan perangkat lunak.

4.1 Analisis Sistem

Analisis dibutuhkan untuk menentukan *user* dan memaparkan kebutuhan *user* dalam pemakaian perangkat lunak yang akan dibuat, selain itu analisis diperlukan untuk menjembatani antara kebutuhan perangkat lunak yang dibuat dengan perancangan perangkat lunak. Sehingga perangkat lunak yang dihasilkan dapat diimplementasikan.

Proses analisis dalam pembuatan sistem ini meliputi analisis kebutuhan perangkat lunak dan implementasi analisis kebutuhan perangkat lunak dalam bentuk *data flow diagram* (DFD).



Gambar 4.1 Diagram Alir Analisis Sistem
Sumber: Perancangan

4.1.1 Analisis Kebutuhan Perangkat Lunak

Penggunaan jaringan *wireless* sebagai media penyaluran sinyal kontrol dan *monitoring* telah berkembang pesat, selaras dengan perkembangan karakteristik masyarakat modern yang memiliki mobilitas tinggi, mencari layanan yang fleksibel dan efisien disegala aspek, serba mudah dan memuaskan dan pada akhirnya penambahan jumlah pemakaian komputerpun tidak dapat dihindarkan

lagi, diantaranya untuk sistem keamanan ruangan, pengaturan peralatan di rumah, kontrol robot dan mesin produksi di Industri. Dengan semakin meningkatnya tindak kejahatan dan pencurian, maka semakin besar pula kebutuhan akan media yang dapat memonitor suatu ruangan. Untuk itu, dibutuhkanlah suatu sistem monitoring ruangan yang mampu memonitor suatu ruangan yang dapat diakses melalui *web* sehingga dapat dimonitor dari jarak jauh melalui jaringan lokal (LAN).

4.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dibutuhkan untuk menjelaskan spesifikasi kebutuhan perangkat lunak yang telah didefinisikan sebelumnya secara lebih detail dan tepat yang akan menjadi dasar bagi perancangan dan implementasi.

Definisi:

1. *User* dapat mengakses sistem monitoring ruangan dan memilih ruangan mana yang akan di tampilkan pada monitor.

Spesifikasi:

- 1.1 *User* harus melakukan setting terhadap *open source software*, dalam hal ini digunakan VLC sebagai perangkat lunak pendukung *video streaming*.
- 1.2 *User* dapat memilih ruangan yang akan *dimonitor*

4.2 Perancangan Perangkat Lunak

Perancangan sistem monitoring ruangan dengan webcam ini dirancang dengan menggunakan bahasa HTML, dengan memanfaatkan *open source software* sebagai media *streaming* gambar dalam hal ini adalah video.

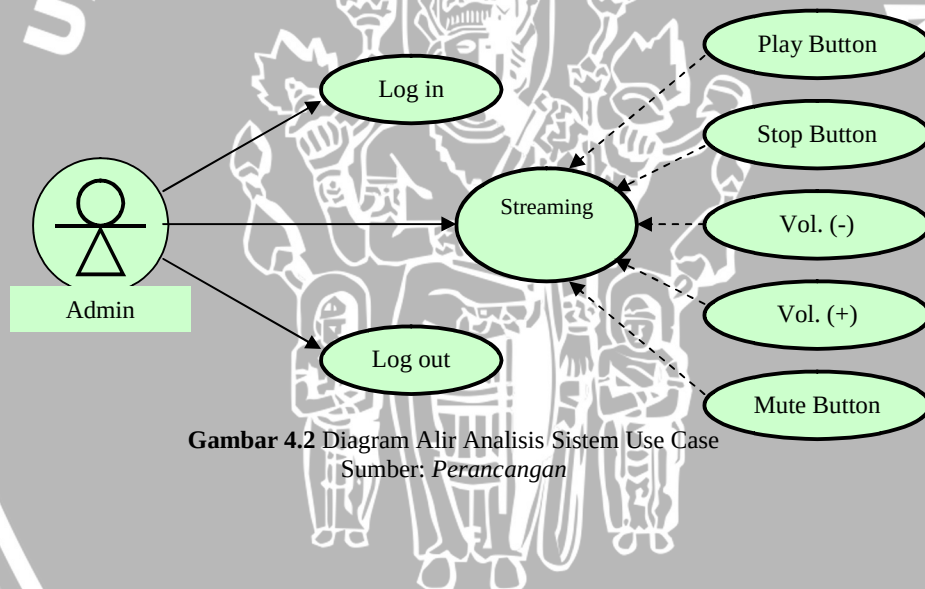
Perancangan perangkat lunak terbagi menjadi tiga yaitu ; perancangan sistem, blok diagram sistem dan cara kerja sistem.

Untuk perangkat lunak sendiri dibagi dua yaitu perangkat lunak pada sisi *server* dan sisi *client*. Berikut perangkat lunak yang digunakan :

1. Windows XP digunakan sebagai *Operating Sistem* untuk *server*. OS ini memiliki fitur yang lengkap dan mudah dalam manajemennya .

2. *VideoLAN Client* (VLC) merupakan perangkat lunak yang dapat berfungsi sebagai *media player* (pemutar file *multimedia*) maupun sebagai *streaming application*. VLC merupakan *software multi platform* yang dapat berjalan di segala macam OS dan memiliki koleksi *multimedia codec* yang lengkap.
3. Browser untuk menampilkan halaman web Kuliah Online. Disarankan menggunakan *Internet Explorer* atau *Mozilla Firefox*.
4. *Client interface* pada halaman web Monitoring ruangan dengan webcam (video streaming) menggunakan bahasa pemrograman HTML, CSS dan PHP.

4.2.1 Perancangan Halaman Utama



Gambar 4.2 Diagram Alir Analisis Sistem Use Case
Sumber: Perancangan

Skenario untuk masing-masing *Use case* akan dijelaskan dalam table scenario *use case*. Skenario *use case* menjelaskan secara detail interaksi dari masing-masing *use case* yang terdapat pada diagram *use case*. Skenario *use case* berisi uraian nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi dan pasca-kondisi yang diharapkan setelah berjalannya fungsional *use case*. Skenario *use case* juga memberikan penjelasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor (aliran utama),

serta kejadian alternatif yang akan terjadi jika suatu kondisi tidak bisa terpenuhi (aliran alternatif). Kebutuhan fungsional untuk proses *log in* yang disediakan oleh system direpresentasikan oleh *use case log in*.

Tabel 4.1 Skenario untuk *use case log in*

Name Use Case	Log in
Aktor	User
Tujuan	Untuk melakukan autentifikasi pemakai
Deskripsi	Proses <i>log-in</i> dilakukan dengan memasukkan <i>user-ID</i> dan <i>password</i> kemudian sistem akan memeriksa keabsahannya, jika <i>user-ID</i> dan <i>password</i> yang dimasukan sah, maka sistem akan menampilkan halaman <i>control panel</i> sesuai dengan level <i>user</i> yang melakukan proses <i>log-in</i> .
Pra-Kondisi	Sistem telah dijalankan pada kondisi <i>use case</i> lihat informasi.
Pasca-Kondisi	<i>User</i> telah melakukan proses <i>log-in</i> , sistem menampilkan halaman <i>control panel</i> .
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. <i>User</i> memilih menu <i>log-in</i> .	2. Sistem menampilkan halaman <i>log-in</i> , halaman <i>log-in</i> berisi <i>textfield</i> untuk mengisi <i>user-ID</i> , <i>password</i> dan sebuah tombol " <i>log-in</i> ".
3. <i>User</i> menuliskan <i>user-ID</i> dan <i>password</i> pada <i>textfield</i> yang disediakan dan menekan tombol " <i>log-in</i> ".	4. Sistem melakukan pemeriksaan karakter terhadap <i>user-ID</i> dan <i>password</i> , selanjutnya sistem melakukan pemeriksaan kedalam basis data, apakah <i>user-ID</i> dan <i>password</i> telah tercatat dalam basis data jika benar maka sistem akan melakukan proses pengaturan <i>session</i> dan selanjutnya melakukan <i>redirection</i> ke halaman <i>control panel</i> sesuai dengan tingkat <i>user level</i> .
Aliran Alternatif 1: Eksepsi jika <i>user-ID</i> atau <i>password</i> salah	
	5. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Tabel 4.2 Skenario untuk *use case log out*

Name Use Case	Log out
Aktor	User
Tujuan	Untuk melakukan autentifikasi pemakai
Deskripsi	Proses <i>log-in</i> dilakukan dengan memasukan <i>user-ID</i> dan <i>password</i> kemudian sistem akan memeriksa keabsahannya, jika <i>user-ID</i> dan <i>password</i> yang dimasukan sah, maka sistem akan menampilkan halaman <i>control panel</i> sesuai dengan level <i>user</i> yang melakukan proses <i>log-in</i> .
Pra-Kondisi	Sistem telah dijalankan pada kondisi <i>use case</i> lihat informasi.
Pasca-Kondisi	<i>User</i> telah melakukan proses <i>log-in</i> , sistem menampilkan halaman <i>control panel</i> .
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. <i>User</i> memilih menu <i>log-in</i> .	2. Sistem menampilkan halaman <i>log-in</i> , halaman <i>log-in</i> berisi <i>textfield</i> untuk mengisi <i>user-ID</i> , <i>password</i> dan sebuah tombol " <i>log-in</i> ".
3. <i>User</i> menuliskan <i>user-ID</i> dan <i>password</i> pada <i>textfield</i> yang disediakan dan menekan tombol " <i>log-in</i> ".	4. Sistem melakukan pemeriksaan karakter terhadap <i>user-ID</i> dan <i>password</i> , selanjutnya sistem melakukan pemeriksaan kedalam basis data, apakah <i>user-ID</i> dan <i>password</i> telah tercatat dalam basis data jika benar maka sistem akan melakukan proses pengaturan <i>session</i> dan selanjutnya melakukan <i>redirection</i> ke halaman <i>control panel</i> sesuai dengan tingkat <i>user level</i> .
Aliran Alternatif 1: Eksepsi jika <i>user-ID</i> atau <i>password</i> salah	
	5. Sistem menampilkan pesan kesalahan dan kembali ke langkah 2.

Tabel 4.3 Skenario untuk *Streaming*

Name Use Case	Streaming
Aktor	User
Tujuan	Untuk melihat tampilan hasil <i>streaming</i>
Deskripsi	Proses <i>streaming</i> dari komputer <i>server</i> ke komputer <i>client</i> .
Pra-Kondisi	Menampilkan <i>display</i> dalam keadaan non-aktif
Pasca-Kondisi	Menampilkan hasil <i>Streaming</i>
Aliran Utama	
Aksi dari Aktor	Tanggapan dari Sistem
1. <i>User</i> menekan tombol Play	2. Sistem menampilkan hasil <i>streaming</i> dari

	computer server
3. User menekan tombol Stop	4. Sistem menghentikan proses streaming
5. User menekan tombol Vol (+/-)	6. Sistem mengatur tingkat volume suara
7. User menekan tombol Mute	8. Sistem mengaktifkan mode silent.

Halaman web monitoring berfungsi untuk menampilkan hasil *streaming* dan mengontrol parameter *audio video* yaitu *audio volume*, *mute*, *stop*, *play* dan *full screen*. Tampilan halaman web yang direncanakan adalah sebagai berikut :



Gambar 4.3. Perancangan Web Monitoring Ruangan
Sumber : Perancangan

Perancangan tampilan halaman web diatas melalui proses berikut :

1. Layout halaman web dibuat dengan bahasa HTML sedangkan untuk mengatur *style* digunakan *Cascading Style Sheet*.
2. Menyisipkan dua *ActiveX plugin* VLC pada halaman web, agar halaman web dapat memutar hasil *streaming audio video* yang di akses.
3. Penambahan fungsi kontrol untuk mengontrol hasil *streaming*. Fungsi kontrol yang tersedia adalah *play*, *stop*, *volume* dan *mute*.

