

**PERANCANGAN DAN IMPLEMENTASI STREAMING VIDEO
MULTICAST DI PTIIK UNIVERSITAS BRAWIJAYA**

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

Nyoman Wira Prasetya

NIM. 0710683030

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2013



LEMBAR PERSETUJUAN

PERANCANGAN DAN IMPLEMENTASI STREAMING VIDEO MULTICAST DI PTIIK UNIVERSITAS BRAWIJAYA

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

Nyoman Wira Prasetya

NIM. 0710683030

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah R. A, S.T, M.Eng

NIP. 820809 06 1 1 0084

Wibisono S W, S.T, M.T

NIK. 82040406110091



KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa (ﷲ) yang telah melimpahkan karuniaNya sehingga penulis dapat menyelesaikan skripsi dengan judul “Perancangan dan Implementasi Streaming Video Multicast di PTIIK Universitas Brawijaya”. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan – bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Orang tua dan kakak-kakak yang selalu sabar menemani, memberikan dorongan dan motivasi, serta membantu secara langsung dalam proses penyusunan skripsi ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Bapak Sabriansyah R. A. S.T, M.Eng dan Bapak Wibisono S W, S.T, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing, memotivasi dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
5. Bapak Ir. Heru Nurwasito, M.Kom selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
6. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.

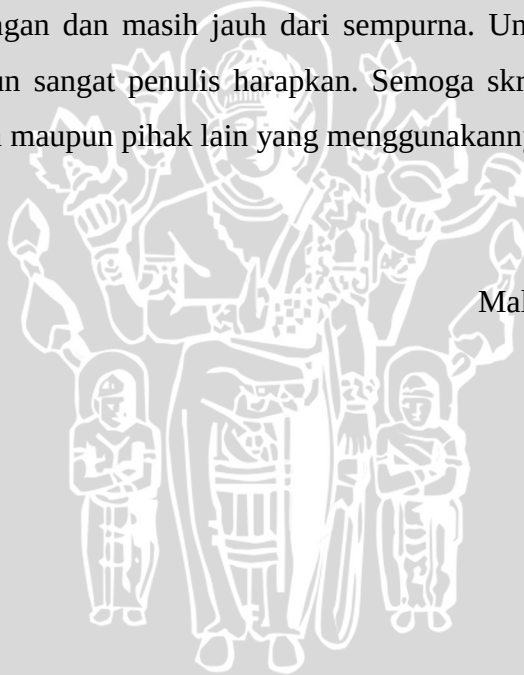
7. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Keluarga TPL-07, terimakasih atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.

Serta seluruh pihak yang tidak mungkin disebutkan satu persatu dalam tulisan ini, semoga diberi kesempatan untuk membantu penulis lagi

Hanya doa yang bisa penulis berikan semoga Tuhan Yang Maha Esa memberikan balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 3 Maret 2013

Penulis



ABSTRAK

Nyoman Wira Prasetya. 2013. : Perancangan dan Implementasi Streaming Video Multicast di PTIIK Universitas Brawijaya.

Dosen Pembimbing : Sabriansyah R A, S.T, M.Eng dan Wibisono S W, S.T, M.T

Video merupakan sarana penyampaian informasi yang sangat lengkap dan dapat diimplementasikan menggunakan teknologi streaming. Layanan jaringan untuk pertukaran media secara lokal sangat membantu kegiatan belajar mengajar, misalnya pemutaran video pembelajaran secara simultan. Penyampaian informasi berupa video ini dapat dilakukan dengan pengiriman paket secara *unicast* maupun *multicast*.

Streaming video multicast menggunakan protokol PIM Sparse mode. Dalam implementasinya router PTIIK dijadikan *Rendezvous Point* (RP). Pengujian dilakukan dengan pemutaran video dengan beberapa penerima sekaligus menggunakan protokol HTTP dan RTP. Pengujian dilakukan pada jam sibuk dilingkungan PTIIK yang telah ditetapkan berada pada pukul 12.00-14.00. *Throughput* yang terjadi pada *multicast* (RTP) sebesar 0.537 Mbit/sec untuk 4 penerima, lebih kecil daripada *unicast* (HTTP) yaitu 1.291 Mbit/sec untuk 4 penerima sehingga jaringan tidak terbebani secara berlebihan. *Delay* dan jitter pada *unicast* lebih besar daripada *multicast* yang disebabkan *header unicast* lebih panjang daripada *multicast*. Pengujian juga dilakukan dengan melakukan *streaming* pada jaringan WiFi di PTIIK dimana terjadi kehilangan paket data yang tinggi. Hal tersebut disebabkan karena WiFi mengalami penurunan kualitas sinyal serta dipengaruhi oleh banyaknya pengguna yang mengakses WiFi tersebut. Dengan mengimplementasikan *multicast* pada jaringan PTIIK video streaming dapat diterima seluruh warga PTIIK secara simultan tanpa membebani lalu lintas jaringan secara berlebihan.

Kata Kunci : *streaming, multicast, video, delay, jitter.*

ABSTRACT

Nyoman Wira Prasetya. 2013. : *Design and Implementation of Video Streaming Multicast in PTIIK UB.*

Advisor : Sabriansyah R A, S.T, M.Eng and Wibisono S W, S.T, M.T

Video is a means of delivering information that is complete and can be implemented using streaming technology. Network services for the exchange of local media is very helpful teaching and learning activities, such as simultaneous playback of video tutorials. Delivery of information in the form of video can be done by sending a unicast or multicast packets.

Streaming video multicast using PIM Sparse mode protocols. In implementation, the router PTIIK used as Rendezvous Point (RP). Testing is done with video playback with multiple recipients using the HTTP protocol and RTP protocols. Tests conducted at busy hours in the environment of PTIIK set is at 12:00 to 14:00. Happens to the multicast throughput (RTP) at 0537 Mbit / sec for 4 receivers, smaller than unicast (HTTP) is a 1291 Mbit / sec for 4 receivers so the network is not burdened unnecessarily. Delay and jitter on unicast greater than multicast caused by unicast header longer than multicast header. Testing is also done by streaming on WiFi networks in PTIIK where there is high packet loss. This was due to decreased quality WiFi signal and is influenced by a number of users accessing the WiFi. By implementing multicast video streaming on a network PTIIK acceptable to all citizens PTIIK simultaneously, in other words PTIIK all citizens can receive streaming video on-the simultaneously without imposing excessive network traffic.

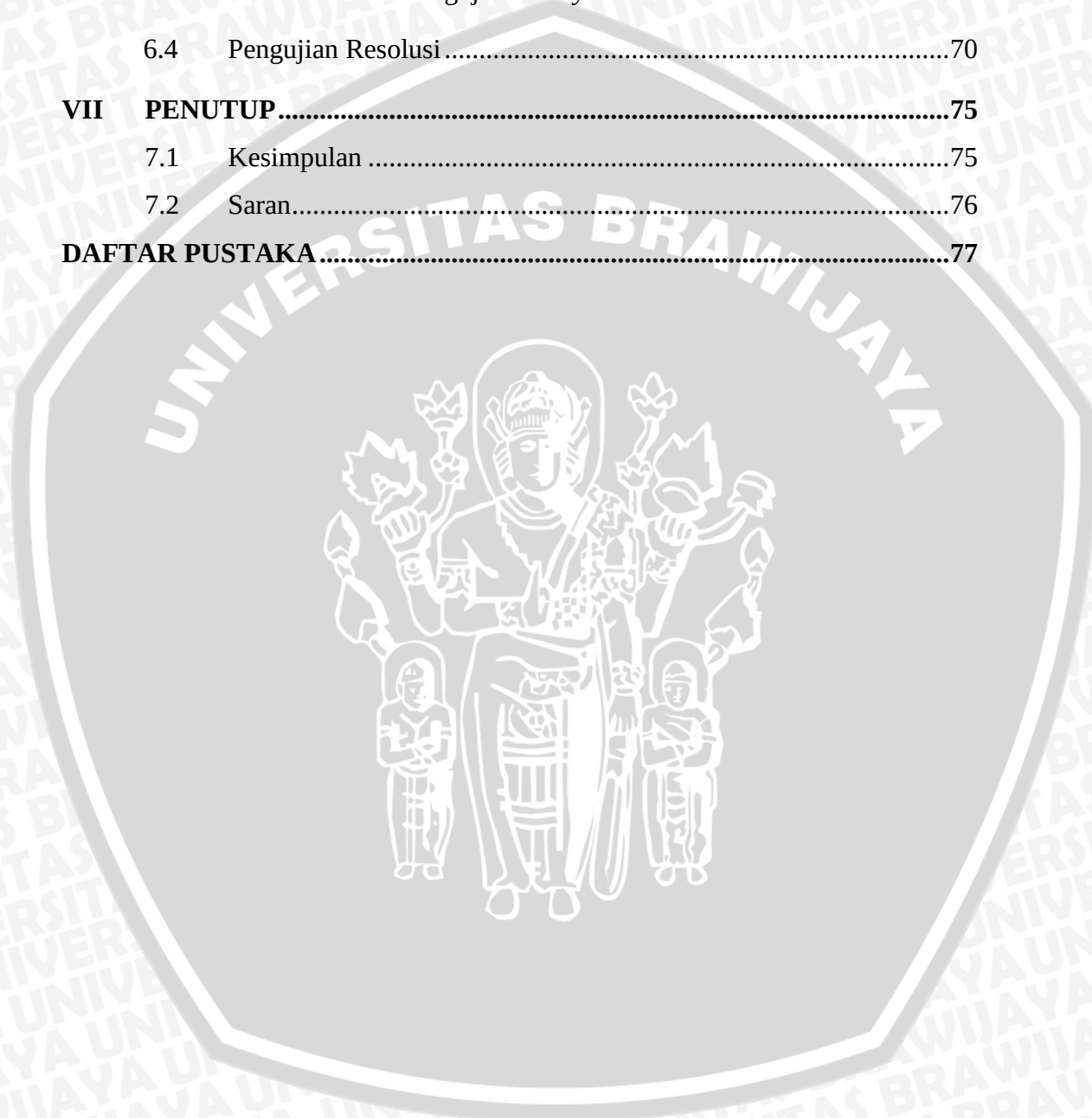
Keywords : streaming, multicast, video, delay, jitter.

DAFTAR ISI

KATA PENGANTAR.....	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR TABEL	viii
DAFTAR GAMBAR.....	ix
DAFTAR LAMPIRAN	xi
DAFTAR ISTILAH	xii
I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	5
II LANDASAN TEORI.....	7
2.1 Streaming	7
2.2 Arsitektur Streaming Video	8
2.2.1 Capture and Encode.....	9
2.2.2 Serving.....	9
2.2.3 Distribution	11
2.3 Prinsip Kerja Streaming Video	11
2.4 Protokol Streaming	12
2.5 Multicast.....	14
2.6 Codec.....	20
2.6.1 Video Codec	20
2.6.2 Audio Codec	21
2.7 Parameter Kualitas Layanan Streaming	22

III	METODE PENELITIAN	25
3.1	Studi Literatur	26
3.2	Perancangan	27
3.3	Implementasi	29
3.5	Pengujian dan Analisis	30
3.6	Pengambilan Kesimpulan dan Saran.....	37
IV	PERANCANGAN	38
4.1	Analisis Kebutuhan	38
4.1.1	Gambaran Umum	38
4.1.2	Identifikasi Pelaku.....	40
4.1.3	Kebutuhan Pelaku	41
4.1.4	Kebutuhan Sistem	42
4.2	Perancangan	44
4.2.1	Perancangan Infrastruktur	44
4.2.2	Penggunaan Perangkat Lunak	44
4.2.3	Perancangan <i>Multicast</i>	45
4.2.4	Proses <i>Streaming Video</i>	46
4.2.5	<i>Transfer Control Protocol (TCP)</i>	47
4.2.6	<i>User Datagram Protocol (UDP)</i>	47
V	IMPLEMENTASI.....	49
5.1	Spesifikasi Sistem	49
5.1.1	Spesifikasi Perangkat Keras	49
5.1.2	Penggunaan Perangkat Lunak	50
5.2	Batasan Implementasi	52
5.3	Topologi	52
5.4	Sistem.....	53
5.5	Kendala Implementasi.....	55
VI	PENGUJIAN DAN ANALISIS.....	56
6.1	Pengujian Dasar Jaringan.....	57
6.2	Pengujian <i>Throughput</i>	62

6.3	Pengujian <i>Delay</i> dan <i>Jitter</i>	64
6.2.1	HTTP	64
6.2.2	RTP	66
6.2.3	Analisa Pengujian <i>Delay</i> dan <i>Jitter</i>	68
6.4	Pengujian Resolusi	70
VII	PENUTUP	75
7.1	Kesimpulan	75
7.2	Saran	76
	DAFTAR PUSTAKA	77



