

**APLIKASI MULTIMEDIA STREAMING SEBAGAI
MEDIA KULIAH ONLINE**

SKRIPSI

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

UNIVERSITAS BRAWIJAYA



Disusun oleh:

GALIH KUSUMO B.

NIM. 0210630053

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2009

APLIKASI MULTIMEDIA STREAMING SEBAGAI MEDIA KULIAH ONLINE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

GALIH KUSUMO B.

NIM. 0210630053

Disahkan dan disetujui oleh dosen pembimbing :

Pembimbing I

Pembimbing II

Rudy Yuwono, ST., M.Sc

NIP. 132 206 465

Rusmi Ambarwati, ST., MT.

NIP. 132 258 188

APLIKASI MULTIMEDIA STREAMING SEBAGAI MEDIA KULIAH ONLINE

Disusun oleh :

GALIH KUSUMO B.

NIM. 0210630053

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 14 Juli 2009

Majelis Penguji :

Ir. Erfan Achmad Dahlan

NIP. 131 124 663

Ir. Wahyu Adi Prijono, MT

NIP. 131 759 602

Gaguk Asmungi, ST

NIP. 132 206 528

Mengetahui :

Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M.Kom

NIP. 131 879 033



PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan kepada *Allah Subhanahu wa Ta'ala* yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Aplikasi *Multimedia Streaming* Sebagai Media Kuliah *Online*” yang diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik.

Penulis ingin mengucapkan terima kasih kepada beberapa pihak yang telah membantu dan mendukung dalam penyelesaian skripsi ini, yaitu:

1. Ibu, Bapak dan segenap keluarga tercinta.
2. Bapak Ir. Heru Nurwarsito., M.Kom. selaku Ketua Jurusan Teknik Elektro.
3. Bapak Rudy Yuwono., ST., MSc. selaku Sekretaris Jurusan Teknik Elektro dan Dosen pembimbing skripsi.
4. Ibu Rusmi Ambarwati, ST., MT selaku dosen pembimbing skripsi.
5. Ibu Ir. Endah Budi P., MT selaku Ketua Kelompok Dosen dan Keahlian Teknik Telekomunikasi.
6. Rekan-rekan TPTIFT, Ristie, Kafet dan dan Teknik Elektro, khususnya angkatan 2002.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan sehingga saran dan kritik membangun sangat penulis harapkan. Akhirnya, semoga skripsi ini bermanfaat untuk kita semua, khususnya Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Malang, Oktober 2009

Penulis



DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
DAFTAR SINGKATAN	xi
ABSTRAKSI	xiii
BAB I	1
PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Kontribusi Penelitian	4
1.6 Sistematika Penulisan	4
BAB II	6
TINJAUAN PUSTAKA	6
2.1 Umum	6
2.2 Konsep Jaringan Komputer	6
2.2.1 <i>Local Area Network (LAN)</i>	7
2.2.2 <i>Metropolitan Area Network (MAN)</i>	8
2.2.3 <i>Wide Area Network (WAN)</i>	8
2.3 <i>Open System Interconnection (OSI) Layer</i>	9
2.4 Jaringan komputer FTUB	10
2.5 <i>Transport Layer</i>	13
2.5.1 <i>Real time Transport Protocol (RTP)</i>	13
2.5.1.1 Aktifitas RTP pada pengirim	14
2.5.1.2 Aktifitas RTP pada penerima	16
2.5.2 <i>User Datagram protocol (UDP/IP)</i>	17



2.6	IP Multicast	18
2.6.1	Pengalamatan Multicast	19
2.6.2	Routing Multicast	20
2.6.3	Protocol Independet Multicast - Sparse Mode	20
2.6.3.1	Internet Group Management Protocol	21
2.6.3.2	Rendezvous Point (RP)	23
2.6.3.3	Reverse Path Forwarding (RPF)	23
2.6.3.4	Mekanisme Protokol Routing PIM SM	24
2.7	Multimedia Streaming	27
2.8	Kompresi multimedia	28
2.8.1	Kompresi Audio	29
2.8.2	Kompresi Video	32
2.8.3	Kompresi MPEG	34
2.8.3.1	MPEG Audio	34
2.8.3.2	MPEG Video	35
2.8.3.3	MPEG Transport Stream	36
2.9	Aplikasi Web	37
2.9.1	HTML	37
2.9.2	PHP (PHP Hypertext Preprocessor)	38
2.10	Performansi Sistem	39
2.10.1	Bandwidth Multimedia Streaming	39
2.10.2	Throughput	40
2.10.3	Kecepatan Transmisi	40
2.10.4	Kapasitas Jaringan	41
2.10.5	Packet Loss	42
2.10.6	Delay End to end	43
2.10.6.1	Delay Codec	43
2.10.6.2	Delay Enkapsulasi dan Dekapsulasi	44
2.10.6.3	Delay Transmisi	45
2.10.6.4	Delay Propagasi	46
2.10.6.5	Delay Antrian	47
2.10.6.6	Delay Jitter	48

BAB III	50
METODOLOGI	50
3.1 Studi Literatur.....	50
3.2 Perancangan sistem.....	50
3.3 Pengujian.....	51
3.4 Pengambilan data.....	51
3.5 Analisis.....	52
3.6 Kesimpulan.....	52
BAB IV	53
PERANCANGAN APLIKASI MULTIMEDIA STREAMING SEBAGAI MEDIA KULIAH ONLINE	53
4.1 Umum.....	53
4.2 Perancangan Sistem Kuliah <i>Online</i>	53
4.2.1 Cara Kerja Aplikasi Kuliah <i>Online</i>	53
4.2.2 Perancangan Perangkat Keras Sistem Kuliah <i>Online</i>	55
4.2.3 Perancangan Perangkat Lunak Sistem Kuliah <i>Online</i>	55
4.2.3.1 Perancangan Halaman Web Kuliah <i>Online</i>	56
4.3 Perancangan Metode <i>Multimedia Streaming</i>	57
4.4 Perancangan <i>Multicast Routing</i> pada jaringan FTUB.....	58
4.4.1 Perancangan Perangkat Keras Jaringan.....	59
4.4.2 Perancangan Perangkat Lunak Jaringan.....	59
4.4.3 Perancangan <i>Multicast Routing</i> PIM SM.....	60
4.4.3.1 Analisa Jaringan Eksisting.....	61
4.4.3.2 Perencanaan topologi jaringan <i>multicast</i>	61
4.4.3.3 Instalasi XORP pada <i>router</i>	63
4.4.3.4 Penerapan <i>Multicast Routing</i> PIM SM.....	63
BAB V	68
PENGUJIAN DAN ANALISA	68
5.1 Pengujian.....	68

5.1.1	Pengujian <i>Multicast Routing</i>	68
5.1.1.1	Pengujian <i>routing PIM SM</i>	70
5.1.1.2	Pengujian <i>PIM Neighbors</i>	70
5.1.1.3	Pengujian <i>Tabel Routing Multicast</i>	72
5.1.1.4	Pengujian <i>Rendezvous point State</i>	72
5.1.1.5	Pengujian <i>Koneksi Multicast</i>	72
5.1.2	Pengujian <i>Multimedia Streaming</i>	74
5.1.2.1	Pengujian <i>Multimedia Streaming dengan Metode Unicast</i>	75
5.1.2.2	Pengujian <i>Multimedia Streaming Dengan Metode Multicast</i> ..	78
5.1.2.3	Pengujian <i>Penggunaan Bandwidth Multimedia Streaming</i>	81
5.1.3	Pengujian <i>Aplikasi Kuliah Online</i>	84
5.1.3.1	Pengujian <i>Koneksi Host Ke Server</i>	84
5.1.3.2	Pengujian <i>Halaman Web Kuliah Online</i>	85
5.1.4	Pengujian <i>Sistem Kuliah Online</i>	86
5.1.5	Pengujian <i>Mean Opinion Square</i>	93
5.1.5.1	Pengujian <i>kualitas audio</i>	94
5.1.5.2	Pengujian <i>kualitas video</i>	95
5.1.5.3	Pengujian <i>pengaruh delay</i>	96
5.2	<i>Analisis</i>	96
5.2.1	<i>Analisa Troughput</i>	97
5.2.2	<i>Analisa Bandwidth Multimedia Streaming</i>	98
5.2.3	<i>Analisa Kapasitas Jaringan</i>	99
5.2.4	<i>Analisa Packet Loss</i>	102
5.2.5	<i>Analisis Delay Enkapsulasi Dan Dekapsulasi</i>	103
5.2.6	<i>Analisis Delay Transmisi</i>	104
5.2.7	<i>Analisis Delay Propagasi</i>	105
5.2.8	<i>Analisis Delay Antrian</i>	106
5.2.9	<i>Analisa Delay Jitter</i>	107
5.2.10	<i>Analisis Delay End-To-End</i>	109
BAB VI		111
PENUTUP		111

6.1	Kesimpulan	111
6.2	Saran.....	113
DAFTAR PUSTAKA		114
LAMPIRAN.....		115



DAFTAR GAMBAR

Gambar 2.1 *Network Topologies* 8

Gambar 2.2 Jaringan komputer FTUB (Eksisting) 12

Gambar 2.3 RTP *sender* 14

Gambar 2.4 Standar IETF dan ITU untuk *protocol transport Audio/Video* pada jaringan Sumber: Collins Perkins, 2003 14

Gambar 2.5 Diagram RTP *sender* 15

Gambar 2.6 Diagram RTP *receiver* 16

Gambar 2.7 Format paket UDP 18

Gambar 2.8 Perbandingan antara *multicast* (atas) dan *unicast* (bawah) 19

Gambar 2.9 Mekanisme IGMP untuk *join group* 22

Gambar 2.10 Struktur paket data IGMPv2 22

Gambar 2.11 Keadaan awal jaringan 25

Gambar 2.12 Proses *join Receiver 1* ke RP 26

Gambar 2.13 Proses pembentukan RPT 26

Gambar 2.14 Proses perpindahan dari RPT ke SPT 27

Gambar 2.15 Proses Konversi dan Kompresi Data 29

Gambar 2.16 Proses *Sampling* 30

Gambar 2.17 Proses *Quantizing* dan Coding 31

Gambar 2.18 Paket data multimedia *streaming* dengan enkapsulasi MPEG TS .. 37

Gambar 2.19 *Model Antrian Single Client* 47

Gambar 4.1 Diagram Alir Sistem Kuliah *Online* 54

Gambar 4.2 Halaman web Kuliah *Online* 56

Gambar 4.3 Blok Diagram proses *streaming* 58

Gambar 4.4 Diagram alir perancangan *multicast routing* 61

Gambar 4.5 Topologi jaringan *multicast* 62

Gambar 4.6 Proses kompilasi kernel FreeBSD 6.2 63

Gambar 4.7 Proses Instalasi XORP 1.4 63

Gambar 4.8 Diagram Alir penerapan *multicast routing PIM SM* 64

Gambar 5.1 Topologi Pengujian 69

Gambar 5.2 Pengujian PIM *Neighbors* pada Router MGW 70

Gambar 5.3 Ilustrasi <i>Neighbouring Routers</i>	71
Gambar 5.4 Hasil pengujian tabel <i>routing multicast</i>	72
Gambar 5.6 Pengujian <i>RP State</i> pada <i>Router Gunawijaya</i>	72
Gambar 5.7 Pengujian Koneksi <i>Multicast</i> pada <i>Host A</i> sebagai <i>Server</i>	73
Gambar 5.8 Pengujian Koneksi <i>Multicast</i> pada <i>Host B</i> sebagai <i>Client</i>	73
Gambar 5.9 Pengujian Koneksi <i>Multicast</i> pada <i>Host B</i> sebagai <i>Server</i>	74
Gambar 5.10 Pengujian Koneksi <i>Multicast</i> pada <i>Host B</i> sebagai <i>Server</i>	74
Gambar 5.11 Pemilihan <i>Capture Device</i> sebagai <i>source streaming</i>	75
Gambar 5.12 Pemilihan <i>device (input) streaming</i>	76
Gambar 5.13 Konfigurasi <i>output streamingunicast</i>	76
Gambar 5.14 Pemilihan <i>Network stream</i> sebagai <i>source streaming</i>	77
Gambar 5.15 Menentukan protokol dan alamat IP <i>streaming source</i>	77
Gambar 5.16 Hasil <i>streaming</i> dari <i>host A</i> ke <i>host B</i>	78
Gambar 5.17 Konfigurasi <i>output streaming multicast</i>	79
Gambar 5.18 Menentukan protokol dan alamat IP <i>streaming source</i>	80
Gambar 5.19 Hasil <i>streaming</i> dari <i>host A</i> ke <i>host B</i>	80
Gambar 5.20 Konfigurasi Pengujian Penggunaan <i>Bandwidth Unicast</i> dan <i>Multicast</i>	82
Gambar 5.21 <i>Streaming</i> dengan Metode <i>Unicast</i> dari <i>Host A</i> ke <i>Host B</i>	82
Gambar 5.22 <i>Streaming</i> dengan Metode <i>Unicast</i> dari <i>Host A</i> ke 4 <i>Host</i>	83
Gambar 5.23 Pengujian <i>Streaming</i> dengan Metode <i>Multicast</i> dari <i>Host A</i> ke <i>Host</i> <i>B</i>	83
Gambar 5.24 Pengujian <i>Streaming</i> dengan Metode <i>Multicast</i> dari <i>Host A</i> ke 4 <i>Host</i>	83
Gambar 5.25 Pengujian Halaman <i>Web Kuliah Online</i>	85
Gambar 5.26 Topologi pengujian Sistem Kuliah Online	86
Gambar 5.27 Screenshot <i>Wireshark</i>	88
Gambar 5.28 Hubungan jumlah hop dengan <i>Packet Loss</i>	103
Gambar 5.29 Hubungan jumlah hop dengan <i>Delay Jitter</i>	109

DAFTAR TABEL

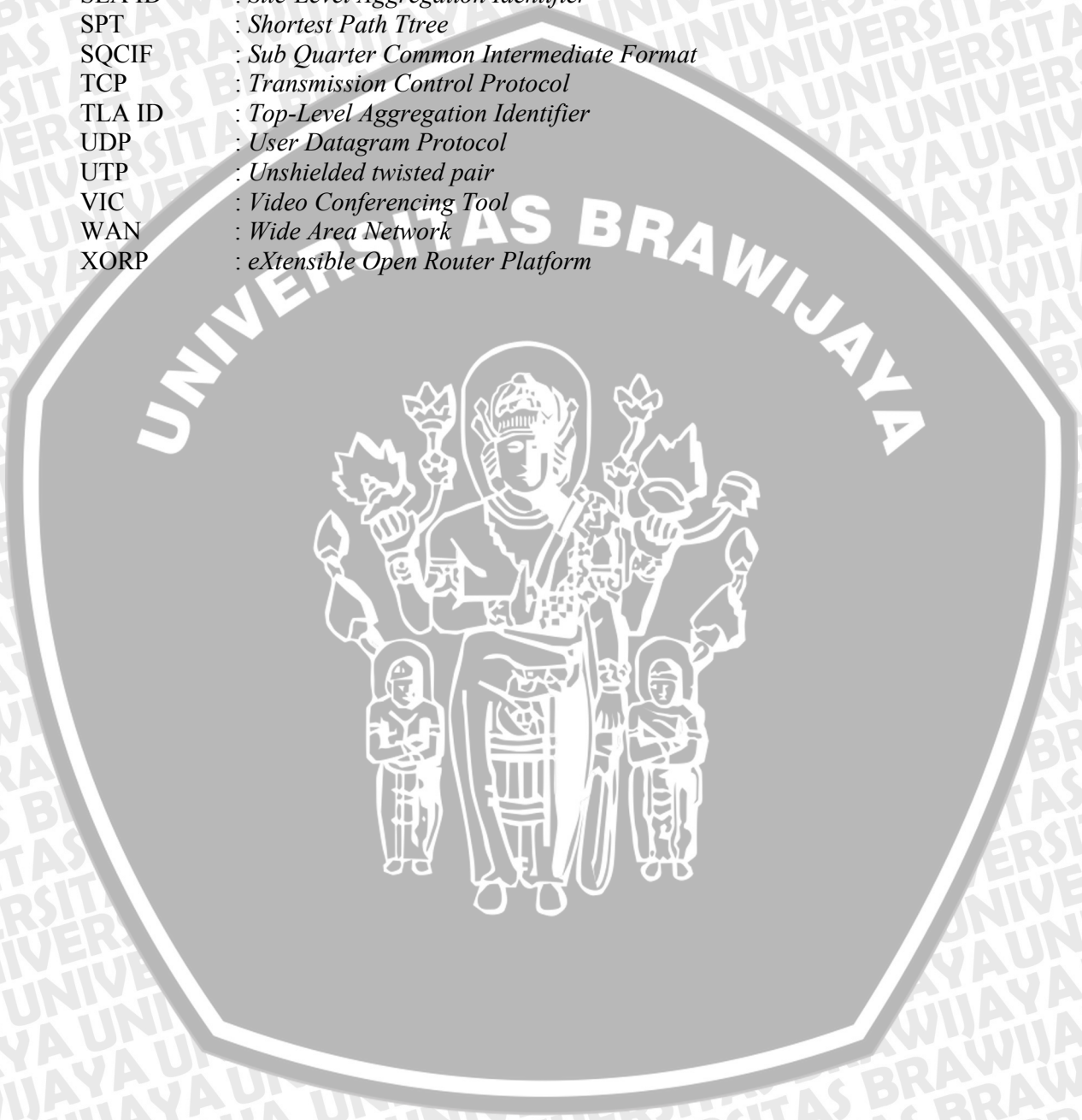
Tabel 2.1 (Open System Interconnection)	10
Tabel 2.2 Spesifikasi Jaringan komputer FTUB	11
Tabel 2.3 Alokasi IP <i>Address</i> jaringan komputer FTUB	11
Tabel 2.4 Type pesan IGMP di gambarkan pada tabel di bawah ini	22
Tabel 2.5 Standar Kompresi <i>Audio</i>	32
Tabel 2.6 Standar Kompresi <i>Video</i>	33
Tabel 2.7 Spesifikasi kompresi MPEG <i>Audio</i>	35
Tabel 2.8 Standar <i>Codec</i> untuk <i>video</i>	36
Tabel 2.9 <i>Delay</i> tipikal maksimum beberapa komponen <i>Fast Ethernet</i>	47
Tabel 5.1 Tabel Pengujian PIM <i>Neighbors</i> pada Tiap <i>Router</i>	71
Tabel 5.2 Hasil Pengujian <i>Streaming</i> dengan Metode <i>Unicast</i>	78
Tabel 5.3 Hasil Pengujian <i>Streaming</i> dengan Metode <i>Unicast</i>	81
Tabel 5.4 Spesifikasi <i>Hardware Host</i> untuk Pengujian <i>Bandwidth</i>	81
Tabel 5.5 Hasil Pengujian <i>Bandwidth Multimedia Streaming</i> dengan Metode <i>Unicast</i>	82
Tabel 5.6 Hasil Pengujian <i>Bandwidth Multimedia Streaming</i> dengan Metode <i>Multicast</i>	83
Tabel 5.7 Pengujian Koneksi <i>Host</i> ke <i>Server</i>	84
Tabel 5.8 Lokasi <i>host</i> dan IP Address yang digunakan	87
Tabel 5.9 Hasil capture data pada <i>host A1</i> dengan filter RTP untuk paket data dari <i>multicast address host A2</i>	89
Tabel 5.10 Hasil capture data pada <i>host A2</i> dengan filter RTP untuk paket data dari <i>multicast address host A1</i>	89
Tabel 5.11 Hasil capture data pada <i>host B1</i> dengan filter RTP untuk paket data dari <i>multicast address host B2</i>	89
Tabel 5.12 Hasil capture data pada <i>host B2</i> dengan filter RTP untuk paket data dari <i>multicast address host B1</i>	90
Tabel 5.13 Hasil capture data pada <i>host C1</i> dengan filter RTP untuk paket data dari <i>multicast address host C2</i>	90

Tabel 5.14 Hasil capture data pada <i>host</i> C2 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> C1	90
Tabel 5.15 Hasil capture data pada <i>host</i> D1 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> D2	91
Tabel 5.16 Hasil capture data pada <i>host</i> D2 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> D1	91
Tabel 5.17 Hasil capture data pada <i>host</i> E1 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> E2.....	91
Tabel 5.18 Hasil capture data pada <i>host</i> E2 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> E1.....	92
Tabel 5.19 Hasil capture data pada <i>host</i> F1 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> F2.....	92
Tabel 5.20 Hasil capture data pada <i>host</i> F2 dengan filter RTP untuk paket data dari <i>multicast</i> address <i>host</i> F1.....	92
Tabel 5.21 Tabulasi hasil pengujian.....	93
Tabel 5.22 Indeks kualitas <i>audio</i> dan <i>video</i>	94
Tabel 5.23 Indeks pengaruh delay	94
Tabel 5.24 Hasil analisa kualitas <i>audio</i> dengan metode MOS.....	94
Tabel 5.25 Hasil analisa kualitas <i>video</i> menggunakan MOS.....	95
Tabel 5.26 Hasil analisa pengaruh delay dengan metode MOS.....	96
Tabel 5.27 Troughput pada <i>host</i>	98
Tabel 5.28 <i>Bandwidth</i> multimedia <i>streaming</i> pada <i>host</i>	99
Tabel 5.29 <i>Packet</i> loss pada <i>host</i>	102
Tabel 5.30 Hubungan antara banyaknya hop dengan nilai <i>packet</i> loss.....	103
Tabel 5.31 Delay jitter pada <i>host</i>	108
Tabel 5.32 Hubungan delay jitter dengan jumlah hop	108

DAFTAR SINGKATAN

BSR	: <i>Bootstrap Router</i>
C-BSR	: <i>Candidat BSR</i>
C-RP	: <i>Candidat-RP</i>
CIDR	: <i>Class-less Interdomain Routing</i>
CIF	: <i>Common Intermediate Format</i>
DHCP	: <i>Dynamic Host Configuration Protocol</i>
DR	: <i>Designated Router</i>
DVMRP	: <i>Distance Vector Multicast Routing Protokol</i>
FIFO	: <i>First In First Out</i>
IANA	: <i>Internet Assigned Name Authority</i>
ICMP	: <i>Internet Control Message Protocol</i>
IETF	: <i>Internet Engineering Task Force</i>
IGMP	: <i>Internet Group Manajemen Protocol</i>
IPv4	: <i>Internet Protocol version 4</i>
IPv6	: <i>Internet Protocol version 6</i>
ISDN	: <i>Integrated Services Digital Network</i>
ISO	: <i>International Organization for Standardization</i>
ISP	: <i>Internet Service Provider</i>
ITU	: <i>International Telecommunication Union</i>
JPEG	: <i>Joint Photographic Experts Group</i>
LAN	: <i>Local Area Network</i>
MAC	: <i>Mandatory Access Control</i>
MFEA	: <i>Multicast Forwarding Engine Abstraction</i>
MIPS	: <i>Mega Instruction per Second</i>
MLD	: <i>Multicast Listener Discovery</i>
MOS	: <i>Mean Opinion Scorel</i>
MOSPF	: <i>Multicast Open Sortheast Path First</i>
MPEG	: <i>Moving Picture Experts Group</i>
MRIB	: <i>Multicast Information Base</i>
MTU	: <i>Maximum Transmission Unit</i>
NAT	: <i>Network Address Translation</i>
NIC	: <i>Network Interface Card</i>
NLA ID	: <i>Next Level Aggregation Identifier</i>
NOS	: <i>Network Operating System</i>
NTP	: <i>Network Time Protocol</i>
OSPF	: <i>Open Sortheast Path First</i>
PCM	: <i>Pulse Code Modulation</i>
PIM-DM	: <i>Protocol-Independent Multicast, Dense Mode</i>
PIM-SM	: <i>Protocol Independent Multicast Sparse Mode</i>
PQ	: <i>Priority Queueing</i>
PSNR	: <i>Peak signal-to-noise ratio</i>
QCIF	: <i>Quarter Common Intermediate Format</i>
QoS	: <i>Quality of Services</i>
RAT	: <i>Robust Audio Tool</i>

- RFC : *Request for Comments*
- RP : *Rendezvous Point*
- RPT : *Rendezvous Point Tree*
- RTP : *Real-time Transport Protocol*
- SLA ID : *Site-Level Aggregation Identifier*
- SPT : *Shortest Path Ttree*
- SQCIF : *Sub Quarter Common Intermediate Format*
- TCP : *Transmission Control Protocol*
- TLA ID : *Top-Level Aggregation Identifier*
- UDP : *User Datagram Protocol*
- UTP : *Unshielded twisted pair*
- VIC : *Video Conferencing Tool*
- WAN : *Wide Area Network*
- XORP : *eXtensible Open Router Platform*



ABSTRAKSI

GALIH KUSUMO B. Juli 2009. *Aplikasi Multimedia Streaming Sebagai Media Kuliah Online*. Dosen Pembimbing: Rudy Yuwono, ST., MSc. Dan Rusmi Ambarwati, ST., MT.