4.2.2 Perancangan Metode *Multimedia Streaming*

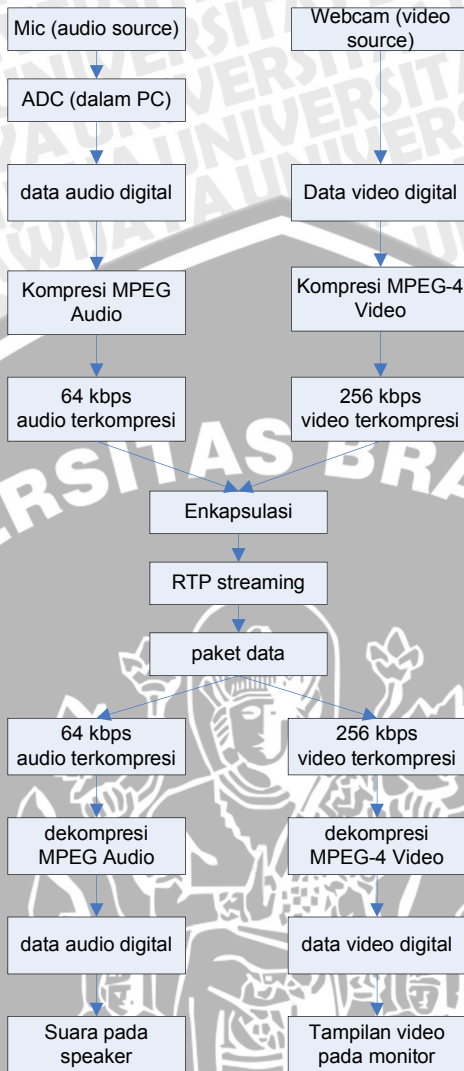
Dalam skripsi ini metode *multimedia streaming* menggunakan teknologi RTP yang berjalan pada jaringan *multicast*. *Video conference* nantinya akan berjalan menggunakan parameter parameter sebagai berikut :

1. Audio
 - *Codec* : MPEG Audio
 - *Bitrate* : 64 kbps
 - *Frequency Sampling* : 44,1 Khz
 - *Channel* : Stereo
2. Video
 - *Codec* : MPEG-4 Video
 - *Data rate* : 256 kbps
 - *Resolusi* : 320 x 480 pixel
 - *Frekuensi frame* : Dapat disesuaikan (minimum 0 fps, maksimum 30 fps).
 - *Bit rate* : dapat disesuaikan (minimum 0 kbps, maksimum 2000 kbps)
3. Protokol komunikasi

Protokol komunikasi yang digunakan untuk proses *streaming* :

 - *Realtime Transport Protocol (RTP)* dengan header 12 byte
 - *User Datagram Protocol (UDP)* dengan header sebesar 8 byte
 - *Internet Protocol version 4 (IPv4)* dengan header sebesar 20 byte
 - *Ethernet II* dengan header sebesar 14 byte

Berikut blok diagram proses *streaming* yang akan dilakukan :



Gambar 4.4 Blok Diagram proses *streaming*
Sumber : Perancangan

4.2.3 Perancangan *Multicast Routing* pada jaringan FTUB

Sistem monitoring menggunakan webcam ini menggunakan teknologi *multicast* pada proses *streaming*nya. Penggunaan teknologi *multicast* dilakukan agar proses *multimedia streaming* menjadi lebih hemat bandwidth. Untuk itu Jaringan komputer FTUB harus memiliki kemampuan *multicast routing*. Pengimplementasian jaringan *multicast* dilakukan di jaringan komputer FTUB. Dalam tahap ini terlebih dahulu dilakukan analisa terhadap jaringan eksisting saat ini, kemudian dilanjutkan dengan proses perancangan *multicast routing* dari sisi *hardware* maupun *software*.

Dari gambar 2.3 dapat kita lihat jaringan komputer FTUB terdiri dari 12 *router* yang tersebar di seluruh area FTUB. Jaringan ini menggunakan teknologi IPv4. Kecepatan data maksimum antara titik terjauh (dari *router* TPTIFT ke *router* Ristie) saat ini berkisar antara 40 – 70 Mbps, sedangkan *ping time* rata – rata sebesar 1.502ms.

Pengimplementasian *routing multicast* pada jaringan komputer FTUB dilakukan dengan menambahkan kemampuan *multicast routing* PIM SM pada setiap *router* yang ada. Sedangkan untuk *routing unicast* tetap mengacu pada tabel *routing unicast* yang sudah ada.

4.2.4 Perancangan Perangkat Keras Jaringan

Perangkat keras yang digunakan dalam jaringan *multicast* tidak berbeda dengan jaringan eksisting. Tidak ada penambahan *hardware* pada jaringan eksisting karena dalam pengimplementasian *multicast* pada jaringan komputer FTUB tidak diperlukan *hardware* tambahan. Perangkat keras yang digunakan terdapat dalam tabel 2.1. Semua *router* dalam jaringan komputer FTUB memiliki spesifikasi yang hampir sama, hanya berbeda pada jumlah NIC yang terpasang saja.

4.2.5 Perancangan Perangkat Lunak Jaringan

Jaringan komputer FTUB menggunakan PC sebagai *router*, oleh karena hal tersebut dibutuhkan perangkat lunak untuk mengatur kinerja PC agar dapat melakukan proses *routing*. Perangkat lunak yang digunakan adalah FreeBSD 6.2 STABLE sebagai *Operating System* (OS). Alasan menggunakan OS ini karena OS ini bersifat *open source* dan memiliki fitur – fitur yang berguna dalam pengaturan jaringan komputer. Selain itu FreeBSD 6.2 STABLE merupakan salah satu OS yang memiliki kinerja paling stabil dan tidak memakan *resource hardware* yang tinggi, sehingga OS ini dapat dijalankan pada PC yang memiliki spesifikasi tidak terlalu tinggi (kecepatan prosesor dibawah 1Ghz).

Perangkat lunak lain yang digunakan dalam perancangan sistem antara lain :

1. *eXtensible Open Router Platform* (XORP) adalah *platform routing open source*. XORP menyediakan fitur-fitur *routing* yang dapat

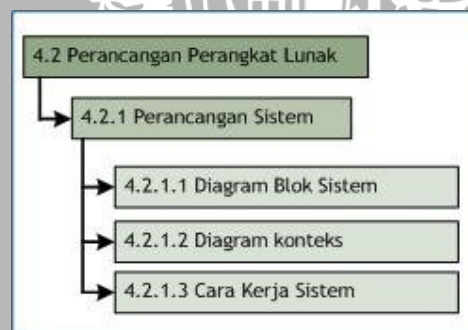
diimplementasikan pada protokol IPv4 dan IPv6, selain itu XORP juga menyediakan *platform* yang sudah terintegrasi untuk mengkonfigurasinya. XORP merupakan salah satu aplikasi *open source platform* yang memiliki fitur untuk *multicast routing*.

2. iftop adalah aplikasi untuk melihat dan mengetahui performansi jaringan komputer pada sebuah *node*, seperti *bandwidth*, koneksi, dan lain lain.
3. ssmpping adalah aplikasi yang digunakan untuk mengetahui konektivitas antara dua *host* melalui jaringan *multicast*. Ssmpping dapat dianalogikan dengan aplikasi ping yang digunakan untuk jaringan *multicast*.

4.3 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui kondisi sistem secara umum. Perancangan sistem meliputi diagram blok sistem dan diagram konteks.

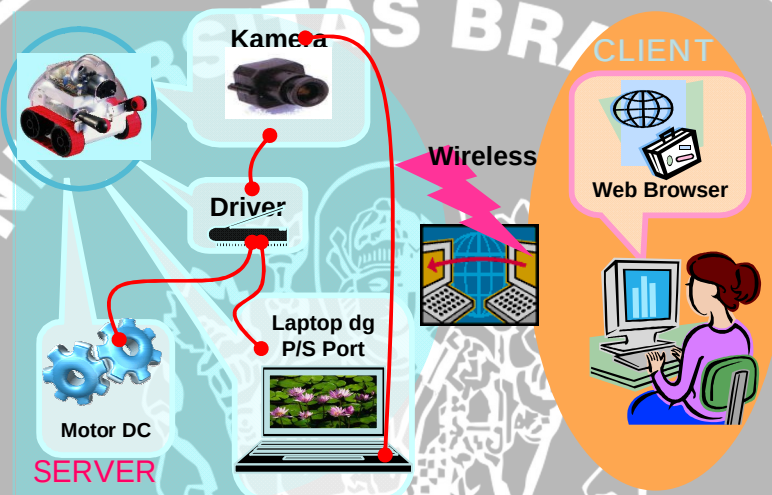
Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang akan dibuat secara umum. Perancangan sistem meliputi diagram blok sistem, diagram konteks dan cara kerja sistem. Diagram pohon perancangan sistem ditunjukkan dalam Gambar 4.9.



Gambar 4. 5. Diagram Pohon Perancangan Sistem
Sumber: Perancangan

4.3.1 Diagram Blok Sistem

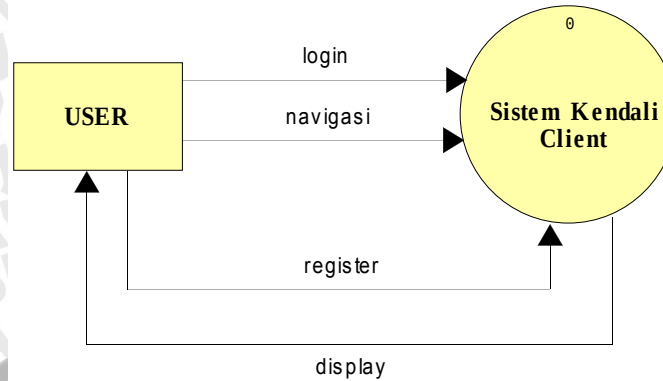
Diagram blok sistem menggambarkan setiap blok atau bagian dari sistem aplikasi. Sistem monitoring ruangan dengan webcam dirancang untuk dapat dijalankan secara *real-time*. Sistem dijalankan secara *online* berarti terdapat di dalam dua atau lebih komputer yang terpisah namun terhubung ke dalam suatu jaringan. Sistem aplikasi yang dijalankan secara *online* dapat digambarkan dengan diagram blok seperti yang diperlihatkan dalam Gambar 4.10.



Gambar 4.6. Koneksi server dan client melalui WirelessLAN
Sumber : Perancangan

4.3.2 Diagram Konteks

Dari diagram blok kemudian direpresentasikan dengan diagram konteks (gambar 4.11.) di bawah ini. Lingkaran tunggal merupakan representasi dari keseluruhan sistem.



Gambar 4.7 Data Flow Diagram Sistem

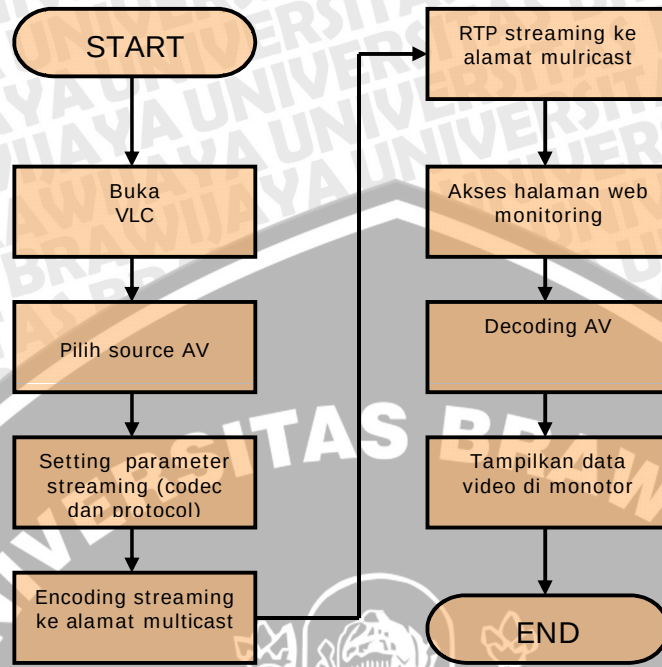
Sumber : Perancangan

4.3.3 Cara Kerja Sistem

Cara kerja sistem monitoring ruangan dengan webcam (video streaming) adalah sebagai berikut :

1. *Server* dan *web server* yang digunakan untuk sistem monitoring ini harus siap (*stand by*).
2. *Server* dan *Client* masing-masing memiliki alamat *multicast* khusus yang digunakan untuk *streaming*.
3. *Client* melakukan inisialisasi *streaming monitoring* dengan *webcam* sebagai sumber *video*. Proses *streaming* menggunakan *VLC* dan ditujukan ke alamat *multicast* untuk *client*.

Secara detail cara kerja sistem monitoring ruangan dengan webcam ditunjukkan dengan diagram alir pada Gambar 4.12.



Gambar 4.8 Diagram alir sistem
Sumber : Perancangan

BAB V

IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini membahas implementasi perangkat lunak. Implementasi adalah suatu proses realisasi dari perancangan perangkat lunak yang telah dilakukan pada bab sebelumnya. Implementasi perangkat lunak meliputi: bahasa HTML, CSS, PHP sebagai aplikasi *client*, dan aplikasi *server*.