DAFTAR TABEL

Tabel 2.1	Rekomendasi ITU-T G.114 untuk <i>delay</i>	24
Tabel 2.2	Standar <i>jitter</i>	24
Tabel 2.3	Standar <i>packet loss</i>	25
Tabel 4.1	Identifikasi pelaku	40
Tabel 4.2	Kebutuhan penyedia	41
Tabel 4.3	Kebutuhan penerima	41
Tabel 6.1	<i>Throughput</i> pada jaringan PTIIK untuk 1 dan 2 Penerima	62
Tabel 6.2	<i>Throughput</i> pada jaringan PTIIK untuk 4 dan 40 Penerima	63
Tabel 6.3	Pengujian HTTP jaringan PTIIK untuk 1 dan 2 Penerima	65
Tabel 6.4	Pengujian HTTP jaringan PTIIK untuk 4 dan 40 Penerima	65
Tabel 6.5	Pengujian RTP jaringan PTIIK untuk 1 dan 2 Penerima	67
Tabel 6.6	Pengujian RTP jaringan PTIIK untuk 4 dan 40 Penerima	67
Tabel 6.7	Tabel <i>delay</i> RTP dan HTTP untuk 1 dan 2 Penerima	68
Tabel 6.8	Tabel <i>delay</i> RTP dan HTTP untuk 4 dan 40 Penerima	68
Tabel 6.9	Tabel <i>jitter</i> RTP dan HTTP untuk 1 dan 2 Penerima	68
Tabel 6.10	Tabel <i>jitter</i> RTP dan HTTP untuk 4 dan 40 Penerima	68
Tabel 6.11	Hasil pengujian throughput dengan berbagai resolusi	71
Tabel 6.12	Tabel <i>delay</i> dari berbagai resolusi	72
Tabel 6.13	Tabel <i>jitter</i> dari berbagai resolusi	72
Tabel 6.14	Tabel <i>packet loss</i> dari berbagai resolusi	73

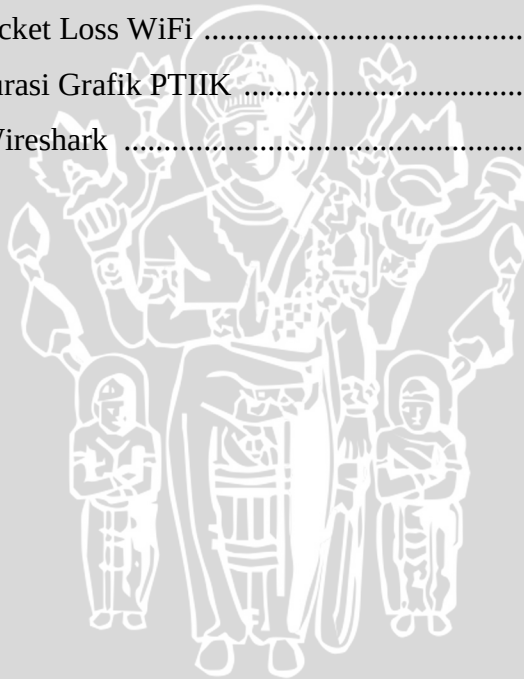
DAFTAR GAMBAR

Gambar 2.1	Arsitektur <i>streaming</i>	8
Gambar 2.2	<i>Live dan On-Demand</i>	10
Gambar 2.3	Prinsip kerja video <i>streaming</i>	12
Gambar 2.4	<i>Multicast dan Unicast</i>	15
Gambar 2.5	<i>Internet Group Management Protocol</i>	17
Gambar 2.6	<i>Sparse mode</i>	18
Gambar 2.7	<i>Dense mode</i>	19
Gambar 2.8	<i>Rendezvous point</i>	20
Gambar 3.1	Diagram alir keseluruhan pelaksanaan penelitian	26
Gambar 3.2	Diagram Alir Perancangan	28
Gambar 3.3	Proses <i>streaming</i>	29
Gambar 3.4	Diagram Alir Implementasi	30
Gambar 3.5	Diagram Alir Pengujian dan Analisis	31
Gambar 3.6	Diagram Alir Pengujian Dasar Jaringan	32
Gambar 3.7	Diagram Alir Pengujian <i>Throughput</i>	33
Gambar 3.8	Diagram Alir Pengujian <i>Deay dan Jitter</i>	35
Gambar 3.9	Diagram Alir Pengujian Resolusi	37
Gambar 4.1	Rencana sistem video <i>streaming multicast</i>	39
Gambar 4.2	Urutan langkah kerja video <i>streaming multicast</i>	40
Gambar 4.3	Penyedia/penerima dapat menerima <i>streaming</i> secara <i>unicast</i> maupun <i>multicast</i>	42
Gambar 4.4	Topologi Jaringan <i>Streaming PTIIK</i>	43
Gambar 4.5	Router PTIIK sebagai RP	46
Gambar 4.6	Proses <i>Streaming</i> melalui HTTP	47
Gambar 4.7	Proses <i>Streaming</i> melalui RTP	47
Gambar 5.1	Topologi PTIIK	53
Gambar 5.2	Desain topologi <i>multicast</i>	54
Gambar 6.1	Lalu lintas data PTIIK	56
Gambar 6.2	Tes ping pada beberapa penerima	58

Gambar 6.3	Video berjalan dengan menggunakan metode <i>unicast</i> pada jaringan PTIIK	59
Gambar 6.4	Penangkapan data <i>unicast</i> menggunakan <i>Wireshark</i>	60
Gambar 6.5	Video berjalan dengan menggunakan metode <i>multicast</i> pada jaringan PTIIK	61
Gambar 6.6	Penangkapan data <i>multicast</i> menggunakan <i>Wireshark</i>	61
Gambar 6.7	Grafik hasil pengujian <i>Throughput</i>	63
Gambar 6.8	Video berjalan menggunakan protokol HTTP pada port 8080	64
Gambar 6.9	Penangkapan data video HTTP	65
Gambar 6.10	Video berjalan menggunakan protokol RTP pada port 5004	66
Gambar 6.11	Penangkapan data video RTP dan analisa	67
Gambar 6.12	Grafik hasil pengujian <i>delay</i>	69
Gambar 6.13	Grafik hasil pengujian <i>jitter</i>	69
Gambar 6.14	Grafik hasil pengujian <i>Throughput</i> dengan berbagai resolusi	71
Gambar 6.15	Grafik hasil pengujian <i>delay</i> dengan berbagai resolusi	72
Gambar 6.16	Grafik hasil pengujian <i>jitter</i> dengan berbagai resolusi	73

DAFTAR LAMPIRAN

Lampiran 1	Instalasi Paket Multicast Mikrotik	80
Lampiran 2	Konfigurasi Multicast Mikrotik	81
Lampiran 3	Data Throughput	85
Lampiran 4	Data Delay	86
Lampiran 5	Data Jitter	87
Lampiran 6	Data Throughput WiFi	88
Lampiran 7	Data Delay WiFi	89
Lampiran 8	Data Jitter WiFi	90
Lampiran 9	Data Packet Loss WiFi	91
Lampiran 10	Konfigurasi Grafik PTIHK	92
Lampiran 11	Filter Wireshark	94



DAFTAR ISTILAH

Streaming

Sebuah metode penyampaian konten multimedia yang merepresentasikan konten yang datang secara langsung tanpa mengunduh seluruhnya.

Multicast

Merupakan transmisi dari satu pengirim ke banyak penerima dalam suatu grup

Unicast

Merupakan transmisi informasi yang dilakukan oleh satu pengirim kepada satu penerima yang lebih dikenal dengan transmisi *point to point*.

Delay

Merupakan jumlah waktu yang dibutuhkan untuk mengantar paket data dari asal ke tujuan

Jitter

Merupakan variasi keterlambatan (*delay*) dalam suatu transmisi data

Throughput

Merupakan kecepatan transfer aktual yang terukur pada suatu ukuran waktu tertentu

Packet Loss

Merupakan paket data yang hilang saat transfer data berlangsung

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejak ditemukan telepon, manusia dapat melakukan komunikasi jarak jauh dengan suara. Kini perkembangan teknologi kian pesat, manusia dapat berkomunikasi tidak hanya dengan teks maupun suara, tetapi juga secara visual dengan menggunakan gambar dan video. Teknologi video sangat berperan dalam ekonomi digital, salah satunya penggunaan video dalam perdagangan secara *online*, konsumen dapat langsung melihat dengan teliti barang yang akan dibeli melalui demonstrasi video yang direkam.

Video dapat dikatakan sebagai gabungan gambar-gambar mati yang dibaca berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame rate*. Sebagai salah satu media, video menjadi layanan komunikasi yang enak dinikmati, interaktif, dan dinamis karena mengandung gambar, teks, dan suara. Saat ini makin banyak masyarakat di Indonesia yang menggunakan aplikasi audio dan video dalam kehidupan sehari-harinya [ABR-10:1] seiring dengan semakin berkembangnya ilmu pengetahuan dan teknologi, khususnya komunikasi jarak jauh atau telekomunikasi, video dapat dengan mudah disebarluaskan melalui internet.

Terdapat dua metode penyampaian konten multimedia ke klien, yakni *streaming* dan *download* [SEP-09:1]. *Download* membutuhkan waktu dan tempat penyimpanan yang relatif besar pada *harddisk*, sedangkan *streaming* merepresentasikan konten yang datang secara langsung tanpa mengunduh seluruhnya. Konten pada *streaming* diletakkan pada memori penyangga (*buffer*) dan langsung direpresentasikan kemudian dibuang. Terdapat teknologi baru bernama *Progressive Download* dimana data disimpan di memori penyangga lalu diputar sebelum *download* selesai. Penyampaian konten multimedia berupa *streaming* memiliki beberapa metode keluaran yaitu HTTP, UDP, MMSH, RTP,

dan RTSP. Metode-metode tersebut mempunyai karakter penyampaian masing-masing dan dapat dilakukan dengan pengiriman paket secara *unicast* maupun *multicast*.

Transmisi *multicast* merupakan transmisi dari satu pengirim ke banyak penerima [SEP-09:2]. Jadi apabila terdapat 3 permintaan dari penerima sebesar 100kb/s maka total *throughput* yang dibutuhkan sebesar 100kb/s, berbeda dengan transmisi *unicast* yang akan memerlukan *throughput* sebesar 300kb/s. Pada dasarnya *multicast* membentuk suatu grup yang disebut *host group* dimana bila *server* mengirim data, grup yang dituju lah yang akan menerima, berbeda dengan *broadcast* yang akan menyampaikan ke semua *client* yang terhubung.

Program Teknologi Informasi dan Ilmu Komputer (PTIIK) sebagai salah satu fakultas di UB mempunyai jumlah mahasiswa dan dosen relatif banyak dan akan semakin bertambah. Hal tersebut menyebabkan penggunaan internet PTIIK menjadi tinggi. Jaringan di PTIIK ini dipakai oleh seluruh warga PTIIK yang pada umumnya digunakan untuk berselancar di dunia maya, melakukan pengunduhan, melakukan pengunggahan, dan memonitor CCTV. Layanan jaringan untuk pertukaran media secara lokal sampai saat ini masih belum ada, padahal pertukaran media secara lokal sangat membantu kegiatan belajar mengajar, misalnya pemutaran video pembelajaran secara simultan. Dengan mengimplementasikan *multicast* pada jaringan PTIIK, kita dapat *streaming*-kan video yang dapat diterima seluruh warga PTIIK secara simultan, dengan kata lain seluruh warga PTIIK dapat menerima siaran video yang *streaming*-kan secara bersamaan tanpa membebani lalu lintas jaringan secara berlebihan.

Sesuai latar belakang diatas, penulis mengambil judul **“PERANCANGAN DAN IMPLEMENTASI STREAMING VIDEO MULTICAST DI PTIIK UNIVERSITAS BRAWIJAYA”** dengan harapan penulis dapat membuat rancangan jaringan *streaming* intranet dalam kampus yang dapat digunakan untuk menyampaikan konten multimedia berbasis *streaming*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka dapat dirumuskan permasalahan sebagai berikut:

1. Pembahasan mengenai penyampaian konten multimedia di lingkungan kampus PTIIK.
2. Pembahasan gambaran umum dari sistem *multicast* yang akan diimplementasikan di PTIIK.
3. Pembahasan perancangan dan implementasi sistem *multicast* yang akan diimplementasikan di PTIIK.
4. Pembahasan sekenario, proses dan hasil dari pengujian sistem *multicast* di PTIIK.
5. Pembahasan mengenai perbandingan *throughput*, *delay*, *jitter*, dan *packet loss* antara *unicast* dengan *multicast*.
6. Membangun jaringan *multicast* menggunakan protokol RTP

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan lebih terfokus maka penelitian tugas akhir *streaming video multicast* dibatasi dalam hal:

1. Protokol jaringan *multicast* yang digunakan adalah IGMPv2 dan PIM
2. Versi IP yang digunakan adalah IPv4
3. Operasi *multicast* yang digunakan adalah *Sparse Mode*
4. Router yang digunakan pada tahap perancangan, implementasi, dan pengujian adalah Mikrobits Ainos 2071
5. Komunikasi yang digunakan dalam perancangan dan implementasi *streaming video multicast* adalah komunikasi suara dan video.