Pemanfaatan Information and Communication Technology (ICT) pada umumnya dan teknologi jaringan komputer pada khususnya, telah membuat perkembangan yang signifikan dalam metode belajar mengajar dalam dunia pendidikan. Penggunaan internet sebagai media pembelajaran sudah bukan hal yang asing lagi. Perpustakaan elektronik, website sekolah, pusat materi belajar, hingga kuliah virtual dapat diaplikasikan di dunia pendidikan yang berbasis pada ICT. Dengan perkembangan dunia pendidikan yang semakin pesat, maka kebutuhan terhadap aplikasi – aplikasi teknologi yang mendukung kegiatan belajar mengajar yang berbasis ICT adalah suatu hal yang mutlak.

Kuliah online adalah suatu bentuk perkuliahan, di mana dosen dan mahasiswa tidak perlu bertatap muka secara langsung, namun secara virtual, atau melalui media telekomunikasi tertentu. Dalam dunia ICT hal ini lazim disebut dengan video conference. Video conference adalah salah satu bentuk multimedia streaming yang digunakan sebagai media interaksi antara dua atau lebih lokasi yang berjauhan melalui media suara dan gambar yang ditransmisikan secara simultan dan realtime. Pada kuliah online dua lokasi tersebut adalah lokasi dosen dan kelas. Skripsi ini akan membahas aplikasi multimedia streaming pada jaringan multicast di FTUB yang akan digunakan sebagai media kuliah online. Sistem kuliah online ini nantinya akan menggunakan teknologi multicast. Agar sistem dapat berjalan, maka pada Jaringan Intranet FTUB harus memiliki kemampuan multicast routing. Analisa sistem dititik beratkan pada keberhasilan routing multicast dan performansi jaringan pada saat sistem berjalan. Pengujian sistem dilakukan dengan men-streaming 256 kbps video (MPEG 4) dan 64 kbps audio (MPEG 4) dari masing masing host. Protokol RTP (Real-time Transport Protocol) digunakan sebagai kontrol pengiriman video dan audio real-time.

Dari hasil pengujian didapat bahwa bandwidth rata-rata yang dibutuhkan untuk melakukan proses streaming dari satu multicast source ke receiver adalah 468,06. Delay end-to-end didapat sebesar 104,71 ms dan paket loss total sebesar 0.518%. Untuk proses streaming dari satu source ke empat receiver, metode streaming multicast membutuhkan bandwidth lebih sedikit daripada metode streaming unicast.

Kata kunci : *Streaming, Conference, Multicast, Virtual Learning*

BAB I PENDAHULUAN

1.1 Latar belakang

Sebagai jawaban atas biaya pendidikan yang murah namun berkualitas, pendayagunaan ICT atau *Information and Communication Technology* merupakan sebuah solusi bagi sistem pendidikan di Indonesia. Motto pendidikan berbasis ICT pun saat ini sedang gencar – gencarnya digaungkan di dunia pendidikan kita. Penggunaan teknologi informasi di bidang pendidikan kini juga sudah sangat meluas. Berbagai jenis aktivitas, mulai dari administrasi hingga kegiatan belajar mengajar kini banyak yang berbasis pada teknologi informasi.

Banyaknya manfaat dari ICT pada umumnya dan teknologi jaringan komputer pada khususnya, telah membuat perkembangan yang signifikan dalam metode belajar mengajar dalam dunia pendidikan. Penggunaan internet sebagai media pembelajaran sudah bukan hal yang asing lagi. Perpustakaan elektronik, *website* sekolah, pusat materi belajar, hingga kuliah *virtual* dapat diaplikasikan di dunia pendidikan yang berbasis pada ICT. Dengan perkembangan dunia pendidikan yang semakin pesat, maka kebutuhan terhadap aplikasi – aplikasi yang mendukung kegiatan belajar mengajar yang berbasis ICT adalah suatu hal yang mutlak.

Sejak tahun 2003 Fakultas Teknik Universitas Brawijaya (FTUB) telah menggunakan teknologi informasi untuk menunjang kegiatan pendidikan. Hal ini diawali dengan pembangunan infrastruktur jaringan komputer yang mencakup semua jurusan di lingkungan FTUB dan pembuatan website resmi FTUB. Saat ini jaringan komputer FTUB menggunakan teknologi IPv4 dan *Fast Ethernet* (100Mbps), dengan topologi *tree*, dimana media yang digunakan antara lain, kabel UTP, *fiber optic*, dan WiFi. Sedangkan untuk *router*, jaringan FTUB menggunakan PC yang difungsikan sebagai *router*. Jaringan komputer FTUB masih menggunakan protokol IPv4, karena teknologi ini dirasa masih mencukupi.

Multimedia streaming merupakan metode untuk memainkan suatu data multimedia dengan cara men-*download* data itu dari sebuah komputer *server*

(langsung/live maupun tidak langsung/*prerecorded*) kemudian memainkannya tanpa menunggu seluruh data tersebut selesai di-*download*. Karakteristik *multimedia streaming* antara lain data yang dikirimkan bersifat *continous* dan *delay* rendah, sehingga membutuhkan data rate yang tinggi. Bentuk aplikasi *multimedia streaming* pun sangat beragam. Mulai dari yang berbasis *audio* seperti *IP Telephony* dan VoIP, maupun yang berbasis *audio/video* seperti teleconference, IPTV dan *TV Over IP*. Saat ini teknologi *multimedia streaming* banyak dikembangkan dengan teknologi *multicast*. *Multicast* merupakan sebuah metode yang digunakan untuk mengatasi tingginya penggunaan sumber daya jaringan komputer pada proses *multimedia streaming*. Teknologi *multicast* memiliki banyak keunggulan jika dibandingkan dengan tipe komunikasi lainnya semisal *unicast* ataupun *broadcast*, terutama dalam hal efisiensi sumber daya jaringan komunikasi. Pada teknologi *multicast*, client yang mengakses *content multimedia streaming* tidak langsung mendownload *content* dari *streaming source*, tetapi pada *router* yang langsung terhubung dengan client tersebut. Hal ini bisa terjadi karena *router* melakukan replikasi paket data yang dikirim sehingga transfer data antar *router* yang berfungsi sebagai *backbone* jaringan menjadi lebih rendah.

Teknologi *multimedia streaming* bersifat sangat fleksibel sehingga dapat dimanfaatkan sebagai media yang dapat menunjang kegiatan belajar mengajar contohnya kuliah. Seiring dengan berkembangnya dunia pendidikan dan teknologi, aplikasi *multimedia streaming* yang berjalan pada jaringan *multicast* akan sangat membantu kegiatan belajar mengajar. Skripsi ini akan membahas aplikasi *multimedia streaming* pada jaringan *multicast* di FTUB yang akan digunakan sebagai media kuliah *online*.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat ditentukan suatu rumusan masalah yaitu:

1. Bagaimana perencanaan dan penerapan jaringan *multicast* pada jaringan komputer FTUB.
2. Bagaimana penerapan *Protocol Independent Multicast-Sparse Mode* (PIM-SM) sebagai protokol *multicast routing* pada *router multicast*.
3. Bagaimana merancang dan membuat aplikasi *multimedia streaming* yang berfungsi sebagai media kuliah *online*.
4. Bagaimana performansi jaringan ditinjau dari kualitas layanan atau *Quality of Service* (QoS) pada komunikasi *audio* dan *video* antara lain *bandwidth* yang dibutuhkan, *delay end-to-end*, *packet loss*, dan *throughput*.

1.3 Batasan Masalah

Berdasarkan permasalahan yang telah disebutkan di atas, maka penulisan skripsi ini dibatasi pada hal-hal sebagai berikut :

1. Perancangan dan penerapan aplikasi pada *server* menggunakan *open source software*.
2. Perancangan dan penerapan sistem dilakukan pada jaringan komputer FTUB.
3. Protokol yang digunakan untuk *multicast routing* menggunakan *protocol* PIM-SM (RFC 2362).
4. Menggunakan *Real time Transport Protokol* (RTP) sebagai protokol level kontrol untuk *multimedia real-time*. (RFC 1889, 3550).
5. Menggunakan teknologi HTML, CSS dan PHP untuk *client interface* pada *server*.

1.4 Tujuan

Tujuan penulisan skripsi ini adalah untuk merancang dan mengimplementasikan aplikasi *multimedia streaming* sebagai media kuliah *online*.

1.5 Kontribusi Penelitian

Penulisan skripsi ini diharapkan dapat memberikan kontribusi:

1. Bagi pembaca, diharapkan dapat memberikan pengetahuan tentang teknologi *multicast* menggunakan PIM-SM untuk *multimedia streaming*.
2. Bagi Fakultas Teknik Universitas Brawijaya, diharapkan hasil skripsi ini dapat membantu kegiatan perkuliahan.
3. Bagi penulis, mampu memahami proses penerapan *multicast* untuk *multimedia streaming* menggunakan PIM-SM dan RTP.

1.6 Sistematika Penulisan

Sistematika penulisan dan gambaran secukupnya yang terdapat dalam setiap bab adalah sebagai berikut :

BAB I Merupakan pendahuluan yang memuat latar belakang, rumusan masalah, ruang lingkup, tujuan penulisan, kontribusi penelitian dan sistematika penulisan.

BAB II Membahas mengenai arsitektur jaringan komputer FTUB, protokol UDP dan RTP, serta membahas mengenai prinsip kerja *multicast* pada jaringan komputer, PIM SM *routing*, serta aplikasi *multimedia streaming* .

BAB III Memuat metodologi penulisan yang menjelaskan tahapan pengerjaan skripsi.

BAB IV Membahas perancangan jaringan *multicast* pada jaringan komputer FTUB, perancangan aplikasi kuliah *online* dan perancangan *multicast routing* dengan PIM SM.

BAB V Membahas pengujian dan analisis performansi jaringan untuk penerapan aplikasi *multimedia streaming* pada jaringan komputer FTUB, meliputi: pengujian dan perhitungan *bandwidth* yang dibutuhkan, kapasitas jaringan, *delay*, *paket loss*, dan *troughput* .

BAB VI Penutup, memuat kesimpulan dan saran.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1 Umum

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* serta tinjauan pustaka tentang jaringan komputer FTUB.

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 telekomunikasi terutama 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.

Salah satu inovasi dalam sistem komunikasi data adalah *audio/video conference* dengan menggunakan IP *multicast* yang memungkinkan beberapa orang bisa saling berinteraksi dalam waktu bersamaan atau *real-time*. IP *multicast* adalah metode pengiriman data yang ditujukan kepada *client* dengan replikasi data pada *router* yang telah bergabung sehingga penggunaan *bandwidth* menjadi lebih efisien.

2.2 Konsep Jaringan Komputer

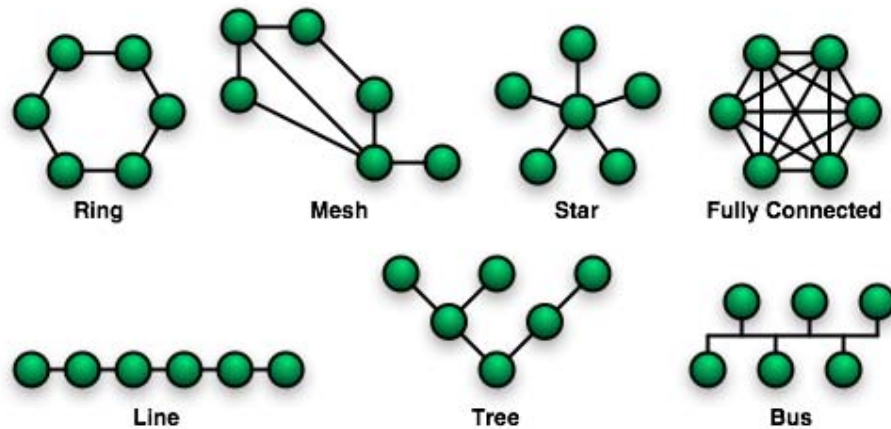
Jaringan komputer adalah kumpulan sekelompok komputer otonom yang saling berhubungan antar satu dengan lainnya menggunakan protokol komunikasi

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)

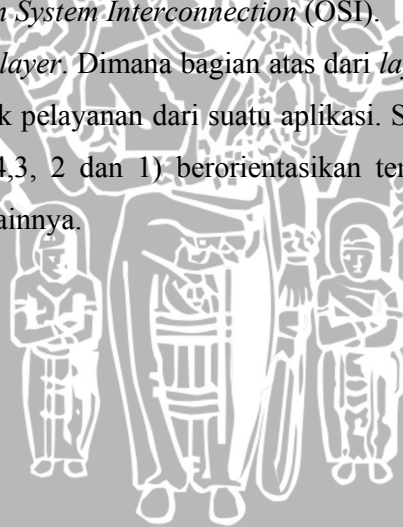
Wide Area Network adalah sebuah jaringan yang memiliki jangkauan radius yang sangat luas, karena jangkauannya tidak lagi mencakup suatu kota, bahkan lebih yaitu bisa mencakup lintas negara atau lintas benua. Pada sebagian besar WAN, komponen yang dipakai dalam berkomunikasi terdiri dari dua komponen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

2.3 *Open System Interconnection (OSI) Layer*

Protokol adalah aturan dalam komunikasi data yang berguna untuk menjaga agar pengiriman data dapat berlangsung. Dalam komunikasi data di perlukan 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 Layer

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 pengalaman. 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 Jaringan komputer FTUB

Sejak tahun 2003 Fakultas Teknik Universitas Brawijaya (FTUB) telah menggunakan teknologi informasi untuk menunjang kegiatan pendidikan. Hal ini diawali dengan pembangunan infrastruktur jaringan komputer yang mencakup semua jurusan di lingkungan FTUB dan pembuatan website resmi FTUB. Saat ini jaringan komputer FTUB berbasis pada teknologi *Fast Ethernet* yang memiliki kecepatan 100Mbps, dengan topologi *tree*, dimana media yang digunakan untuk jaringan *backbone* antara lain, kabel UTP dan fiber optic, sedangkan media yang digunakan untuk akses *client* adalah kabel UTP dan WiFi.

Jaringan komputer FTUB mencakup seluruh gedung – gedung yang berada di lingkungan FTUB. Pada tiap gedung terdapat titik akses utama berupa *switch* atau *router*. *Router* yang digunakan saat ini adalah *PC router*. *PC Router* adalah PC yang difungsikan sebagai *router* dengan menambahkan aplikasi – aplikasi tertentu. Spesifikasi Jaringan komputer FTUB saat ini adalah sebagai berikut :

Tabel 2.2 Spesifikasi Jaringan komputer FTUB

Media	Kabel UTP CAT 5E <i>Fiber Optic</i>
Switch	Switch yang digunakan dari berbagai jenis merk dengan kecepatan 10/100Mbps dan memenuhi standar IEEE 802.3 (10T <i>Ethernet</i>), 802.3u (100TX <i>Fast Ethernet</i>), 802.3x (Flow Control)
Router	Prosesor Pentium III (667Mhz) Memori 128 MB NIC 3Com dan Allied Telesis (<i>Fast Ethernet</i>) <i>Hardisk</i> IBM 4G Sistem Operasi FreeBSD 6.2

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

Pembagian alokasi IP Address dan Subnet pada jaringan komputer FTUB berdasarkan lokasi dan jurusan. Pada tabel 2.3 ditunjukkan Alokasi penggunaan IP Address pada jaringan komputer FTUB.

Tabel 2.3 Alokasi IP Address jaringan komputer FTUB

No	Zona	Subnet	Gateway
1	Net Ristie	172.17.67.32/27	172.17.67.41
2	Net Elektro 2 (Rekording Elektro)	172.17.63.128/25	172.17.63.129
3	Net Elektro 3 (Gedung Kuliah)	172.17.67.64/27	172.17.67.65
4	Net Mesin	172.17.62.0/24	172.17.62.41
5	Net Elektro 1 (GBE)	172.17.63.0/25	172.17.63.1
6	Net Pengairan	172.17.64.0/24	172.17.64.41
7	Net Sipil	172.17.61.0/24	172.17.61.41
8	Net DT	172.17.69.0/24	172.17.69.41
9	Net Arsitek	172.17.65.0/24	172.17.65.129
10	Net Dekanat	172.17.60.0/24	172.17.60.41
11	Net PWK	172.17.66.0/24	172.17.66.129
12	Net TPTIFT	172.17.68.0/24	172.17.68.129
13	Net HMP	172.17.67.128/27	172.17.67.129

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

2.5 *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.

2.5.1 *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 awalnya 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*.

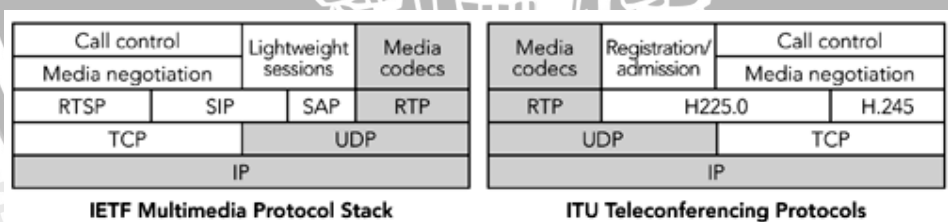
RTP dikembangkan oleh *Audio/Video Transport working group* oleh IETF dan telah diadopsi oleh ITU sebagai bagian dari seri rekomendasi dari H.323, serta badan standar Internasional lainnya. Disamping RTP, sebuah sistem yang lengkap biasanya membutuhkan protokol yang lain dan standar sesi

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 memperlakukan bagaimana *session* bernegosiasi, dan terlepas dari dasar *transport* jaringan.

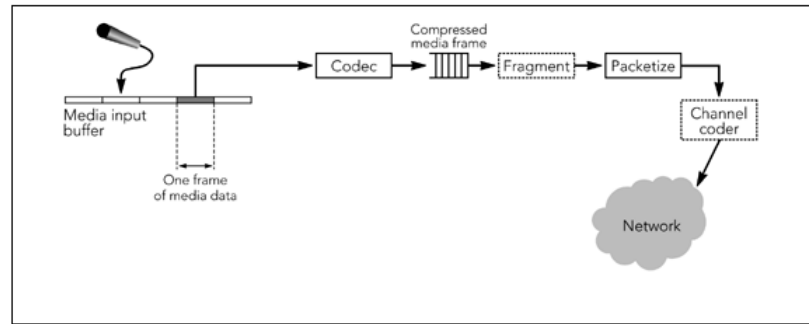


Gambar 2.4 Standar IETF dan ITU untuk *protocol transport Audio/Video* pada jaringan
 Sumber: Collins Perkins, 2003

2.5.1.1 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

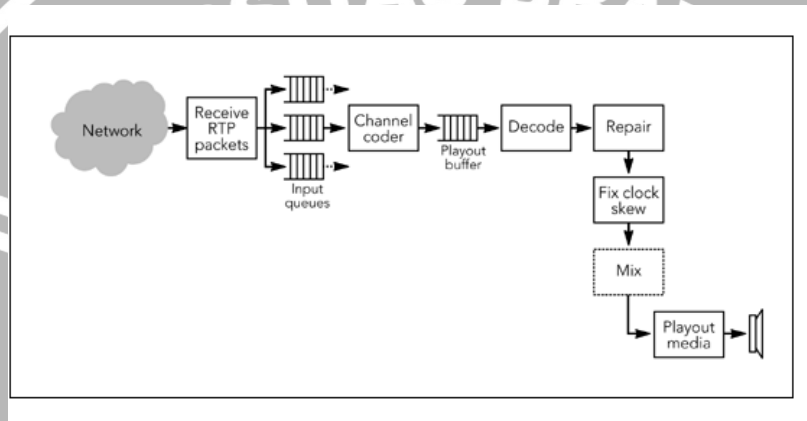
Media data yang tak terkompres *audio/video* ditangkap ke dalam *buffer*, dari *frame* dikompres yang diproduksi. *Frame* mungkin diekodekan pada beberapa jalan tergantung dari algoritma pemrosesan yang digunakan, dan *frame* yang terencode boleh di gantungkan pada awal atau akhir data.

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.

2.5.1.2 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*, mengizinkan 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* membutuhkan 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 diobservasi 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.

2.5.2 *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

2.6 IP Multicast

IP Multicast sebuah teknik yang digunakan untuk komunikasi data satu ke banyak (*one to many*). Untuk aplikasi yang melibatkan proses komunikasi satu-ke banyak titik, seperti aplikasi *video/audio conference*, teknologi *multicast* jelas mempunyai banyak keunggulan daripada dengan sistem *unicast* atau *broadcast*, terlebih dalam kondisi cakupan area yang luas. *Multicast* memungkinkan sebuah *node* mengirim paket yang dapat menjangkau banyak penerima dalam jaringan hanya dengan mengirimkan paket ke alamat tertentu yaitu alamat grup *multicast*. Disamping hal tersebut, dengan berbagai karakteristik lainnya teknologi *IP multicast* ini akan menawarkan biaya yang relatif rendah dan masih ditambah dengan fitur-fitur lainnya yang membawa proses komunikasi menjadi lebih efektif dan efisien. Selain itu pada umumnya untuk aplikasi komunikasi *one to many* atau *many to many* memerlukan manajemen dan kontrol yang spesifik. Saat ini aplikasi *IP Multicast* lebih banyak dilakukan pada jaringan yang menggunakan teknologi IPv6.

Standar *IP Multicast service model* adalah sebagai berikut :

1. Mengacu pada RFC1112 (*Host Extension of IP Multicasting*).
2. Diagram IP ditransmisikan ke sebuah *group multicast*.
3. *Group Multicast* di identifikasikan dengan IP Address kelas D

4. *Member* dari sebuah *group multicast* bisa berada dimana saja.
5. *Member* melakukan *join* (bergabung) dan *leave/prune* (keluar) dari *group* dengan *memberikan* sinyal pada *router*.
6. *Sender* dan *receiver* terpisah, sehingga *sender* tidak perlu menjadi *member group*.
7. *Router* *listen* ke semua alamat *multicast* dan menggunakan *protocol routing multicast* untuk mengatur *group*. (IGMP, Internet Group Management Protocol, RFC2236).



Gambar 2.8 Perbandingan antara *multicast* (atas) dan *unicast* (bawah)

Sumber: wikipedia

Penerapan IP *multicast* pada sebuah jaringan memiliki beberapa keunggulan. Pada umumnya aplikasi yang dikembangkan dengan *multicast*, merupakan aplikasi yang banyak mengkonsumsi *bandwidth* dan sumber daya jaringan, misalnya untuk keperluan aplikasi *streaming*, *audio-video conference* ataupun pada *virtual learning* dimana pada aplikasi tersebut dituntut agar *delay* ditekan sekecil mungkin.