5.1 Lingkungan Implementasi

Sistem ini dibuat dengan menggunakan aplikasi pemrograman *Macromedia Flash*, web PHP, HTML dan CSS. Sistem ini juga menggunakan *XMPPA* sebagai *web server* dan VLC Player sebagai media *streaming*. Sistem ini diimplementasikan dengan menggunakan spesifikasi sebagai berikut :

1. Konfigurasi server

Spesifikasi *hardware* :

- Laptop Axioo M7S2 Intel Core 2 Duo
- Memori 1024 Mb SD RAM
- Webcam
- Alamat IP :10.10.21.3

Spesifikasi *Software*

- Microsoft Windows XP Profesional Service Pack 2
- XAMPP Control Panel V.2.5
- Firefox IE / Windows IE
- VLC Player

2. Konfigurasi client

Spesifikasi *hardware*

- Laptop Axioo MS Series Intel Core 2 Duo
- Memori 512 Mb SD RAM
- Alamat IP : 10.10.21.2

Spesifikasi *Software*

- Microsoft Windows XP Profesional Service Pack 2
- Firefox IE / Windows IE

Selain implementasi yang sudah disebutkan sebelumnya, untuk memudahkan implementasi, khususnya dalam pembuatan algoritma dan kode program, dalam hal ini digunakan tool sebagai berikut :

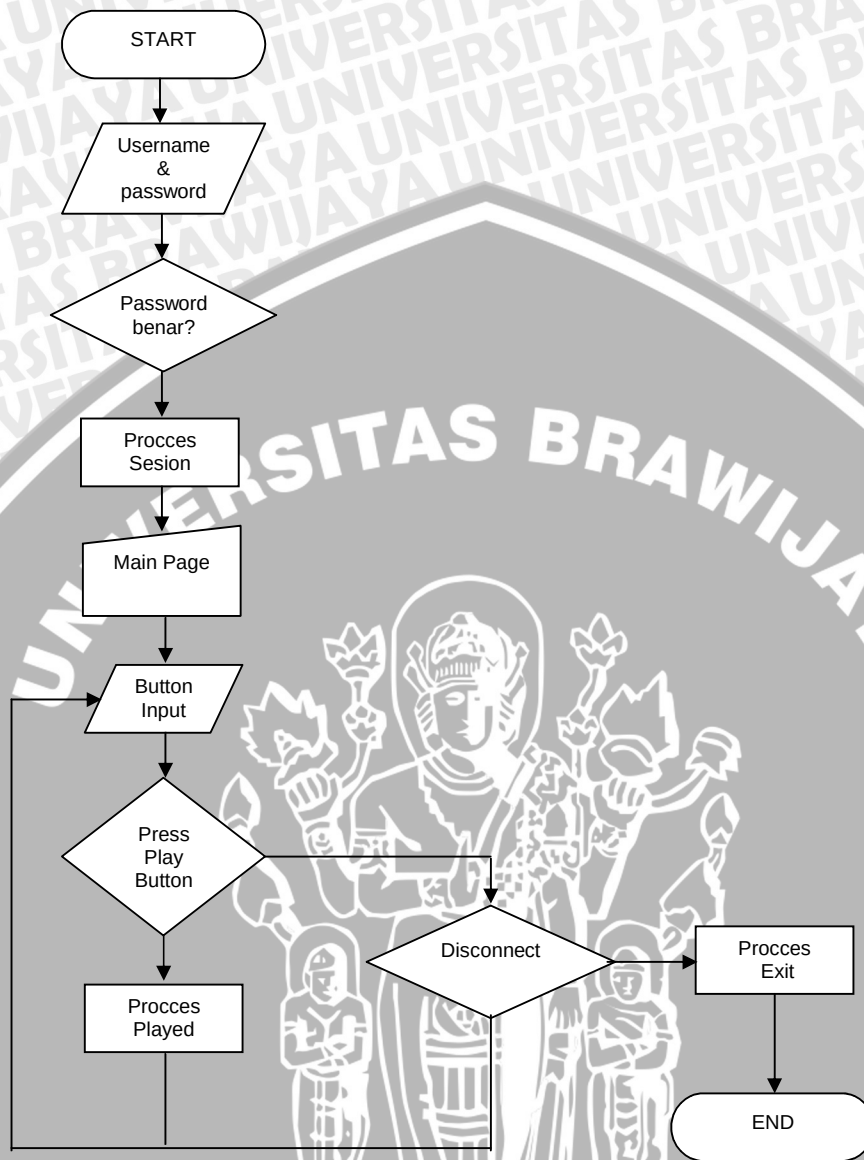
1. Microsoft Visio 2003
Merupakan *tool* untuk menggambarkan algoritma dalam bentuk *flow chart*.
2. Adobe Photoshop 7.0
Digunakan untuk mendesain tampilan dalam bentuk *image*.
3. Macromedia Dreamweaver MX
Digunakan untuk merancang tampilan web.

5.2 Algoritma Pemrograman

Algoritma pemrograman yang telah dibuat agar dapat lebih mudah dipahami maka dalam hal ini dibuat *flow chart* pada setiap proses yang ada.

5.2.1 Proses Sistem Video Steaming

Pada bagian ini akan ditampilkan algoritma sistem *video streaming* secara garis besar dan untuk setiap proses yang terdapat pada sistem *video streaming* ini akan dijelaskan dengan *flowchart* pada sub bab berikutnya. Proses sistem *video streaming* tersebut ditunjukkan pada *flowchart* dalam Gambar 5.1



Gambar 5.1 Algoritma Sistem Video Streaming
Sumber : Implementasi

5.2.2 Aplikasi Video Streaming

Pada bagian ini akan dijelaskan beberapa proses beserta bagaimana mengatur aplikasi ini agar dapat menampilkan video streaming sesuai dengan kebutuhan.

5.2.2.1 Proses Menampilkan Video

VLC media player adalah aplikasi yang akan digunakan sebagai media streaming. Selain disediakan secara gratis (*free software*), aplikasi ini mudah dalam mengoperasikan.

Berikut adalah langkah-langkah bagaimana kita mengatur VLC agar dapat menampilkan video streaming pada web nantinya :

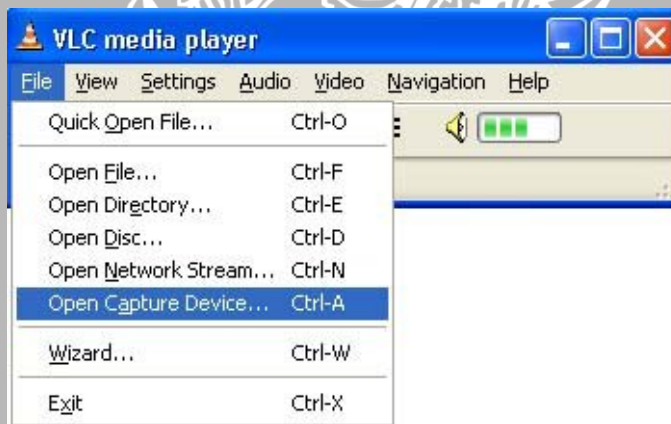
1. Buka Aplikasi VLC



Gambar 5.2 VLC Application

Sumber : Perancangan

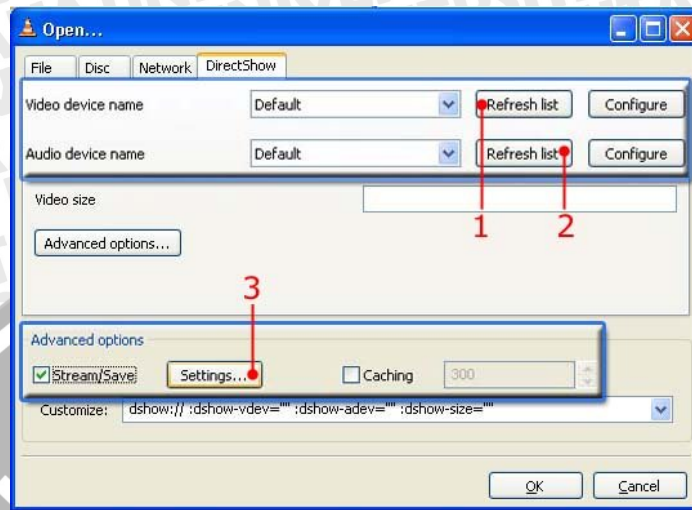
2. Pilih menu “File” kemudia klik “Open Capture Device” <Ctrl+A>



Gambar 5.3 VLC Application > File

Sumber : Perancangan

3. Dari langkah no. 2 diatas akan ditampilkan *form* seperti pada gambar 5.3

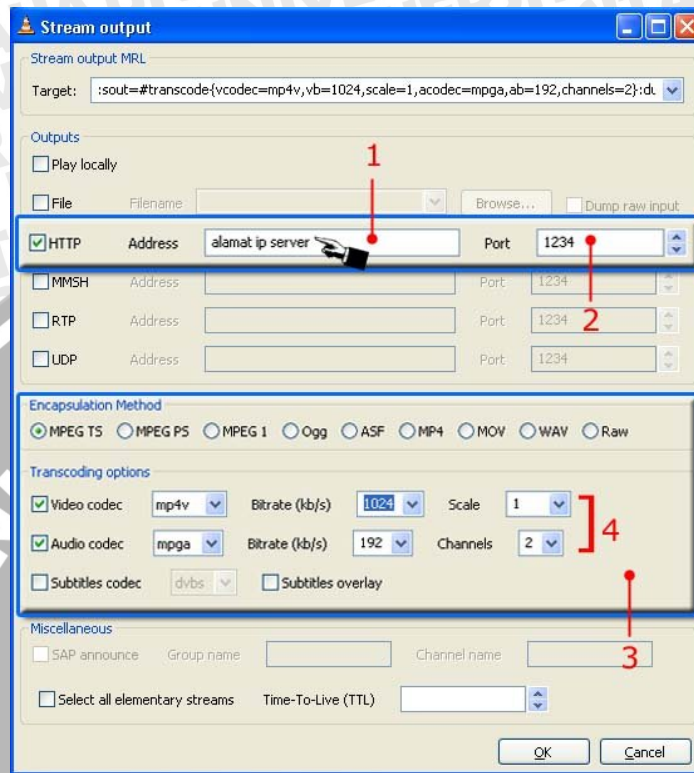


Gambar 5.4 VLC Application > File > Open Capture Device
Sumber : Perancangan

Keterangan :

- 1) *Video device name* adalah untuk memilih *input video device* yang akan dijadikan masukan berupa video. Misalnya kalau kita menggunakan *webcam*, dll.
- 2) *Audio device name* adalah untuk memilih *input audio device* yang akan dijadikan masukan berupa suara. Misalnya kalau kita menggunakan *speaker*, dll.
- 3) Check list “*Stream/Save*” kemudian tekan tombol “*Setting*” untuk mengatur *Stream Output* yang akan ditampilkan

4. Dari langkah no. 2 diatas akan ditampilkan *form* seperti pada gambar 5.3

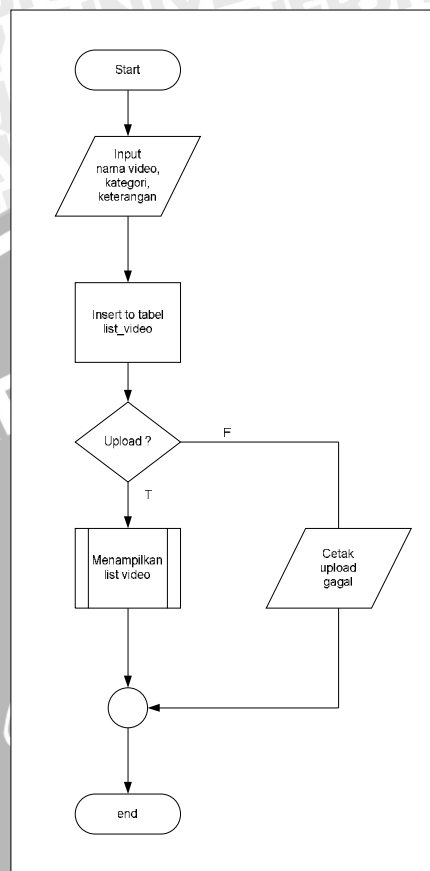


Gambar 5.5 VLC Application > File > Open Capture Device : Setting
Sumber : Perancangan

Keterangan :

1. Masukkan alamat IP atau *IP Address server* sebagai alamat yang nantinya akan di tuju oleh *client*. Misalkan alamat IP kita atur sebagai berikut :
Alamat IP Server A : 10.10.21.3.
2. *Port* adalah
3. *Encapsulation Method* digunakan untuk mengatur kualitas streaming output yang kita inginkan.
4. a). *Video codec*, untuk mengatur *codec* untuk *video*.
b). *Audio codec*, untuk mengatur *codec* untuk *audio* / suara

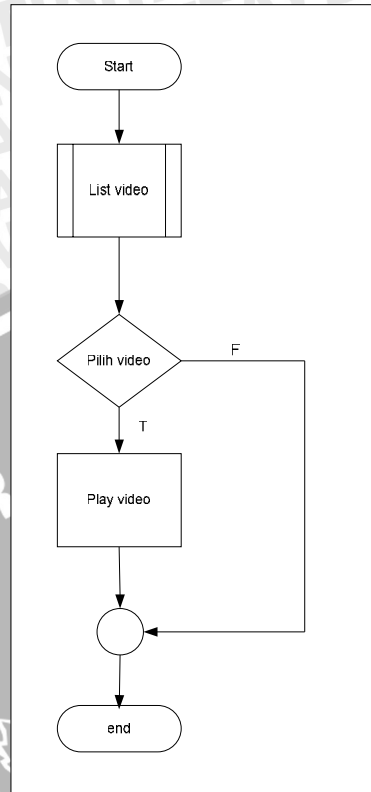
Proses menampilkan video ditunjukkan pada *flowchart* dalam Gambar 5.2



Gambar 5.6 Algoritma *Menampilkan Video*
Sumber: *Implementasi*

5.2.2.2 Proses *Play*

Proses *play* video merupakan proses yang terjadi pada saat video dimainkan oleh *user*. Proses *play* video ditunjukkan pada *flowchart* dalam Gambar 5.3



Gambar 5.7 Algoritma *Play Video*
Sumber: Implementasi

Proses *Play video* dapat dilakukan setelah proses untuk menampilkan *streaming* berhasil dan video dapat dimainkan setelah user menekan *play button*.