6. Pengujian dilakukan pada saat jam sibuk dimana telah ditetapkan pada pukul 12.00 hingga 14.00 lingkungan kampus PTIIK

1.4 Tujuan

Sesuai dengan latar belakang dan rumusan masalah, tujuan dari skripsi ini adalah untuk menyediakan sarana penyampaian konten multimedia di dalam kampus berbasis *streaming* menggunakan *multicast*. *Multicast* dirancang dengan menggunakan metode *PIM Sparse Mode*. Dalam implementasinya *router* PTIIK dijadikan *Rendezvous Point* (RP). Pengujian dilakukan berdasarkan *throughput*, *delay*, *jitter*, dan *packet loss* serta akan dilakukan perbandingan antara *unicast* dan *multicast*. Diharapkan sistem *multicast* ini dapat mendukung proses belajar mengajar di lingkungan PTIIK.

1.5 Manfaat

Manfaat yang nantinya dapat diambil dari pengembangan sistem *multicast* ini adalah sebagai berikut:

1. Bagi penulis
 - a) Menerapkan ilmu yang telah didapat di Teknik Informatika konsentrasi Komputasi Berbasis Jaringan.
 - b) Mendapatkan pemahaman lebih lanjut mengenai *streaming* video menggunakan *multicast*
2. Bagi Pengguna
 - a) Dosen dapat memberikan materi berupa video melalui *streaming* menggunakan jaringan *multicast* kepada mahasiswa, misalnya video tutorial
 - b) Mahasiswa dapat menikmati siaran *streaming* video *multicast* yang disiarkan *server*.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir *streaming video multicast* adalah:

Bab I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Dasar Teori

Berisi tentang dasar teori secara luas mengenai perangkat lunak maupun perangkat keras yang diperlukan untuk instalasi jaringan *server streaming* yang baik serta referensi mengenai metode *streaming*.

Bab III Metode Penelitian

Berisi tentang langkah-langkah dalam merancang jaringan *streaming* dan analisis metode keluaran dari *streaming*. Pada bab ini dijelaskan langkah-langkah yang akan dilakukan yaitu studi literatur, analisis kebutuhan, perancangan jaringan *multicast*, implementasi jaringan *multicast*, analisis dan pengujian dari jaringan *multicast* serta analisis dari metode keluaran *streaming*. Kesimpulan dan saran disertakan sebagai catatan dan kemungkinan arah pengembangan selanjutnya.

Bab IV Perancangan

Bab perancangan berisi tentang perencanaan sistem yang akan dibangun, meliputi perancangan infrastruktur, penggunaan perangkat lunak, dan proses *streaming video*.

Bab V Implementasi

Bab implementasi berisi tentang implementasi sistem yang dibangun, meliputi spesifikasi sistem, batasan implementasi, topologi, sistem, dan kendala dalam implementasi.

Bab VI Pengujian dan Analisis

Bab pengujian dan analisis berisi tentang pengujian dasar jaringan, pengujian *throughput*, pengujian *delay*, *jitter*, dan *packet loss* beserta analisisnya.

Bab VII Penutup

Pada bab penutup berisi kesimpulan yang diambil berdasarkan implementasi dan pengujian dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



BAB II

LANDASAN TEORI

Beberapa penelitian sebelumnya terhadap *streaming* video telah menginspirasi untuk meneliti *streaming* video menggunakan *multicast*. Penelitian sebelumnya antara lain *streaming* video pada *wireless* 802.11b [SAS-07] menghasilkan *jitter* yang fluktuatif dan terdapat banyak data yang hilang, penggunaan *live streaming* CCTV pada sistem keamanan proyek [DAR-11], serta *streaming* pada jaringan MPLS VPN [ABR-10.] dimana dikatakan mengalami kehilangan data yang besar karena proses enkripsi.

2.1 *Streaming*

Terdapat dua metode penyampaian konten multimedia ke klien, yakni *streaming* dan *download* [SEP-09:1] [FRA-02:1]. *Download* membutuhkan waktu dan tempat penyimpanan yang besar pada *harddisk*, sedangkan *streaming* merepresentasikan konten yang datang secara langsung tanpa mengunduh seluruhnya. Konten pada *streaming* diletakkan pada memori penyangga dan langsung direpresentasikan kemudian dibuang. Terdapat teknologi baru bernama *Progressive Download* dimana data disimpan di *buffer* lalu diputar sebelum proses pengunduhan selesai.

Streaming merupakan transportasi media secara *real-time (live)* atau media yang disimpan (misalnya video, audio dan setiap data terkait) melalui internet, antara penerima/pengguna dan komputer *server* [FRA-02:1]. *Streaming* merupakan teknik untuk mengirim *file* dari satu alat ke alat lain yang berjalan secara terus menerus sehingga pengguna di sisi tujuan bisa menjalankan *file* yang dikirim dengan tidak menunggu semua *file* dari tempat asal selesai dikirimkan. *Streaming* juga dapat di artikan sebagai proses penghantaran data dalam aliran berkelanjutan dan tetap yang memungkinkan pengguna mengakses dan menggunakan *file* sebelum data dihantar sepenuhnya.

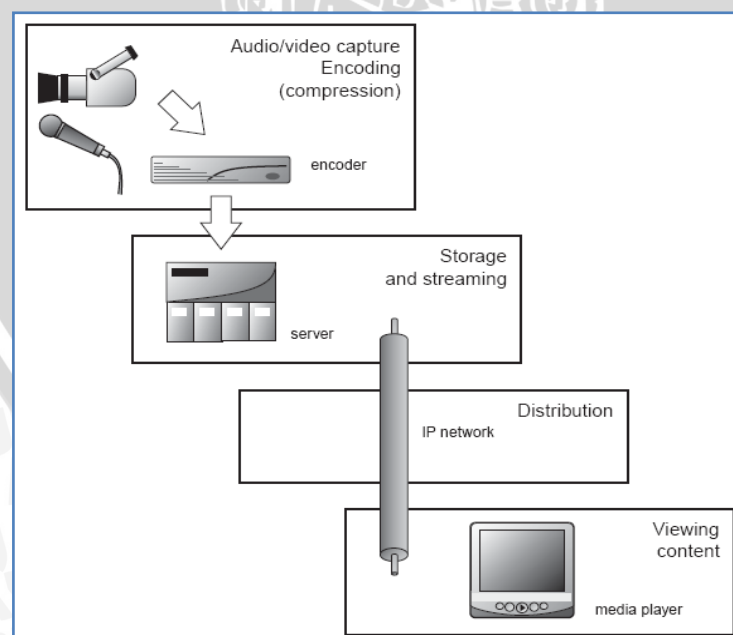
2.2 Arsitektur Streaming Video

Media *streaming* terdiri dari *streaming* video dan *streaming* audio [SEP-09:1]. *Streaming* menunjukkan transmisi satu arah dari *server* ke *client*. Pada sisi *client*, data masuk ke *buffer* selama beberapa detik sebelum mulai dikirim ke layar, sehingga terdapat *delay* pada pengiriman data paket [SEP-09:2].

Arsitektur media *streaming* terdiri dari [VAS-05:138] :

- *Capture and encoding*
- *Serving*
- *Distribution and Delivery*
- *Media player*

Capture and Encoding mengambil audio dan video mentah dari mikrofon dan kamera lalu memprosesnya kedalam *file* komputer. *File* tersebut disimpan dalam *server* yang mana terdapat perangkat lunak khusus yang dapat mengontrol pengiriman secara *streaming*. Dari *server*, data akan didistribusikan kepada *client* yang akan diterima oleh *Media Player*. Untuk lebih jelasnya dapat dilihat pada gambar 2.1.



Gambar 2.1 Arsitektur Streaming [VAS-05:139]

2.2.1 Capture and Encode

Proses *Capture and Encode* mengambil sinyal audio dan video biasa dalam format televisi (analog) lalu mengubahnya menjadi *file streaming* [VAS-05:140]. Dalam hal ini terdapat beberapa tahap:

- *Capture to computer file format*
- *Compress the media data*
- *Packetize data with hints or index*

Sebuah sinyal video langsung tidak bisa ditangani oleh perangkat lunak hingga sinyal tersebut diubah kedalam format tertentu (umumnya AVI). Ketika melakukan konversi, umumnya akan terjadi pengecilan skala ukuran gambar hingga sekitar seperempat dari resolusi asli.

2.2.2 Serving

File yang telah di *encode* diunggah ke *server* untuk di distribusikan ke *client*. Sebuah *streaming server* mengendalikan *file* yang di-*streaming*-kan secara *real-time*, hal ini yang membedakan dengan *server* lain seperti *server web*. *Server* dapat mengatur metode penyampaian *content multimedia* yaitu:

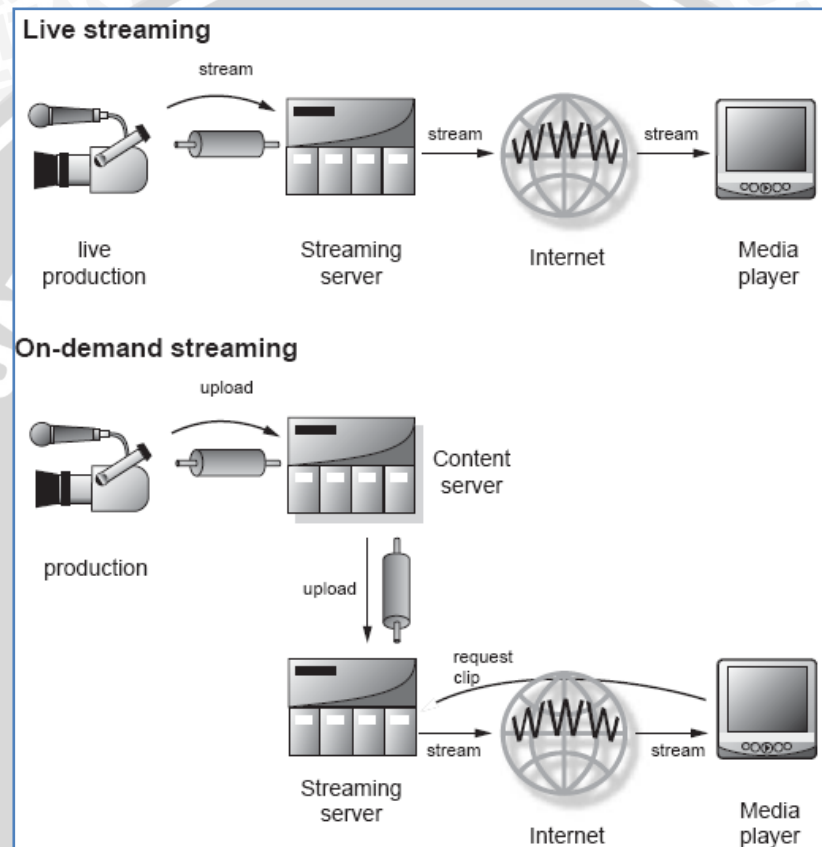
- ***Progressive Download***

Progressive download merupakan langkah penengah antara *download-and-play* dengan *streaming* sesungguhnya [VAS-05:142]. Daripada menunggu *file* lengkap di *download* ke *disk* lokal, *file* dapat dibuka sementara sisanya masih dalam proses *download*.

Perbedaannya yaitu *server* tidak memberikan *file* secara *realtime* sehingga *media player* bisa kehabisan bahan bila transfer lebih lambat dari *decoded rate*. Seluruh *file* media disimpan secara lokal oleh *media player*.

- **Live and On-Demand**

Streaming dapat digunakan dengan dua cara [VAS-05:142]. Media dapat diteruskan ke penonton seperti siaran TV pada umumnya (*live streaming*). Alternatif lain penonton dapat memilih atau meminta kepada *server* seperti dalam perpustakaan (*on-demand*) seperti pada gambar 2.2.



Gambar 2.2 Live dan On-Demand [VAS-05:144]

Keduanya dapat dipadukan misalnya dalam *simulated-live*. Misalnya pembicara dapat diperkenalkan berupa promo, lalu pembicara berbicara secara *live* di kamera, dan kemudian menunjukkan klip rekaman dari *server*.

- **On-Demand and Interactivity**

Salah satu kelebihan dari *streaming* bila dibandingkan siaran TV biasa adalah interaktivitas [VAS-05:144]. Penonton dapat berinteraksi dengan file yang

ada di *server*. Penonton mempunyai kontrol seperti *PLAY*, *STOP*, *FF*, *REW*, dan akses acak dari setiap bagian *clip*. Untuk mendukung fasilitas ini, *server* khusus untuk *streaming* harus digunakan. File *streaming* harus mengalami pemberian indeks agar dapat diakses secara acak.