2.6.1 Pengalamatan *Multicast*

Multicast menggunakan pengalamatan yang khusus dalam pengimplementasiannya. Pada IPv4 alamat *multicast* dialokasikan dalam IPv4 kelas D. Alamat *multicast* berada dalam *range* 224.0.0.0 hingga 239.255.255.255 atau jika ditulis menggunakan CIDR 224.0.0.0/8.

2.6.2 Routing Multicast

Ada beberapa pilihan protokol untuk implementasi IP *multicast routing* antara lain:

1. *Distance Vector Multicast Routing* Protokol (DVMRP)
2. *Multicast OSPF* (MOSPF)
3. *Core-Base Trees* (CBT)
4. *Protocol-Independent Multicast, Dense Mode* (PIM-DM)
5. *Protocol-Independent Multicast, Sparse Mode* (PIM-SM)

2.6.3 Protocol Independet Multicast - Sparse Mode

Dari contoh beberapa protokol untuk *routing multicast* yang dapat digunakan di atas, PIM-SM merupakan salah satu protokol yang saat ini banyak digunakan. PIM SM didesain untuk membangun sebuah *routing multicast* yang efisien dalam sebuah jaringan. Sesuai dengan namanya PIM SM merupakan sebuah protokol yang independen. Artinya *multicast routing* yang dilakukan PIM SM tidak terpengaruh oleh bagaimana *unicast routing* dijalankan pada jaringan. PIM SM menggunakan informasi *routing unicast* yang sudah ada, baik *dynamic* maupun *static routing* untuk meneruskan paket data ke alamat *multicast* yang dituju.

Secara khusus PIM SM dirancang untuk memenuhi tujuan berikut:

1. Mempertahankan sistem IP *multicast* tradisional dimana *join* ke *group multicast* merupakan inisiatif dari *receiver*. Dalam sistem ini *multicast source* hanya menaruh paket data pada *first-hop Ethernet* tanpa pensinyalan sedangkan *receiver* memberikan sinyal tertentu kepada *router* untuk *join* ke *group multicast* yang dan mengambil data yang diinginkan.
2. Tidak merubah konfigurasi *host*. PIM SM adalah protokol komunikasi *router* ke *router* , yang artinya *host* tidak perlu melakukan perubahan tertentu agar dapat bergabung dalam jaringan

multicast. Yang dibutuhkan adalah semua *router* memiliki kemampuan *routing* PIM SM.

3. Mendukung *shared and source distribution trees*. Untuk *shared trees*, PIM-SM menggunakan *router central* yang disebut *Rendezvous Point* (RP), sebagai titik utama dari *shared tree*. Hubungan yang dibentuk oleh *source* dan *receiver* melalui RP disebut *Rendezvous Point Tree* (RPT). Untuk *source trees*, *router* menghubungkan *receiver* langsung ke *source*. *Source tree* ini nantinya akan dicari jalur terpendeknya atau *Shortest Path Tree* (SPT) dari *source* ke *receiver*. PIM-SM dapat menggunakan RPT dan SPT tersebut salah satu atau dua-duanya secara bersamaan.
4. Mempertahankan sifat independen dari berbagai bentuk *unicast routing protocol*.
5. Menggunakan mekanisme *soft-state mechanism* untuk beradaptasi terhadap perubahan jaringan dan perubahan *multicast group*. *Soft-state* berarti, jika konfigurasi *router* tidak di diubah maka konfigurasi bersifat jangka pendek dan akan *expired* setelah periode waktu tertentu..
6. Standar protokol PIM SM mengacu pada RFC 2362 dan RFC 4601.

2.6.3.1 *Internet Group Management Protocol*

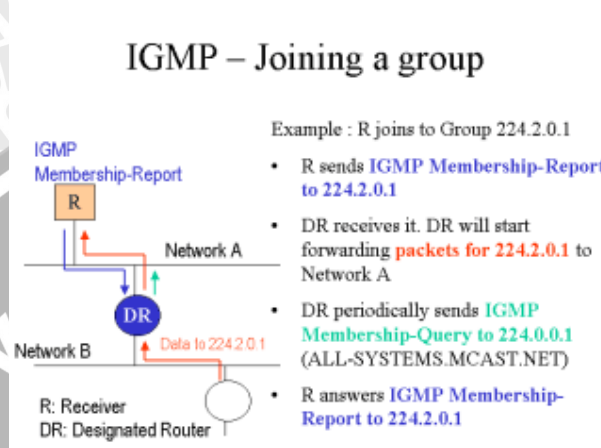
Internet Group Management Protocol (IGMP) adalah *protocol* komunikasi yang digunakan untuk mengatur *membership* sebuah *host* terhadap *group multicast* tertentu. IGMP digunakan oleh *host* dan *router* untuk mengatur keanggotaan terhadap sebuah *group multicast*.

Ilustrasi bagaimana IGMP berjalan pada jaringan adalah sebagai berikut :

Contoh : R join ke *Group* 224.2.0.1

1. R mengirim *IGMP Membership-Report* ke 224.2.0.1
2. *Designated Router* (DR) akan menerima paket IGMP tersebut. DR akan memulai *memforward* paket untuk 224.2.0.1 ke *Network A*.

3. DR secara *periodic* mengirim IGMP *Membership-Query* to 224.0.0.1 (ALL-SYSTEMS.MCAST.NET)
4. R menerima IGMP *Membership-Report* dari 224.2.0.1



Gambar 2.9 Mekanisme IGMP untuk *join group*

Sumber : <http://soi.wide.co.jp>

+	Bits 0 - 7	8 - 15	16 - 23	24 - 31
0	Type	Max Resp Time	Checksum	
32	Group Address			

Gambar 2.10 Struktur paket data IGMPv2

Sumber : Wikipedia

Tabel 2.4 Type pesan IGMP di gambarkan pada tabel di bawah ini

Type pesan IGMP	Keterangan
<i>Host Membership Query</i>	Dikirimkan oleh <i>router multicast</i> secara periodik untuk menyediakan <i>membership</i> bagi sebuah <i>group</i> .
<i>Host Membership Report</i>	Dikirim oleh <i>host</i> ketika bergabung dalam sebuah grup <i>multicast</i> tertentu.
<i>Leave Group</i>	Dikirim oleh sebuah <i>host</i> ketika meninggalkan grup dan mungkin <i>member</i> terakhir dari grup dalam sebuah segmen jaringan

Sumber : <http://technet.microsoft.com/en-us/library/cc787925.aspx>

2.6.3.2 *Rendezvous Point (RP)*

Rendezvous Point digunakan dalam PIM-SM sebagai titik bermuaranya paket *multicast*. Selain itu digunakan oleh *router first-hop* mengirimkan pesan register PIM pada *host* yang mengirim sebuah paket kepada sebuah *group*. Alamat RP juga digunakan oleh *router last-hop* untuk mengirim sinyal *join* PIM dan pesan *prune* kepada RP untuk *memberikan* informasi tentang kenggotaan *group*. Alamat RP harus dikonfigurasi pada semua *router* (termasuk *router RP*).

Untuk setiap satu *group multicast* hanya boleh terdapat satu RP. RP dapat dikonfigurasi secara *static* ataupun *dynamic*. Untuk mekanisme pemilihan RP secara dinamik (*bootstrap*) terdapat satu atau lebih *router* yang dikonfigurasi untuk menjadi RP berdasarkan parameter tertentu. Sedangkan untuk *router* yang berada pada cabang hanya menerima pesan RP dari *bootstrap router*.

2.6.3.3 *Reverse Path Forwarding (RPF)*

Pada proses *routing* terjadi proses meneruskan paket data dari jaringan satu ke jaringan lainnya melalui *interface* tertentu. Mekanisme ini disebut *forwarding*. Pada mekanisme *routing* dengan flooding, semua paket data yang masuk diterima. Kemudian paket data di *forward* ke semua *interface* kecuali *interface* paket data tersebut berasal. Hal ini mengakibatkan jaringan menjadi mudah mengalami *overhead*. RPF memiliki proses yang berbeda. Paket data dari sebuah *source* yang masuk ke sebuah *interface* akan diterima jika *router* memang menggunakan *interface* tersebut untuk terhubung dengan *source*. Kemudian paket data akan diteruskan ke *receiver* lainnya dengan proses yang sama. RPF mengacu pada tabel *routing unicast* yang dimiliki oleh *router*.

Pada *multicast routing*, RPF akan mengacu pada RP jika koneksi dari *source* ke *receiver* menggunakan *shared tree* (*,G). Jika koneksi dari *source* ke *receiver* menggunakan *source tree* maka RPF akan mengacu pada *source address* (S,G).

2.6.3.4 Mekanisme Protokol Routing PIM SM

Proses *routing* PIM SM adalah sebagai berikut :

1. PIM Neighbor Discovery

Paket PIM Hello dikirimkan ke *group* "All-PIM-Routers" (224.0.0.13, ff02::d) secara *periodic*. IP address tertinggi dipilih sebagai Designated Router. Jika DR mengalami times-out, akan dipilih DR baru. Router yang terhubung secara langsung akan terdeteksi sebagai PIM Neighbour.

2. PIM SM Forwarding

PIM-SM router memforward trafik *multicast* ke seluruh *interface* yang menuju ke *multicast group*. Router melakukan pengecekan RPF sebelum paket di forward. Jika menggunakan *shared tree* maka paket akan diforward mengacu pada RPT jika menggunakan *source tree* maka akan mengacu pada SPT. Paket data *multicast* yang datang ke sebuah *interface* akan di forward ke *interface* lainnya jika :

- Terdapat router lain yang *join* melalui *interface* tersebut.
- Terdapat receiver yang terkoneksi secara langsung dan *join* ke *group*.
- *Interface* secara manual dikonfigurasi untuk *join* ke *group* lainnya.

3. PIM SM Joining

4. PIM SM Registering

5. PIM SM SPT-Switchover

6. PIM SM Pruning

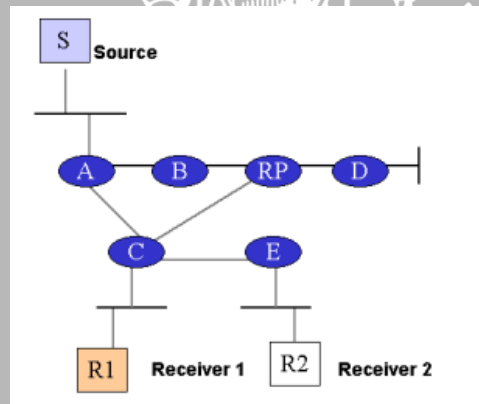
7. PIM SM Bootstrap

- Beberapa router dikonfigurasi sebagai *bootstrap router* (BSR).
- Satu BSR dipilih dalam sebuah domain.
- *Candidate RP* adalah router yang dipilih oleh BSR untuk menjadi RP. *Candidate RP* secara periodik mengirim *Candidate-RP-Advertisement messages* (C-RP-Advs) ke BSR.
- BSR secara periodik mengirim *Bootstrap messages* ke semua anggota *group* dan menginformasikan alamat *Candidate-RP*.

- Router menerima *bootstrap message* dari BSR dan menyimpan alamat RP.
- 8. PIM SM State Maintenance.
 - *Periodic Join/Prunes* dikirim ke semua PIM neighbors
 - *Periodic Joins refresh interfaces* di daftar PIM Interface
 - *Periodic Prunes refresh prune state* ke PIM Neighbors
 - Menerima *Multicast packets* reset(S,G) entry expiration timers.

Ilustrasi proses *routing* protokol PIM SM dalam jaringan adalah sebagai berikut :

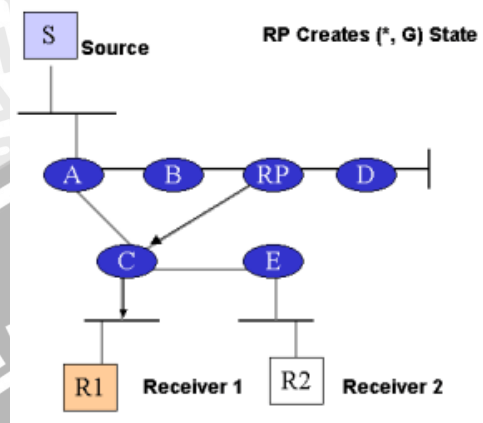
A. Keadaan awal jaringan, *Receiver 1* terhubung langsung dengan C dan *join*.



Gambar 2.11 Keadaan awal jaringan

Sumber : <http://soi.wide.co.jp>

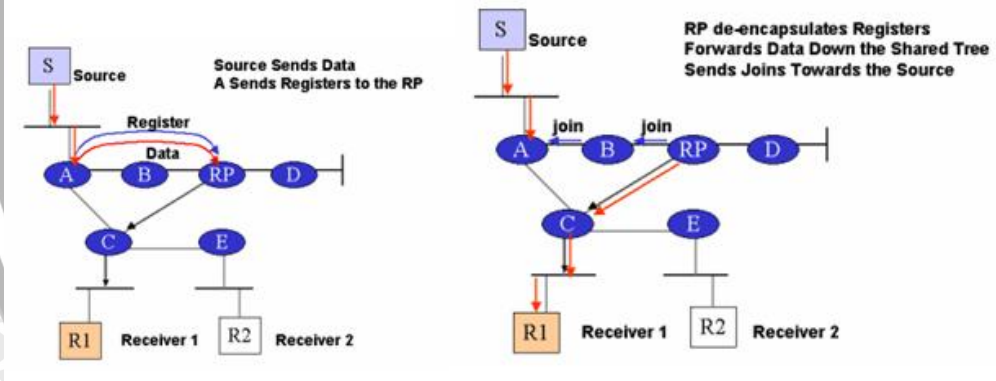
B. Proses *Join Receiver 1* ke RP dan entri (*,G) pada RP.



Gambar 2.12 Proses *join Receiver 1* ke RP

Sumber : <http://soi.wide.co.jp>

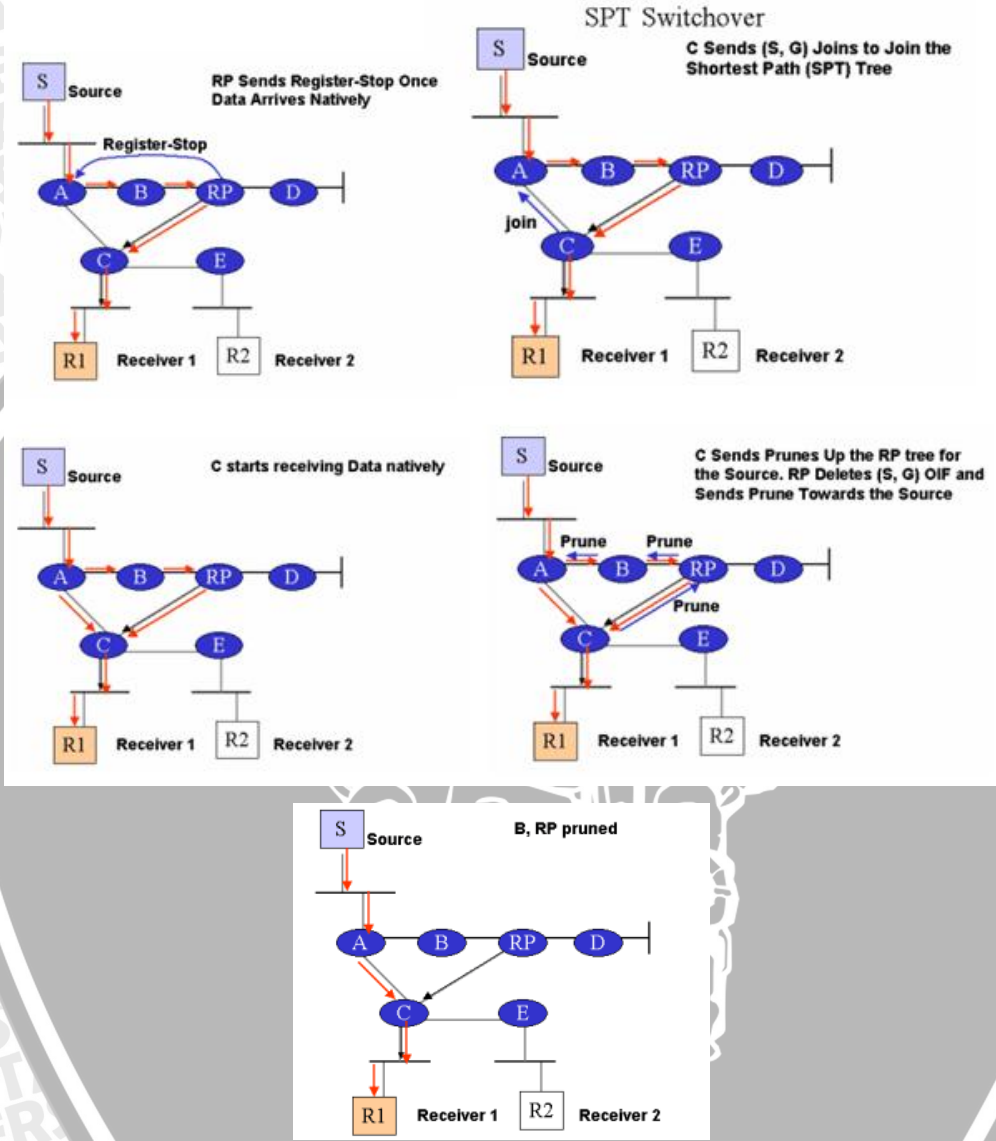
C. Proses *join source* ke RP dan pembentukan RPT



Gambar 2.13 Proses pembentukan RPT

Sumber : <http://soi.wide.co.jp>

D. Proses *Switchover* dari RPT ke SPT.



Gambar 2.14 Proses perpindahan dari RPT ke SPT

Sumber : <http://soi.wide.co.jp>

2.7 *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.

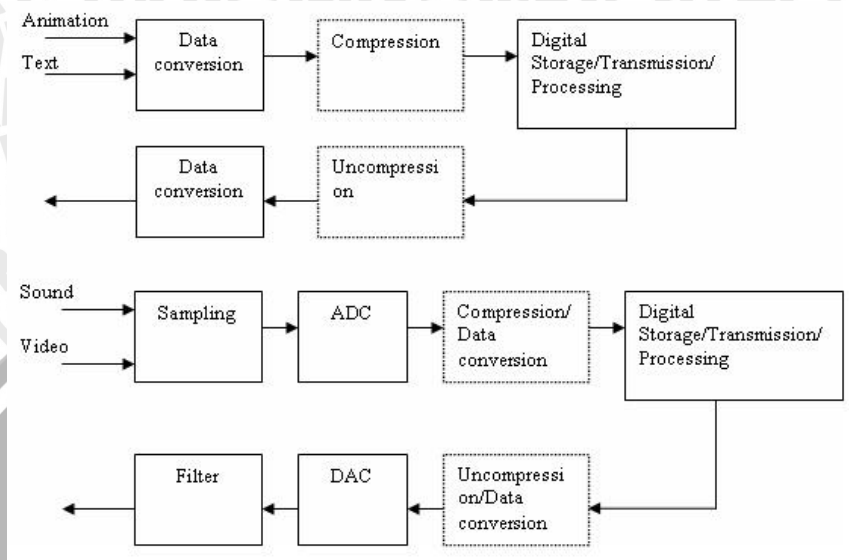
Paket-paket yang ditransmisikan memiliki *delay end to end* yang fluktuatif. Variasi dari *delay end to end* ini menyebabkan *delay jitter*. *Delay* ini menjadi masalah karena *client* harus men-*decode* dan menampilkan paket-paket pada kecepatan yang konstan dan keterlambatan paket akan menyulitkan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

2.8 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 2.15 Proses Konversi dan Kompresi Data

Sumber : W. Buchanan, 1997 : 11

2.8.1 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)

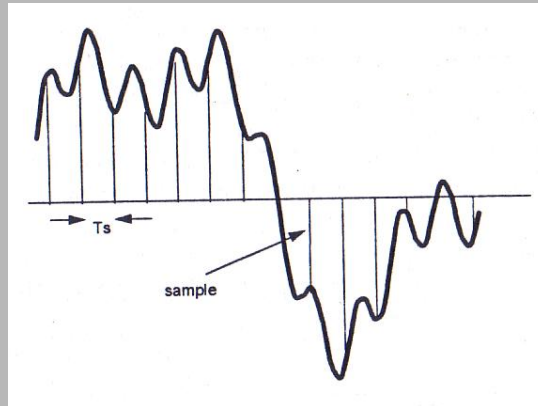
$$f_s = 2f(t)_{maks} \quad \dots \dots \dots (2-1)$$

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 2.16 Proses *Sampling*

Sumber : W. Buchanan, 1997 : 12

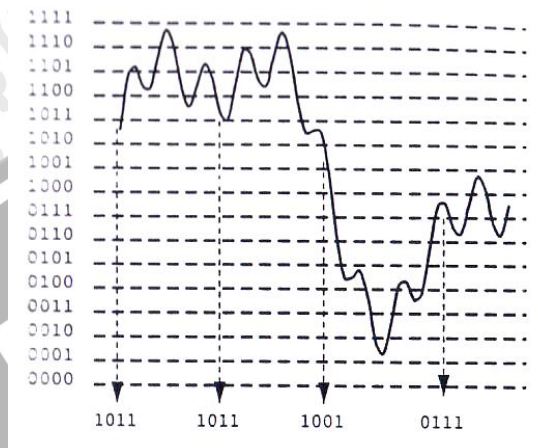
2. *Quantizing* (kuantisasi)

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 2.17 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 :

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

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 2.5.

Tabel 2.5 Standar Kompresi Audio

Kompresi <i>Audio</i>	Kecepatan <i>Encoding</i>	Rasio Kompresi
<i>Uncompressed audio</i> (mono)	705.6 kbps	1:1
<i>Uncompressed audio</i> (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

2.8.2 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 2.6 Standar Kompresi Video

Kompresi <i>Video</i>	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 detiknya. Semakin tinggi laju *frame*, semakin tinggi pula kualitas *videonya*. Namun, hal ini akan mengakibatkan jumlah bit yang diperlukan untuk menyimpan atau mentransmisikannya meningkat.

2.8.3 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.

2.8.3.1 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 2.7 Spesifikasi kompresi MPEG Audio

	Bit rate range (kbit/s)	Target bit rate (kbit/s)	Typical compression ratio
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) 128 (stereo)	1:10 to 1:12
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

2.8.3.2 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.

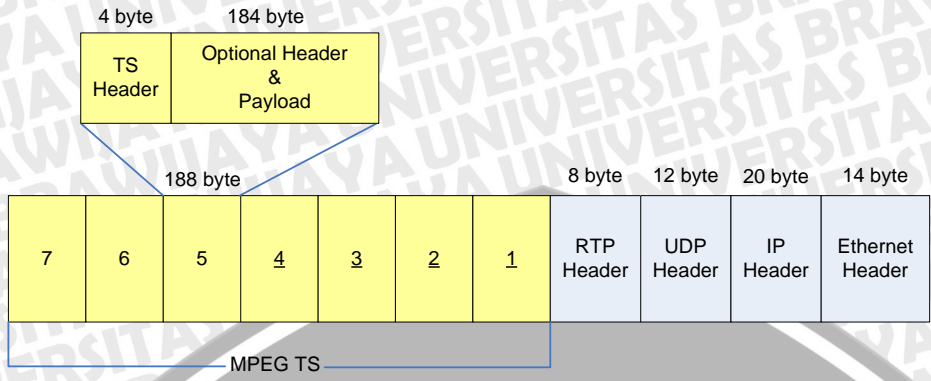
Tabel 2.8 Standar Codec untuk video

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

2.8.3.3 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 2.18 Paket data multimedia streaming dengan enkapsulasi MPEG TS
 Sumber : Vigato, 2005

2.9 Aplikasi Web

Aplikasi web atau dalam bahas Inggris disebut *web application* adalah sebuah aplikasi yang dapat diakses melalui *web browser* melalui sebuah jaringan intranet 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.

2.9.1 HTML

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.