5.3 Implementasi Antarmuka

Implementasi web untuk menampilkan hasil *streaming* akan di tunjukkan pada gambar 5.4. *Web monitoring* tersebut dirancang dengan menggunakan aplikasi Dreamweaver MX. Bahasa yang digunakan adalah bahasa HTML dan CSS.

5.3.1 Algoritma Log In

Algoritma *log in* adalah algoritma yang disusun khusus untuk pelaksanaan proses login. Algoritma ini diimplementasikan pada *method login* dalam class *user*.

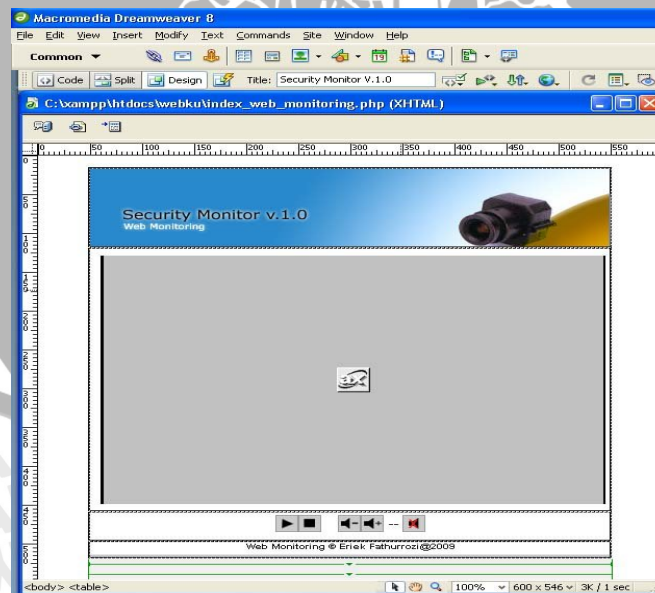
```

1      :: FUNGSION LOGIN ::
2
3      SET user_id value TO value of user_id textfield
4      SET password value TO value of passwordfield
5      SET new password TO encrypted password with md5 algorithm
6      CREATE new database object as login
7      SET query TO "select all from user where user_id='user_id' and password='password'"
8      CALL connect_db method
9      CALL select_db method
10     CALL run_query method
11     IF query_result is not empty string OR query_result is not null THEN
12         SET result TO query_result
13         SET session id TO user_id
14     END IF
15     explode query_result TO array
16     SET level TO value of 2nd index array
17     IF level is 1 THEN
18         redirect TO cpadmin page
19     ELSE IF level is 2 THEN
20         redirect TO cpspdk page
21     ELSE IF level is 3 THEN
22         redirect TO cpskc page
23     ELSE IF level is 4 THEN
24         redirect TO cpscs page
25     ELSE IF level is 5 THEN
26         redirect TO cpskua page
27     ELSE
28         back TO login page
29     END IF
30
31     :: END FUNGSION ::

```

Gambar 5.8 Algoritma Log In

Sumber: Implementasi



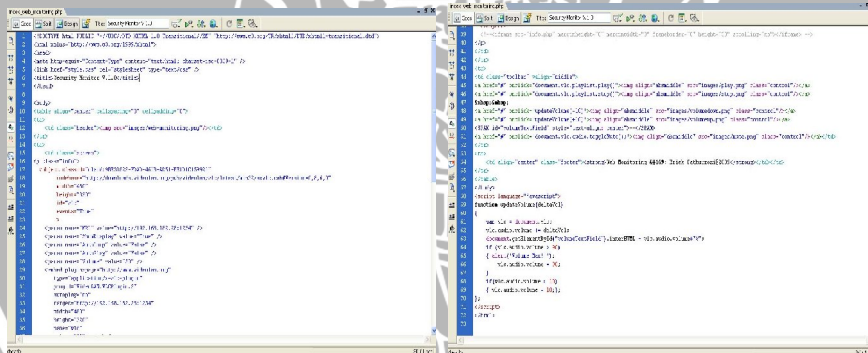
Gambar 5.9 Antarmuka Web

Sumber : Perancangan


```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
02 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
03 <html xmlns="http://www.w3.org/1999/xhtml">  
04 <head>  
05 <meta http-equiv="Content-Type" content="text/html; charset=iso-  
06 8859-1" />  
07 <link href="style.css" rel="stylesheet" type="text/css" />  
08 <title>Security Monitor V.1.0</title>  
09 </head>  
10  
11 <body>  
12 <table align="center" cellpadding="0" cellspacing="0">  
13 <tr>  
14 <td class="header"></td>  
16 </tr>  
17 <tr>  
18 <td class="screen">  
19 <p class="info">  
20 <object classid="clsid:9BE31822-FDAD-461B-AD51-BE1D1C159921"  
21  
22 codebase="http://downloads.videolan.org/pub/videolan/vlc/latest/  
23 win32/axvlc.cab#Version=0,8,6,0"  
24 width="480"  
25 height="320"  
26 id="vlc"  
27 events="True"  
28 >  
29 <param name="MRL" value="http://192.168.182.28:1234" />  
30 <param name="ShowDisplay" value="True" />  
31 <param name="AutoLoop" value="False" />  
32 <param name="AutoPlay" value="False" />  
33 <param name="Volume" value="50" />  
34 <embed pluginspage="http://www.videolan.org"  
35 type="application/x-vlc-plugin"  
36 progid="VideoLAN.VLCPlugin.2"  
37 autoplay="no"  
38 target="http://192.168.182.28:1234"  
39 width="480"  
40 height="320"  
41 name="vlc"  
42 volume="50"> </embed>  
43 </object>  
44 <!--<iframe src="info.php" marginheight="0" marginwidth="0"  
45 frameborder="0" height="30" scrolling="no"></iframe> -->  
46 </p>  
47 </td>  
48 </tr>  
49 <tr>  
50 <td class="toolbar" valign="middle">  
51 <a href="#" onclick=document.vlc.playlist.play()></a>  
53 <a href="#" onclick=document.vlc.playlist.stop()></a>  
55 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
56 <a href="#" onclick='updateVolume(-10)'></a>  
58 <a href="#" onclick='updateVolume(+10)'></a>  
60 <SPAN id="volumeTextField" style="text-align:center">--</SPAN>  
61 <a href="#" onclick=document.vlc.audio.toggleMute()'></a></td>
64 </tr>
65 <tr>
66     <td align="center" class="footer"><strong>Web Monitoring
67     &#169; Eriek Fathurrozi@2009</strong></td></tr>
68 </tr>
69 </table>
70 </body>
71 <script language="javascript">
72     function updateVolume(deltaVol)
73     {
74         var vlc = document.vlc;
75         vlc.audio.volume += deltaVol;
76         document.getElementById("volumeTextField").innerHTML =
77         vlc.audio.volume+"%";
78         if (vlc.audio.volume > 90)
79         { alert('Volume Max!!!');
80           vlc.audio.volume = 90;
81         }
82         if(vlc.audio.volume < 10)
83         { vlc.audio.volume = 10;};
84     };
85 </script>
86 </html>

```

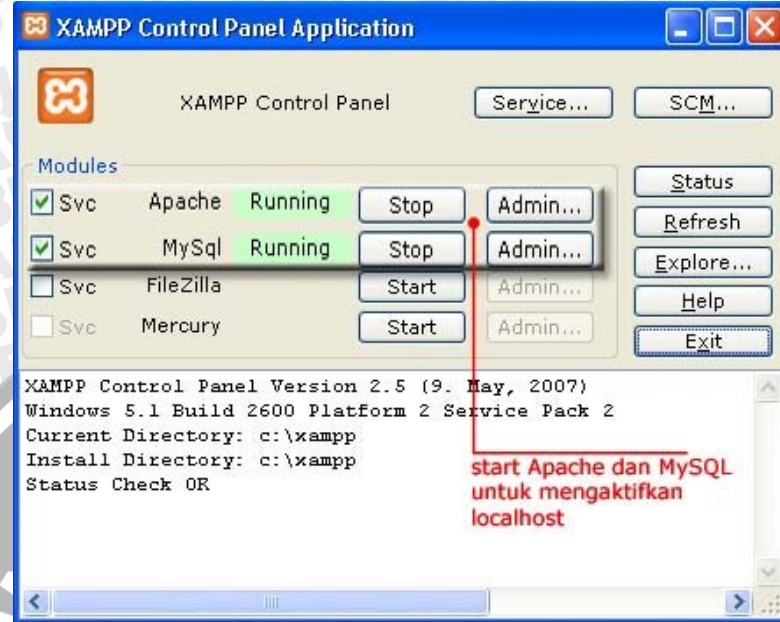
Listing di atas di implementasikan pada aplikasi Macromedia MX seperti yang ditunjukkan pada gambar 5.5.



Gambar 5.10 HTML Code
Sumber : Perancangan

5.4. Implementasi Web Server

Server adalah sebuah komputer yang menjalankan *Video Steaming Server* yang menjadi pusat *transfer* data. Dalam komputer *server* tersebut terdapat sebuah aplikasi *AXMPP Control*. Aplikasi ini secara opsional dapat berisi file *Server-Side Communication Action Script* (ASC) dan tempat menyimpan data. Tampilan XAMPP Control Panel ditunjukkan pada gambar 5.6.



Gambar 5.11. Aplikasi XAMPP
Sumber : Perancangan

BAB VI

PENGUJIAN DAN ANALISIS

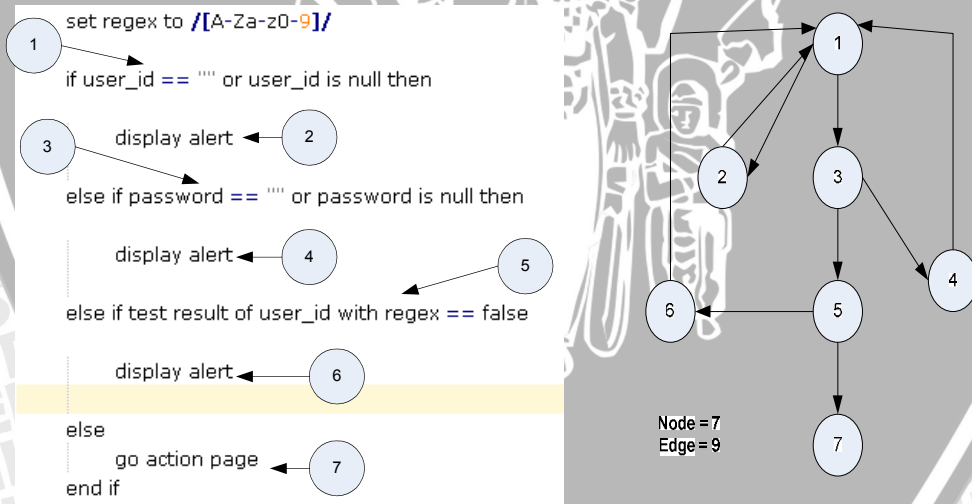
6.1. Pengujian

Pada bab ini akan dibahas tentang pengujian dan analisis Sistem Monitoring Ruangan dengan Menggunakan Webcam melalui jaringan intranet. Pengujian perangkat lunak ditujukan untuk menemukan kesalahan, sehingga perangkat lunak yang dihasilkan dapat terbebas dari segala kesalahan pada saat dieksekusi oleh pengguna.

Pada pengujian *multimedia streaming* dilakukan perbandingan jika proses *streaming* dilakukan dengan menggunakan metode *unicast* dengan proses *streaming* yang menggunakan metode *multicast*. Yang terakhir adalah pengujian aplikasi Sistem Monitoring Ruangan dengan Menggunakan Webcam untuk mengetahui apakah aplikasi yang telah dibuat dapat berfungsi.