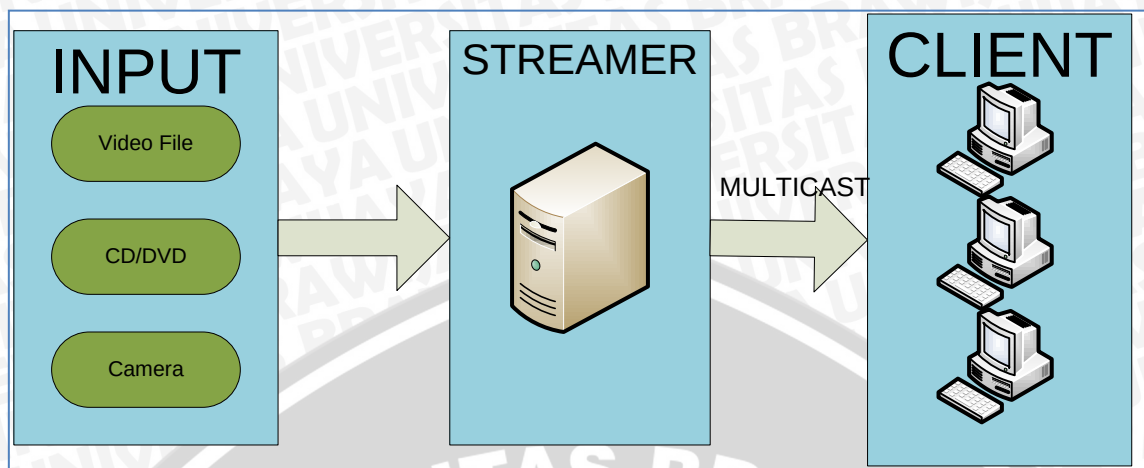
2.2.3 Distribution

Prinsip distribusi sebenarnya sederhana. Selama terdapat konektivitas IP antara *server* dengan *media player/client*, paket data akan sampai pada *media player* [VAS-05:144]. Namun pada prakteknya hal tersebut tidak mudah karena internet sejak awal tidak didesain untuk mengirimkan paket data secara terus menerus selama presisten koneksi. Hal ini berdasarkan sistem *packet-switched* untuk data *asynchronous*.

Beberapa perkembangan membantu perbaikan kualitas pengiriman. Melalui kabel, modem, DSL, *download* 1Mbit/s dapat dicapai. Hingga kini perkembangan masih terus berlanjut.

2.3 Prinsip Kerja Streaming Video

Pada dasarnya video *streaming* adalah sebuah teknologi multimedia untuk memainkan *file* video atau audio secara langsung atau dengan *prerecorded* dari sebuah *server* [SEP 09:1]. Prinsip kerja dari video *streaming* seperti pada gambar 2.3 adalah dari objek yang di-shoot oleh sumber media, dalam hal ini berupa kamera kemudian dikompres menjadi suatu sinyal yang dapat dilewatkan ke jaringan. Setelah data dikompres, kemudian masuk ke media *streaming server* yang kemudian diteruskan ke *client*, dimana pada sisi *client*, sinyal data dikembalikan ke format awal sebelum ditampilkan pada *handset*.



Gambar 2.3 Prinsip kerja video streaming

2.4 Protokol Streaming

Dalam metode penyampaian *content multimedia* yang berupa *streaming* terdapat beberapa metode keluaran yaitu HTTP, UDP, MMSH, RTP, dan RTSP [VAS-05:20]. Metode-metode tersebut mempunyai karakter penyampaian masing-masing. Misalnya HTTP menggunakan sistem TCP/IP sedangkan MMSH (bekerja pada *linux*) dapat menggunakan TCP/IP atau UDP/IP.

- **HTTP (Hyper Text Transfer Protocol)**

Protokol ini digunakan untuk transmisi melalui *web page*. Protokol ini berada diatas *layer TCP (Transport Control Protocol)* [SEP-09:02]. Karena menggunakan TCP, maka data yang sampai akan berurutan dan minim data *loss*. Hal tersebut disebabkan karena terdapat nomor paket dan *acknowledgments*. Namun karena hal tersebut pula, menjadi tidak cocok digunakan untuk *real-time* atau *video conferencing* karena akan terjadi *delay* yang cukup besar.

- **UDP Legacy (User Datagram Protocol)**

UDP merupakan protokol transmisi cepat yang cocok untuk konferensi dan aplikasi yang bersifat *real-time* [JAN-10:01]. Sedikit kehilangan paket data tidak banyak mempengaruhi media yang diputar. UDP dapat menghemat waktu transmisi tanpa pemulihan dan transmisi ulang.

Kelemahan UDP adalah tidak menjamin pengiriman data [JAN-10:01]. Hal itu disebabkan karena tidak ada kontrol lalu lintas diatas lapisan UDP sehingga *server* tidak menyadari adanya kegagalan transmisi dan akan terus mengirim tanpa menahan dengan resiko membebani jaringan.

- **MMSH/MS-WMSP**

MMS adalah klien / *server* berbasis protokol yang digunakan untuk *streaming* data *real-time* antara *client* (penerima data *streaming*) dan *server* (pengirim data *streaming*) [MMS-11:09].

Microsoft Media Server (MMS) adalah nama dari protokol jaringan milik Microsoft *streaming* yang digunakan untuk mentransfer data *unicast* pada *Windows Media Service*. MMS dapat diangkut melalui UDP atau TCP. Port *default* MMS adalah UDP/TCP1755. Pada tahun 2003 MMS mendukung RTSP (TCP /UDP port554) lalu diubah pada tahun 2008.

Microsoft masih merekomendasikan menggunakan "mms: //" sebagai "protokol rollover URL". Bila membuka sebuah "mms: //" URL pada Windows Media Player akan mencoba untuk menghubungkan pertama dengan RTSP melalui UDP dan jika gagal akan mencoba versi modifikasi dari HTTP melalui koneksi TCP akan dicoba. Versi modifikasi disebut oleh beberapa pihak ketiga sebagai MMSH [AST-11], dan oleh Microsoft sebagai MS-WMSP (*Windows Media HTTP Streaming Protocol*).

- **RTP (*Real-Time Transport Protocol*)**

Merupakan protokol yang berada diatas *layer* IP (*Internet Protocol*) yang mendukung pengiriman data secara *real-time* (waktu nyata) [SEP-09:02]. RTP sesuai untuk telepon, media, konferensi, dan transmisi data lain yang bersifat *real-time* [SOI-08:05]. RTP dapat menggunakan baik *unicast* maupun *multicast* dimana dalam RTP telah ditambahkan RTCP (*Real-Time Transport Control Protocol*). Dengan adanya RTCP, memungkinkan urutan paket data direkonstruksi kembali oleh penerima. RTP juga mendukung enkripsi data.

RTCP digunakan untuk mengontrol data yang ditransfer yaitu dengan memberikan informasi kepada aplikasi, mengidentifikasi pengirim dan penerima RTP, mengontrol interval dan rating transmisi, dan menyampaikan informasi sesi kontrol yang minimal. Paket RTCP ada 5 yaitu : SR: Sender report, RR: Receiver report, SDES: Source description, BYE: End of participation, APP: Application specific function.

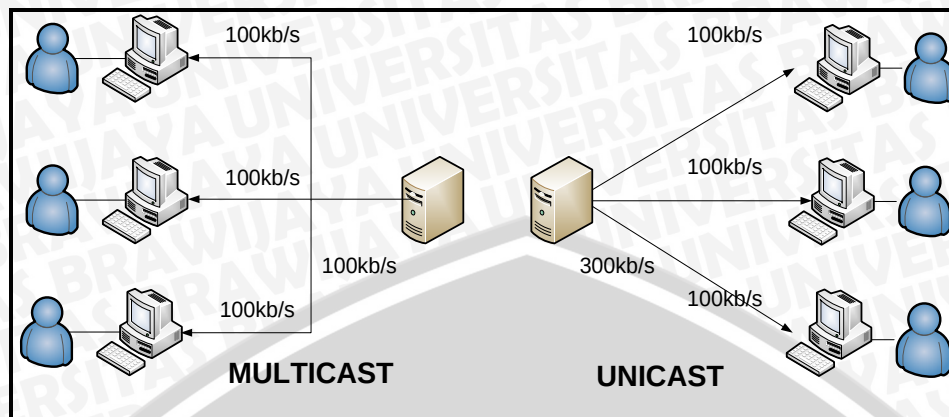
- **RTSP (*Real-Time Transport Streaming Protocol*)**

Protokol ini berada diatas lapisan RTP yang digunakan untuk media *streaming* [SEP-09:02]. RTSP menetapkan dan mengontrol sinkronisasi waktu dari media *streaming* seperti audio dan video [RTSP-98:05]. RTSP server umumnya menggunakan RTP untuk pengiriman media *streaming*. Dengan adanya RTSP maka terdapat metode yang dapat dijalankan untuk mengontrol *streaming* seperti: DESCRIBE, ANNOUNCE, GET_PARAMETER, OPTION, PAUSE, PLAY, RECORD, REDIRECT, SETUP, SET_PARAMETER, dan TEARDOWN

2.5 Multicast

Pengiriman data pada jaringan dengan IP versi 4 mempunyai 3 metode yaitu: *broadcast*, *unicast*, dan *multicast*. *Broadcast* merupakan transmisi untuk menyampaikan informasi yang sama kepada seluruh *host* [DAR-11:02], yang terdapat dalam 1 jaringan. *Unicast* merupakan transmisi informasi yang dilakukan oleh satu pengirim kepada satu penerima [SEP-09:02]. Lebih dikenal dengan transmisi *point to point*.

Transmisi *multicast* merupakan transmisi dari satu pengirim ke banyak penerima [SEP-09:2]. Jadi apabila terdapat 3 permintaan dari penerima sebesar 100kb/s maka total *bandwidth* yang dibutuhkan sebesar 100kb/s, berbeda dengan *unicast* yang akan membutuhkan *bandwidth* sebesar 300kb/s seperti pada gambar 2.4. Pada dasarnya *multicast* membentuk suatu grup yang disebut *host group* dimana bila server mengirim data , grup yang dituju lah yang menerima, berbeda dengan *broadcast* yang akan menyampaikan ke semua *client* yang terhubung.



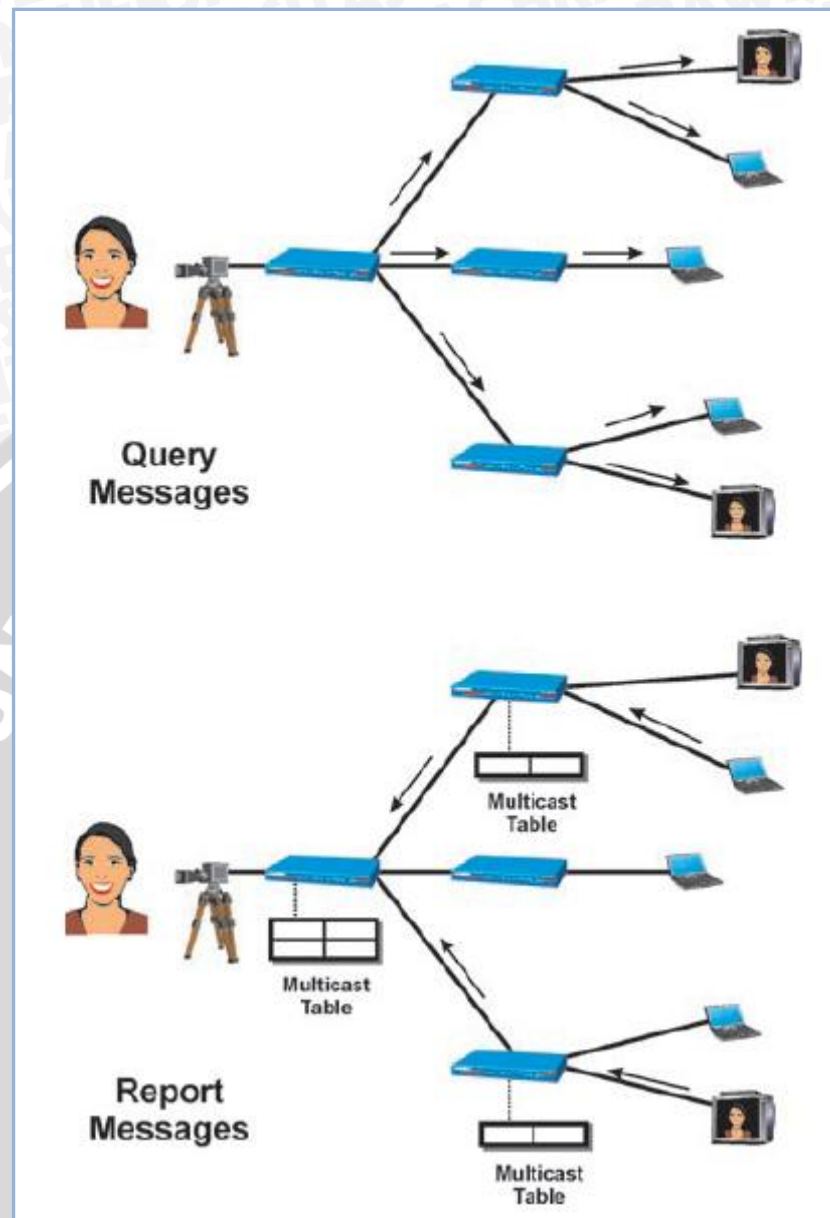
Gambar 2.4 Multicast dan Unicast

Multicasting menggunakan satu set alamat IP yang dicadangkan dalam kelas D, mulai dari 224.0.0.0 ke 239.255.255.255 [VAS-05:22]. Untuk pengiriman datanya, kita perlu mencantumkan satu alamat *multicast* sebagai alamat tujuan dimana alamat tersebut merepresentasikan satu grup yang terdiri dari sejumlah *host* yang bergabung.