2.9.2 PHP (*PHP Hypertext Preprocessor*)

PHP (*Peripheral 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 *embeded* 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.

2.10 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.

2.10.1 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 :

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

2.10.2 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 = *throughput* (paket/s)
- t_v = waktu rata-rata untuk mentransmisikan 1 paket yang benar (s)
- t_i = waktu yang dibutuhkan untuk mentransmisikan 1 paket (s)
- a = konstanta transmisi ternormalisasi
- p = probabilitas kesalahan paket

2.10.3 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)

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

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

2.10.4 Kapasitas Jaringan

Kapasitas jaringan adalah kemampuan jaringan dalam memberikan suatu layanan. Besarnya kapasitas jaringan dipengaruhi oleh bandwidth maksimum saluran transmisi, *bandwidth* aplikasi, dan efisiensi jaringan. Efisiensi jaringan adalah kemampuan saluran transmisi untuk melewati data tanpa error. Nilai efisiensi dipengaruhi oleh beberapa hal seperti *throughput*, *delay*, panjang saluran transmisi dan selisih kedatangan antar paket data.

Persamaan untuk efisiensi jaringan adalah seperti pada persamaan 2-5

$$E = \left(1 + \frac{2BLE}{cF}\right)^{-1} \dots\dots\dots (2-5)$$

Dimana :

- E = Efisiensi jaringan
- B = *throughput* total yang melalui saluran transmisi
- L = panjang saluran transmisi
- c = *delay* propagasi
- F = panjang paket
- e = selisih waktu kedatangan antar paket

Sedangkan untuk kapasitas jaringan dapat diketahui dengan persamaan (2-6)

$$\text{Kapasitas}_{\max} = \frac{BW_{\text{jaringan}} \times E}{BW_{\text{streaming}}} \dots\dots\dots (2-6)$$

Dimana :

- BW_{link} = *Bandwidth* maksimum jaringan
- E = Efisiensi jaringan
- $BW_{\text{streaming}}$ = *Bandwidth* aplikasi



Kapasitas optimum jaringan adalah kemampuan jaringan untuk dapat melewati data pada saluran transmisi secara optimal. Parameter optimal adalah jika koefisien *packet collision* kurang dari 10%. Kapasitas optimum jaringan dapat diketahui dengan Persamaan (2-7)

$$\text{Koefisien}_{\text{packet collision}} = \frac{\text{Jumlah Data}_{\text{max}}}{\text{BW}_{\text{jaringan}}} \times 100\% \dots\dots\dots (2-7)$$

- Jumlah Data_{max} : Jumlah data total
- BW_{jaringan} : *Bandwidth* jaringan
- Koefisien_{packet collision} : Koefisien *packet collision* ≤ 10%

2.10.5 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 (2-8)

$$N_{\text{packet loss}} (\%) = \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100\% \dots\dots\dots (2-8)$$

- Dengan :
- $N_{\text{packet loss}} (\%)$ = prosentase *packet loss*
 - $N_{\text{packet loss}}$ = jumlah *packet loss*
 - N_{paket} = jumlah paket *multimedia* yang diterima



2.10.6 Delay End to end

Delay pada sistem ini adalah waktu yang dibutuhkan untuk mengirimkan sebuah paket *multimedia*. *Delay end to end* merupakan *delay* antara *node* sumber dan *node* tujuan. *Delay end to end* pada sistem ini dihitung dari *server* ke *client*. Adapun batasan *delay* yang terjadi bisa dikelompokkan sebagai berikut (Monteleus, 2001) :

1. *Delay* di bawah 150 ms tidak akan terasa pengaruhnya oleh *client*
2. *Delay* antara 150 – 400 ms masih dapat ditolerir
3. *Delay* di atas 400 ms tidak dapat ditolerir.
4. *Delay end to end* ditentukan dengan Persamaan (2-9)

$$t_{end-to-end} = t_{codec} + t_{enc} + t_i + t_p + t_w + t_{dec} + t_{jitter} + t_{codec} \dots\dots\dots (2-9)$$

Dengan :

- t_{codec} = *delay codec* (s)
- $t_{end to end}$ = *delay end to end* (s)
- t_{enc} = *delay enkapsulasi* (s)
- t_i = *delay transmisi* (s)
- t_p = *delay propagasi* (s)
- t_w = *delay antrian* (s)
- t_{jitter} = *delay jitter* (s)
- t_{dec} = *delay dekapsulasi* (s)

2.10.6.1 Delay Codec

Delay codec adalah waktu yang dibutuhkan aplikasi untuk melakukan proses *encoding* dan *decoding* pada data *multimedia* pada komputer . *Delay codec* dipengaruhi berbagai macam faktor seperti algoritma kompresi, rasio kompresi, proses *sampling* dan lain lain. Setiap metode *encoding* memiliki *delay codec* yang berbeda beda. Pada metode *encoding* menggunakan MPEG dengan *Simple*



Profiles tanpa *B-frames* akan mengalami *delay* sebesar 50 ms (Vigato, 2006). Profil ini cocok digunakan untuk aplikasi yang menuntut *delay* rendah seperti *video conference*. *Delay* coder pada MPEG relatif lebih besar daripada metode *encoding* lainnya karena encoding MPEG ditujukan untuk kualitas gambar yang baik dengan *bitrate* yang rendah. Hal ini mengakibatkan algoritma kompresi MPEG yang lebih kompleks daripada algoritma kompresi lainnya.

2.10.6.2 *Delay* Enkapsulasi dan Dekapsulasi

Enkapsulasi adalah proses menambahkan *header* pada paket data, sehingga paket data tersebut dapat tepat sampai ke tujuan. *Delay* enkapsulasi adalah waktu yang dibutuhkan untuk menambahkan keseluruhan *header* pada sebuah paket. Sedangkan *delay* dekapulasi adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket. Pada saat pembentukan hubungan (*connection established*) antara *client* dengan *server* terjadi kesepakatan dalam enkapsulasi dan dekapulasi paket *multimedia*.

Apabila *node* sumber ingin mengirim paket *multimedia* ke *node* tujuan, maka proses yang terjadi adalah paket tersebut dikirimkan ke *transport layer*. Di *transport layer* dengan menggunakan *protocol* RTP dan UDP, paket *multimedia* dienkapsulasi menjadi segmen UDP dengan *header* RTP 8 byte dan *header* UDP 12 byte.

Dari *transport layer*, segmen UDP kemudian diteruskan ke *internet layer*. Di *internet layer* dengan menggunakan protokol IP, segmen UDP dienkapsulasi menjadi datagram IP dengan panjang IP *header* 20 byte.

Dari *internet layer*, datagram IP dikirimkan ke *network interface layer*. Di *network interface layer* dengan menggunakan protokol *Ethernet* II, datagram IP dienkapsulasi menjadi *frame Ethernet*. Panjang *frame Ethernet* ditentukan dengan Persamaan (2-10)

$$W_{frame} = W_{datagram} + Header_{Ethernet} \dots\dots\dots (2-10)$$

Dengan :

W_{frame} = panjang *frame Ethernet* (byte/paket)

$W_{datagram}$ = panjang datagram IP (byte/paket)

$Header_{Ethernet II}$ = 14 byte/paket

Delay enkapsulasi ditentukan dengan Persamaan (2-11)

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8 \quad (2-11)$$

Dengan :

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

W_{frame} = panjang *frame Ethernet* (byte/paket)

MSS = Maximum Segment Size (byte/paket)

C_{pros} = kecepatan pemrosesan data di *server* (bps)

Delay dekapsulasi ditentukan dengan Persamaan (2-12)

$$t_{dec} = \frac{W_{frame} - MSS}{C_{pros2}} \times 8 \quad (2-12)$$

Dengan :

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

W_{frame} = panjang *frame Ethernet* (byte/paket)

MSS = Maximum Segment Size (byte/paket)

C_{pros2} = kecepatan pemrosesan data di *client* (bps)

2.10.6.3 *Delay Transmisi*

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket *multimedia* ke media transmisi. Dipengaruhi ukuran paket *multimedia* dan kecepatan transmisi. *Delay* transmisi ditentukan dengan Persamaan (2-13)

$$t_i = \frac{(L+L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8 \quad \dots\dots\dots (2-13)$$

Dengan :

- t_i = delay transmisi (s)
- L = panjang paket *multimedia* (byte/paket)
- L' = panjang *header* (byte/paket)
- C = kapasitas kanal (bps)
- W_{frame} = panjang *frame Ethernet* (byte/paket)

2.10.6.4 Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket *multimedia* melalui media transmisi dari *server* ke *client*. *Delay propagasi* ditentukan dengan Persamaan (2-14)

$$t_p = (N_{DTE} \times t_{DTE}) + \left(\frac{L_{UTP}}{100} \times t_{UTP}\right) + (N_{router} \times t_{router}) \quad \dots\dots\dots (2-14)$$

Dengan :

- t_p = delay propagasi (s)
- N_{DTE} = jumlah *Data Terminal Equipment* seperti NIC dan Switch.
- t_{NIC} = delay pada NIC (s)
- L_{UTP} = panjang kabel UTP yang digunakan.
- t_{UTP} = delay pada UTP CAT 5 (s)
- N_{router} = jumlah *router* yang dilewati paket data.
- t_{router} = delay pada *router* (s)



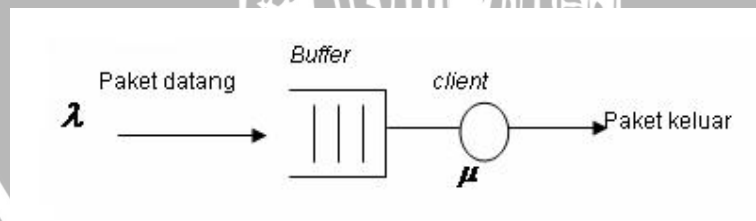
Tabel 2.9 Delay tipikal maksimum beberapa komponen *Fast Ethernet*.

Component	Maximum Delay (μs)
Fast Ethernet NIC	0.25
Fast Ethernet Switch Port	0.25
Class I Repeater	0.7 max
Class II Repeater	0.46 max
UTP Cable (per 100 meters)	0.55
Multimode Fiber (per 100 meters)	0.50

Sumber : Reynders, 2003

2.10.6.5 Delay Antrian

Delay antrian adalah waktu di mana paket *multimedia* berada dalam antrian untuk diproses oleh *client*. *Delay* antrian dapat dihitung dengan menggunakan *model* antrian M/M/1. M pertama menunjukkan distribusi kedatangan Poisson, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan bahwa jumlah *client* adalah tunggal. Disiplin antrian yang digunakan dalam skripsi ini yaitu FIFO (First In First Out), seperti yang terlihat dalam Gambar 2.19



Gambar 2.19 Model Antrian Single Client

Sumber : Mischa Schwartz, 1987 : 22

Besarnya *delay* antrian yang terjadi pada *client* ditentukan dengan Persamaan (2-15) (I Made Wiryana, 1992:2)

$$t_w = t_{queue} + t_{serv} \\ = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \quad \text{..... (2-15)}$$

Dengan:

t_w = delay antrian pada *client* (s)

t_{queue} = waktu tunggu paket pada *client* (s)

t_{serv} = waktu rata-rata pelayanan *client* (s)

λ = kecepatan kedatangan paket pada *client* (paket/s)

μ = kecepatan pelayanan *client* (paket/s)

Sedangkan kecepatan pelayanan ditentukan dengan Persamaan (2-16)
(Mischa Schwartz, 1987 : 23)

$$\mu = \frac{C}{L} \quad \text{..... (2-16)}$$

Dengan :

μ = kecepatan pelayanan *client* (paket/s)

C = kapasitas kanal (bps)

L = panjang paket *multimedia* (bit/paket)

2.10.6.6 Delay Jitter

Delay jitter disebabkan oleh kedatangan paket yang acak karena setiap paket melewati jalur yang berbeda-beda di internet. *Delay jitter* rata-rata ditentukan dengan Persamaan (2-17)

$$\theta = \frac{1}{N_{paket} - 1} \sum_{n=0}^{N_{paket} - 1} t_{end\ to\ end(n+1)} - t_{end\ to\ end(n)}$$

$$= \frac{t_v}{N_{paket} - 1} \dots \dots \dots (2-17)$$

Dengan :

- θ = delay jitter rata-rata (s)
- t_v = waktu transmisi (s)
- $t_{end\ to\ end}$ = delay end to end pada urutan ke-n (s)
- $t_{end\ to\ end(n+1)}$ = delay end to end pada urutan ke-(n+1) (s)
- N_{paket} = jumlah paket multimedia yang diterima



BAB III METODOLOGI

Tujuan penulisan skripsi ini adalah untuk mengaplikasikan multimedia *streaming* sebagai media kuliah online di jaringan komputer FTUB. Untuk merancang sistem tersebut metodologi yang digunakan adalah

3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan dasar teori yang mendukung dalam perancangan dan implementasi sistem, yaitu :

1. Kajian Pustaka mengenai konsep jaringan komputer, yaitu suatu cara untuk menghubungkan antara satu komputer dengan komputer yang lainnya.
2. Kajian Pustaka mengenai arsitektur jaringan komputer FTUB.
3. Kajian pustaka mengenai konsep dari *routing multicast* yang menggunakan *Protocol Independent Multicast Sparse Mode (PIM-SM)*.
4. Kajian Pustaka mengenai konsep multimedia *streaming*, menggunakan protokol RTP sebagai standar komunikasi multimedia yang real-time.

3.2 Perancangan sistem

Dari literatur-literatur yang dipelajari maka perancangan aplikasi multimedia *streaming* sebagai media kuliah online terdiri dari :

1. Perancangan sistem Kuliah Online, meliputi perancangan hardware dan software.
2. Perancangan metode multimedia *streaming*.
3. Perancangan jaringan *multicast* di Jaringan Komputer FTUB, dengan menambahkan fitur *multicast routing* pada *router*.

3.3 Pengujian

Setelah perancangan sistem selesai maka untuk mengetahui apakah sistem berjalan dengan baik maka diperlukan pengujian, pengujian meliputi :

1. Pengujian interkoneksi *multicast source* dengan masing-masing *multicast receiver*.
2. Pengujian keberhasilan *routing multicast*.
3. Pengujian aplikasi kuliah online.

3.4 Pengambilan data

Data-data yang digunakan dalam penyusunan skripsi ini adalah data primer dan data sekunder meliputi :

1. Data primer

Data primer adalah data yang didapatkan dari hasil pengujian, data primer yang diperoleh adalah :

- Jumlah paket rata rata yang diterima
- *Packet Loss* rata rata
- Jumlah data yang diterima (byte)
- Kecepatan rata rata transmisi data
- Kualitas *Audio* dan *Video*

2. Data Sekunder

Data sekunder adalah data yang diperoleh dari berbagai buku, jurnal-jurnal dan dari internet . beberapa data sekunder yang diperoleh adalah :

- Panjang *header* UDP 8 byte [Tanenbaum, 2003:232]
- Panjang *header* IPv4 20 byte [Riza, 2002:51]
- Panjang *header* RTP 12 byte [Wesley, 2003:156]
- Panjang *Ethernet header* 14 byte [Newport-Network, 2005:2]
- Kapasitas maksimum fast Ethernet adalah 100 Mbps [Stallings, 1996:479]
- Panjang maksimum MTU (Maximum Transmission Unit) = 1500 byte [Stevens, 1994:29]

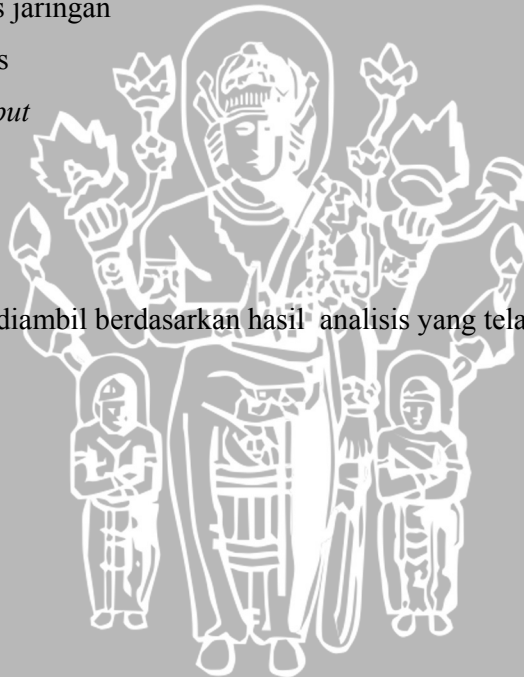
3.5 Analisis

Setelah sistem terimplementasi dan berjalan, kemudian sistem dianalisis hasilnya digunakan sebagai bahan pengambilan kesimpulan. Analisis dilakukan secara matematis terhadap hasil pengujian dengan menggunakan persamaan-persamaan yang ada pada bagian tinjauan pustaka dan parameter-parameter aplikasi pada bagian perancangan. Analisis dilakukan pada :

1. *Routing Multicast*
2. *Delay end-to-end*
3. *Bandwidth Multimedia Streaming*
4. Kapasitas jaringan
5. Paket loss
6. *Throughput*

3.6 Kesimpulan

Kesimpulan diambil berdasarkan hasil analisis yang telah dilakukan



BAB IV

PERANCANGAN APLIKASI *MULTIMEDIA STREAMING* SEBAGAI MEDIA *KULIAH ONLINE*

4.1 Umum

Kuliah *online* adalah suatu bentuk perkuliahan, di mana dosen dan mahasiswa tidak perlu bertatap muka secara langsung, namun secara *virtual*, atau melalui media telekomunikasi tertentu. Dalam dunia ICT hal ini lazim disebut dengan *video conference*. *Video conference* adalah salah satu bentuk *multimedia streaming* yang digunakan sebagai media interaksi antara dua atau lebih lokasi yang berjauhan melalui media suara dan gambar yang ditransmisikan secara simultan dan *realtime*. Sistem kuliah *online* ini nantinya akan menggunakan teknologi *multicast*. Agar sistem dapat berjalan, maka pada Jaringan komputer FTUB harus memiliki kemampuan *multicast routing*.

Pada bab ini akan dijelaskan mekanisme perancangan *multimedia streaming* sebagai media kuliah *online* yang meliputi perancangan sistem kuliah *online*, perancangan metode *streaming*, serta perancangan *multicast routing* di Jaringan komputer FTUB.

4.2 Perancangan Sistem Kuliah *Online*

Sistem kuliah *online* pada dasarnya adalah sebuah sistem *video conference* antara dua lokasi yang berbeda yaitu lokasi dosen dan kelas. Dua subyek kuliah *online* ini nantinya akan menggunakan sistem kuliah *online* yang akan dibangun untuk dapat berinteraksi secara *realtime* melalui media *audio* dan *video streaming*.

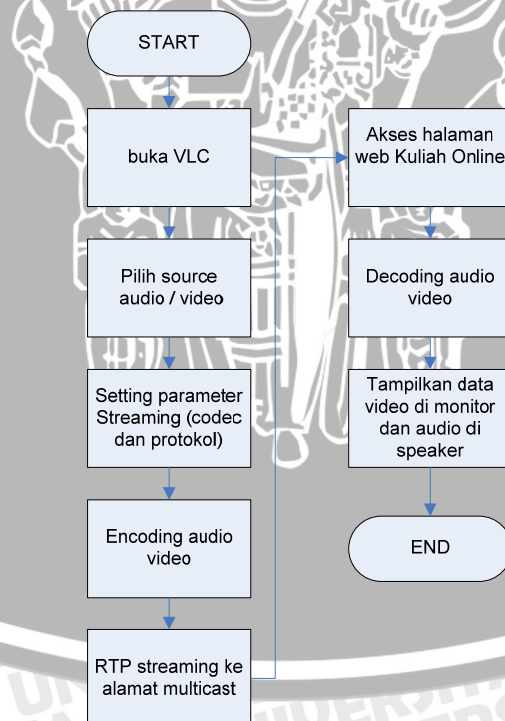
4.2.1 Cara Kerja Aplikasi Kuliah *Online*

Cara kerja Aplikasi Kuliah *Online* adalah sebagai berikut:

1. *Server* dan *web server* yang digunakan untuk sistem kuliah *online* harus siap (*stand by*).

2. Dosen dan kelas masing – masing memiliki alamat *multicast* khusus yang digunakan untuk *streaming*.
3. Dosen melakukan inisiasi *streaming* dengan *webcam* sebagai sumber *video* dan *mic* sebagai sumber *audio*. Proses *streaming* menggunakan VLC dan ditujukan ke alamat *multicast* untuk dosen.
4. Untuk kelas, salah seorang peserta kuliah melakukan inisiasi *streaming* dengan *webcam* sebagai sumber *video* dan *mic* sebagai sumber *audio*. Proses *streaming* menggunakan VLC dan ditujukan ke alamat *multicast* untuk kelas.
5. Peserta kuliah lainnya atau orang lain yang tidak berada di dalam kelas dapat mengikuti perkuliahan dengan membuka halaman web pada *server* kuliah *online*.

Secara detail cara kerja sistem kuliah *online* ditunjukkan dengan diagram alir pada Gambar 4.1



Gambar 4.1 Diagram Alir Sistem Kuliah *Online*

Sumber: Perancangan

4.2.2 Perancangan Perangkat Keras Sistem Kuliah Online

Sistem Kuliah *Online* menggunakan sebuah web *server* yang digunakan untuk menempatkan halaman web yang berfungsi sebagai *user interface* untuk aplikasi kuliah *online*. Spesifikasi perangkat keras yang digunakan untuk *server* adalah sebagai berikut :

Prosesor	: Intel Pentium D 3.0 Ghz
Memori	: 1 GB
NIC	: 3com 10/100 Fast Ethernet
Mainboard	: Asrock
Hard disk	: Hitachi 200 GB SATA

Untuk sisi *host* baik dosen maupun kelas, perangkat keras yang digunakan bebas namun disarankan untuk memenuhi spesifikasi minimum sebagai berikut :

Prosesor	: Kecepatan diatas 1 Ghz
Memori	: Diatas 128 MB
NIC	: <i>Fast Ethernet</i> NIC dengan kecepatan 100Mbps atau lebih
Video	: USB <i>Webcam</i> atau <i>Built In Webcam</i>
Audio I/O	: Stereo <i>Audio</i> (Mic dan Speaker)

4.2.3 Perancangan Perangkat Lunak Sistem Kuliah Online

Selain perangkat keras, pada Aplikasi Kuliah *Online* juga dibutuhkan perangkat lunak. Untuk perangkat lunak sendiri dibagi dua yaitu perangkat lunak pada sisi *server* dan sisi *client*. Berikut perangkat lunak yang digunakan :

1. Ubuntu 8.04 *Server Edition* (hardy heron) digunakan sebagai *Operating System* 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 sebagi *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 *Kuliah Online* menggunakan bahasa pemrograman HTML, CSS dan PHP.

4.2.3.1 Perancangan Halaman Web *Kuliah Online*

Halaman web *Kuliah Online* berfungsi untuk menampilkan hasil *streaming* dan mengontrol parameter *audio video* yaitu *audio volume*, *mute*, *stop*, *play* dan *full screen*. Selain itu halaman web ini juga memiliki fasilitas *shoutbox* yaitu fasilitas bertukar pesan berbasis teks untuk mendukung jika diperlukan komunikasi secara tertulis antara Dosen dengan Kelas. Tampilan halaman web yang direncanakan adalah sebagai berikut :



Gambar 4.2 Halaman web *Kuliah Online*

Sumber : Perancangan

Perancangan tampilan halaman web pada Gambar 4.2 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. Penambahan script PHP untuk *message box* untuk bertukar pesan.