6.1.1 Pengujian Halaman Web

Pengujian integrasi proses login:



Dari gambar diatas diperoleh kompleksitas siklomatis sebagai berikut:

$$V(G) = E - N + 2$$

$$V(G) = 9 - 7 + 2$$

$$V(G) = 4$$

Dengan demikian diperoleh 4 buah basis set dari jalur independent yaitu:

Jalur 1 : 1-2-1

Jalur 2 : 1-3-4-1

Jalur 3 : 1-3-5-6-1

Jalur 4 : 1-3-5-7

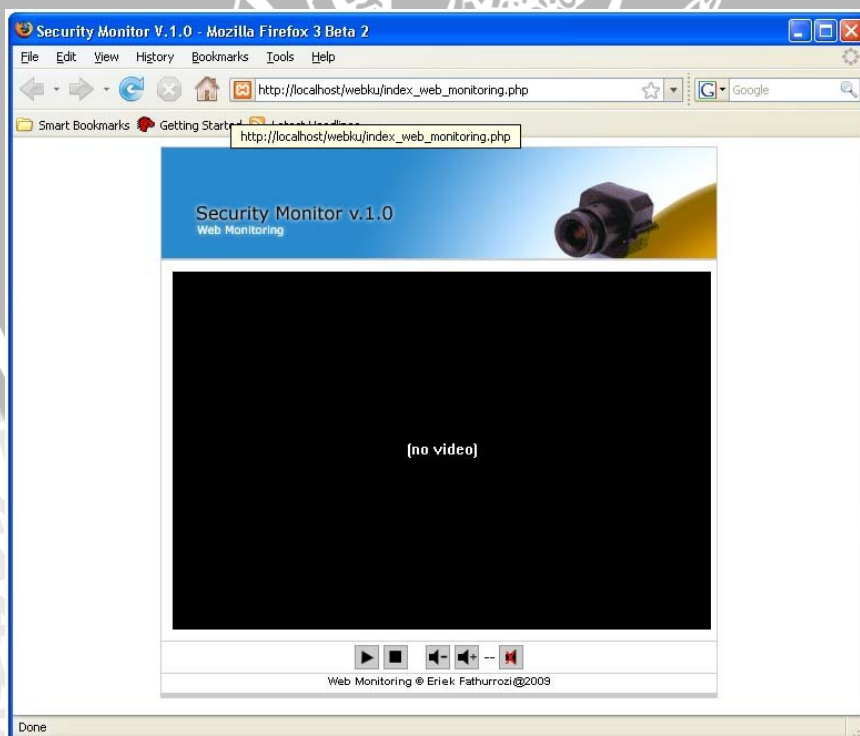
- Jalur 1:
 - o (1) Diberikan user name "" → "" = "" atau "" = null
 - o (2) muncul peringatan kesalahan
 - o (1) kembali ke halaman login
- Jalur 2:
 - o (1) Diberikan user name "@#\$" → "@#\$" != "" atau "@#\$" != null
 - o (3) Diberikan password "" → "" = "" atau "" = null
 - o (4) muncul peringatan kesalahan
 - o (1) kembali ke halaman login
- Jalur 3:
 - o (1) Diberikan user name "@#\$" → "@#\$" != "" atau "@#\$" != null
 - o (3) Diberikan password "@#\$" → "@#\$" != "" atau "@#\$" != null
 - o (5) "@#\$" tidak sesuai dengan regular expretion /[A-Za-z0-9]/
 - o (6) muncul peringatan kesalahan
 - o (1) kembali ke halaman login
- Jalur 4:
 - o (1) Diberikan user name "admin" → "admin" != "" atau "admin" != null
 - o (3) Diberika password "admin" → "admin" != "" atau "admin" != null
 - o (5) "admin" sesuai dengan regular expretion /[A-Za-z0-9]/
 - o (7) Masuk ke halaman admin

Tabel 6.1 Tabel aliran proses login

input	hasil yang diharapkan	hasil yang didapatkan	status
user = ""	menampilkan pesan "user id	menampilkan pesan "user id	valid

password =""	belum terisi" dan kembali ke halaman login	belum terisi" dan kembali ke halaman login	
user = admin password =""	menampilkan pesan password belum terisi dan kembali ke halaman login	menampilkan pesan password belum terisi dan kembali ke halaman login	valid
user ="" password =admin	menampilkan pesan "user id belum terisi" dan kembali ke halaman login	menampilkan pesan "user id belum terisi" dan kembali ke halaman login	valid
user = !@# password =!@#	menampilkan pesan user_id tidak valid dan kembali ke halaman login	menampilkan pesan user_id tidak valid dan kembali ke halaman login	valid

Pengujian halaman sistem monitoring ruangan dilakukan dengan mengakses server Monitoring ruangan melalui *browser*. Pengujian dilakukan dengan menggunakan *browser Internet Explorer* dan mengakses URL http://10.10.21.3/webku/index_web_monitoring.php. Hasil pengujian adalah sebagai berikut :



Gambar 6.1 Pengujian Halaman Sistem monitoring ruangan
Sumber : Pengujian

Dari hasil pengujian menunjukkan jika belum terjadi proses monitoring ruangan maka halaman tidak akan menampilkan *content multimedia*.

6.1.2 Pengujian Sistem Monitoring Ruangan

Pengujian sistem Monitoring ruangan meliputi pengujian fungsi keseluruhan sistem yang telah dirancang. Proses pengujian yang dilakukan adalah sebagai berikut :

1. *Host A* melakukan proses *streaming* dengan aplikasi VLC. *Video source* berasal dari webcam sedangkan *audio source* berasal dari *microphone*. Proses *streaming* menggunakan protokol RTP dan ditujukan pada alamat *multicast 239.1.1.1 port 2222*. *Video* dan *audio* codec menggunakan MPEG 4 dengan *bitrate video 256kbps* dan *audio 64kbps*. Nilai TTL sebesar 12. Proses konfigurasi dengan cara :
 - *Open Capture Device* dengan menekan *Ctrl + A*.
 - Pilih device yang digunakan pada pilihan *Video device name* dan *Audio device name*.
 - Pilih menu *Stream/Save* kemudian klik *Settings*.
2. *Host B* melakukan proses *streaming* dengan aplikasi VLC. *Video source* berasal dari webcam sedangkan *audio source* berasal dari *microphone*. Proses *streaming* menggunakan protokol RTP dan ditujukan pada alamat *multicast 239.1.1.2 port 2222*. *Video* dan *audio* codec menggunakan MPEG 4 dengan *bitrate video 256kbps* dan *audio 64kbps*. Proses yang dilakukan sama dengan pada *host A* hanya pada bagian RTP, IP Address yang digunakan berbeda.
3. *Host A* dan *Host B* mengakses server aplikasi sistem monitoring ruangan dan menyaksikan *multimedia streaming* yang sedang dilakukan melalui browser.



Gambar 6.2 Pengujian Sistem Monitoring Ruangan
Sumber : Pengujian

6.1.3 Pengujian Performansi Sistem Aplikasi

Tabel.6.2 Performansi Video Streaming

LT.	KELAS	KUALITAS SINYAL	STREAMING		KETERANGAN
			Bisa	Tidak Bisa	
1	1-1	4-5	✓		Posisi AP, Client dan Server : • AP diletakkan di sekitar tangga. • Komputer <i>client</i> diletakkan di ruang pengajaran. • Komputer <i>server</i> di setiap kelas.
	1-2	5	✓		
	1-3	5	✓		
	1-4	3-5	✓		
	1-5	3-4	✓		
	1-6	3-4	✓		
2	2-1	5	✓		Kualitas Sinyal : • (1) Very Low • (2) Low • (3) Good • (4) Very Good • (5) Exelent
	2-2	5	✓		
	2-3	5	✓		
	2-4	4	✓		
	2-5	3-5	✓		
	2-6	4-5	✓		
	2-7	4	✓		
	2-8	3-4	✓		

6.1.4 Pengujian Performansi Jaringan

Pengujian dilakukan pada masing-masing kelas.

Tabel 6.3 Hasil capture data pada kelas 2-1

Pengujian ke	t (detik)	Captured packet	Packet Loss	Jumlah data	Jitter
1	60	2592	2	3549070	0,91
2	60	2609		3573810	0,95
3	60	2665	1	3649170	0,85
4	60	2613	2	3577370	0,90
5	60	2652		3631023	0,94
6	60	2643		3620838	0,92
7	60	2658		3640650	0,86
8	60	2663		3647920	0,88
9	60	2617	1	3583780	0,87
10	60	2627	1	3597512	0,87

Sumber : Pengujian

Tabel 6.4 Hasil capture data pada kelas 2-2

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2624	1	3593704	0,98
2	60	2653	1	3633413	0,86
3	60	2661		3645221	0,88
4	60	2604		3567347	0,85
5	60	2591		3547728	0,90
6	60	2569	2	3518202	0,94
7	60	2609	3	3572082	0,92
8	60	2605		3566382	0,86
9	60	2602		3562576	0,97
10	60	2598	2	3557627	1,01

Sumber : Pengujian

Tabel 6.5 Hasil capture data pada kelas 2-3

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2614		3579579	1,02
2	60	2691	1	3685319	0,99
3	60	2499	2	3422006	0,95
4	60	2675	1	3662507	0,86
5	60	2677		3667245	0,95
6	60	2688		3682092	0,85
7	60	2661	1	3644286	0,90
8	60	2656	2	3636599	0,94
9	60	2626		3596971	0,89
10	60	2659		3642285	0,86

Sumber : Pengujian

Tabel 6.6 Hasil capture data pada kelas 2-4

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2649	2	3627474	0,98
2	60	2612		3577441	0,99
3	60	2666	4	3649822	0,94
4	60	2587		3543816	0,97
5	60	2608		3572668	0,99
6	60	2682		3674183	1,03
7	60	2596		3554795	0,96
8	60	2615	3	3580171	0,95
9	60	2664	1	3648545	0,99
10	60	2679		3668167	0,94

Sumber : Pengujian

Tabel 6.7 Hasil capture data pada kelas 2-5

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2630		3602545	0,86
2	60	2627	4	3596620	0,95
3	60	2608	1	3572811	0,85
4	60	2659		3642760	0,90
5	60	2633		3607116	0,94
6	60	2658		3641459	0,92
7	60	2634	3	3606311	0,86
8	60	2673	3	3659642	0,88
9	60	2591	1	3548406	0,87
10	60	2647		3625541	0,87

Sumber : Pengujian

Tabel 6.8. Hasil capture data pada kelas 2-6

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2619	1	3586731	0,91
2	60	2639	1	3614185	0,89
3	60	2602	1	3563479	0,88
4	60	2593	3	3550390	0,91
5	60	2669		3655959	0,89
6	60	2667		3652719	0,87
7	60	2646		3624517	0,87
8	60	2648	4	3625122	0,86
9	60	2683		3675653	0,89
10	60	2645	1	3622552	0,94

Sumber : Pengujian

Tabel 6.9 Hasil capture data pada kelas 2-7

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2602		3564198	0,85
2	60	2592		3550463	0,85
3	60	2665	1	3649961	0,85
4	60	2613		3579152	0,87
5	60	2657	2	3638587	0,86
6	60	2652	2	3631879	0,85
7	60	2637		3611986	0,86
8	60	2640		3616760	0,85
9	60	2666		3652039	0,85
10	60	2662	4	3644316	0,85

Sumber : Pengujian

Tabel 6.10 Hasil capture data pada kelas 2-8

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2647	3	3624482	0,85
2	60	2638		3613039	0,85
3	60	2593	1	3551077	0,84
4	60	2648	2	3626118	0,84
5	60	2565	1	3512966	0,84
6	60	2590		3548202	0,85
7	60	2620		3588763	0,86
8	60	2627		3598698	0,86
9	60	2606	2	3569280	0,86
10	60	2556	3	3499913	0,82

Sumber : Pengujian

Tabel 6.11 Hasil capture data pada kelas 1-1

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2654		3635807	0,96
2	60	2573	1	3523949	0,95
3	60	2635		3609818	0,85
4	60	2584		3539079	1,01
5	60	2618		3585913	0,94
6	60	2569	2	3517488	0,92
7	60	2649	1	3627864	0,86
8	60	2634	4	3606047	0,88
9	60	2665		3650498	0,87
10	60	2579		3532514	0,87

Sumber : Pengujian

Tabel 6.12 Hasil capture data pada kelas 1-2

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2674	1	3661584	0,89
2	60	2627		3597972	0,99
3	60	2601		3562635	0,95
4	60	2585	1	3540114	0,99
5	60	2606		3569011	0,95
6	60	2608	3	3573026	0,85
7	60	2601	1	3561989	0,90
8	60	2612		3577360	0,94
9	60	2642		3618310	1,02
10	60	2678	3	3669027	0,99

Sumber : Pengujian

Tabel 6.13 Hasil capture data pada kelas 1-3

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2560	2	3504970	0,93
2	60	2610	3	3573313	0,99
3	60	2616	1	3582694	0,95
4	60	2591		3549320	0,90
5	60	2636		3610800	0,95
6	60	2587		3543282	0,85
7	60	2590	1	3546827	0,90
8	60	2600	1	3559994	0,94
9	60	2611		3576704	0,88
10	60	2653		3633575	0,92