- **IGMP**

Internet Group Management Protocol adalah perintah, proses dan prosedur yang digunakan untuk mengirim pesan kontrol dan mengkoordinasikan transfer data multicast (distribusi simultan) melalui jaringan protokol internet [HAR-08:29]. IGMP digunakan untuk membangun keanggotaan ke dalam sebuah grup multicast yang beroperasi dalam jaringan. Dengan IGMP, pengguna dapat menginformasikan router dalam jaringan yang pengguna ingin menerima media dan pesan kontrol dari kelompok multicast tertentu. IGMP didefinisikan dalam RFC 1112 dan 2236.

Gambar 2.5 menunjukkan bagaimana IGMP mengatur sesi komunikasi multicast ke host multicast lokal. Pada gambar 2.5 pengatur IGMP mengirimkan permintaan *query* kepada anggota kelompok agar dapat menentukan apakah anggota tersebut tertarik (atau masih tertarik) untuk berpartisipasi dalam sesi multicast. Anggota grup mengirim laporan untuk mengidentifikasi bahwa mereka tertarik untuk berpartisipasi dalam sesi *multicast*. Router *multicast* menggunakan laporan pesan untuk membangun tabel router *multicast* mereka.



Gambar 2.5 Internet Group Management Protocol [HAR-08:30]

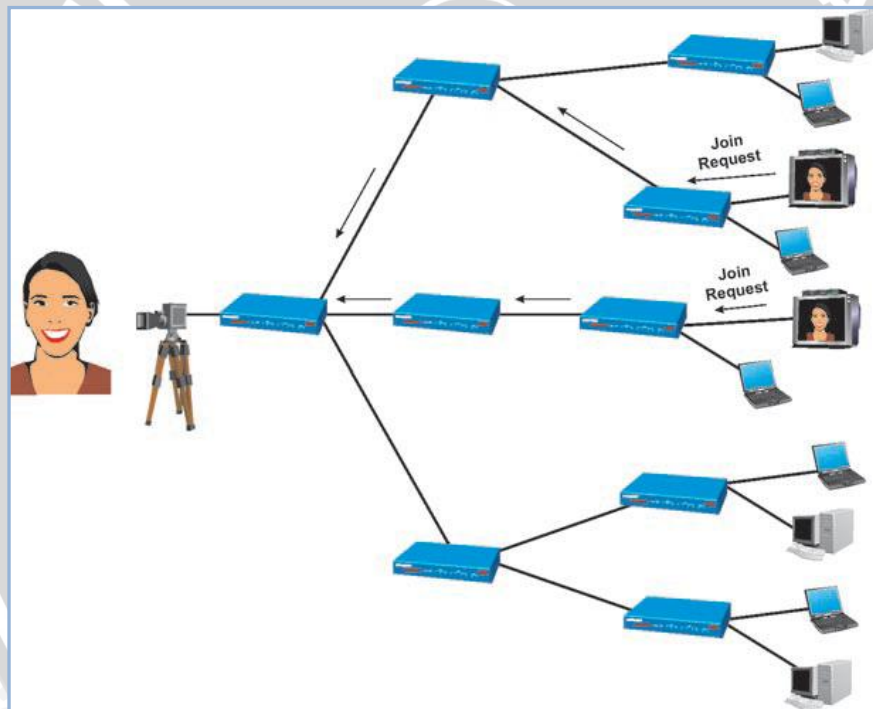
- PIM

Protocol Independent Multicasting merupakan pengaturan dan pendistribusian paket dalam struktur pohon *multicast*. *Protocol independent multicasting* menggunakan tabel routing yang ada untuk menentukan apakah paket harus disalin dan diarahkan di pohon *multicast*. Hal ini memungkinkan *multicasting protocol independen* (PIM) untuk diterapkan pada jaringan yang menggunakan protokol berbeda [HAR-08:35].

Saat paket *multicast* diterima router, tabel routing yang telah ada digunakan untuk menentukan dari alamat mana paket tersebut berasal. Asal paket tersebut disebut *upstream* pada pohon dan paket tersebut akan disalin dan diteruskan ke port lainnya.

○ Sparse Mode

Sparse mode merupakan distribusi media untuk beberapa pengguna dalam jaringan data yang mana sebagian kecil dari pengguna tersebut terhubung ke dalam jaringan grup *multicast* [HAR-08:12]. Setiap anggota mode koneksi *sparse mode* melakukan permintaan koneksi kepada grup *multicast* lalu menetapkan jalur *multicast*.

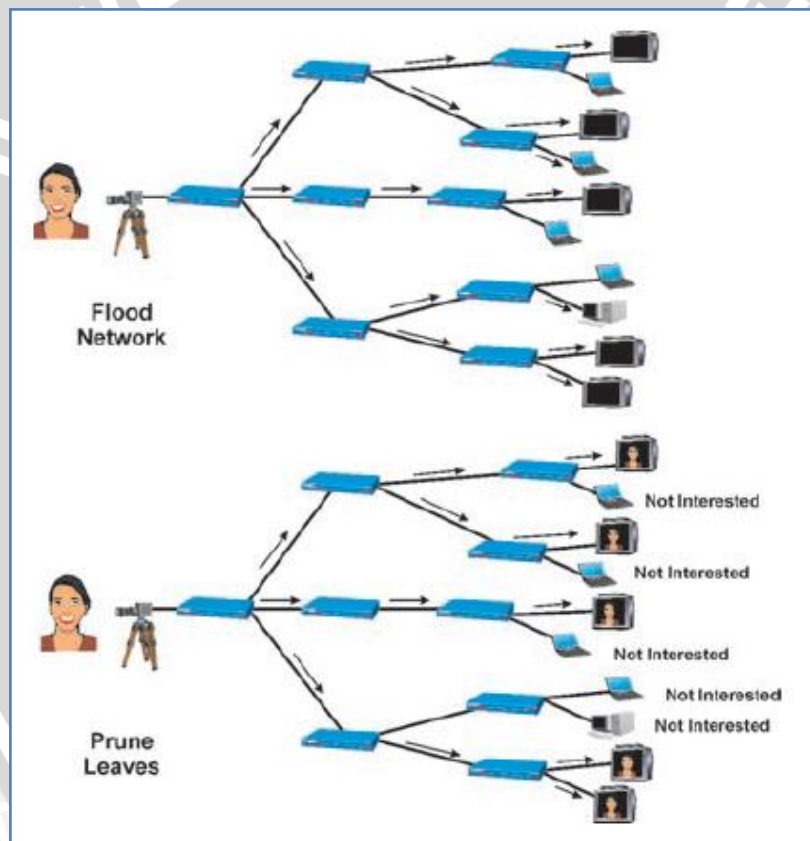


Gambar 2.6 *Sparse mode* [HAR-08:13]

Pada gambar 2.6 menunjukkan bagaimana jaringan data terbentuk dengan *sparse mode*. *Multicasting sparse mode* efektif bila ada sejumlah kecil perangkat yang ingin menerima sumber media bersama. Setiap perangkat melakukan koneksi permintaan dengan sumber media ketika mereka ingin melihatnya. Hanya router yang menjadi bagian dari pohon distribusi perlu berkomunikasi dengan anggota kelompok.

○ **Dense Mode**

Dense mode merupakan distribusi media untuk beberapa pengguna dalam jaringan data di mana banyak atau sebagian besar pengguna yang terhubung ke jaringan adalah bagian dari grup *multicast* [HAR-08:13]. Dalam sistem *multicast dense mode*, jaringan dibanjiri pesan *multicast* dan anggota kelompok yang tidak terhubung ke jaringan dipangkas dari pohon *multicast*. Anggota dari suatu sistem *multicasting dense mode* sering meminta koneksi ke grup *multicast* setelah pohon telah dibuat.



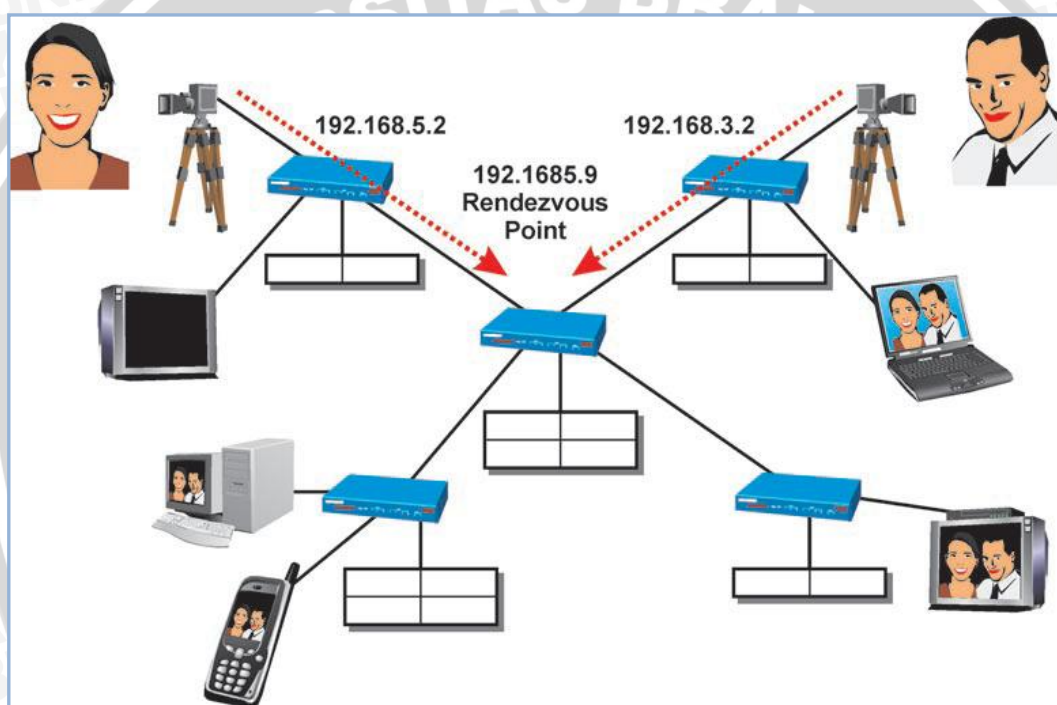
Gambar 2.7 *Dense mode* [HAR-08:15]

Gambar 2.7 menunjukkan bagaimana sebuah jaringan data *multicast* terbentuk dengan menggunakan *dense mode*. Gambar 2.9 menunjukkan bahwa *dense mode* membanjiri semua router hingga penerima/pengguna dalam jaringan dengan informasi *multicast*.

Perangkat yang tidak tertarik menerima informasi tersebut akan dikeluarkan (dipotong) dari pohon distribusi.

- **Rendezvous Point (RP)**

Rendezvous point merupakan alamat jaringan data dari sebuah node (seperti router) dalam jaringan komunikasi di mana satu atau beberapa sumber data akan diarahkan sehingga mereka dapat dipancarkan kembali ke titik lain dalam pohon distribusi *multicast* [HAR-08:29].



Gambar 2.8 *Rendezvous point* [HAR-08:30]

Gambar 2.8 menunjukkan bagaimana sebuah sesi sumber data *multicast* memungkinkan berbagai sumber untuk mengirimkan informasi ke beberapa penerima melalui penggunaan alamat referensi (titik pertemuan). Dalam contoh ini, data dari masing-masing sumber dikirim ke titik pertemuan (RP), yang kemudian mendistribusikan informasi melalui struktur pohon ke penerima grup *multicast*. Diagram ini menunjukkan bahwa ada potensi untuk beberapa transmisi duplikat di sesi berbagi sumber *multicast* sebagai sumber dan tujuan dapat dikirim dalam arah yang berbeda melalui router yang sama

2.6 Codec

Dalam bidang multimedia, *codec* adalah suatu metode atau algoritma yang terdapat pada sebuah *access player* yang fungsinya untuk melakukan proses pengompresan dan pendekompresan [GAT-09:2].

2.6.1 Video Codec

- **H.264**

H.264 di kembangkan oleh ITU-T *Video Coding Expert Group* (VCEG) bersama-sama dengan ISO/IEC *Moving Picture Expert Group* (MPEG) yang dinamakan *joint video team* (JVC). H.264 juga disebut sebagai *Advance Video Coding* (AVC). H.264 terdiri dari dua definisi yang terpisah:

- *Video Coding Layer* (VCL)
- *Network Abstraction Layer* (NAL)

VCL merupakan konten video, dan NAL mendefinisikan format *packetization* untuk transportasi protokol seperti RTP. Semua data yang terkandung dalam unit NAL. *Bytestream* H.264 dapat dari dua format : Unit aliran NAL dan format *bytestream* sebagaimana ditentukan dalam RFC3984 [THI-07:133].