4.3 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 :

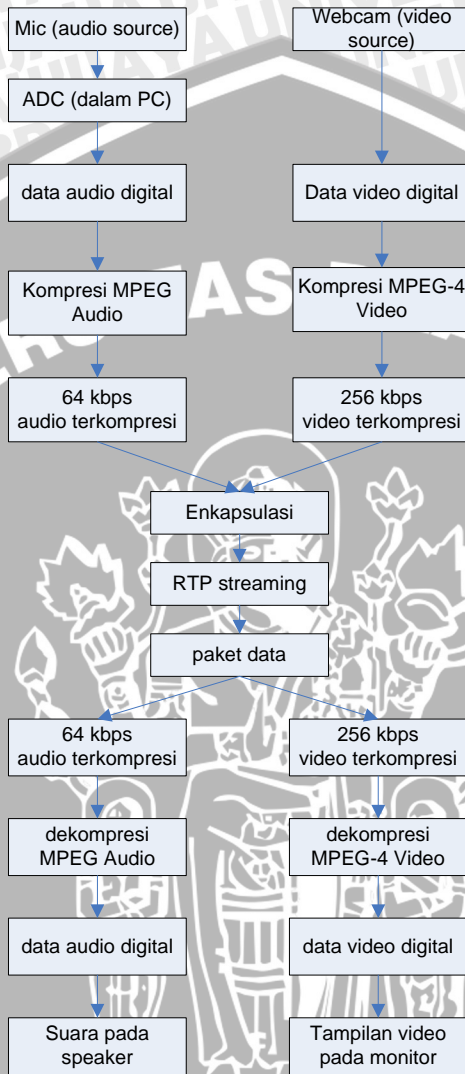
1. *Audio*
 - *Codec* : MPEG *Audio*
 - *Bitrate* : 64 kbps
 - *Frequency Sampling* : 44,1 Khz
 - *Channel* : Stereo
2. *Video*
 - *Codec* : MPEG-4 *Video*
 - *Data rate* : 256 kbps
 - *Resolusi* : 320 x 480 *pixel*
 - *Frekuensi frame* : Dapat disesuaikan (minimum 0 fps, maksimum 30 fps).
 - *Bit rate* : 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 4.3 Blok Diagram proses *streaming*

4.4 Perancangan *Multicast Routing* pada jaringan FTUB

Sistem Kuliah *Online* 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.

4.4.1 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.

4.4.2 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

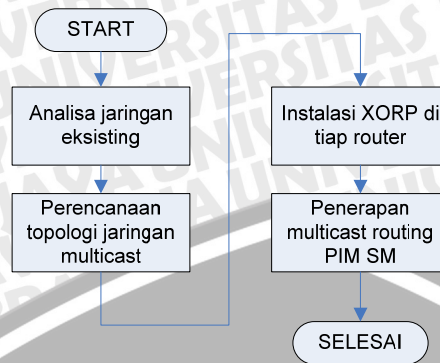
tinggi, sehingga OS ini dapat dijalankan pada PC yang memiliki spesifikasi tidak terlalu tinggi (kecepatan prosesor dibawah 1Ghz).

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. *ssmping* adalah aplikasi yang digunakan untuk mengetahui konektivitas antara dua *host* melalui jaringan *multicast*. Ssm ping dapat dianalogikan dengan aplikasi ping yang digunakan untuk jaringan *multicast*.

4.4.3 Perancangan *Multicast Routing PIM SM*

Pada jaringan komputer FTUB terdapat 12 *router* yang melakukan proses *routing* dalam jaringan. Proses *routing* ini menggunakan metode *static routing*. Tabel *routing unicast ipv4* mengacu pada *static routing* yang sudah ada. Untuk proses *multicast routing*, proses *routing* menggunakan metode PIM SM dengan *router daemon XORP*. Berikut ini adalah proses perancangan *multicast routing* :



Gambar 4.4 Diagram alir perancangan *multicast routing*

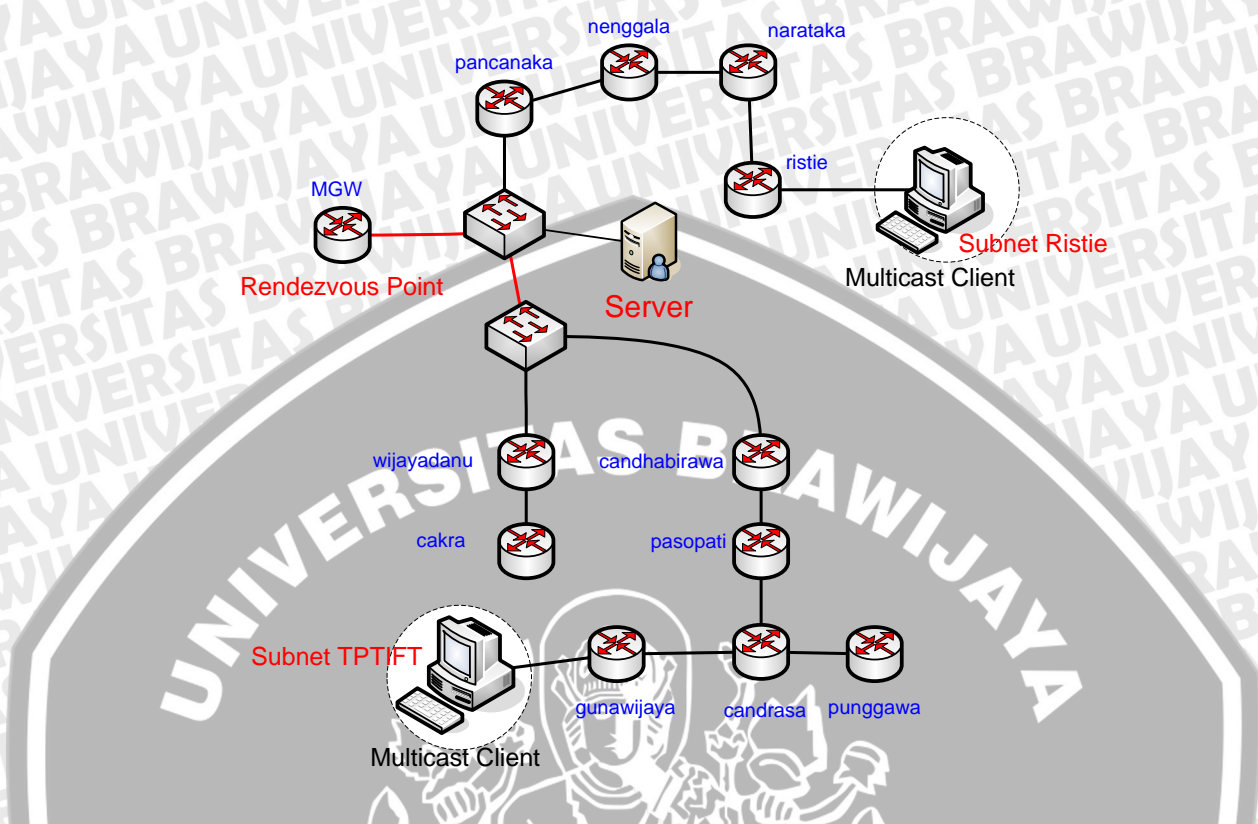
Sumber : Perancangan

4.4.3.1 Analisa Jaringan Eksisting

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. Hop terbanyak adalah 8 hop yaitu dari *router* TPTIFT ke *router* RISTIE.

4.4.3.2 Perencanaan topologi jaringan *multicast*

Untuk jaringan *multicast* topologi secara fisik tidak berubah. Penambahan yang dilakukan hanya menentukan *rendezvous point* untuk PIM SM dan menentukan posisi *server* pada jaringan *multicast*. Berikut topologi jaringan *multicast* yang direncanakan :



Gambar 4.5 Topologi jaringan *multicast*
Sumber : Perancangan

Proses perancangan jaringan *multicast* adalah sebagai berikut :

1. Perencanaan topologi jaringan : Topologi jaringan tidak mengalami perubahan. Topologi tetap menggunakan topologi jaringan eksisting.
2. Penentuan titik *rendezvous point* : *Main Gateway* FTUB dijadikan *rendezvous point* karena posisi *Main Gateway* FTUB berada di pusat jaringan komputer FTUB. *IP Address* *Rendezvous Point* adalah 172.17.7.250.
3. Posisi *Multicast Client* berada pada Subnet TPTIFT dan Subnet Ristie. Kedua subnet merupakan jarak terjauh antara dua *router* dengan hop maksimum adalah 8 hop.
4. Penentuan posisi *Server* : *Server* berada pada zona *Main Gateway* dengan konfigurasi IP sebagai berikut :

<i>IP Address</i>	: 172.17.7.252
<i>Subnet Mask</i>	: 255.255.255.248
<i>Default Gateway</i>	: 172.17.7.250

4.4.3.3 Instalasi XORP pada *router*.

Routing multicast pada skripsi ini menggunakan XORP sebagai *routing daemon*. XORP ini nantinya akan melakukan fungsi *multicast routing* PIM SM pada *router*. Proses instalasi XORP adalah sebagai berikut :

1. Kompilasi ulang kernel FreeBSD 6.2 agar dapat melakukan proses *multicast routing*. Hal ini dilakukan dengan menambahkan *option MROUTING* dan PIM pada konfigurasi kernel . Kemudian dilakukan proses kompilasi kernel.

```
[root@rendezvous /usr/src]# make buildkernel
KERNCONF=MULTICAST
[root@rendezvous /usr/src]# make installkernel
KERNCONF=MULTICAST
```

Gambar 4.6 Proses kompilasi kernel FreeBSD 6.2

Sumber : Perancangan

2. Instalasi XORP versi 1.3 menggunakan *port* pada FreeBSD 6.2. Proses ini dilakukan dengan perintah sebagai berikut :

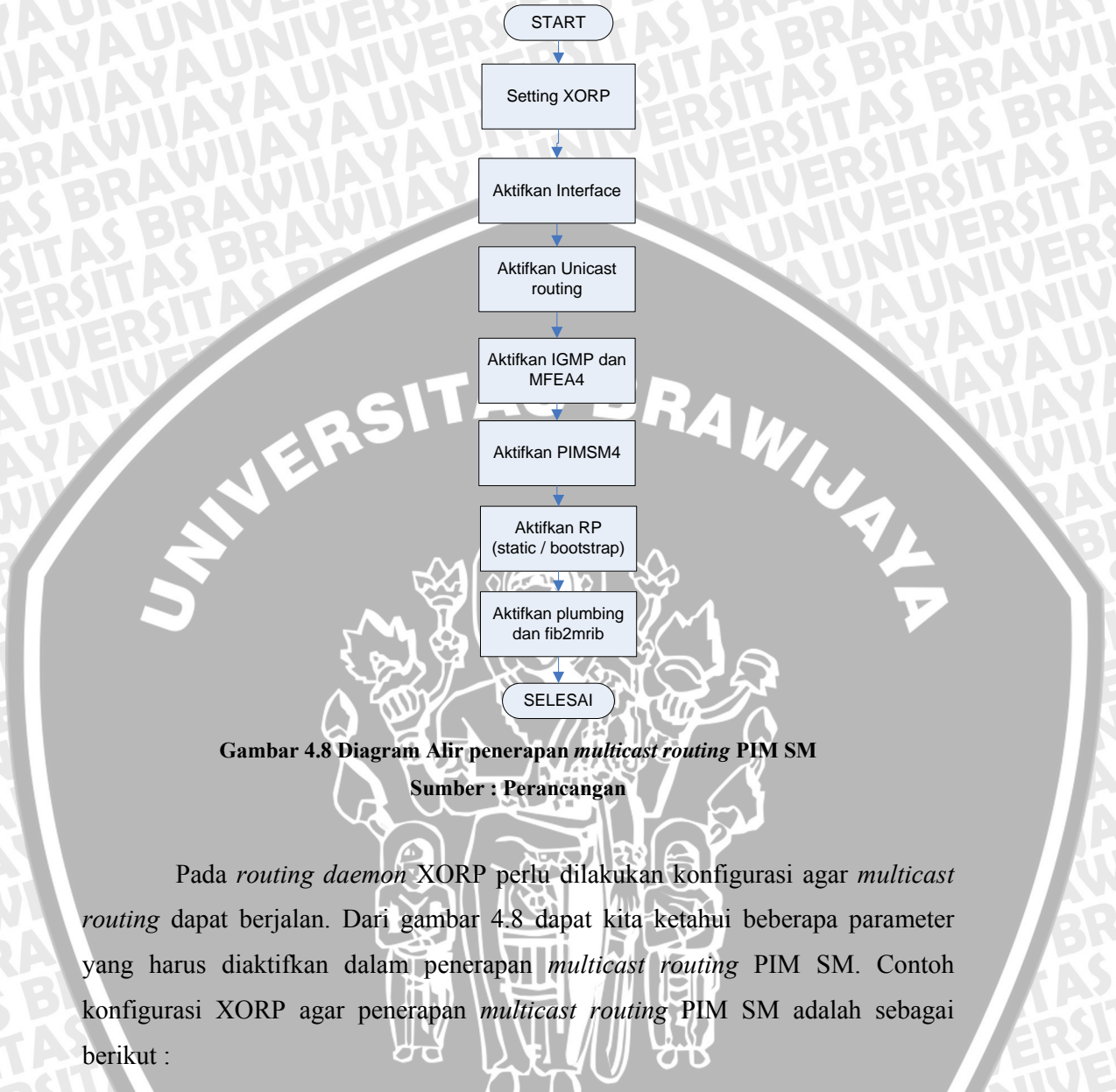
```
[root@rendezvous ~]# cd /usr/ports/net/xorp/
[root@rendezvous /usr/ports/net/xorp]# make install
```

Gambar 4.7 Proses Instalasi XORP 1.4

Sumber : Perancangan

4.4.3.4 Penerapan *Multicast Routing* PIM SM

Routing multicast tidak dapat diterapkan seperti *routing unicast*. Proses *routing* tidak dapat langsung dimasukkan ke dalam tabel *routing unicast* IPv4 yang sudah ada. Hal ini karena *routing multicast* tidak mengacu pada alamat *multicast source* atau *multicast source* tetapi pada *group multicast*. Untuk menerapkan *multicast routing* langkah langkahnya adalah sebagai berikut :



Gambar 4.8 Diagram Alir penerapan *multicast routing* PIM SM
 Sumber : Perancangan

Pada *routing daemon* XORP perlu dilakukan konfigurasi agar *multicast routing* dapat berjalan. Dari gambar 4.8 dapat kita ketahui beberapa parameter yang harus diaktifkan dalam penerapan *multicast routing* PIM SM. Contoh konfigurasi XORP agar penerapan *multicast routing* PIM SM adalah sebagai berikut :

```

/*XORP Configuration File, v1.0*/
protocols {
  fib2mrib {
    disable: false
  }
  igmp {
    disable: false
    interface r10 {
      vif r10 {
  
```

```
        di sabl e: fal se
    }
}
pi msm4 {
    di sabl e: fal se
    i nterface r10 {
        vi f r10 {
            di sabl e: fal se
        }
    }
    s tatic-rps {
        rp 172.17.7.252 {
            g roup-prefi x 224.0.0.0/4 {
            }
        }
    }
    swi tch-to-spt-threshol d {
        di sabl e: fal se
        i nterval : 100
        bytes: 102400
    }
}
fea {
    u ni cast-forwardi ng4 {
        di sabl e: fal se
    }
}
i nterfaces {
    restore-ori gi nal -confi g-on-shutdown: fal se
    i nterface r10 {
        di sabl e: fal se
        di scard: fal se
        descri pti on: "upl i nk"
```

```
default t-system-config {
}
}
pimbinding {
  mfea4 {
    disable: false
    interface r10 {
      vif r10 {
        disable: false
      }
    }
  }
}
```

Dari file konfigurasi tersebut terdapat beberapa *option* yang di harus atur.

Yaitu :

1. protocols : Digunakan untuk mengatur protokol protokol yang digunakan dalam *multicast routing*. Protokol IGMP, pimsm4, dan fib2mrib dikonfigurasi pada bagian ini.
2. fib2mrib : Digunakan untuk mengaktifkan proses injeksi tabel routing multicast ke tabel routing unicast yang ada.
3. igmp : Digunakan untuk mengatur interface – interface yang terdapat dalam sebuah komputer agar dapat melewati paket IGMP.
4. pimsm4 : Digunakan untuk mengatur parameter – parameter routing PIM SM. Parameter ini harus di enable agar multicast routing bisa berjalan. Beberapa parameter

yang harus dikonfigurasi agar PIM SM dapat berjalan adalah interfaces, static rps, dan switch to spt treshold.

5. static rps : Digunakan untuk mengatur alamat router yang dijadikan rendezvous point. Pada skripsi ini rendezvous point (RP) di setting secara static dengan alamat RP adalah 172.17.7.250 (MGW). RP dapat di setting secara statis maupun dinamis.
6. switch to spt treshold : Digunakan untuk mengatur waktu yang perpindahan dari RPT ke SPT . Terdapat dua opsi yang di setting yaitu berdasarkan interval waktu atau berdasarkan besar data yang sudah dikirim.
7. fea : FEA merupakan kependekan dari Forwarding Engine Abstraction yang digunakan untuk mengaktifkan forwarding pada router. Pada file konfigurasi diatas opsi FEA yang digunakan adalah unicast-forwarding4 yang artinya kita menggunakan engine forwarding untuk IPv4.
8. interfaces : Digunakan untuk mengaktifkan interfaces pada router dan juga untuk mengatur IP Address, Subnetmask dan parameter interfaces lainnya. Pada skripsi ini parameter interfaces menggunakan opsi default-system-config yang berarti kita menggunakan setting IP Address dari sistem yang sudah ada. (native IP Addressing).
9. mfea4 : Digunakan untuk mengaktifkan proses forwarding paket data menggunakan metode multicast pada interface. MFEA4 merupakan kependekan dari Multicast Forwarding Engine Abstraction 4 yang berarti proses multicast forwarding berjalan pada teknologi IPv4.

BAB V

PENGUJIAN DAN ANALISA

Pengujian pada skripsi ini bertujuan untuk mengetahui apakah sistem yang dirancang dan diimplementasikan dapat berfungsi. Sedangkan pada analisa bertujuan untuk mengetahui performansi sistem berdasarkan hasil pengujian.

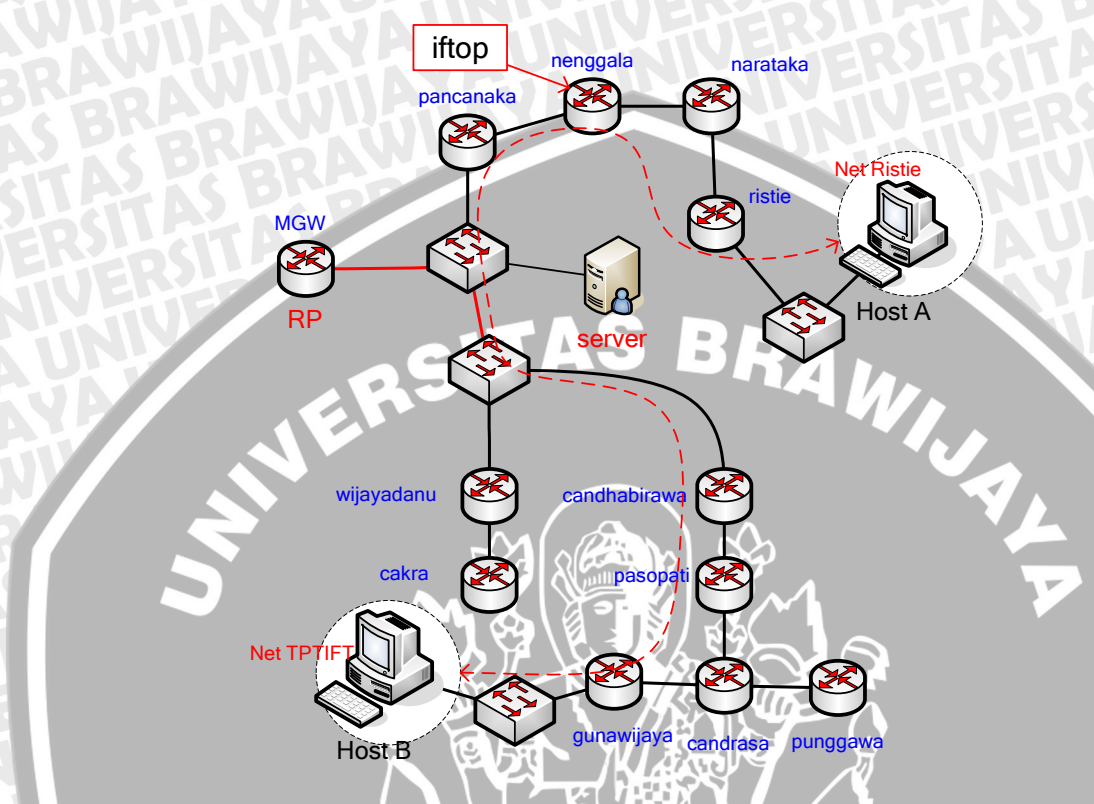
5.1 Pengujian

Pengujian pada skripsi ini dibagi menjadi 3 bagian yaitu pengujian *multicast routing* untuk mengetahui hasil implementasi *multicast routing* dari perancangan. Kemudian dilakukan pengujian *multimedia streaming* pada jaringan *multicast* untuk mengetahui apakah proses *multimedia streaming* dapat berjalan pada jaringan *multicast* yang sudah dibangun. 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 Kuliah *Online* untuk mengetahui apakah aplikasi yang telah dibuat dapat berfungsi.

5.1.1 Pengujian *Multicast Routing*

Pengujian *multicast routing* dapat dikatakan berhasil jika jaringan *multicast* mampu melewati paket data *multicast* hingga mencapai tujuan yang benar. Pengujian dilakukan dengan konfigurasi sebagai berikut :

1. Menggunakan 2 komputer sebagai *host*. *Host A* berada dalam Net Ristie dengan IP Address 172.17.67.38. *Host B* berada dalam Net TPTIFT dengan IP Address 172.17.68.212. Dua *host* diletakkan dalam jaringan dengan jarak terjauh yaitu 8 hop. *Host A* dan *Host B* dapat berfungsi sebagai *multicast receiver* maupun *multicast source*.



Gambar 5.1 Topologi Pengujian
 Sumber : Perancangan

2. Pengujian interkoneksi *Host A* dan *Host B* menggunakan aplikasi *ssmpping*.
3. Monitoring *bandwidth* yang dibutuhkan untuk *streaming* menggunakan aplikasi *iftop* yang dijalankan pada *router* *Nenggala*.
4. *IP Address server* adalah 172.17.7.252.
5. *Multicast Address* yang digunakan oleh *host A* adalah 239.1.1.1 dan sedangkan untuk *host B* adalah 239.1.1.2 dan *port* yang digunakan untuk proses *streaming* adalah *port 2222*.

5.1.1.1 Pengujian *routing* PIM SM

Pengujian *routing* PIM SM dilakukan dengan menguji kemampuan *routing multicast* PIM SM. Parameter keberhasilan *routing* PIM SM adalah *router - router* yang terhubung satu dengan lainnya dapat melakukan fungsi *multicast routing*.

Pengujian dilakukan dengan *login* ke xorpsh dan melihat parameter parameter *multicast routing*, antara lain PIM *neighbors*, tabel *routing multicast*, dan PIM *rendezvous point states*.

5.1.1.2 Pengujian PIM *Neighbors*

Pengujian PIM *neighbors* bertujuan untuk mengetahui apakah *router router* yang terhubung secara langsung (*peer to peer*) dapat saling mengenali sebagai *router* yang dapat melakukan *routing* PIM SM. Pengujian meliputi pengecekan status PIM *Neighbors* pada setiap *router* dan melihat berapa jumlah PIM *neighbors* yang terhubung.