Sumber : Pengujian

Tabel 6.14 Hasil capture data pada kelas 1-4

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2567		3516010	0,95
2	60	2638	2	3613528	0,85
3	60	2629	2	3600790	0,90
4	60	2573		3524350	0,94
5	60	2565		3513633	0,88
6	60	2637	1	3611576	0,92
7	60	2552		3496078	0,95
8	60	2564	4	3510313	0,85
9	60	2612		3577645	0,90
10	60	2574	3	3524819	0,94

Sumber : Pengujian

Tabel 6.15 Hasil capture data pada kelas 1-5

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2630		3602545	0,86
2	60	2627	4	3596620	0,95
3	60	2608	1	3572811	0,85
4	60	2659		3642760	0,90
5	60	2633		3607116	0,94
6	60	2658		3641459	0,92
7	60	2634	3	3606311	0,86
8	60	2673	3	3659642	0,88
9	60	2591	1	3548406	0,87
10	60	2647		3625541	0,87

Sumber : Pengujian

Tabel 6.16 Hasil capture data pada kelas 1-6

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2624	1	3593704	0,98
2	60	2653	1	3633413	0,85
3	60	2661		3645221	0,90
4	60	2604		3567347	0,94
5	60	2591		3547728	0,92
6	60	2569	2	3518202	0,86
7	60	2609	3	3572082	0,88
8	60	2605		3566382	0,85
9	60	2602		3562576	0,90
10	60	2598	2	3557627	1,01

Setelah data dari tabel 6.3 hingga tabel 6.16 ditabulasi didapat hasil sebagai berikut :

Tabel 6.17 Tabulasi hasil pengujian

NO	KELAS	JUMLAH PAKET DATA TOTAL (PAKET)	JUMLAH PAKET DATA RATA RATA (PAKET)	JUMLAH PACKET LOSS TOTAL (PAKET)	JUMLAH DATA TERCAPTURE (BYTE)	DELAY JITTER RATA RATA (MS)
1	1-1	26339	2633,9	8	2972446,443	0,924
2	1-2	26116	2611,6	9	2980356,833	0,891
3	1-3	26446	2644,6	8	2936056,673	0,970
4	1-4	26358	2635,8	12	2950098,167	0,974
5	1-5	26360	2636,0	12	2966788,917	0,890
6	1-6	26411	2641,1	9	2914275,583	0,893
7	2-1	26386	2638,6	7	2911611,750	0,854
8	2-2	26090	2609,0	9	2977711,500	0,974
9	2-3	26160	2616,0	7	2985748,083	0,893

10	2-4	26234	2623,4	10	2994252,333	0,899
11	2-5	26054	2605,4	12	2973456,583	0,984
12	2-6	25911	2591,1	11	2957395,167	0,970
13	2-7	26351	2635,1	9	2940056,221	0,893
14	2-8	26224	2622,4	12	2992114,866	0,932
	TOTAL	367440	36744	147,000	41452369,12	0,924

Sumber : Pengujian

Keterangan :

- Jumlah paket data total : Jumlah seluruh paket data yang tercapture pada *host* selama pengujian berlangsung.
- Jumlah paket data rata rata : Jumlah paket data rata rata yang diterima *host* selama pengujian berlangsung.
- Jumlah *packet loss* total : Jumlah seluruh *packet loss* total pada *host* selama pengujian berlangsung.
- Jumlah data tercapture : Jumlah total data yang tercapture pada *host* selama pengujian berlangsung.
- Delay jitter rata rata : Delay jitter rata rata pada *host* selama pengujian berlangsung.

6.2. Analisis

Analisis sistem dilakukan secara matematis dengan menggunakan persamaan-persamaan yang ada pada bagian tinjauan pustaka. Analisis dilakukan terhadap parameter-parameter sistem yang digunakan dan hasil yang didapat dari proses pengujian.

6.2.1 Delay

6.2.1.1 Analisis Delay Enkapsulasi Dan Dekapsulasi

Paket *multimedia* yang akan dikirim dari *source* menuju *destination* mengalami penambahan *header*. *Header* yang ditambahkan pada paket *multimedia* murni antara lain :

1. *Header* RTP sebesar 8 byte/paket
2. *Header* UDP sebesar 8 byte/paket
3. *Header* IP sebesar 20 byte/paket

4. *Header Ethernet II* sebesar 14 byte/paket.

Waktu yang dibutuhkan dalam proses penambahan *header* pada paket data *multimedia* disebut delay enkapsulasi. Sedangkan waktu yang dibutuhkan pada proses pelepasan *header* pada destination hingga didapatkan paket *multimedia* murni disebut delay dekapsulasi.

Besar delay enkapsulasi dapat dihitung dari Persamaan (2-10) :

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8$$

Dimana :

t_{enc} = Delay enkapsulasi

W_{frame} = Panjang *frame* data sesudah penambahan *header* sebesar 1370 byte

MSS = Maximum Segment Size 1288 byte

C_{pros} = Kecepatan paket data ditransmisikan atau bitrate yaitu 483,062 kbps

Maka besar delay enkapsulasi adalah :

$$\begin{aligned} t_{enc} &= \frac{1370 - 1288}{483062} \times 8 \\ &= 1,48 \times 10^{-3} \text{ s} \\ &= 1,48 \text{ ms} \end{aligned}$$

Dengan asumsi bahwa proses penambahan *header* dan pelepasan *header* membutuhkan waktu yang sama, maka nilai delay enkapsulasi dan dekapsulasi diasumsikan sama. Dari sini kita dapatkan bahwa besar delay enkapsulasi dan delay dekapsulasi adalah 1,48ms.

6.2.1.2 Analisis Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket ke media transmisi. *Delay transmisi* dihitung dengan menggunakan persamaan (2-13):

$$t_i = \frac{W_{frame}}{C} \times 8$$

Dengan asumsi :

1. Panjang $frame$ (W_{frame}) adalah 1370 byte
2. Kapasitas media transmisi (C) adalah 100 Mbps

Bila diketahui :

$$1024 \times 1024 \text{ bit} = 2^{10} \times 2^{10} \text{ bit} = 1 \text{ Mb (Mega bit)}$$

maka kapasitas media transmisi (C) tersebut dapat dituliskan kembali menjadi

$$C = 100 \times 1024 \times 1024 = 104857600 \text{ bps}$$

Dengan mensubstitusikan nilai W_{frame} , dan C ke dalam Persamaan (2-18) diperoleh nilai $delay$ transmisi (t_i) sebagai berikut :

$$\begin{aligned} t_i &= (1370 \times 8) / 104857600 \\ &= 0,015 \text{ ms} \end{aligned}$$

Dari perhitungan di dapatkan bahwa $delay$ transmisi sebesar 0,105 ms

6.2.1.3 Analisis Delay Propagasi

$Delay$ propagasi adalah waktu yang dibutuhkan untuk merambatkan sinyal yang berisi data *multimedia* dari *node* sumber (*server*) ke *node* tujuan (*client*). $Delay$ propagasi dihitung dengan menggunakan persamaan 2-14 :

$$t_p = (N_{DTE} \times t_{DTE}) + \left(\frac{L_{UTP}}{100} \times t_{UTP} \right) + (N_{router} \times t_{router})$$

Dengan asumsi :

1. Jumlah DTE yang terlewati saat paket ditransmisikan dari *host* A ke *host* B dan sebaliknya adalah 24 DTE. Terdiri dari 16 buah *Fast Ethernet* NIC dan 8 *Fast Ethernet Switch Port*. Dari tabel 2.9 diketahui nilai t_{DTE} sebesar 0,25 μ s
2. Total kabel yang digunakan dari *host* A ke *host* B diasumsikan sepanjang 1300m (panjang maksimum) . Dari tabel 2.9 diketahui nilai t_{UTP} per 100m adalah 0,55 μ s.
3. t_{router} adalah $delay$ proses pada *router*. Nilai ini didapat dari selisih ping time antara 2 *router*. Pada pengujian paket data melewati 8 *router* (N_{router}) dan t_{router} rata rata adalah 0,194ms.

$$\begin{aligned} t_p &= (24 \times 0,25 \mu\text{s}) + ((1300/100) \times 0,55 \mu\text{s}) + (8 \times 0,194\text{ms}) \\ &= 0,006 \text{ ms} + 0,00715 \text{ ms} + 1,552\text{ms} \end{aligned}$$

$$= 1,565\text{ms}$$

Dari analisa didapat nilai delay propagasi sebesar 1,565ms.

6.2.1.4 Analisis Delay Antrian

Delay antrian adalah *delay* yang terjadi karena keterbatasan *node* tujuan (*client*) dalam memproses paket *multimedia* yang datang. *Delay* antrian dihitung dengan menggunakan Persamaan (2-15) seperti yang terlihat di bawah ini :

$$t_w = t_{\text{queue}} + t_{\text{serv}}$$

$$= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu}$$

Dengan asumsi :

1. Kapasitas media transmisi (C) adalah 100 Mbps

Bila diketahui :

$$1024 \times 1024 \text{ bit} = 2^{10} \times 2^{10} \text{ bit} = 1 \text{ Mb (Mega bit)}$$

maka kapasitas media transmisi (C) tersebut dapat dituliskan kembali menjadi

$$C = 100 \times 1024 \times 1024 = 104857600 \text{ bps}$$

2. Panjang paket rata-rata (l) adalah 1370 byte/paket
3. *Throughput* (λ) rata rata adalah 43,81 paket/s.

Kecepatan pelayanan *client* (μ) dihitung dengan menggunakan Persamaan seperti yang terlihat dibawah ini

$$\mu = \frac{C}{l \times 8}$$

Dengan mensubstitusikan nilai C dan l ke dalam persamaan diatas diperoleh nilai kecepatan pelayanan *client* (μ) sebagai berikut :

$$\mu = \frac{C}{l \times 8}$$

$$= \frac{104857600}{1370 \times 8}$$

$$= 9567.29 \text{ paket / s}$$

Kemudian dengan mensubstitusikan nilai λ , dan μ ke dalam Persamaan (2-15) diperoleh nilai *delay* antrian (t_w) sebagai berikut :

$$\begin{aligned}
 t_w &= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \\
 &= \frac{43,81}{(9567,29)(9567,29 - 40,026)} + \frac{1}{9567,29} \\
 &= 0,000105 \text{ s}
 \end{aligned}$$

Dari perhitungan di atas didapat nilai delay antrian adalah sebesar 0,000105 s atau sebesar 0,105 ms.

6.2.1.5 Analisis Delay End-To-End

Delay end to end merupakan nilai delay total dari delay yang ada dihitung dari *source* ke *receiver*. Sesuai dengan persamaan 2-7 besar *delay end to end* adalah :

$$t_{\text{end-to-end}} = t_{\text{codec}} + t_{\text{enc}} + t_i + t_p + t_w + t_{\text{dec}} + t_{\text{jitter}} + t_{\text{codec}}$$

Dimana :

1. t_{codec} adalah waktu yang dibutuhkan untuk memproses data *multimedia* digital menjadi payload data MPEG. Menurut referensi, untuk metode encoding MPEG, *delay codec* sebesar 100ms (Vigato, 2006).
2. t_{enc} dan t_{dec} didapat dari analisa hasil pengujian sebesar 1,48ms.
3. t_p didapat dari analisa hasil pengujian sebesar 1,565ms.
4. t_i didapat dari analisa hasil pengujian sebesar 0,105ms
5. t_w didapat dari analisa hasil pengujian sebesar 0,105ms.

$$\begin{aligned}
 t_{\text{end-to-end}} &= 50\text{ms} + 1,48\text{ms} + 0,105\text{ms} + 1,565\text{ms} + 0,105\text{ms} + 0,709\text{ms} + 50\text{ms} \\
 &= 103,255 \text{ ms}
 \end{aligned}$$

Dari analisa didapat delay end to end sebesar 103,964ms. Pada sebuah sistem *real time multimedia streaming delay end-to-end* sebesar 104,71ms masih dalam ambang batas yang diperbolehkan yaitu masih di bawah 150ms

6.2.2 Pengujian Delay (Manual)

Tabel 6.18 pengujian delay secara manual

NO.	PENGUJIAN KE	WAKTU (DETIK)
1.	1	2,7
2.	2	3,0
3.	3	2,9
4.	4	2,9
5.	5	2,8
6.	6	3,0
7.	7	3,0
8.	8	2,8
9.	9	2,9
10.	10	2,9
	RATA-RATA	2,9

Dari perhitungan secara teori dan manual terdapat perbandingan perbedaan yang cukup jauh, ini menunjukkan bahwa perhitungan secara teori tidak sesuai dengan hasil percobaan di lapangan. Hal ini disebabkan oleh beberapa faktor di lapangan, seperti ; topografi gedung (meliputi dinding pembatas seperti beton), spesifikasi komputer dan beberapa hal lainnya.