- **WMV**

Windows Media Video adalah sebuah *codec* untuk meng-*encode* film dan mentransform *slide show* yang berisi format bitmap kedalam video terkompres yang merupakan bagian dari sistem *Windows Media* buatan *Microsoft*. WMV sebenarnya adalah versi *proprietary* dari MPEG-4 [AST-11]. Video Stream sering dikombinasikan dengan Audio Stream dalam format WMA, dengan video WMV yang dikemas kedalam kontainer AVI atau ASF.

- **DIV 3**

Fourcc singkatan dari “*Four Character Code*” merupakan sebuah *identifier* untuk video *codec*, format kompresi, warna atau format pixel yang digunakan dalam *file* media. Karakter tersebut bernilai 1byte/8 bit, sehingga fourcc selalu memakan tepat 32bit/4 byte dalam sebuah *file*.

Salah satu *codec* tersebut adalah DIV3 yang dikembangkan oleh Divx berdasarkan MPEG-4. DIV 3 merupakan *low motion codec* yang digunakan Divx untuk mengoptimasi *low motion materials* [FOU-11].

2.6.2 Audio Codec

- **WMA**

Windows Media adalah nama generik untuk set teknologi video dan audio *streaming* yang dikembangkan oleh Microsoft. Ini mencakup *Windows Media Video* (WMV) dan *Windows Media Audio* (WMA) *codec* [AST-11].

Versi pertama dirilis pada tahun 1999, WMA disukai oleh para pendiri musik online karena dukungannya terhadap *Digital Rights Management* (DRM). DRM adalah fitur untuk mengunci *file* audio sehingga dapat mencegah terjadinya pembajakan musik digital. Kelebihan dari WMA lainnya adalah kualitas musik yang lebih baik daripada MP3 maupun AAC.

- **MP3**

Merupakan *codec* yang paling terkenal dan berlisensi, serta distandardisasikan tahun 1991. MP3 hanya memiliki maksimal 2 channel. MP3 adalah jenis kompresi audio populer berdasarkan spesifikasi MPEG-1 (nama lengkap MP3 adalah MPEG-1 Layer 3 Audio) [AST-11]

- **AAC (Advance Audio Coding)**

AAC didesain untuk menghasilkan kualitas suara yang lebih bagus daripada mp3 dalam *bitrate* yang sama yang dikembangkan oleh MPEG. AAC memiliki rentang frekuensi sampel yang lebih besar hingga 48 *channel*. AAC juga berlisensi [AST-11]. AAC dikembangkan dari MPEG-2 namun tidak kompatibel dengan format MPEG-1 [MPG-11].

2.7 Parameter Kualitas Layanan Streaming (QoS)

Quality of Service (QoS), sebagaimana dijelaskan dalam rekomendasi CCITT E.800 adalah : “*Efek kolektif dari kinerja layanan yang menentukan derajat kepuasan seorang pengguna terhadap suatu layanan*” [ITU-95]

Penelitian ini menggunakan 4 (empat) parameter QoS yang diujikan dalam 2 (dua) jenis pengujian yaitu *delay*, *jitter*, *Packet loss*, dan *throughput* pada pengujian sistem dan pengujian protokol.

- **Delay**

Delay dialami oleh paket data saat menyeberang melalui jaringan yang merupakan jumlah waktu yang dibutuhkan untuk mengantar paket data dari asal ke tujuan [TRI-08:155]. *Delay* yang terjadi merupakan jumlah *delay* dari tiap *link* dan *node* yang dilintasi oleh paket data.

Tabel 2.1 Rekomendasi ITU-T G.114 untuk delay [DIK-11:314]

Range in milisecon	Description
0-150 msec	Acceptable for most user
150-400 msec	Acceptable provided that administrator that aware of the transmission time and it's impact on transmission quality of user qualification.
>400 msec	Unacceptable for general network planning purpose, it is recognized that in some exceptional cases this limit will be exceeded

- **Jitter**

Jitter mengacu pada variasi dalam jumlah keterlambatan. Hal tersebut sering disebut sebagai nilai absolut dari variasi satu arah keterlambatan [TRI-08:157]. Paket sering tidak sampai pada tujuan sesuai dengan rute yang sama, atau akan ditunda di *router* untuk jangka waktu yang berbeda, karena itu paket tidak mencapai target pada tingkat yang stabil. Untuk mengatasi masalah ini dapat digunakan *buffer jitter* sebagai solusi karena *buffer* ini menerima paket dan mengirimkan ke penerima dengan penundaan yang lebih kecil.

Tabel 2.2 Standar *jitter* [DIK-11:311]

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	75 ms
Sedang	125 ms
Jelek	225.ms

- **Throughput**

Throughput, adalah kecepatan (*rate*) transfer data efektif, yang diukur dalam bps [DIK-11:309]. *Header* paket data mempengaruhi *throughput* karena semakin panjang *header* maka akan semakin besar paket data. Throughput dapat pula dikatakan *bandwidth* aktual yang terukur pada suatu ukuran waktu tertentu.

Throughput (S) adalah total waktu yang digunakan mengirim paket dengan sukses per satuan waktu tertentu yang dapat dihitung dengan :

$$S = \frac{\text{Jumlah paket sukses} \times \text{waktu transmisi paket}}{\text{Lama pengamatan}}$$

- **Packet Loss**

Sebuah paket dikatakan hilang jika tidak tiba di tempat tujuan. Hal ini dapat disebabkan bahwa paket mengandung kesalahan/eror atau datang terlambat juga dapat dikatakan hilang [TRI-08:159]. Dalam komunikasi *real time*, bila paket data datang terlambat maka paket tersebut tidak berguna lagi

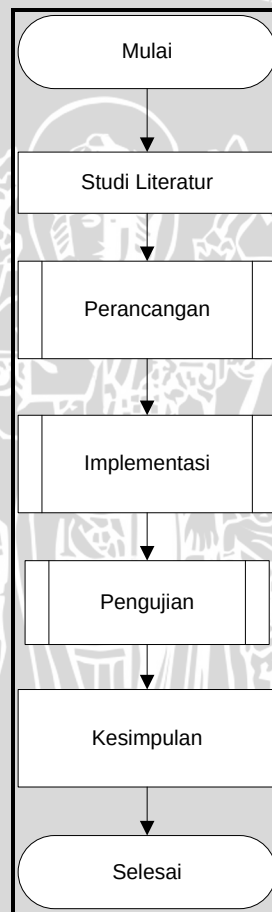
Tabel 2.3 Standar *packet loss* [DIK-11:312]

Kategori Degradasi	Packet Loss
Sangat Bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

BAB III

METODOLOGI PENELITIAN

Pada bab metodologi penelitian dijelaskan langkah-langkah yang akan dilakukan dalam perancangan, implementasi, analisis dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Diagram alir dari pelaksanaan penelitian secara keseluruhan dapat dilihat pada gambar 3.1



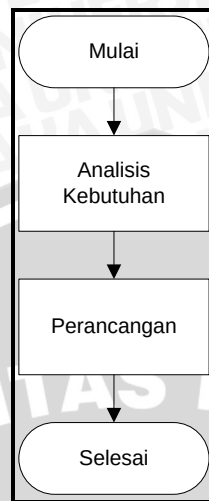
Gambar 3.1 Diagram alir keseluruhan pelaksanaan penelitian

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

- a. Jaringan *Streaming*
 - Arsitektur *Streaming* Video
 - Prinsip Kerja *Streaming*
- b. Protokol *Streaming*
 - UDP
 - RTP
 - RTSP
 - MMSH
 - HTTP
- c. Multicast
 - IGMP
 - PIM
 - Sparse Mode
 - Dense Mode
 - *Rendezvous Point*
- d. Kualitas Layanan *Streaming* (QoS)
 - *Delay*
 - *Jitter*
 - *Troughput*
 - *Packet Loss* (Tingkat Paket Hilang)
- e. Perangkat Lunak Pendukung
 - VLC
 - Wireshark
- f. Pengujian
 - VLC
 - Wireshark

3.2 Perancangan



Gambar 3.2 Diagram Alir Perancangan

Perancangan dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Penelitian dan analisis mengenai pengembangan infrastruktur penunjang *streaming* dengan memanfaatkan jaringan kampus, dalam upaya efisiensi biaya telekomunikasi di PTIIK, dilakukan di wilayah PTIIK dengan memusatkan tempat penelitian dan penempatan *server* di Gedung PTIIK.

- **Analisis Kebutuhan (*Requirement Analysis*)**

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada sehingga dapat diketahui implementasi baik desain sistem maupun perangkat yang akan digunakan.

- **Perancangan Infrastruktur**

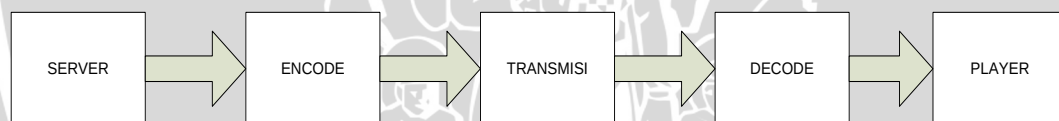
Perancangan infrastruktur menjelaskan kebutuhan yang didapat dari analisis kebutuhan dari segi infrastruktur. Hal tersebut meliputi spesifikasi komputer, perangkat jaringan, dan desain infrastruktur.

- **Penggunaan Perangkat Lunak**

Perancangan perangkat lunak menjelaskan kebutuhan yang didapat dari analisis kebutuhan dari segi perangkat lunak. Hal tersebut meliputi spesifikasi perangkat lunak yang digunakan, fungsi dan tujuan perangkat lunak yang digunakan.

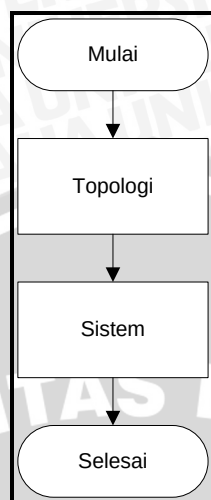
- **Proses Streaming Video**

Pada proses *streaming* akan dijelaskan gambaran dasar *streaming* video. Secara umum proses *streaming* yang terjadi berawal dari server yang melakukan siaran *streaming*. Server tersebut melakukan *encoding* terhadap berkas media yang akan disiarkan. Setelah proses *encoding* selesai, berkas media di transmisikan ke tujuan, proses transmisi ini dapat menggunakan *unicast* maupun *multicast*. Setelah sampai di sisi penerima/pengguna, berkas media tersebut di *decode* oleh pemutar berkas media lalu ditampilkan. Gambar 3.2 menunjukkan alur proses *streaming*.



Gambar 3.3 Proses streaming

3.3 Implementasi

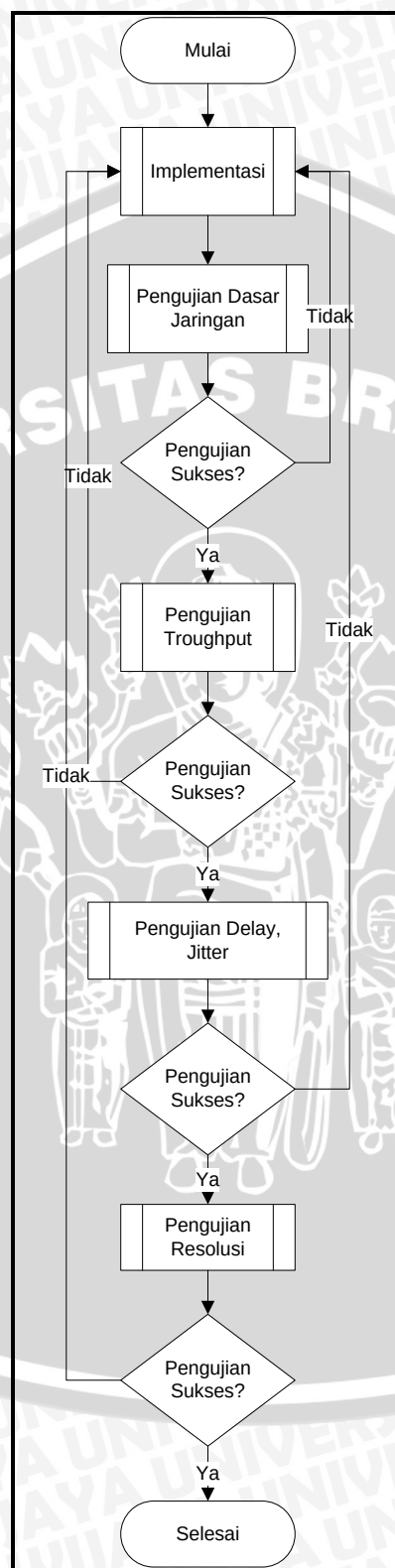


Gambar 3.4 Diagram Alir Implementasi

Implementasi mengacu pada perancangan. Implementasi diawali dengan penjabaran spesifikasi lingkungan implementasi, kemudian dilanjutkan dengan implementasi sistem pada topologi jaringan yang telah ada di PTIIK. Pada tahap implementasi akan dijelaskan lebih detail mengenai spesifikasi sistem yang mengacu pada perancangan infrastruktur dan penggunaan perangkat lunak beserta desain sistem.