Contoh hasil pengujian PIM *neighbors* pada *router* MGW yang berfungsi sebagai *rendezvous point* didapat sebagai berikut :

```

root@mgw> show PIM neighbors
Interface  DRpriority NeighborAddr  V Mode  Holdtime Timeout
eth0      1 172.17.7.253  2 Sparse  105     94
eth0      1 172.17.7.251  2 Sparse  105     89
eth0      1 172.17.7.254  2 Sparse  105     55
root@mgw>

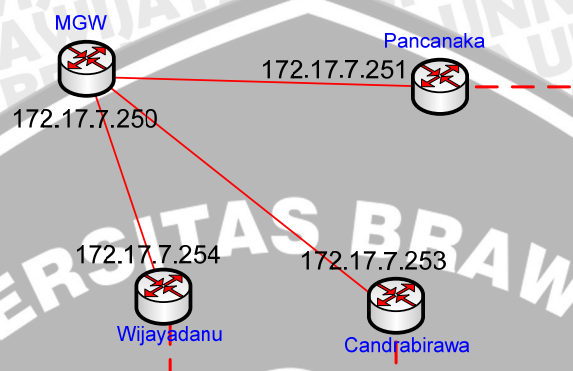
```

Gambar 5.2 Pengujian PIM *Neighbors* pada *Router* MGW

Sumber : Pengujian

Dari hasil pengujian diatas dapat diketahui bahwa *router* MGW memiliki tiga PIM *neighbors* yang terhubung secara *peer to peer*. Dari pengujian diatas juga dapat kita ketahui nilai *holdtime* dan *timeout*. Nilai *holdtime* menunjukkan berapa lama *router* akan menyimpan informasi PIM *neighbors*, sedangkan nilai *timeout* menunjukkan berapa lama lagi (dalam satuan detik) informasi PIM

neighbors tersebut akan dihapus. Pada saat nilai *timeout* 0 maka *router* akan mengirimkan pesan PIM *hello* lagi kepada *neighbors router* untuk mengecek statusnya.



Gambar 5.3 Ilustrasi *Neighbouring Routers*
 Sumber : Pengujian

Tabel 5.1 Tabel Pengujian PIM *Neighbors* pada Tiap Router

No	Hostname router	Jumlah PIM neighbors	No	Hostname Router	Jumlah PIM neighbors
1	MGW	3	8	Cakra	1
2	Pancanaka	4	9	Pasopati	2
3	Candrabirawa	4	10	Candrasa	3
4	Wijayadanu	4	11	Punggawa	1
5	Nenggala	2	12	Gunawijaya	1
6	Narataka	2			
7	Ristie	1			

Sumber : Pengujian



5.1.1.3 Pengujian Tabel *Routing Multicast*

Pengujian tabel *routing multicast* digunakan untuk mengetahui apakah tabel *routing multicast* pada *router* dapat terbentuk.

Tabel *routing multicast* pada *router* MGW dapat dilihat sebagai berikut:

```

root@gunawijaya> show pim mrib
DestPrefix      NextHopRouter  VifName VifIndex MetricPref Metric
0.0.0.0/0       172.17.7.21   r10     0         254  65535
172.17.7.20/30  172.17.7.22   r10     0         0     0
172.17.67.128/27 172.17.67.129 r12     2         0     0
172.17.68.0/27  172.17.68.130 r11     1         254  65535
172.17.68.128/25 172.17.68.129 r11     1         0     0
root@gunawijaya>

```

Gambar 5.4 Hasil pengujian tabel *routing multicast*

Sumber : Pengujian

5.1.1.4 Pengujian *Rendezvous point State*

Pengujian RP *State* bertujuan untuk mengetahui apakah untuk *group multicast* yang ditentukan *router* dapat mengetahui *router* mana yang menjadi RP. RP untuk setiap *router* sama karena diatur secara statik agar mengarah ke alamat IP yang sudah ditentukan. Berikut ini hasil pengujian RP *state* pada *router* Gunawijaya :

```

root@gunawijaya.ftub> show pim rps
RP          Type      Pri Holdtime Timeout ActiveGroups GroupPrefix
172.17.7.2  static   192    -1    -1          1 224.0.0.0/4

```

Gambar 5.5 Pengujian RP *State* pada *Router* Gunawijaya

Sumber : Pengujian

5.1.1.5 Pengujian Koneksi *Multicast*

Pengujian koneksi *multicast* bertujuan untuk mengetahui apakah *host* yang terhubung dengan jaringan komputer dapat melakukan komunikasi data dengan teknologi *multicast*. Pengujian menggunakan aplikasi *ssmpping* dan

melibatkan 2 komputer sebagai *host*, yaitu *Host A* dan *Host B*. Pengujian dilakukan dengan menguji koneksi *multicast* menggunakan *ssmping* dengan satu *host* sebagai *server* (*multicast source*) dan *host* lainnya menjadi *client* (*multicast receiver*). *Host* yang menjadi *server* akan menjalankan aplikasi *ssmpingd* (*ssmping daemon*) sedangkan *host* yang menjadi *client* menggunakan aplikasi *asmping*. Hasil pengujian adalah sebagai berikut :

```
root@host-A:~# ssm pingd
received request from 172.17.68.200
received request from 172.17.68.200
received request from 172.17.68.200
received request from 172.17.68.200
received request from 172.17.68.200
```

Gambar 5.6 Pengujian Koneksi *Multicast* pada *Host A* sebagai *Server*

Sumber : Pengujian

```
root@host-B:~# asmping -4 -c 5 224.1.2.234 172.17.67.61
ssm ping joined (S,G) = (172.17.67.61,224.1.2.234)
pinging S from 172.17.68.200
  unicast from 172.17.67.61, seq=1 dist=8 time=2.783 ms
  unicast from 172.17.67.61, seq=2 dist=8 time=1.520 ms
  multicast from 172.17.67.61, seq=2 dist=6 time=1.666 ms
  unicast from 172.17.67.61, seq=3 dist=8 time=2.235 ms
  multicast from 172.17.67.61, seq=3 dist=6 time=2.557 ms
  unicast from 172.17.67.61, seq=4 dist=8 time=1.961 ms
  multicast from 172.17.67.61, seq=4 dist=6 time=2.078 ms
  unicast from 172.17.67.61, seq=5 dist=8 time=2.575 ms
  multicast from 172.17.67.61, seq=5 dist=6 time=2.852 ms

--- 172.17.67.61 ssm ping statistics ---
5 packets transmitted, time 5003 ms
unicast:
  5 packets received, 0% packet loss
  rtt min/avg/max/std-dev = 1.520/2.214/2.783/0.451 ms
multicast:
  4 packets received, 0% packet loss since first mc packet (seq 2) recvd
  rtt min/avg/max/std-dev = 1.666/2.288/2.852/0.454 ms
```

Gambar 5.7 Pengujian Koneksi *Multicast* pada *Host B* sebagai *Client*

Sumber : Pengujian

```

root@host-B:~# ssm pingd
received request from 172.17.67.61
received request from 172.17.67.61
received request from 172.17.67.61
received request from 172.17.67.61
received request from 172.17.67.61

```

Gambar 5.8 Pengujian Koneksi *Multicast* pada *Host B* sebagai *Server*

Sumber : Pengujian

```

root@host-A:~# asmping -4 -c 5 224.1.2.234 172.17.68.200
ssmping joined (S,G) = (172.17.68.200,224.1.2.234)
pinging S from 172.17.67.61
  unicast from 172.17.68.200, seq=1 dist=8 time=1.172 ms
  unicast from 172.17.68.200, seq=2 dist=8 time=1.357 ms
  multicast from 172.17.68.200, seq=2 dist=6 time=1.501 ms
  unicast from 172.17.68.200, seq=3 dist=8 time=1.128 ms
  multicast from 172.17.68.200, seq=3 dist=6 time=1.295 ms
  unicast from 172.17.68.200, seq=4 dist=8 time=1.094 ms
  multicast from 172.17.68.200, seq=4 dist=6 time=1.293 ms
  unicast from 172.17.68.200, seq=5 dist=8 time=1.915 ms
  multicast from 172.17.68.200, seq=5 dist=6 time=2.155 ms

--- 172.17.68.200 ssmping statistics ---
5 packets transmitted, time 5000 ms
unicast:
  5 packets received, 0% packet loss
  rtt min/avg/max/std-dev = 1.094/1.333/1.915/0.305 ms
multicast:
  4 packets received, 0% packet loss since first mc packet (seq 2) recvd
  rtt min/avg/max/std-dev = 1.293/1.561/2.155/0.353 ms

```

Gambar 5.9 Pengujian Koneksi *Multicast* pada *Host B* sebagai *Server*

Sumber : Pengujian

5.1.2 Pengujian *Multimedia Streaming*

Pengujian *multimedia streaming* dilakukan dengan melakukan proses *streaming* dari satu *host* ke *host* lainnya dengan satu *host* sebagai *streaming source* sedangkan *host* lainnya sebagai *receiver* (satu arah). Pengujian ini bertujuan untuk mengetahui apakah proses *streaming* pada jaringan *multicast* dapat dilakukan. Kemudian pengujian dilakukan satu *host* sebagai *source* dan empat *host* menjadi *receiver*. Pengujian ini dilakukan dengan metode *unicast* dan *multicast* untuk mengetahui perbandingan penggunaan *bandwidth* jika menggunakan dua metode tersebut.

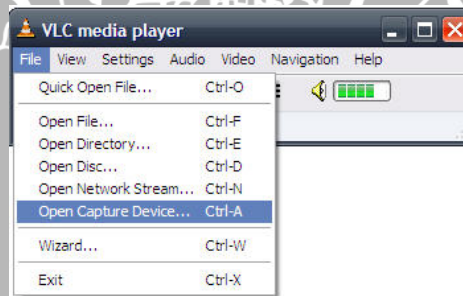
5.1.2.1 Pengujian *Multimedia Streaming* dengan Metode *Unicast*

Pengujian dilakukan dengan *host A* sebagai *source* dan *host B* sebagai *receiver* kemudian sebaliknya. Proses *streaming* menggunakan aplikasi VLC sebagai *streaming server (source)* maupun *streaming client (receiver)*.

Prosedur pengujian :

6. *Host A* menjadi *streaming source* dan *host B* menjadi *streaming receiver*. *Host A* melakukan proses *streaming* dengan aplikasi VLC. *Video source* berasal dari webcam sedangkan *audio source* berasal dari *microphone*. Proses *streaming* menggunakan protokol HTTP dan ditujukan pada alamat IP 172.17.67.38 *port* 2222. *Video* dan *audio* codec menggunakan MPEG 4 dengan *bitrate video* 256kbps dan *audio* 64kbps. Proses *streaming* adalah sebagai berikut :

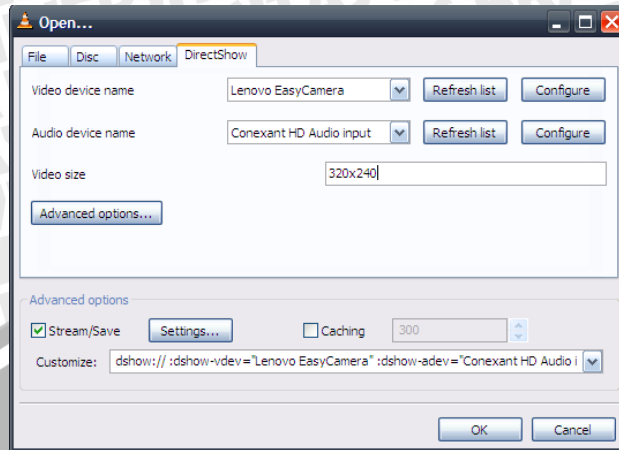
- *Open Capture Device* dengan menekan Ctrl + A.



Gambar 5.10 Pemilihan *Capture Device* sebagai *source streaming*

Sumber : Pengujian

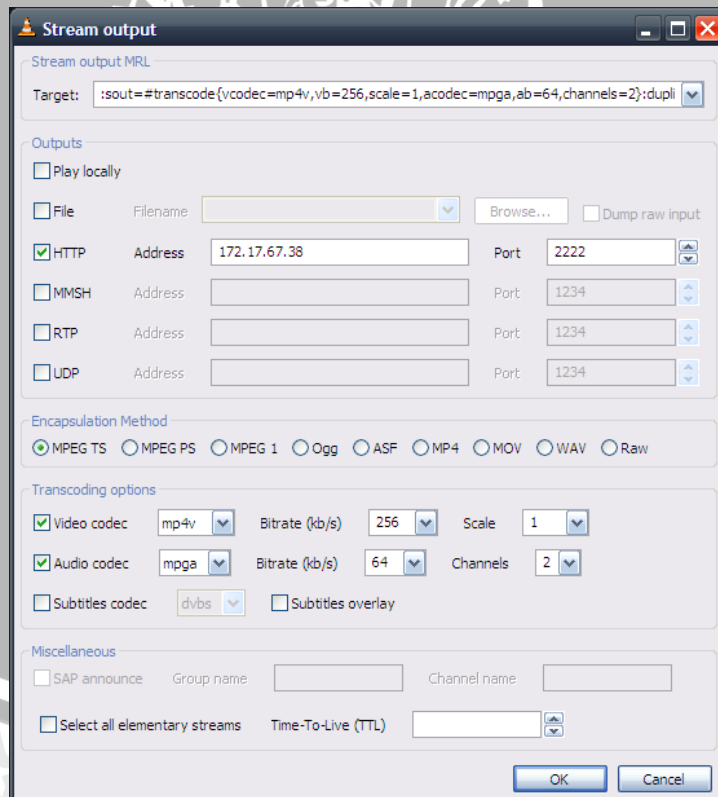
- Pilih *device* yang digunakan pada pilihan *Video device name* dan *Audio device name*.



Gambar 5.11 Pemilihan device (input) streaming

Sumber : Pengujian

- Pilih menu *Stream/Save* kemudian klik *Settings*. Pada gambar berikut ditunjukkan konfigurasi pada *stream* output untuk *host A*.

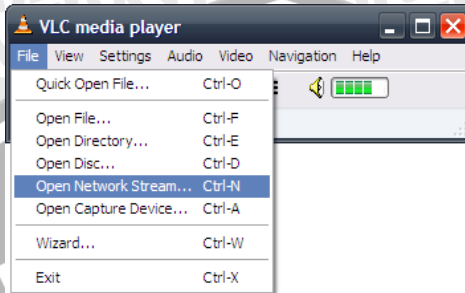


Gambar 5.12 Konfigurasi output streamingunicast

Sumber : Pengujian

7. *Host B* menjadi *streaming client* dengan membuka *streaming* dari *host A* dengan aplikasi VLC. Proses *play stream* dari *host A* adalah sebagai berikut :

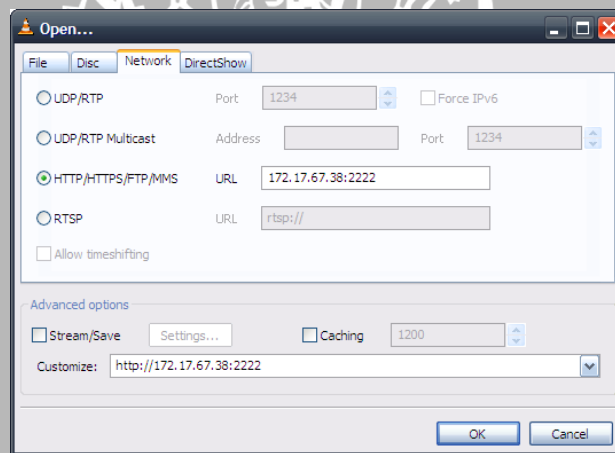
- *Open Network Stream* dengan menekan tombol Ctrl+N



Gambar 5.13 Pemilihan *Network stream* sebagai *source streaming*

Sumber : Pengujian

- Menentukan protokol dan alamat IP yang digunakan *streaming source*.



Gambar 5.14 Menentukan protokol dan alamat IP *streaming source*.

Sumber : Pengujian

Jika proses *streaming* berhasil maka pada *host B* akan dapat menampilkan hasil *streaming* dari *host A*. Gambar 5.14 menunjukkan tampilan hasil *streaming* pada *host B* dari *host A*.



Gambar 5.15 Hasil streaming dari host A ke host B.

Sumber : Pengujian

8. Host B menjadi *streaming source* dan host A menjadi *streaming receiver*. Prosedur pengujian sama dengan prosedur sebelumnya dengan alamat IP yang digunakan adalah alamat IP host B yaitu 172.17.68.212.

Tabel 5.2 Hasil Pengujian Streaming dengan Metode Unicast

Pengujian	Host A Source Host B Receiver	Host B Source Host A Receiver
Pengujian ke 1	Streaming dapat berjalan.	Streaming dapat berjalan.
Pengujian ke 2	Streaming dapat berjalan.	Streaming dapat berjalan.
Pengujian ke 3	Streaming dapat berjalan.	Streaming dapat berjalan.
Pengujian ke 4	Streaming dapat berjalan.	Streaming dapat berjalan.

Sumber : Pengujian

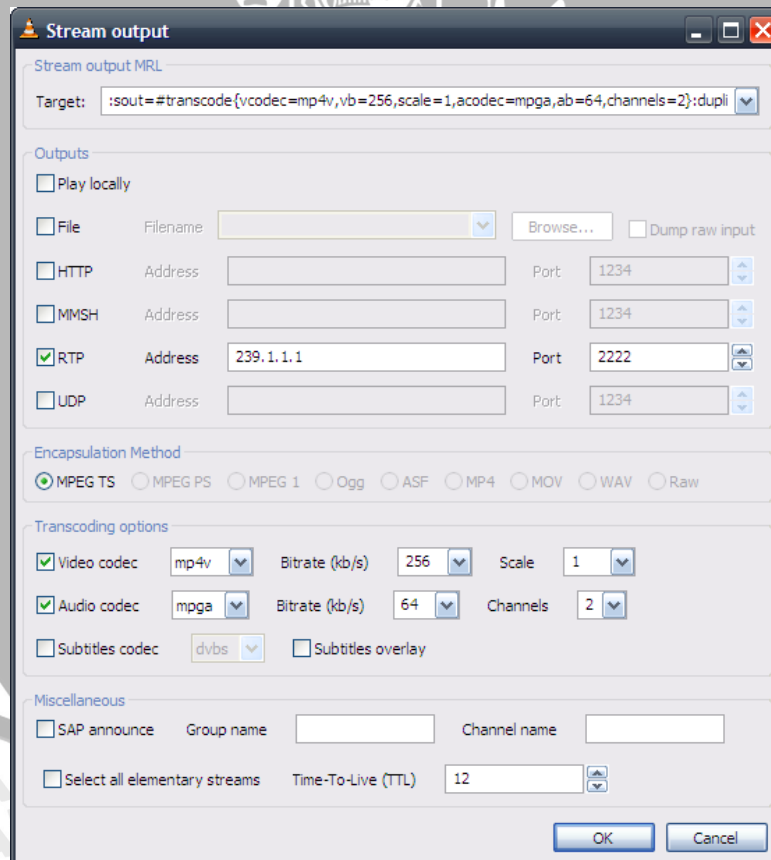
5.1.2.2 Pengujian Multimedia Streaming Dengan Metode Multicast

Pengujian dilakukan dengan host A dan host B bergantian menjadi *multicast source* dan *multicast receiver*. Proses *streaming* menggunakan aplikasi VLC sebagai *streaming server (source)* maupun *streaming client (receiver)*.

Prosedur pengujian :

1. *Host A* menjadi *streaming source* dan *host B* menjadi *streaming receiver*. *Host A* melakukan proses *streaming* dengan aplikasi VLC. *Video source* berasal dari webcam sedangkan *audio source* berasal dari *microphone*. Proses *streaming* menggunakan protokol HTTP 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 adalah 12. Proses *streaming* adalah sebagai berikut :

- *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*. Pada gambar berikut ditunjukkan konfigurasi pada *stream output* untuk *host A*.

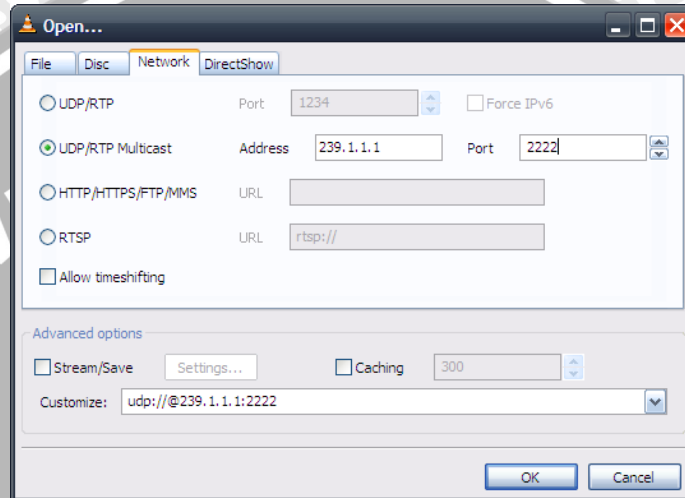


Gambar 5.16 Konfigurasi *output streaming multicast*

Sumber : Pengujian

2. *Host B* menjadi *streaming client* dengan membuka *streaming* dari *host A* dengan aplikasi VLC. Proses *play stream* dari *host A* adalah sebagai berikut :

- *Open Network Stream* dengan menekan tombol Ctrl+N
- Menentukan protokol dan alamat IP yang digunakan *streaming source*.



Gambar 5.17 Menentukan protokol dan alamat IP *streaming source*.

Sumber : Pengujian

Jika proses *streaming* berhasil maka pada *host B* akan dapat menampilkan hasil *streaming* dari *host A*. Gambar 5.17 menunjukkan tampilan hasil *streaming* pada *host B* dari *host A*.



Gambar 5.18 Hasil *streaming* dari *host A* ke *host B*.

Sumber : Pengujian

3. *Host B* menjadi *streaming source* dan *host A* menjadi *streaming receiver*. Prosedur pengujian sama dengan prosedur sebelumnya dengan alamat *multicast* yang digunakan adalah 239.1.1.2.

Tabel 5.3 Hasil Pengujian *Streaming* dengan Metode *Unicast*

Pengujian	<i>Host A Multicast Source</i> <i>Host B Multicast Receiver</i>	<i>Host B Multicast Source</i> <i>Host A Multicast Receiver</i>
Pengujian ke 1	<i>Streaming</i> dapat berjalan.	<i>Streaming</i> dapat berjalan.
Pengujian ke 2	<i>Streaming</i> dapat berjalan.	<i>Streaming</i> dapat berjalan.
Pengujian ke 3	<i>Streaming</i> dapat berjalan.	<i>Streaming</i> dapat berjalan.
Pengujian ke 4	<i>Streaming</i> dapat berjalan.	<i>Streaming</i> dapat berjalan.

Sumber : Pengujian

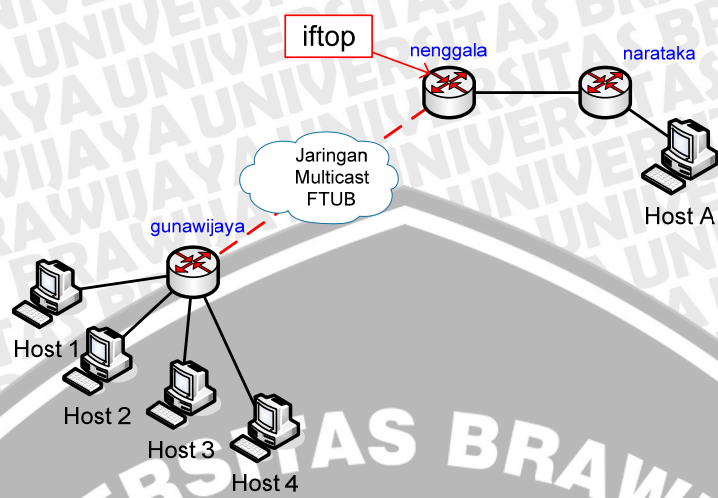
5.1.2.3 Pengujian Penggunaan *Bandwidth Multimedia Streaming*

Pengujian dilakukan dengan melakukan proses *streaming* dari *host A* sebagai *source* ke *host B*. Kemudian proses *streaming* dilakukan dengan *host A* sebagai *source* dan 4 *host* pada Net TPTIFT. Keempat *host* tersebut adalah sebagai berikut :

Tabel 5.4 Spesifikasi *Hardware Host* untuk Pengujian *Bandwidth*

	Operating System	IP Address	Spesifikasi Hardware
<i>Host 1</i>	Ubuntu 8.04	172.17.68.157	Pentium III 900Mhz, 256MB, 20GB
<i>Host 2</i>	Ubuntu 8.04	172.17.68.200	Pentium 4 2.6 Ghz, 512MB, 200GB
<i>Host 3</i>	Windows XP SP2	172.17.68.204	Intel Core to Duo, 2GB, 160 GB
<i>Host 4</i>	Windows XP SP2	172.17.68.212	Centrino, 768MB, 80GB

Sumber : Pengujian



Gambar 5.19 Konfigurasi Pengujian Penggunaan *Bandwidth Unicast dan Multicast*

Sumber : Pengujian

Pengukuran *bandwidth* menggunakan aplikasi iftop yang dijalankan di router Nenggala selama 30 menit pada saat proses *streaming* berlangsung.