6.2.3 Packet Loss

Packet loss adalah banyaknya paket data yang hilang saat transmisi. Semakin besar prosentase *packet loss* maka semakin buruk performansi sebuah saluran transmisi. Sesuai persamaan maka nilai packet loss adalah :

Packet loss proses *streaming* dari *Server* ke *Client* (contoh pada kelas 1.6), sampel data diambil dari pengujian kelas 1-6.

Tabel 6.19. Hasil capture data pada kelas 1-6

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2624	1	3593704	0,98
2	60	2653	1	3633413	0,85
3	60	2661		3645221	0,90
4	60	2604		3567347	0,94
5	60	2591		3547728	0,92
6	60	2569	2	3518202	0,86
7	60	2609	3	3572082	0,88

8	60	2605		3566382	0,85
9	60	2602		3562576	0,90
10	60	2598	2	3557627	1,01
RATA-RATA					
TOTAL		29116	9		

$$N_{packet\ loss} (\%) = \frac{N_{packet\ loss}}{N_{paket} + N_{packet\ loss}} \times 100 \%$$

Dengan demikian Paket Loss per 60 detik adalah sebagai berikut :

$$\begin{aligned} N_{packet\ loss} (\%) &= (9 / (29116+9)) \times 100\% \\ &= 0,031 \% \end{aligned}$$

6.3 Pengujian Mean Opinion Square (MOS)

Pengujian MOS dilakukan untuk mengetahui kualitas *video* multimedia *streaming*. Pengujian dilakukan pada kualitas *video* dan delay. Pengujian MOS dilakukan pada *stream* dari *Server* ke *Client* dengan menggunakan opini dari 24 responden. Standar penilaian kualitas untuk *video* adalah sebagai berikut :

Tabel 6.20 Indeks kualitas *video*

Skor	Kualitas
5	Sangat Baik
4	Baik
3	Sedang
2	Buruk
1	Sangat Buruk

Sumber : Pengujian

Sedangkan untuk penilaian kualitas delay adalah sebagai berikut.

Tabel 6.21 Indeks pengaruh delay

Skor	Kualitas
5	Tidak terasa
4	Terasa
3	Sedikit mengganggu
2	Mengganggu
1	Sangat mengganggu

Sumber : Pengujian

6.3.1 Pengujian kualitas video

Pengujian kualitas video dilakukan dengan meng-capture hasil *streaming* kemudian menyimpannya dalam bentuk file JPEG. Kemudian gambar hasil capture diberikan kepada responden untuk dinilai kualitasnya. Hasil pengujian ditampilkan pada tabel berikut

Tabel 6.22 Hasil analisa kualitas video menggunakan MOS

Skor	Kualitas	Responden	Total
5	Sangat Baik	-	0
4	Baik	7	21
3	Sedang	17	51
2	Buruk	-	0
1	Sangat Buruk	-	0
		TOTAL	79

Sumber : Pengujian

Dari tabel 6.21 dapat diketahui nilai MOS adalah

Jika dihitung dengan persamaan berikut maka :

$$\begin{aligned}
 \text{MOS} &= \frac{\text{Total skor}}{\text{Jumlah responden}} \\
 &= 79 / 24 \\
 &= 3,29
 \end{aligned}$$

Nilai MOS untuk kualitas video adalah 3,29. (Sedang)

6.3.2 Pengujian pengaruh delay

Pengujian pengaruh delay dilakukan dengan meminta responden untuk menyaksikan langsung proses *streaming*. Hasil pengujian ditampilkan pada Tabel 6.22.

Tabel 6.23 Hasil analisa pengaruh delay dengan metode MOS

Skor	Kualitas	Responden	Total
5	Tidak terasa	-	0
4	Terasa	-	0
3	Sedikit mengganggu	9	27
2	Mengganggu	13	26
1	Sangat Mengganggu	2	2
		TOTAL	55

Sumber : Pengujian

Dari Tabel 6.22 dapat diketahui nilai MOS adalah

Jika dihitung dengan persamaan berikut maka :

$$\begin{aligned}\text{MOS} &= \frac{\text{Total skor}}{\text{Jumlah responden}} \\ &= 55 / 24 \\ &= 2,29\end{aligned}$$

Nilai MOS untuk pengaruh delay adalah 2,29 (Mengganggu).



BAB VII

PENUTUP

Setelah melalui tahap perancangan dan implementasi sistem, *multicast routing* pada jaringan komputer lokal dapat berjalan dengan baik sehingga proses multimedia *streaming* dengan metode *multicast* menggunakan protokol HTTP dapat berjalan dengan baik. Pengujian pada sistem juga telah dilakukan dan diperoleh hasil pengujian yang telah dianalisa sesuai dengan parameter-parameter yang telah ditentukan. Dari semua proses yang telah dilakukan mulai awal hingga akhir pada skripsi ini dapat ditarik kesimpulan dan saran yang diharapkan dapat bermanfaat di masa mendatang.

7.1. Kesimpulan

Beberapa kesimpulan yang dapat diambil dari laporan tugas akhir ini antara lain:

1. Video streaming yang dihasilkan tidak sepenuhnya real-time. Hal ini disebabkan adanya delay, baik delay proses maupun delay transmisi
2. Dari pengujian didapat nilai *packet loss* rata rata sebesar 0,026%. Nilai packet loss ini masih jauh dibawah nilai packet loss yang diperbolehkan yaitu 1%. Hal ini menunjukkan bahwa jaringan yang ada telah dibangun *reliable* dalam proses *packet delivery*.

7.2. Saran

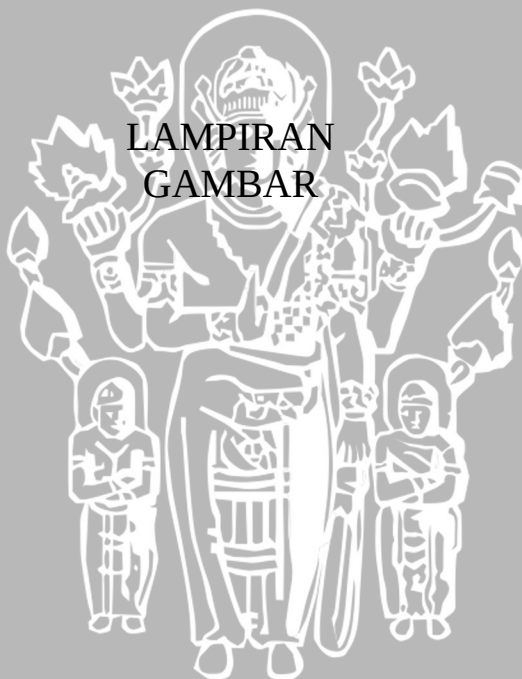
Dalam perencanaan di laporan akhir ini masih didapatkan hal-hal yang belum sempurna, untuk itu perencanaan ini masih bisa disempurnakan, yaitu :

1. Aplikasi sistem web monitoring yang dibuat masih sangat sederhana dengan fitur fitur yang belum lengkap. Diharapkan aplikasi yang telah dibuat dapat dikembangkan lebih lanjut.
2. Untuk mendapatkan sistem transmisi yang lebih ringan sebaiknya tidak menggunakan jenis kompresi video yang terlalu memakan bandwidth yang besar.

DAFTAR PUSTAKA

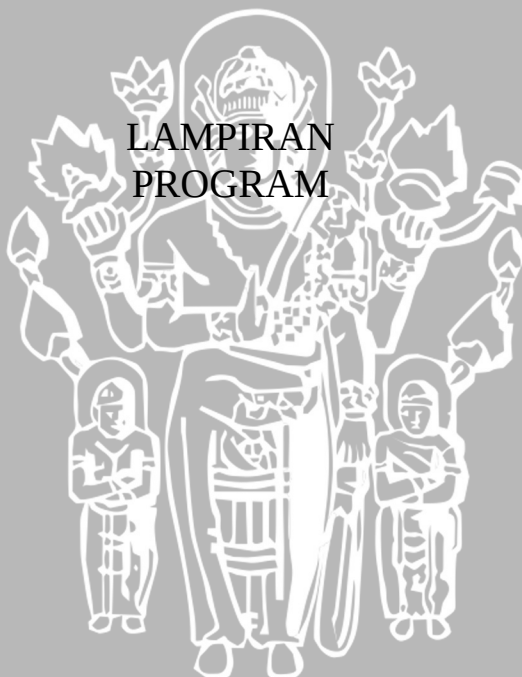
- [AND-04] "Wireless Atasi Keterbatasan Jangkauan", Andi Yogyakarta, 2004
- [ABD-03] Abdul Kadir, "Dasar Pemrograman WEB Dinamis menggunakan PHP", Andi Yogyakarta, 2003
- [ARF-05] Arif Lutfanda, "Pembuatan Monitoring Ruang Berbasis Camera Server", Proyek Akhir PENS-ITS, 2005
- [ASK-05] Askari Azikin, "Kamera Pengawas Berbasis Open Source", PT Elex Media Komputindo, Jakarta, 2005
- [ONN-02] Onno W. Purbo, "Konferensi Video Melalui Internet", Andi Yogyakarta, 2002
- [SID-05] Sidik, Betha, Ir, Pohan, Husni I, Ir., M. Eng., 2005, *Pemrograman Web dengan HTML*, Informatika, Bandung
- [ANY-01] "Modul Pelatihan Pengenalan Linux", <http://ftp.ui.edu/bebas/v01/TimPandu/linux-dasar-single.pdf>
- [ANY-02] "Camserv", <http://cserv.sourceforge.net/>
- [ANY-03] "Package Camserv", <http://packages.debian.org/testing/web/Camserv>
- [ANY-04] "Pengantar JavaScript", <http://www.ilmukomputer.com/>
- [ANY-05] "Dasar Pemrograman PHP dan MySQL", <http://www.ilmukomputer.com/>
- [ANY-06] "Aplikasi web dengan PHP", <http://www.ilmukomputer.com/>
- [ANY-07] "Pengenalan TCP/IP", http://www.ai3.itb.ac.id/Tutorial/pengenalan_tcp.htm
- [ANY-08] www.ilkom.unsri.ac.id/TA&KP/abstract/TK/FOGI_CHANDRA.doc