Pada tahap implementasi dijelaskan juga mengenai topologi yang menjabarkan implementasi yang terjadi pada topologi jaringan yang ada di PTIIK. Terdapat pula kendala-kendala yang merupakan kendala yang dialami selama proses implementasi dilakukan.

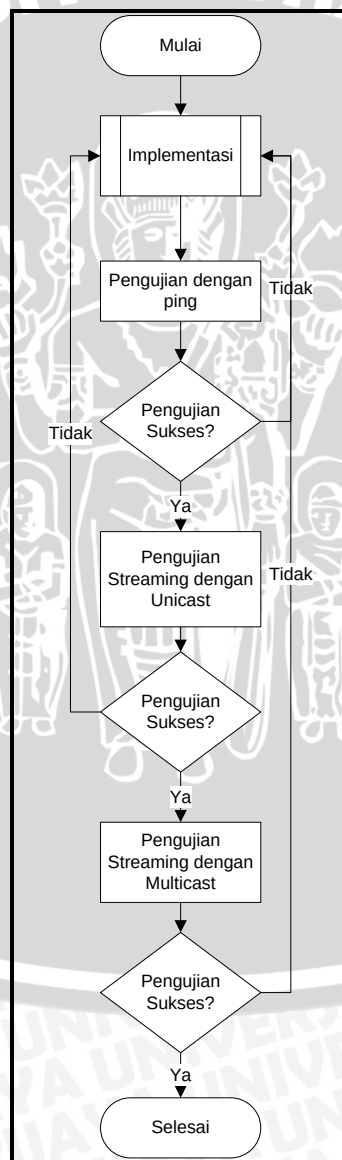
3.4 Pengujian dan Analisis



Gambar 3.5 Diagram Alir Pengujian dan Analisis

- **Pengujian Dasar Jaringan**

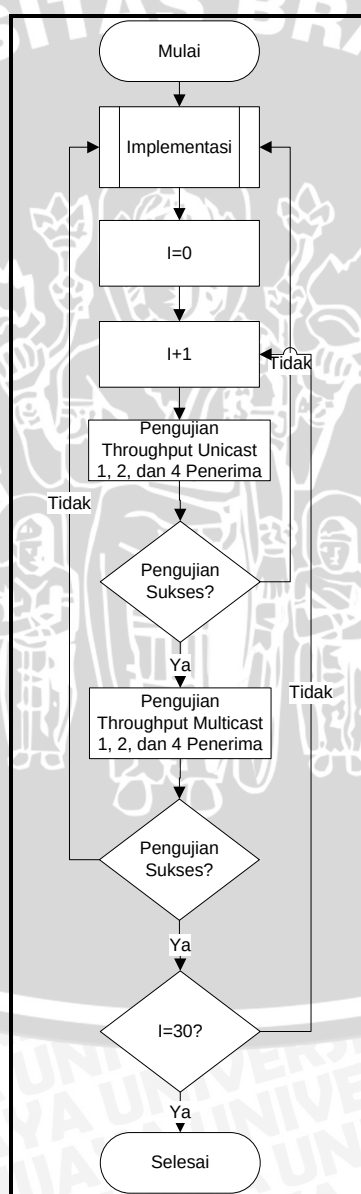
Pengujian kinerja jaringan dan server dimaksudkan untuk menguji apakah sistem telah berhasil terimplementasi. Pengujian dilakukan dengan melakukan ping ke semua komputer yang dibutuhkan pada proses pengujian untuk menguji interkoneksi dari jaringan. Kemudian pengujian dilakukan dengan melakukan *streaming* menggunakan unicast dari server ke pengguna untuk menguji kelancaran *unicast*. Kemudian pengujian dilakukan dengan melakukan streaming menggunakan multicast dari server ke pengguna untuk menguji kelancaran *multicast*.



Gambar 3.6 Diagram Alir Pengujian Dasar Jaringan

- **Pengujian Troughput**

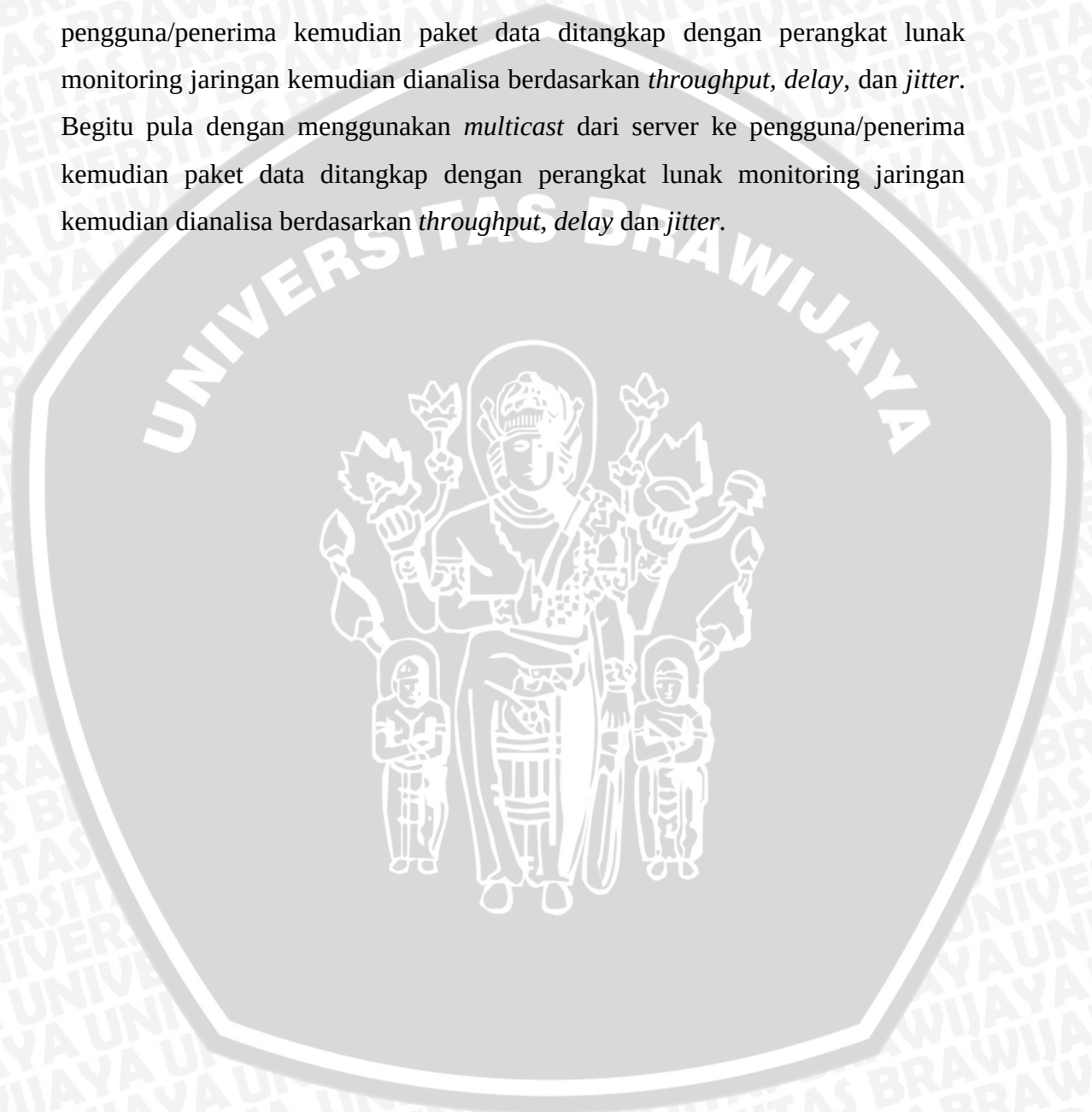
Pengujian sistem dimaksudkan untuk menguji apakah jaringan *multicast* benar-benar lebih baik untuk dilakukan *streaming* video secara simultan apabila dibandingkan dengan *unicast*. Pengujian dilakukan dengan melakukan *streaming* menggunakan *unicast* dengan jumlah pengguna/penerima berbeda yaitu 1, 2, 4 lalu dilihat *throughput* pada pengujian tersebut dan dianalisa. Begitu pula dengan *streaming* menggunakan *multicast*, dengan jumlah pengguna/penerima berbeda 1, 2, 4, dan 40 lalu dilihat *throughput* pada pengujian tersebut dan dianalisa.

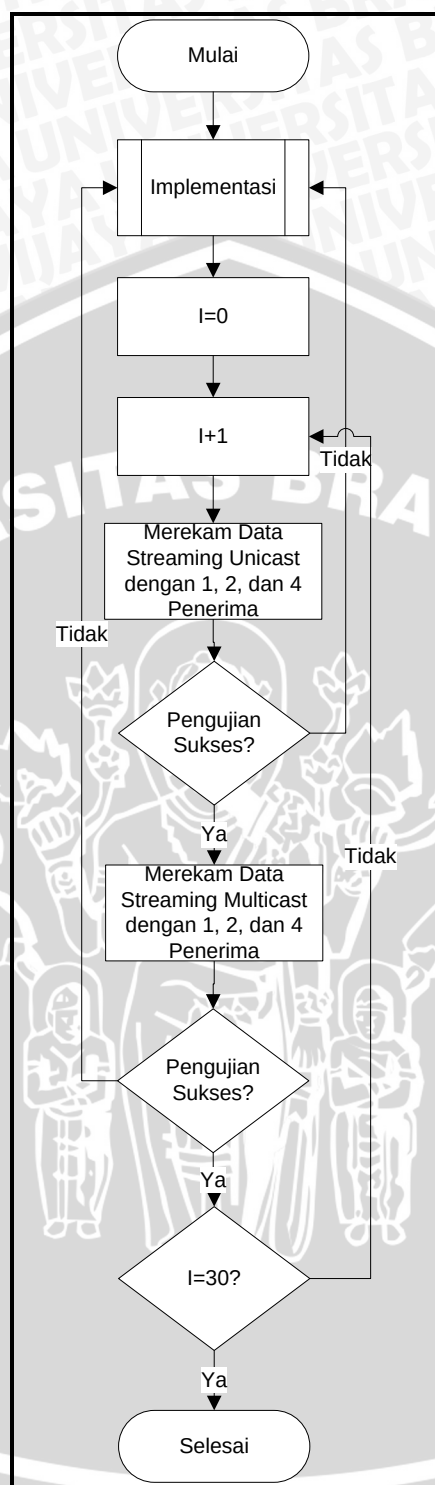


Gambar 3.7 Diagram Alir Pengujian *Throughput*

- **Pengujian Delay dan Jitter**

Pengujian *streaming* video dimaksudkan untuk menguji bagaimanakah kondisi data apabila dilakukan streaming menggunakan *unicast* atau *multicast*. Pengujian dilakukan dengan *streaming* menggunakan *unicast* dari server ke pengguna/penerima kemudian paket data ditangkap dengan perangkat lunak monitoring jaringan kemudian dianalisa berdasarkan *throughput*, *delay*, dan *jitter*. Begitu pula dengan menggunakan *multicast* dari server ke pengguna/penerima kemudian paket data ditangkap dengan perangkat lunak monitoring jaringan kemudian dianalisa berdasarkan *throughput*, *delay* dan *jitter*.