Tabel 5.5 Hasil Pengujian *Bandwidth Multimedia Streaming* dengan Metode *Unicast*

Keterangan	Average rates	Peak rates
Pengujian <i>streaming</i> dari <i>host A</i> ke <i>host B</i>	431 Kbps	446 Kbps
Pengujian <i>streaming</i> dari <i>host A</i> ke 4 <i>host</i> .	1,81 Mbps	1.87 Mbps

Sumber : Pengujian

```

172.17.68.212:40983 => 172.17.67.38:2222      8.94Kb  8.54Kb  6.09Kb
                    <=                          431Kb   446Kb   311Kb
-----
TX:          628KB  peak:   10.4Kb  rates:  8.94Kb  8.54Kb  6.09Kb
RX:          49.8MB  529Kb   431Kb   446Kb   311Kb
TOTAL:      50.4MB  539Kb   440Kb   455Kb   317Kb
    
```

Gambar 5.20 *Streaming* dengan Metode *Unicast* dari *Host A* ke *Host B*

Sumber : Pengujian

```

172.17.68.157:42309 => 172.17.67.38:2222      7.52Kb  8.47Kb  8.81Kb
                    <=<                    440Kb  460Kb  452Kb
172.17.68.100:46772 => 172.17.67.38:2222      9.89Kb  9.12Kb  9.55Kb
                    <=<                    453Kb  471Kb  461Kb
172.17.68.204:33344 => 172.17.67.38:2222     10.2Kb  10.1Kb  6.88Kb
                    <=<                    448Kb  465Kb  455Kb
172.17.68.212:50923 => 172.17.67.38:2222      9.9Kb   10.3Kb  6.88Kb
                    <=<                    468Kb  476Kb  470Kb
-----
TX:          cumm:   759Kb  peak:   37.99Kb  rates:  37.51Kb  37.99Kb  32.12Kb
RX:          56.3MB   1.08Mb   1.81Mb   1.87Mb   1.84Mb
TOTAL:      57.0MB   1.10Mb   1.84Mb   1.91Mb   1.87Mb
    
```

Gambar 5.21 Streaming dengan Metode Unicast dari Host A ke 4 Host

Sumber : Pengujian

Tabel 5.6 Hasil Pengujian Bandwidth Multimedia Streaming dengan Metode Multicast

Keterangan	Average rates	Peak rates
Pengujian streaming dari host A ke host B	456 Kbps	478 Kbps
Pengujian streaming dari host A ke 4 host.	461 Kbps	482 Kbps

Sumber : Pengujian

```

239.1.1.1          => 172.17.67.38          0b      0b      0b
                    <=<                    456Kb   478Kb   466Kb
-----
TX:          cumm:   0B   peak:   0b   rates:   0b   0b   0b
RX:          34.2MB   478Kb   456Kb   478Kb   466Kb
TOTAL:      34.2MB   478Kb   456Kb   478Kb   466Kb
    
```

Gambar 5.22 Pengujian Streaming dengan Metode Multicast dari Host A ke Host B

Sumber : Pengujian

```

239.1.1.1          => 172.17.67.38          0b      0b      0b
                    <=<                    461Kb   482Kb   459Kb
-----
TX:          cumm:   0B   peak:   0b   rates:   0b   0b   0b
RX:          35.1MB   482Kb   461Kb   482Kb   459Kb
TOTAL:      35.1MB   482Kb   461Kb   482Kb   459Kb
    
```

Gambar 5.23 Pengujian Streaming dengan Metode Multicast dari Host A ke 4 Host

Sumber : Pengujian

Dari pengujian ini dapat kita lihat bahwa untuk proses *streaming* dari *Host A* ke *Host B* (*one to one*) dengan metode *unicast* dan *multicast* tidak terdapat perbedaan signifikan dalam penggunaan *bandwidth*. Tetapi untuk proses *streaming* dari *Host A* ke 4 *Host* (*one to many*) terdapat perbedaan signifikan dalam penggunaan *bandwidth* yang dibutuhkan. Pada proses pengujian *streaming one to many*, metode *unicast* membutuhkan yang kurang lebih sama dengan 4 kali *bandwidth streaming* pada pengujian *one to one*. Sedangkan dengan menggunakan metode *multicast bandwidth* yang diperlukan pada *streaming one to one* maupun *one to many* tidak terdapat perbedaan signifikan.

5.1.3 Pengujian Aplikasi Kuliah Online

Pengujian dilakukan untuk mengetahui apakah aplikasi Kuliah *Online* dapat berfungsi. Pengujian meliputi pengujian koneksi *host* ke *server*, pengujian halaman *web* Kuliah *Online* dan yang terakhir pengujian sistem Kuliah *Online*.

5.1.3.1 Pengujian Koneksi Host Ke Server

Pengujian dilakukan dengan melakukan ping dari *Host A* dan *Host B* ke alamat IP *server* yaitu 172.17.7.252. Hasil pengujian adalah sebagai berikut:

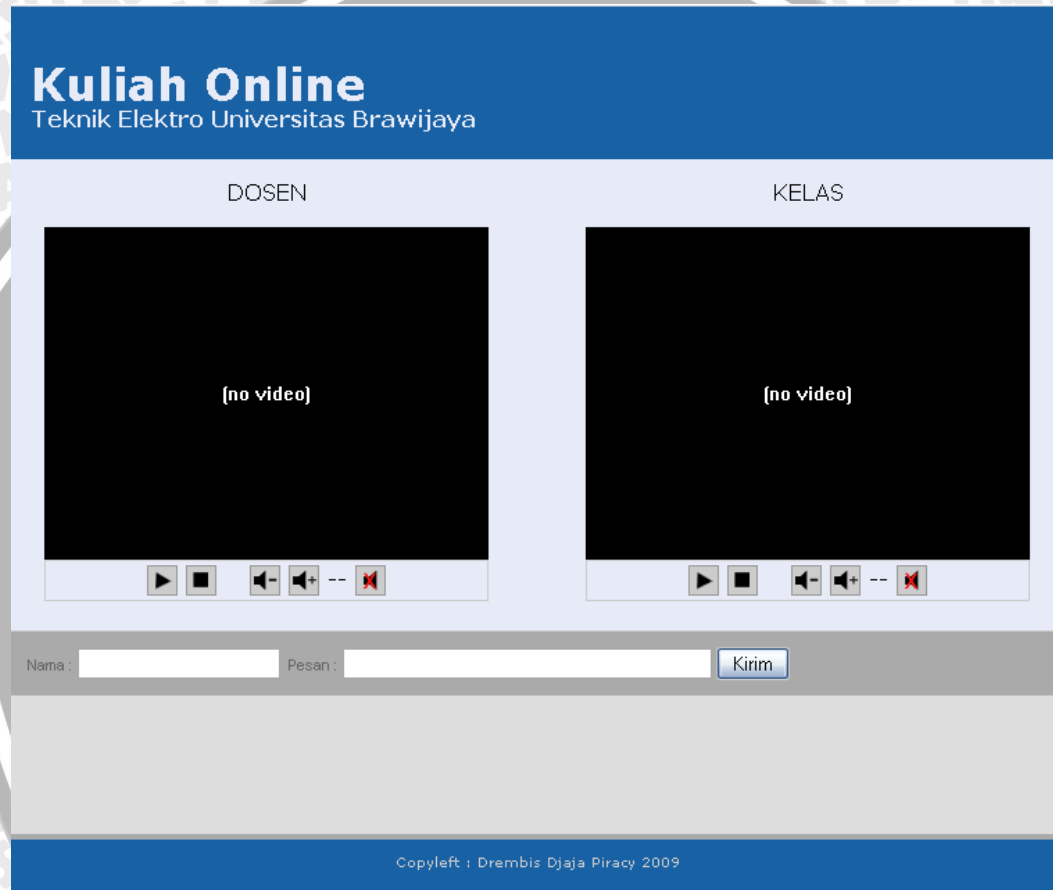
Tabel 5.7 Pengujian Koneksi Host ke Server

Pengujian Ke-	Keterangan
1	Berhasil
2	Berhasil
3	Berhasil
4	Berhasil
5	Berhasil

Sumber : Pengujian

5.1.3.2 Pengujian Halaman Web Kuliah Online

Pengujian halaman *web* Kuliah *Online* dilakukan dengan mengakses *server* Kuliah *Online* melalui *browser*. Pengujian dilakukan dengan menggunakan *browser Internet Explorer* dan mengakses URL <http://172.17.7.252/Kuliah/>. Hasil pengujian adalah sebagai berikut :



Gambar 5.24 Pengujian Halaman Web Kuliah Online

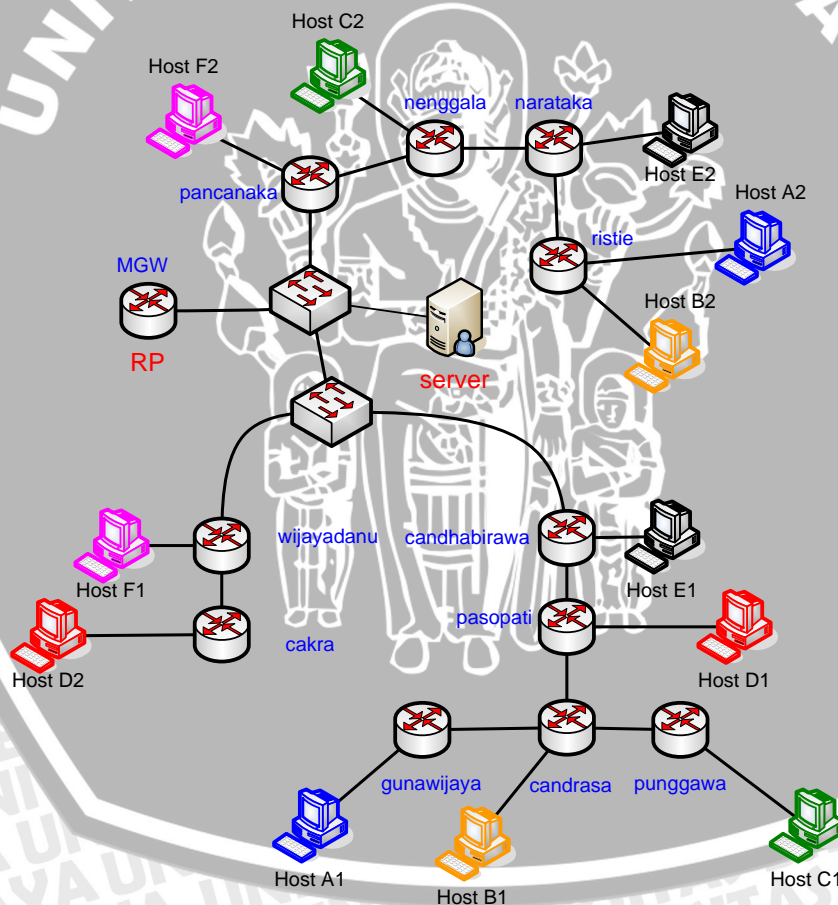
Sumber : Pengujian

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

5.1.4 Pengujian Sistem Kuliah Online

Pengujian sistem Kuliah *Online* meliputi pengujian fungsi keseluruhan sistem yang telah dirancang. Proses pengujian yang dilakukan adalah sebagai berikut :

1. Pengujian dilakukan pada saat jam sibuk pada jaringan komputer FTUB yaitu pada pukul 11.30 hingga 13.00. Pengujian dilaksanakan pada tanggal 16 Juli 2009.
2. Pengujian dilakukan dengan 12 *host* yang melakukan proses steaming secara bersamaan. Topologi pengujian adalah sebagai berikut :



Gambar 5.25 Topologi pengujian Sistem Kuliah Online

3. Proses *streaming* antara 6 pasang *host* yaitu :
 - *Host A1 – Host A2*

- *Host B1 – Host B2*
- *Host C1 – Host C2*
- *Host D1 – Host D2*
- *Host E1 – Host E2*
- *Host F1 – Host F2*

4. Lokasi *host* dan IP Address yang digunakan adalah sebagai berikut :

Tabel 5.8 Lokasi *host* dan IP Address yang digunakan.

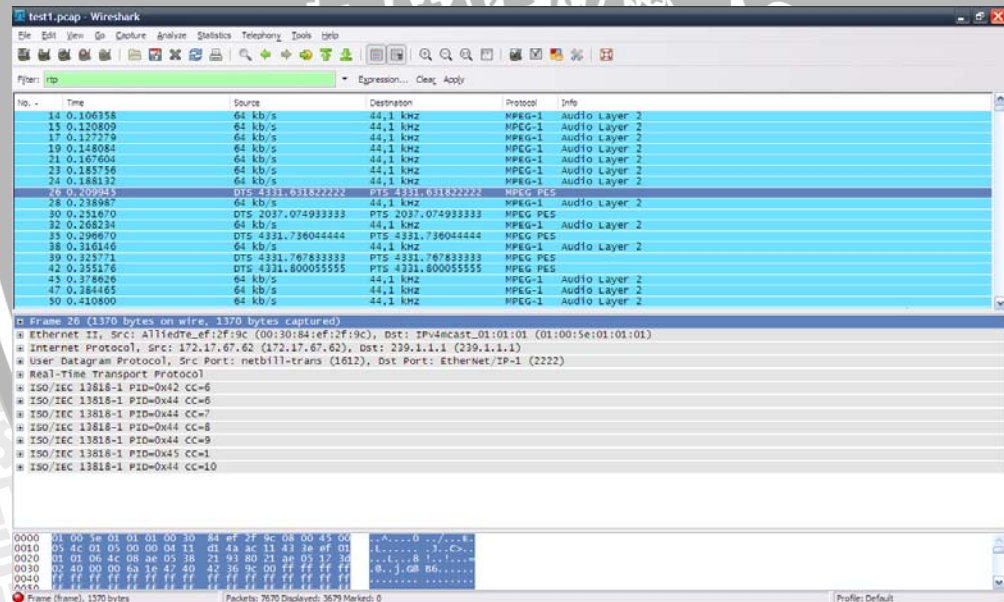
No	Host	Lokasi	IP Address	Multicast Address	Gateway
1	Host A1	TPTIFT	172.17.68.212	239.1.1.1	172.17.68.129
2	Host A2	RisTIE (Gedung Kemahasiswaan HME Lt.2)	172.17.67.38	239.1.1.2	172.17.67.41
3	Host B1	Ruangan Sub Bag Kepegawaian FTUB	172.17.60.152	239.1.1.3	172.17.60.41
4	Host B2	RisTIE (Gedung Kemahasiswaan HME Lt.2)	172.17.67.37	239.1.1.4	172.17.67.41
5	Host C1	Lab. Komputer PWK (Gedung Baru FT Lt.3)	172.17.66.131	239.1.1.5	172.17.66.129
6	Host C2	Ruangan Router Teknik Mesin (Gedung Mesin I Lt.2)	172.17.62.214	239.1.1.6	172.17.62.41
7	Host D1	Lab. Komputer Arsitek (Gedung Kuliah Arsitek Lt.2)	172.17.65.156	239.1.1.7	172.17.65.129
8	Host D2	Lab. DKP Teknik Elektro (GBE Lt.1)	172.17.63.32	239.1.1.8	172.17.63.1
9	Host E1	Perpustakaan Teknik Pengairan (Gedung Lab. Hidrolika Lt.2)	172.17.64.35	239.1.1.9	172.17.64.1
10	Host E2	Gedung Kuliah Teknik Elektro Lt.2	172.17.67.72	239.1.1.10	172.17.67.65
11	Host F1	Ruangan Router Teknik Pengairan (Gedung Rekording Pengairan Lt.2)	172.17.64.201	239.1.1.11	172.17.64.129
12	Host F2	Ruangan Lab. Telekomunikasi Teknik Elektro (GBE Lt.2)	172.17.63.42	239.1.1.12	172.17.63.1

Sumber : Pengujian

5. Masing masing *host* 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* yang telah ditentukan dan menggunakan *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*.
6. Masing masing *host* mengakses hasil *multimedia streaming* yang sedang dilakukan melalui *client interface* pada browser.
 7. Meng-*capture* paket data yang melalui *interface* jaringan pada masing masing *host* dengan menggunakan aplikasi *Wireshark*. Hasil capture di decode menjadi paket RTP. Proses peng-capture-an paket data pada tiap *host* dilakukan selama 10 kali dengan periode masing masing 60 detik.



Gambar 5.26 Screenshot Wireshark

Sumber : Pengujian

Dari pengujian menggunakan *software wireshark* di dapat hasil sebagai berikut :

Tabel 5.9 Hasil capture data pada *host A1* dengan filter RTP untuk paket data dari *multicast address host A2*

Pengujian ke	t (detik)	Captured packet	Packet Loss	Jumlah data	Jitter
1	60	2592	3	3549070	1,05
2	60	2609		3573810	1,04
3	60	2665	3	3649170	1,02
4	60	2613	4	3577370	1,00
5	60	2652	3	3631023	0,98
6	60	2643		3620838	1,02
7	60	2658		3640650	0,99
8	60	2663		3647920	1,03
9	60	2617	1	3583780	1,02
10	60	2627	1	3597512	1,02

Sumber : Pengujian

Tabel 5.10 Hasil capture data pada *host A2* dengan filter RTP untuk paket data dari *multicast address host A1*

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2624	1	3593704	0,98
2	60	2653	1	3633413	1,05
3	60	2661		3645221	1,05
4	60	2604		3567347	1,03
5	60	2591	3	3547728	1,00
6	60	2569	2	3518202	1,06
7	60	2609	3	3572082	1,03
8	60	2605	4	3566382	1,01
9	60	2602	3	3562576	0,97
10	60	2598	2	3557627	1,01

Sumber : Pengujian

Tabel 5.11 Hasil capture data pada *host B1* dengan filter RTP untuk paket data dari *multicast address host B2*

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2614	3	3579579	1,02
2	60	2691	1	3685319	0,99
3	60	2499	2	3422006	0,95
4	60	2675	4	3662507	1,01
5	60	2677		3667245	0,93
6	60	2688		3682092	1,00
7	60	2661	1	3644286	0,96
8	60	2656	2	3636599	1,00
9	60	2626		3596971	0,90
10	60	2659		3642285	0,94

Sumber : Pengujian

Tabel 5.12 Hasil capture data pada *host* B2 dengan filter RTP untuk paket data dari *multicast address host* B1

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	2	3554795	0,96
8	60	2615	3	3580171	0,95
9	60	2664	1	3648545	0,99
10	60	2679	2	3668167	0,94

Sumber : Pengujian

Tabel 5.13 Hasil capture data pada *host* C1 dengan filter RTP untuk paket data dari *multicast address host* C2

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 5.14 Hasil capture data pada *host* C2 dengan filter RTP untuk paket data dari *multicast address host* C1

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 5.15 Hasil capture data pada *host* D1 dengan filter RTP untuk paket data dari *multicast address host* D2

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 5.16 Hasil capture data pada *host* D2 dengan filter RTP untuk paket data dari *multicast address host* D1

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		3569280	0,86
10	60	2556	3	3499913	0,82

Sumber : Pengujian

Tabel 5.17 Hasil capture data pada *host* E1 dengan filter RTP untuk paket data dari *multicast address host* E2

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2654		3635807	0,64
2	60	2573	1	3523949	0,66
3	60	2635		3609818	0,65
4	60	2584		3539079	0,63
5	60	2618		3585913	0,63
6	60	2569	2	3517488	0,64
7	60	2649	1	3627864	0,68
8	60	2634	4	3606047	0,64
9	60	2665		3650498	0,64
10	60	2579		3532514	0,64

Sumber : Pengujian

Tabel 5.18 Hasil capture data pada *host* E2 dengan filter RTP untuk paket data dari *multicast address host* E1

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2674	1	3661584	0,63
2	60	2627		3597972	0,68
3	60	2601		3562635	0,63
4	60	2585	1	3540114	0,63
5	60	2606		3569011	0,63
6	60	2608	3	3573026	0,66
7	60	2601	1	3561989	0,64
8	60	2612		3577360	0,70
9	60	2642		3618310	0,65
10	60	2678	3	3669027	0,62

Sumber : Pengujian

Tabel 5.19 Hasil capture data pada *host* F1 dengan filter RTP untuk paket data dari *multicast address host* F2.

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2560	2	3504970	0,53
2	60	2610	3	3573313	0,51
3	60	2616	1	3582694	0,50
4	60	2591		3549320	0,49
5	60	2636		3610800	0,50
6	60	2587		3543282	0,49
7	60	2590	1	3546827	0,49
8	60	2600	1	3559994	0,53
9	60	2611		3576704	0,53
10	60	2653		3633575	0,53

Sumber : Pengujian

Tabel 5.20 Hasil capture data pada *host* F2 dengan filter RTP untuk paket data dari *multicast address host* F1

Pengujian ke	t (detik)	Captured	Packet Loss	Jumlah data	Jitter
1	60	2567		3516010	0,49
2	60	2638		3613528	0,53
3	60	2629		3600790	0,54
4	60	2573		3524350	0,48
5	60	2565		3513633	0,53
6	60	2637	1	3611576	0,47
7	60	2552		3496078	0,52
8	60	2564	4	3510313	0,50
9	60	2612		3577645	0,52
10	60	2574	3	3524819	0,51

Sumber : Pengujian



Setelah data dari tabel 5.10 hingga tabel 5.20 ditabulasi didapat hasil sebagai berikut :

Tabel 5.21 Tabulasi hasil pengujian

No	Host	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	Host A1	26339	2633,9	15	3005928,583	1,017
1	Host A2	26116	2611,6	19	2980356,833	1,019
3	Host B1	26446	2644,6	13	3018240,750	0,970
4	Host B2	26358	2635,8	14	3008090,167	0,974
5	Host C1	26360	2636,0	12	3008600,917	0,890
6	Host C2	26411	2641,1	11	3014275,583	0,893
7	Host D1	26386	2638,6	9	3011611,750	0,854
8	Host D2	26090	2609,0	10	2977711,500	0,847
9	Host E1	26160	2616,0	8	2985748,083	0,645
10	Host E2	26234	2623,4	9	2994252,333	0,647
11	Host F1	26054	2605,4	8	2973456,583	0,510
12	Host F2	25911	2591,1	8	2957395,167	0,509
	TOTAL	314865,000	31486,500	136,000	35935668,250	9,775

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.

5.1.5 Pengujian Mean Opinion Square

Pengujian MOS dilakukan untuk mengetahui kualitas *audio video* multimedia *streaming*. Pengujian dilakukan pada kualitas *audio*, kualitas *video*

dan delay. Pengujian MOS dilakukan pada *stream* dari *host* A1 dan *host* A2 dengan menggunakan opini dari 24 responden. Standar penilaian kualitas untuk *audio* dan *video* adalah sebagai berikut :

Tabel 5.22 Indeks kualitas *audio* dan *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 5.23 Indeks pengaruh delay

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

Sumber : Pengujian

5.1.5.1 Pengujian kualitas *audio*

Pengujian kualitas *audio* dilakukan dengan merekam *audio* hasil *streaming* dengan durasi selama 60 detik kemudian memperdengarkan rekaman *audio* tersebut kepada 24 responden. Hasil pengujian seperti pada tabel 5.24.