UNIVERSITAS BRAWIJAYA



LAMPIRAN
GAMBAR

UNIVERSITAS BRAWIJAYA



LAMPIRAN
PROGRAM

UNIVERSITAS BRAWIJAYA

LAMPIRAN
QUESIONER METODE MOS

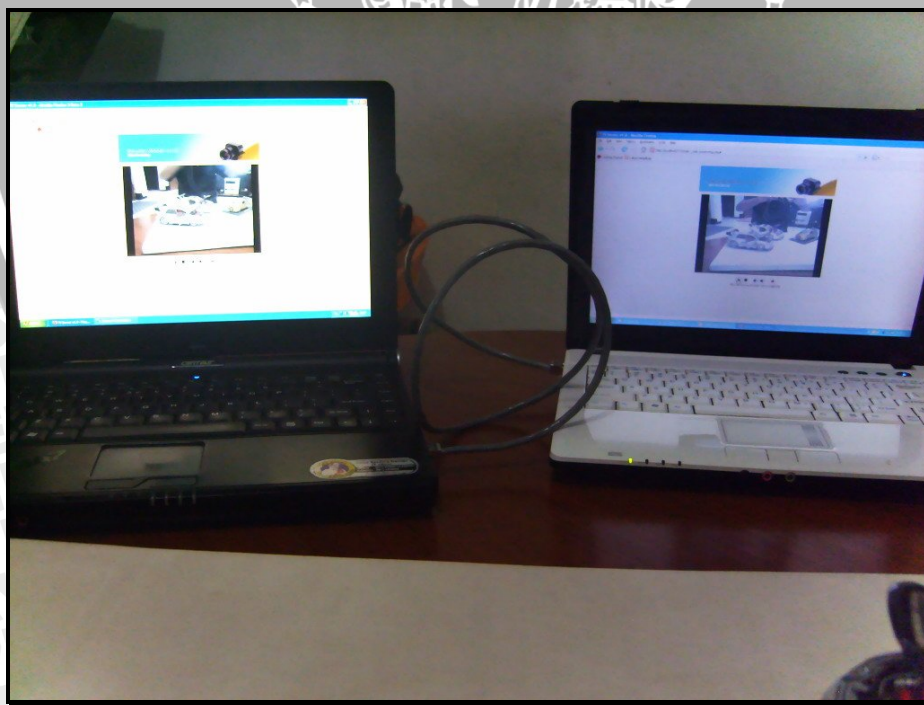




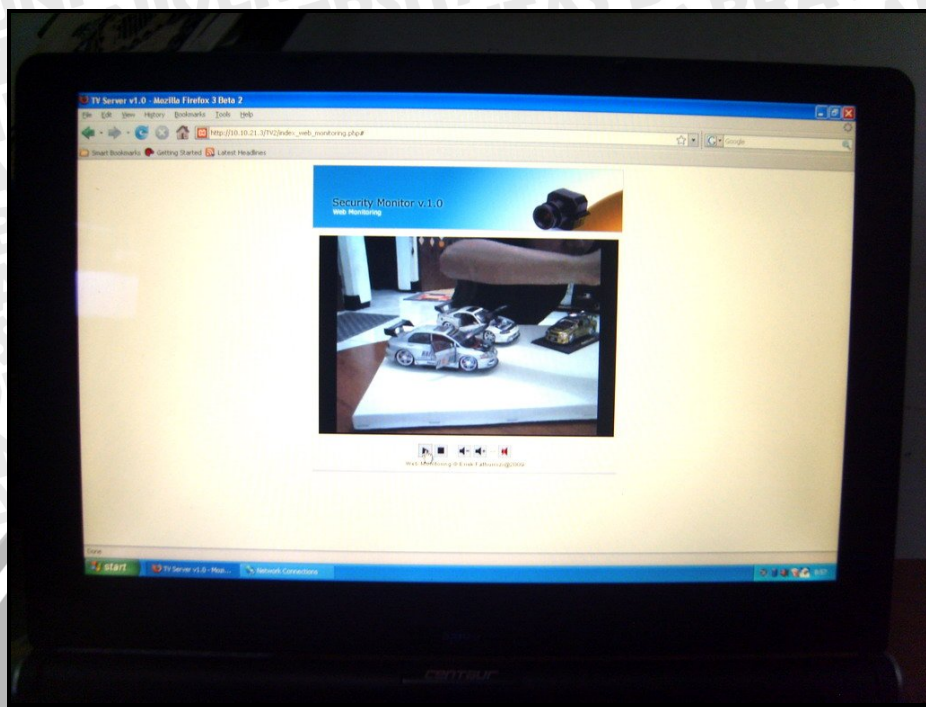
DAFTAR GAMBAR



Gambar A. Objek yang akan di monitor



Gambar B. Hasil Streaming dari Host A ke Host B



Gambar C. Hasil Streaming pada komputer Client



LAMPIRAN

Tabel. Hasil kuesioner kualitas video dengan MOS

No	Sangat Baik	Baik	Sedang	Buruk	Sangat Buruk
1			√		
2			√		
3			√		
4		√			
5			√		
6			√		
7			√		
8		√			
9			√		
10			√		
11			√		
12			√		
13			√		
14			√		
15		√			
16			√		
17			√		
18		√			
19		√			
20			√		
21			√		
22			√		
23		√			
24		√			

Tabel. Hasil kuesioner pengaruh delay dengan metode MOS

No	Tidak terasa	Terasa	Sedikit mengganggu	Mengganggu	Sangat mengganggu
1			√		
2			√		
3			√		
4			√		
5				√	
6				√	
7			√		
8				√	
9			√		
10			√		
11			√		
12				√	
13				√	
14				√	
15				√	
16				√	
17			√		
18				√	
19				√	
20				√	
21				√	
22				√	
23					√
24					√

Index - Home

```
<?php
session_start();
session_destroy();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/>
<link href="style.css" rel="stylesheet" type="text/css" />
<title>Home - Log in</title>
<style type="text/css">
body{ background-image:url(images/ziron_1152.jpg); background-
attachment:fixed;}
.box{width:800px; margin:auto;}
.box_judul{width:800px; height:auto; font-family:Tahoma; font-
size:36px; font-weight:bold; text-align:center; color:#fff; padding-
top:20px; padding-bottom:20px;}
.box_login{width:600px; margin:auto; height:auto;}
.bl1{width:64px; height:auto; float:left;}
.blr{width:500px; height:auto; float:right;}
</style>
</head>

<body>
<div class="box">
    <div class="box_judul">SECURITY MONITORING</div>
    <div class="box_login">
<form method="post" action="login.php">
<table width="250" border="0" align="center">
    <tr>
        <td width="85" style="color:#fff;">User Id </td>
        <td width="10">:</td>
        <td width="155"><input type="text" name="user" size="25" /></td>
    </tr>
    <tr>
        <td style="color:#FFFFFF;">Password</td>
        <td>:</td>
        <td><input type="password" name="pass" size="25" /></td>
    </tr>
    <tr>
        <td colspan="3" align="right"><input name="submit" type="submit"
value="Log In" /></td>
    </tr>
</table>
</form>
    </div>
</div>
</body>
</html>
```

Login

```
=====
<?php
session_start();
class lock{
public $nama;
public $pass;
function login()
{
    //echo $this->pass;
    if(($this->pass != NULL || $this->pass != '')&&($this->nama != '' || $this->nama != NULL))
    {
        //echo $this->pass;
        $pass_valid = md5($this->nama);
        $nama_valid = md5($this->pass);
        if($pass_valid && $nama_valid)
        {
            //echo $pass_valid;
            $_SESSION['valid']=$pass_valid;
            if($_SESSION['valid']!= '' || $_SESSION['valid']!=NULL)
            {
                //echo $this->pass;
                header('location:index_web_monitoring.php');
            }
        }
        else
        {
            header('location:index.php');
        }
    }
}
}
$lock = new lock();
$lock->nama=$_POST['user'];
$lock->pass=$_POST['pass'];
$lock->login();
?>
```


Main Program

```

=====
<?php
session_start();
if($_SESSION['valid']==NULL||$_SESSION['valid']=='')
{
    header('location:index.php');
}
elseif($_SESSION['valid']==md5('user'))
{
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/>
<link href="style.css" rel="stylesheet" type="text/css" />
<title>Security Monitor V.1.0</title>
</head>

<body>
<table align="center" cellspacing="0" cellpadding="0">
<tr>
<td class="header"></td>
</tr>
<tr>
<td id="header" class="text"><a href="index.php">Log Out</a></td>
</tr>
<tr>
<td class="screen">
<p class="info">
<object classid="clsid:9BE31822-FDAD-461B-AD51-BE1D1C159921"
codebase="http://downloads.videolan.org/pub/videolan/vlc/latest/win32/a
xvlc.cab#Version=0,8,6,0"
width="480"
height="320"
id="vlc"
events="True"
>
<param name="MRL" value="http://192.168.182.28:1234" />
<param name="ShowDisplay" value="True" />
<param name="AutoLoop" value="False" />
<param name="AutoPlay" value="False" />
<param name="Volume" value="50" />
<embed pluginspage="http://www.videolan.org"
type="application/x-vlc-plugin"
progid="VideoLAN.VLCPlugin.2"
autoplay="no"
target="http://192.168.182.28:1234"
width="480"

```


CSS Script for Button Panel

```
body {  
font-family:Arial, sans-serif;  
font-size:12px;  
text-align:left;  
padding:0;  
color:#000000;  
background:#FFFFFF;  
background:url(images/spaceh_head.png) repeat-x;  
}
```

```
ul {  
list-style:decimal;  
margin:0px 5px 0px 30px;  
padding:0;  
}
```

```
.header {  
border-bottom:2px solid #CCCCCC;  
border-top:1px solid #CCCCCC;  
border-left:1px solid #CCCCCC;  
border-right:1px solid #CCCCCC;  
}  
#header{border-bottom:2px solid #CCCCCC;  
border-top:1px solid #CCCCCC;  
border-left:1px solid #CCCCCC;  
border-right:1px solid #CCCCCC;  
background-color:#FFFFFF;}  
.insert {  
height:20px;  
text-align:center;  
border-left:1px solid #CCCCCC;  
border-right:1px solid #CCCCCC;  
}
```

```
.login {  
padding:3px;  
border-left:1px solid #CCCCCC;  
}
```

```
.textbox {  
padding:3px;  
border-right:1px solid #CCCCCC;  
}
```

```
.footer {  
font-size:10px;  
padding-bottom:5px;  
border-left:1px solid #CCCCCC;  
border-right:1px solid #CCCCCC;  
border-bottom:5px solid #CCCCCC;  
}
```

```
.button {
```

```
margin:0;
padding:0px 2px 0px 2px;
color:#000;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight:bold;
background-color:#DDDDDD;
border:1px solid #000000;
}
.title {
padding:5px;
border-left:1px solid #CCCCCC;
border-right:1px solid #CCCCCC;
text-align:center;
font-weight:bold;
color:#666666;
}

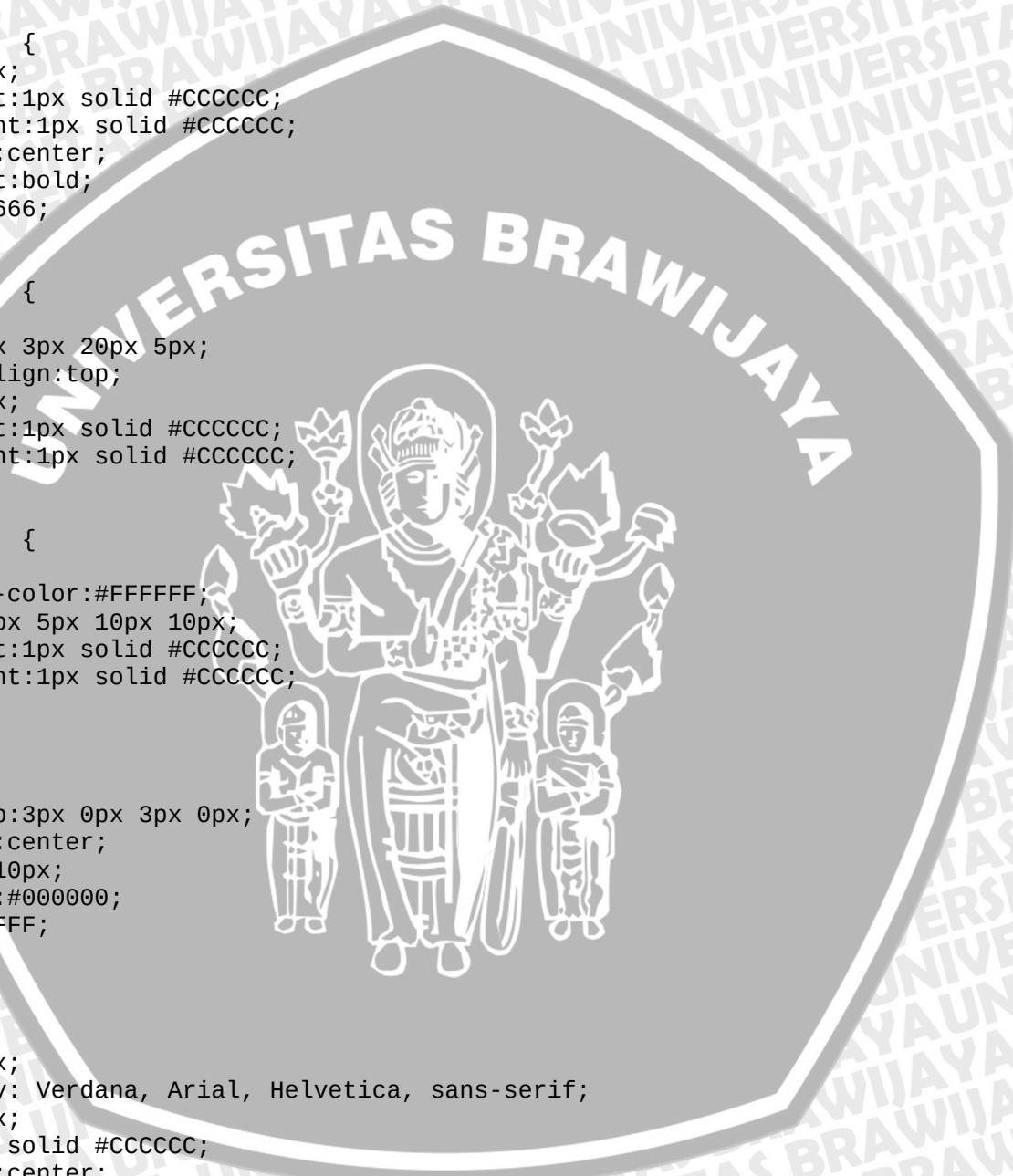
.utama {
margin:0;
padding:0px 3px 20px 5px;
vertical-align:top;
width:478px;
border-left:1px solid #CCCCCC;
border-right:1px solid #CCCCCC;
}

.screen {
margin:0;
background-color:#FFFFFF;
padding:10px 5px 10px 10px;
border-left:1px solid #CCCCCC;
border-right:1px solid #CCCCCC;
}

.info {
margin:0;
padding-top:3px 0px 3px 0px;
text-align:center;
font-size:10px;
background:#000000;
color:#FFFFFF;
}

.toolbar {
margin:0;
height:30px;
font-family: Verdana, Arial, Helvetica, sans-serif;
padding:3px;
border:1px solid #CCCCCC;
text-align:center;
vertical-align:middle;
}

.control {
margin:0;
padding:0;
border:1px #999999 solid;
```




```

}
.ganti {
margin:0;
padding:0;
background:#000000;
text-align:center;
}
.text a:link,.text a:visited{font-size:16px; font-family:verdana; font-
weight:bold; padding:5px; padding-right:10px; float:right; text-
decoration:none; color:#666666;}

```