Tabel 5.24 Hasil analisa kualitas *audio* dengan metode MOS

Skor	Kualitas	Responden	Total
5	Sangat Baik	-	
4	Baik	9	36
3	Sedang	15	45
2	Buruk	-	
1	Sangat Buruk	-	
		TOTAL	81

Sumber : Pengujian

Dari tabel 5.24 dapat diketahui nilai MOS adalah sebagai berikut :

Jika dihitung dengan persamaan berikut maka :

$$\begin{aligned} \text{MOS} &= \frac{\text{Total skor}}{\text{Jumlah responden}} \\ &= 81 / 24 \\ &= 3,37 \end{aligned}$$

Nilai MOS untuk kualitas *audio* adalah 3,37. (Sedang)

5.1.5.2 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 5.25.

Tabel 5.25 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 5.25 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)

5.1.5.3 Pengujian pengaruh delay

Pengujian pengaruh delay dilakukan dengan meminta responden untuk menyaksikan langsung proses *streaming*. Hasil pengujian ditampilkan pada Tabel 5.26

Tabel 5.26 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 5.26 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).

5.2 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.

Pada skripsi ini kompresi *video* yang digunakan adalah MPEG-4 dengan bitrate 256 Kbps, sedangkan untuk kompresi *audio* menggunakan teknologi MPEG *Audio* dengan bitrate 64 Kbps. Paket data *audio* dan *video* di enkapsulasi menggunakan teknologi MPEG *Transport Stream* (MPEG TS). Pada metode enkapsulasi MPEG TS, data *audio* dan *video* di enkapsulasi menjadi satu data *multimedia* kemudian di pecah menjadi beberapa *transport stream* kecil untuk

dikirim. MPEG TS membatasi panjang *transport stream* (TS) sebesar 188 byte yang terdiri dari *header* sebesar 4 byte dan optional *header* serta payload dengan panjang maksimum 184 byte. Dalam satu MTU, MPEG TS membatasi jumlah TS yang boleh dikirim adalah 7 paket TS. Dari sini kita bisa menghitung berapa besar panjang *frame* dan berapa payload data yang bisa ditampung dalam satu *frame*.

Panjang *frame* didapat dari perhitungan sebagai berikut :

$$\begin{aligned} \text{panjang frame} &= (7 \times \text{paket}_{\text{MPEG TS}}) + \text{Header}_{(\text{RTP} + \text{UDP} + \text{IP} + \text{Ethernet})} \\ &= (7 \times 188 \text{ byte}) + (8 + 12 + 20 + 14) \text{ byte} \\ &= 1370 \text{ byte} \end{aligned}$$

Dari perhitungan di atas dapat kita lihat bahwa MPEG TS membatasi jumlah paket TS yang dikirim dalam satu MTU sebanyak 7 paket TS agar nilai panjang *frame* tidak melebihi dari nilai MTU yang diperbolehkan, yaitu 1500 byte. Dengan panjang satu paket TS sebesar 188 bytes, *header* TS sebesar 4 byte, dan dalam satu MTU terdapat 7 paket TS maka jika diasumsikan optional *header* tidak ada, maka payload maksimum data *multimedia* dalam satu MTU adalah sebagai berikut :

$$\begin{aligned} \text{Payload} &= 7 \times (\text{Panjang paket}_{\text{MPEG TS}} - \text{Header}_{\text{MPEG TS}}) \\ &= 7 \times (188 - 4) \text{ byte} \\ &= 1288 \text{ byte} \end{aligned}$$

Sehingga payload maksimum data *multimedia* dalam satu MTU adalah sebesar 1288 byte.

5.2.1 Analisa Troughput

Throughput dihitung dengan membagi jumlah paket yang diterima dengan waktu transmisi sesuai dengan Persamaan (2-3).

Dari perhitungan didapat troughput untuk tiap tiap *host* adalah sebagai berikut :

Troughput *Host A1*

$$\bar{\lambda} = \frac{1}{t_v} = \frac{2633,9}{60} = 43,90 \text{ paket / s}$$

Tabel 5.27 Troughput pada *host*

No	Host	Troughput
1	Host A1	43,90
1	Host A2	43,53
3	Host B1	44,08
4	Host B2	43,93
5	Host C1	43,93
6	Host C2	44,02
7	Host D1	43,98
8	Host D2	43,48
9	Host E1	43,60
10	Host E2	43,72
11	Host F1	43,42
12	Host F2	43,19
	Total	524,78
	Rata Rata	43,73

Sumber : Pengujian

Dari hasil perhitungan diketahui bahwa troughput rata – rata untuk tiap *stream* adalah 43,73 paket/s sedangkan troughput total pada jaringan pada saat pengujian adalah 524,78 paket/s.

5.2.2 Analisa Bandwidth Multimedia Streaming

Pada skripsi ini kompresi *video* yang digunakan adalah MPEG-4 dengan bitrate 256 Kbps, sedangkan untuk kompresi *audio* menggunakan teknologi MPEG *Audio* dengan bitrate 64 Kbps. Paket data *audio* dan *video* di enkapsulasi menggunakan teknologi MPEG TS. Besar panjang paket *multimedia* adalah sebesar 1316 bytes.

Dari data tersebut dapat kita hitung *bandwidth multimedia streaming* dari masing masing *stream* pada tiap *host* sesuai dengan persamaan 2-2 :

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

Dari hasil perhitungan berdasarkan persamaan diatas untuk tiap tiap *host* didapat hasil berikut:

Bandwidth multimedia streaming host A1

$$\begin{aligned} BW_{multimedia} &= N_{paket\ multimedia} \times (M_{paket\ multimedia} + Header_{(RTP+UDP+IP+Ethernet)}) \times 8 \\ &= 43,90 \text{ paket/s} \times (1370 \text{ byte}) \times 8 \\ &= 469,85 \text{ kbps} \end{aligned}$$

Tabel 5.28 Bandwidth multimedia streaming pada host

No	Host	BW Multimedia (Kbps)
1	Host A1	469,85
1	Host A2	465,87
3	Host B1	471,76
4	Host B2	470,19
5	Host C1	470,22
6	Host C2	471,13
7	Host D1	470,69
8	Host D2	465,41
9	Host E1	466,66
10	Host E2	467,98
11	Host F1	464,77
12	Host F2	462,21
	Total	5616,73
	Rata Rata	468,06

Sumber : Pengujian

Dari tabel diatas didapat didapat *bandwidth* total yang dibutuhkan pada saat pengujian adalah sebesar 5,616 Mbps. Sedangkan *bandwidth* rata rata untuk satu *stream multimedia* adalah 468,06 Kbps.

5.2.3 Analisa Kapasitas Jaringan

Pada analisa kapasitas jaringan dilakukan perhitungan untuk mengetahui dua hal. Yang pertama adalah kapasitas maksimum jaringan komputer FTUB jika digunakan untuk *multimedia streaming*. Dari hasil analisa tersebut kita akan mengetahui berapa jumlah maksimum *stream* yang dapat dilakukan dan berapa jumlah Kuliah Online yang bisa dilakukan. Yang kedua adalah perhitungan kapasitas optimum jaringan FTUB jika digunakan untuk *multimedia streaming*. Kapasitas optimum adalah jika koefisien collision tidak lebih dari 10%.

Untuk perhitungan kapasitas maksimum dihitung dengan menggunakan Persamaan (2-5):

$$\text{Kapasitas}_{\max} = \frac{\text{BW}_{\text{jaringan}} \times E}{\text{BW}_{\text{streaming}}}$$

Dimana :

BW_{link} = Bandwidth maksimum jaringan yaitu 70Mbps (eksisting)

E = Efisiensi jaringan

$\text{BW}_{\text{streaming}}$ = Bandwidth rata rata untuk satu *stream* yaitu 468,06 Kbps

Dari persamaan diatas kita belum mengetahui besar efisiensi jaringan. Efisiensi jaringan pada saat pengujian dapat dihitung dengan persamaan (2-6) :

$$E = \left(1 + \frac{2BLE}{cF}\right)^{-1}$$

Dimana :

E = Efisiensi jaringan

B = troughput total dua *stream* (87 paket/s)

L = panjang kabel (1300 m)

c = delay propagasi (1,565 ms)

F = panjang paket (1370 byte)

e = selisih waktu kedatangan antar paket sebesar 9,6 μ s (Minolli, 2001)

Sehingga didapat efisiensi jaringan sebesar 88,71 %. Setelah kita mengetahui efisiensi jaringan maka kita dapat mengetahui kapasitas maksimum jaringan untuk aplikasi multimedia *streaming* adalah :

$$\begin{aligned} \text{Kapasitas}_{\max} &= \frac{\text{BW}_{\text{jaringan}} \times E}{\text{BW}_{\text{streaming}}} \\ &= \frac{70 \text{ Mbps} \times 94,02\%}{468,06 \text{ Kbps}} \\ &= 136 \text{ stream} \end{aligned}$$

Dari perhitungan diatas dapat diketahui kapasitas maksimum jaringan untuk multimedia *streaming* adalah 136 *stream* atau 68 pasang kelas dan dosen.

Sedangkan untuk kapasitas optimum dapat diketahui dengan menggunakan persamaan berikut :

$$\text{Koefisien}_{\text{packet collision}} = \frac{\text{Jumlah Data}_{\text{max}}}{\text{BW}_{\text{jaringan}}} \times 100\%$$

Jumlah Data_{max} : Jumlah data total yang tercapture selama pengujian

BW_{jaringan} : *Bandwidth* jaringan sebesar 70 Mbps.

Koefisien_{packet collision} : Koefisien *packet collision* ≤ 10%

Dari persamaan diatas BW_{jaringan} dan Koefisien_{packet collision} telah kita ketahui dan yang kita cari adalah besar Jumlah Data_{max} yang dapat dilewatkan jaringan dengan optimal. Setelah dilakukan perhitungan sesuai dengan persamaan diketahui besar Jumlah Data_{max} yang bisa dilewatkan adalah :

$$10\% = \frac{\text{Jumlah Data}_{\text{max}}}{70\text{Mbps}} \times 100\%$$

$$\begin{aligned} \text{Jumlah Data}_{\text{max}} &= 0,1 \times 70\text{Mbps} \\ &= 7340032 \text{ bit.} \end{aligned}$$

Untuk menghitung jumlah *stream* optimum maka kita membandingkan antara

Jumlah Data max dengan BW_{streaming}

$$\begin{aligned} \text{Jumlah stream optimum} &= \frac{\text{Jumlah Data}_{\text{max}}}{\text{BW}_{\text{link}}} \\ &= \frac{7340032 \text{ bit}}{468,06 \text{ Kbps}} \end{aligned}$$

$$= 22 \text{ stream}$$

Jumlah optimum *streaming* adalah 22 *stream* atau 11 *stream* full duplex.

5.2.4 Analisa Packet Loss

Packet loss adalah banyaknya paket data yang hilang saat transmisi. Semakin besar prosentase *packet loss* maka semakin buruk performansi sebuah saluran transmisi. Besar *packet loss* dapat dihitung dengan Persamaan (2-8)

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

Dari hasil perhitungan berdasarkan persamaan diatas untuk tiap tiap *host* didapat hasil berikut:

Packet loss host A1

$$\begin{aligned} N_{\text{packet loss}} (\%) &= (15 / (26339+15)) \times 100\% \\ &= 0,057 \% \end{aligned}$$

Tabel 5.29 *Packet loss pada host*

No	Host	Pakcket Loss (%)
1	Host A1	0,057
1	Host A2	0,073
3	Host B1	0,049
4	Host B2	0,053
5	Host C1	0,046
6	Host C2	0,042
7	Host D1	0,034
8	Host D2	0,038
9	Host E1	0,031
10	Host E2	0,034
11	Host F1	0,031
12	Host F2	0,031
	Total	0,518
	Rata Rata	0,043

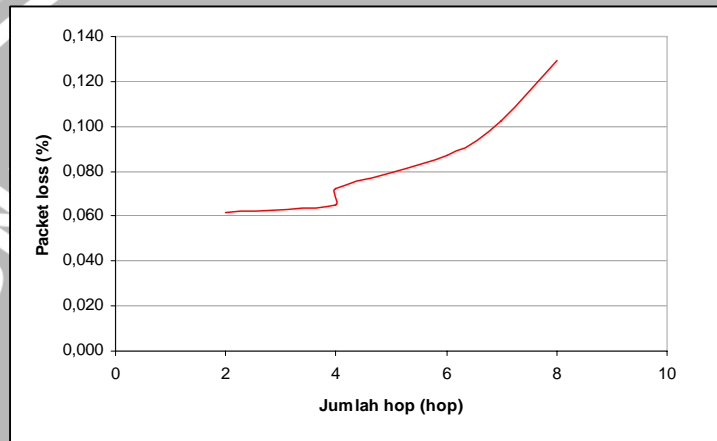
Sumber : Pengujian

Prosentase paket loss total pada saat pengujian adalah 0,518% sedangkan prosentase pakcet loss rata rata untuk tiap *stream* adalah 0,043%. Dari hasil analisa dapat diketahui semakin banyak hop yang dilewati paket data maka *packet loss* akan semakin besar. Grafik hubungan antara banyaknya hop dengan *packet loss* adalah sebagai berikut :

Tabel 5.30 Hubungan antara banyaknya hop dengan nilai *packet loss*

No	Stream dari	Jumlah hop (hop)	Packet loss (%)
1	Host F1 - Host F2	2	0,062
2	Host E1 - Host E2	4	0,065
3	Host D1 - Host D2	4	0,072
4	Host C1 - Host C2	6	0,087
5	Host B1 - Host B2	7	0,102
6	Host A1 - Host A2	8	0,130

Sumber : Pengujian



Gambar 5.27 Hubungan jumlah hop dengan *Packet Loss*

Sumber : Pengujian

5.2.5 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.

5.2.6 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

5.2.7 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 $0,25 \mu s$

- 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 $0,55 \mu s$.
- 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 $0,194ms$.

$$\begin{aligned} t_p &= (24 \times 0,25 \mu s) + ((1300/100) \times 0,55 \mu s) + (8 \times 0,194ms) \\ &= 0,006 ms + 0,00715 ms + 1,552ms \\ &= 1,565ms \end{aligned}$$

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

5.2.8 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 :

$$\begin{aligned} t_w &= t_{queue} + t_{serv} \\ &= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \end{aligned}$$

Dengan asumsi :

- 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

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

- 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

$$\mu = \frac{C}{lx8}$$

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

$$\begin{aligned} \mu &= \frac{C}{lx8} \\ &= \frac{104857600}{1370 \times 8} \\ &= 9567,29 \text{ paket / s} \end{aligned}$$

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.

5.2.9 Analisa Delay Jitter

Aplikasi *Wireshark* mampu melakukan perhitungan *delay jitter* dengan menghitung selisih waktu kedatangan paket sebuah paket (n) dengan paket

sebelumnya (n-1). Perhitungan delay jitter pada aplikasi wireshark menggunakan persamaan 2-14 . Dari pengujian didapat nilai rata rata untuk *delay jitter* transmisi dari tiap tiap *host* adalah sebagai berikut :

Tabel 5.31 Delay jitter pada host

No	Host	Delay Jitter (ms)
1	Host A1	1,017
1	Host A2	1,019
3	Host B1	0,970
4	Host B2	0,974
5	Host C1	0,890
6	Host C2	0,893
7	Host D1	0,854
8	Host D2	0,847
9	Host E1	0,645
10	Host E2	0,647
11	Host F1	0,510
12	Host F2	0,509
	Rata Rata	0,978

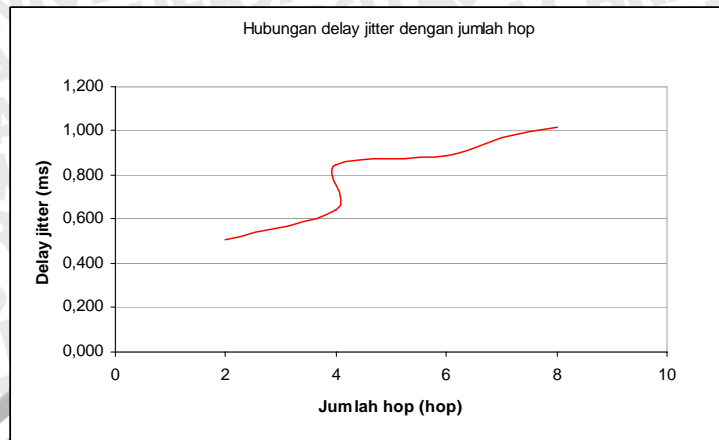
Sumber : Pengujian

Dari hasil analisa dapat diketahui bahwa delay jitter rata rata untuk tiap *stream* adalah 0,978 ms. Dari hasil analisa dapat diketahui semakin banyak hop yang dilewati paket data maka delay jitter juga akan semakin besar. Grafik hubungan antara banyaknya hop dengan delay jitter adalah sebagai berikut :

Tabel 5.32 Hubungan delay jitter dengan jumlah hop

No	Stream dari	Jumlah hop (hop)	Delay jitter rata rata (ms)
1	Host F1 - Host F2	2	0,510
2	Host E1 - Host E2	4	0,646
3	Host D1 - Host D2	4	0,850
4	Host C1 - Host C2	6	0,891
5	Host B1 - Host B2	7	0,972
6	Host A1 - Host A2	8	1,018

Sumber : Pengujian



Gambar 5.28 Hubungan jumlah hop dengan Delay Jitter

Sumber : Pengujian

5.2.10 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 :

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

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.
6. t_{jitter} didapat dari hasil pengujian sebesar 0,709 ms.

$$\begin{aligned} t_{end-to-end} &= 50ms + 1,48ms + 0,105ms + 1,565ms + 0,105ms + 0,709ms + 50ms \\ &= 103,964 \text{ ms} \end{aligned}$$

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.



BAB VI PENUTUP

Setelah melalui tahap perancangan dan implementasi sistem, *multicast routing* pada jaringan *intranet* FTUB dapat berjalan dengan baik sehingga proses multimedia *streaming* dengan metode *multicast* menggunakan protokol RTP 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.

6.1 Kesimpulan

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

1. Topologi jaringan *multicast* tidak berbeda dengan jaringan eksisting. Agar jaringan dapat mendukung *multicast routing* perlu ditambahkan aplikasi XORP pada *router* yang dapat melakukan *multicast forwarding*.
2. *Routing multicast* dengan XORP mengambil informasi dari *routing unicast* untuk melewati data pada tujuan yang benar. Pada *routing multicast* dengan menggunakan protokol PIM-SM diperlukan sebuah titik sebagai muara dari paket *multicast* yang disebut *Rendezvous Point*.
3. Pengiriman paket *video* dan *audio* dari *multicast source* ke masing-masing *receiver* atau sebaliknya, menggunakan protokol RTP yang ditumpangkan pada protokol UDP sebagai protokol kontrol dalam pengirimannya data multimedia.
4. Data *audio* di kompresi menggunakan *codec* MPEG *Audio* dengan *bitrate* 64kbps. Sedangkan untuk data *video* dikompresi

5. menggunakan kompresi MPEG 4 dengan bitrate 256kbps dan di enkapsulasi dengan menggunakan teknologi MPEG TS.
6. Dengan metode *multicast*, proses *multimedia streaming one-to-one* ataupun *one-to-many* tidak terdapat perbedaan signifikan dalam kebutuhan *bandwidth*. Sedangkan dengan metode *unicast*, kebutuhan *bandwidth* pada saat proses *multimedia streaming one-to-many* hampir setara dengan 4 kali kebutuhan *bandwidth* untuk *streaming one-to-one*. Hal ini membuktikan bahwa *multicast* dengan metode replikasinya lebih efisien dalam penggunaan *bandwidth*.
7. *Bandwidth* rata-rata yang dibutuhkan untuk aplikasi multimedia *streaming* untuk Kuliah Online adalah 468,06 kbps untuk satu *stream*. Sedangkan *throughput* rata-rata untuk tiap satu *stream* adalah 43,90 paket/s.
8. Dari pengujian didapat nilai *packet loss* untuk tiap *stream* rata-rata sebesar 0,043% sedangkan *packet loss* total pada saat pengujian berlangsung adalah 0,518%. Hal ini menunjukkan bahwa jaringan *multicast* yang dibangun *reliable* dalam proses *packet delivery*.
9. Kapasitas maksimum jaringan FTUB untuk aplikasi Kuliah Online adalah 136 *stream*. Sedangkan kapasitas optimumnya adalah 22 *stream*.
10. Dari analisa MOS didapat bahwa kualitas *audio* dikategorikan sedang (nilai MOS 3,37), kualitas *video* sedang (nilai MOS 3,29), sedangkan untuk pengaruh delay dikategorikan mengganggu (nilai MOS 2,29).
11. *Delay end-to-end* merupakan penjumlahan keseluruhan *delay* yang ada. Berdasarkan hasil pengujian didapatkan *delay end-to-end* adalah sebesar 104,71 ms yang masih di dalam ambang batas *delay* yang diperbolehkan untuk *realtime streaming*.
12. Dari keseluruhan hasil pengujian dan analisa dapat ditarik kesimpulan bahwa proses Kuliah *Online* pada jaringan komputer FTUB dengan metode *multicast* dapat dijalankan dengan baik.

6.2 Saran

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

1. Aplikasi Kuliah Online yang dibuat masih sangat sederhana dengan fitur-fitur yang belum lengkap. Diharapkan aplikasi yang telah dibuat dapat dikembangkan lebih lanjut.
2. Teknologi *multicast* yang sudah ada dapat dikembangkan hingga ke seluruh jaringan *intranet* Universitas Brawijaya.
3. *Multicast routing* dapat diterapkan juga untuk jaringan komputer yang menggunakan media *wireless*.
4. Dapat dikembangkan untuk menghubungkan jaringan *multicast* yang dihubungkan oleh *network non-multicast*.



DAFTAR PUSTAKA

- Anonim. 2007. ” *eXtensible Open Router Platform (XORP)* ”. XORP.
<http://xorp.org/>
- Anonim. 2007. *Network Topology* .
http://en.wikipedia.org/wiki/Network_topology
- Anonim. 2007. ” *User Datagram Protocol* “. Wikipedia Encyclopedia.
http://en.wikipedia.org/wiki/User_Datagram_Protocol.
- Austerberry, David. 2005. “The Technology of Video and Audio Streaming”.
USA : Focal Press
- Dhoto. 2003. ” *Jaringan Komputer dan Internet* ”. Ilmukomputer.com. 2003
- Latre, Alexis de, et al., 2005. “*VideoLAN Streaming How To*”.
<http://videolan.org/streaming/>
- Murai, Jun., 1999. “IP Multicast Routing”, SOI Japan .
<http://www.soi.wide.ad.jp/class/99007/slides/25/>
- Perkins, Colin. 2003. ”RTP: Audio and Video for the Internet”. USA : Addison
Wesley
- RFC 2362, “Protokol Independent Multicast-Sparse Mode (PIM-SM): Protokol
Specification,” D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering,
M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, 1998
- RFC 2117, “Protokol Independent Multicast-Sparse Mode (PIM-SM): Protokol
Specification,” D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering,
M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, 1997
- RFC 1889. 1996. “*RTP: A Transport Protocol for Real-Time Applications*”. H.
Schulzrinne, S. Casner, R. Frederick, V. Jacobson
- TPTIFT UB, 2008. Data Inventaris Jaringan Intranet FTUB 2008. Malang :
Fakultas Teknik Universitas Brawijaya.
- Vigato, Alberto. 2005. “IPTV Broadcasting Over Satellite using current MPEG-2
Transport Stream”. Padova : University of Padova

LAMPIRAN

Tabel 6.1 Hasil kuesioner kualitas audio 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 6.2 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 6.3 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					√