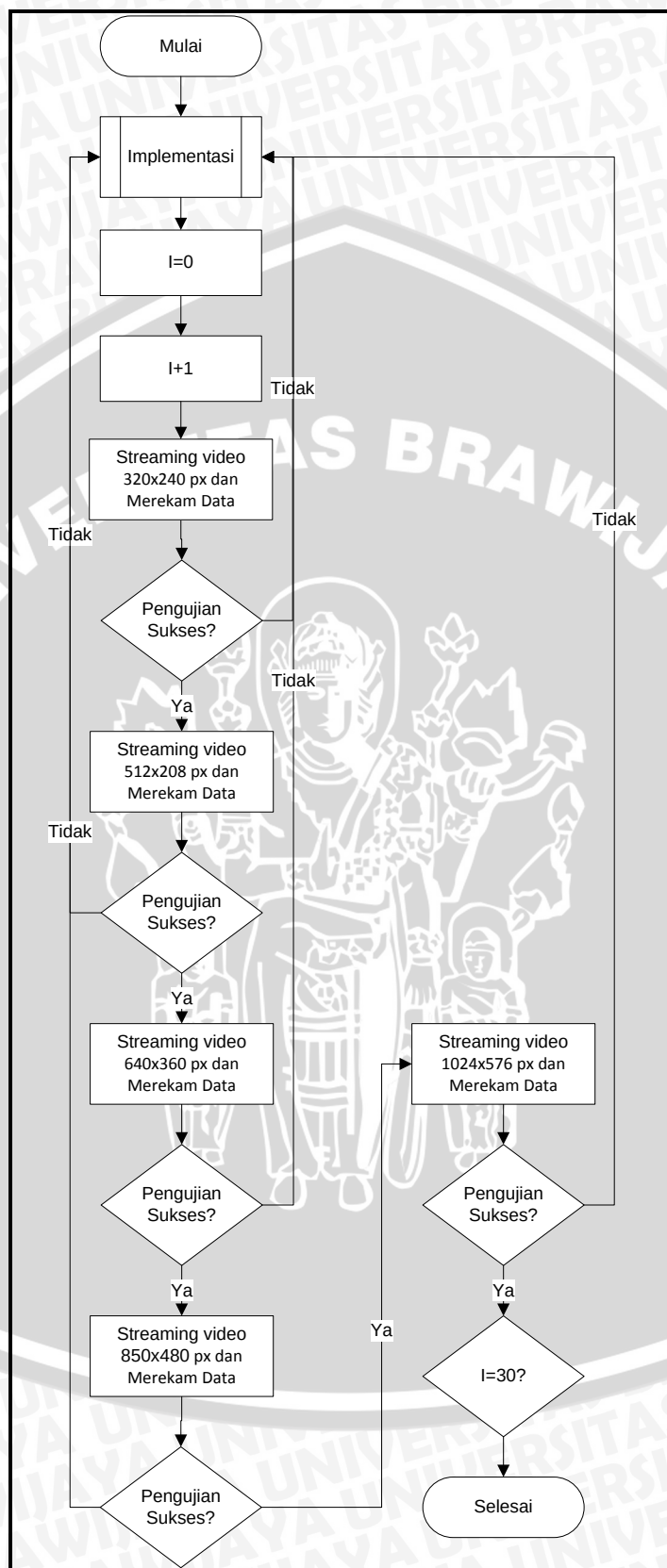


Gambar 3.8 Diagram Alir Pengujian Delay dan Jitter

- **Pengujian Resolusi**

Pengujian resolusi dimaksudkan untuk menguji bagaimanakah kondisi data apabila dilakukan *streaming* menggunakan *multicast* dengan resolusi video yang berbeda-beda di jaringan WiFi di PTIIK. Pengujian dilakukan dengan *streaming multicast* menggunakan protokol RTP dari server ke pengguna/penerima kemudian paket data ditangkap dengan perangkat lunak monitoring jaringan kemudian dianalisa berdasarkan *throughput*, *delay*, *jitter*, dan *packet loss*.





Gambar 3.9 Diagram Alir Pengujian Resolusi

3.5 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan sistem selanjutnya.

UNIVERSITAS BRAWIJAYA



BAB IV

PERANCANGAN

Pada Bab perancangan dijelaskan tentang perancangan sistem komunikasi *streaming video multicast* menggunakan VLC sebagai server dan penerimanya. Perancangan ini dilakukan untuk mengetahui mekanisme bagaimana suatu proses komunikasi dapat berlangsung pada jaringan *streaming* mulai dari pembuatan server, konfigurasi di sisi penerima, konfigurasi *multicast*, hingga setiap aktivitas yang terjadi pada jaringan *streaming* tersebut dapat didokumentasikan.

Perancangan Sistem komunikasi *streaming* ini dibagi menjadi dua tahap, yaitu : analisis kebutuhan dan perancangan. Analisis kebutuhan berisi semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis Kebutuhan meliputi gambaran umum, identifikasi pelaku, kebutuhan pelaku, dan kebutuhan sistem. Perancangan meliputi perancangan infrastruktur, penggunaan perangkat lunak, perancangan *multicast* dan proses *streaming*. Tahapan perancangan seperti pada bab 3 gambar 3.2.

4.1 Analisis kebutuhan

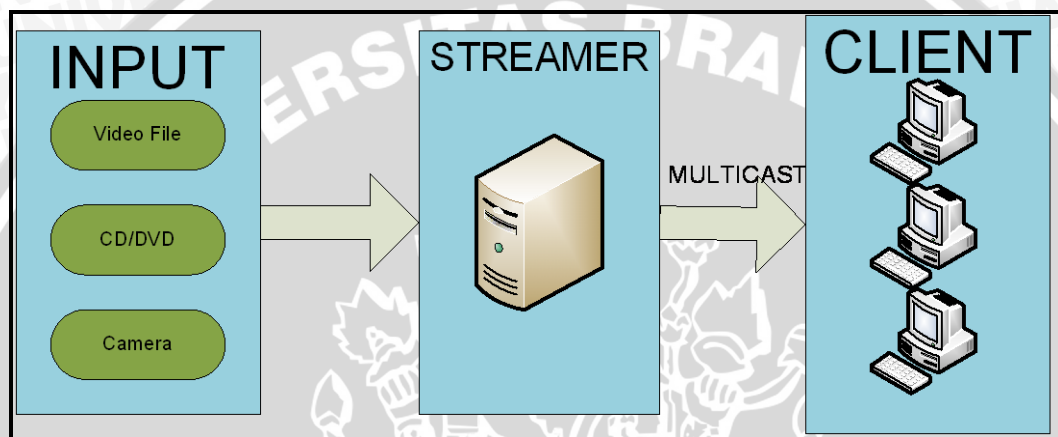
Analisis kebutuhan dapat dibedakan menjadi 2 yaitu kebutuhan pengguna dan kebutuhan sistem [SOM-11:83]. Pada kebutuhan pengguna menjelaskan kebutuhan *streaming* secara global sedangkan kebutuhan sistem menjelaskan kebutuhan *streaming* secara lebih spesifik.

4.1.1 Gambaran Umum

PTIIK sebagai salah satu fakultas di UB mempunyai jumlah mahasiswa dan dosen yang relatif banyak. Hal tersebut menyebabkan penggunaan internet PTIIK menjadi tinggi. Jaringan di PTIIK ini dipakai oleh seluruh warga PTIIK yang pada umumnya digunakan untuk browsing, mengunduh, mengunggah, monitoring CCTV. Dengan mengimplementasikan *multicast* pada jaringan PTIIK, kita dapat men-*streaming*-kan video yang dapat diterima seluruh warga PTIIK

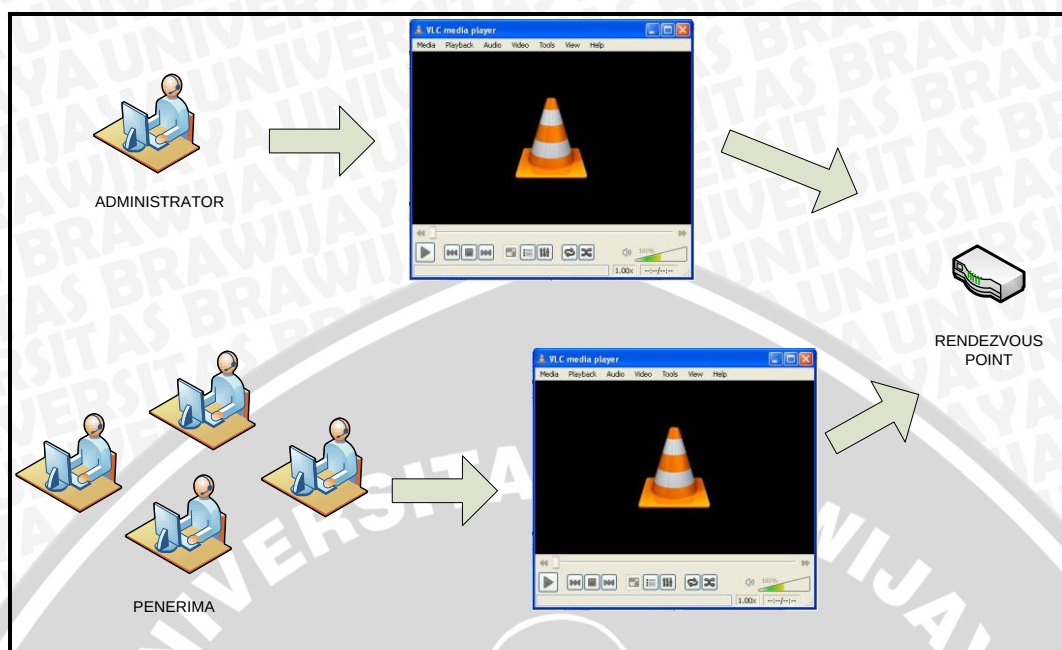
secara simultan, dengan kata lain seluruh warga PTIIK dapat menerima siaran video yang di-streaming-kan secara bersamaan tanpa membebani lalu lintas data jaringan secara berlebihan

Secara global, rencana sistem video *streaming multicast* yang akan diterapkan seperti pada gambar 4.1 Server melakukan *streaming* suatu video kepada jaringan data yang kemudian diterima oleh sejumlah penerima/pengguna secara simultan.



Gambar 4.1 Rencana sistem video *streaming multicast*

Streaming video multicast mempunyai urutan langkah kerja seperti gambar 4.2. Terdapat penyedia *streaming* yang menyiapkan media yang akan di-streaming-kan. penyedia *streaming* tersebut melakukan konfigurasi pada perangkat lunak VLC yang berfungsi sebagai alat *streaming*. Media tersebut akan dikirim ke Rendezvous Point (RP). Penerima siaran *streaming* melakukan konfigurasi terlebih dahulu pada perangkat lunak VLC kemudian siaran *streaming* dapat diterima dari Rendezvous Point (RP).



Gambar 4.2 Urutan langkah kerja video streaming multicast

4.1.2 Identifikasi Pelaku

Pada tahap ini dilakukan identifikasi terhadap pelaku yang akan berinteraksi dengan *streaming* video multicast. Tabel 4.1 memperlihatkan pelaku-pelaku yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari identifikasi pelaku.

Tabel 4.1 Identifikasi pelaku

Pelaku	Deskripsi
Penyedia	Administrator adalah pengguna yang melakukan siaran <i>streaming</i>
Penerima	Penerima adalah pengguna yang menerima siaran <i>streaming</i>

4.1.3 Kebutuhan Pelaku

Kebutuhan penyedia dari *streaming video multicast* ini terdiri sebagai berikut:

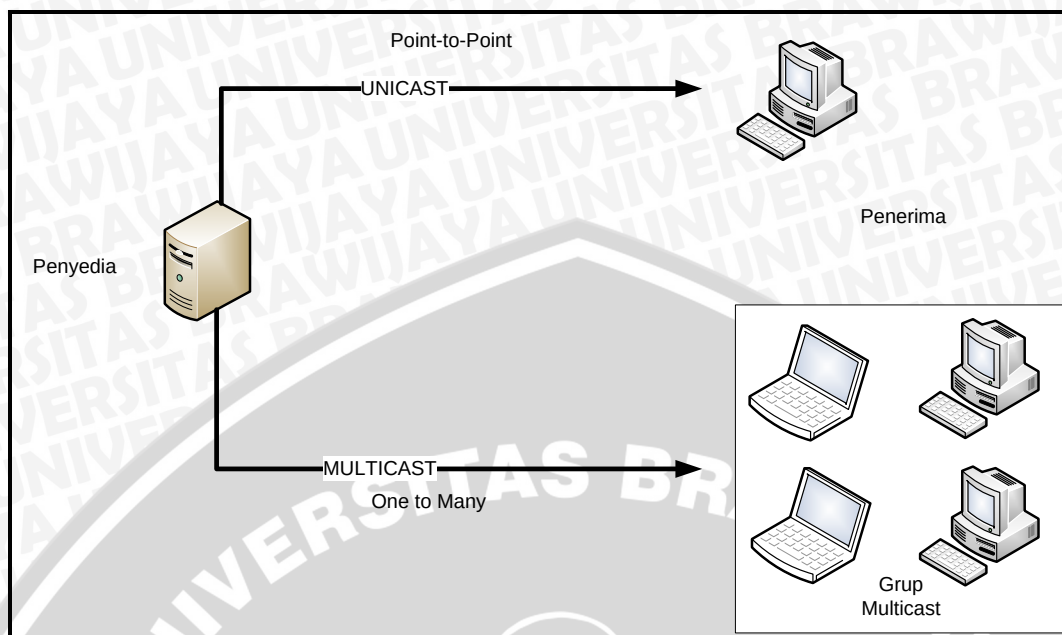
Tabel 4.2 Kebutuhan penyedia

Kebutuhan	Deskripsi
Streaming Unicast	<i>Server streaming</i> dapat melakukan <i>streaming video</i> menggunakan <i>unicast</i>
Streaming Multicast	<i>Server streaming</i> dapat melakukan <i>streaming video</i> menggunakan <i>multicast</i> .
Koneksi	Pengguna terhubung dalam jaringan dengan menerima IP dhcp di lingkungan PTIIK seperti pada umumnya

Kebutuhan penerima dari *streaming video multicast* ini terdiri sebagai berikut:

Tabel 4.3 Kebutuhan Penerima

Kebutuhan	Deskripsi
Streaming Unicast	Penerima <i>streaming</i> dapat menerima siaran <i>streaming video</i> menggunakan <i>unicast</i>
Streaming Multicast	Penerima <i>streaming</i> dapat menerima siaran <i>streaming video</i> menggunakan <i>multicast</i>
Koneksi	Pengguna terhubung dalam jaringan dengan menerima IP dhcp di lingkungan PTIIK seperti pada umumnya



Gambar 4.3 Penyedia/penerima dapat menerima *streaming* secara *unicast* maupun *multicast*

4.1.4 Kebutuhan Sistem

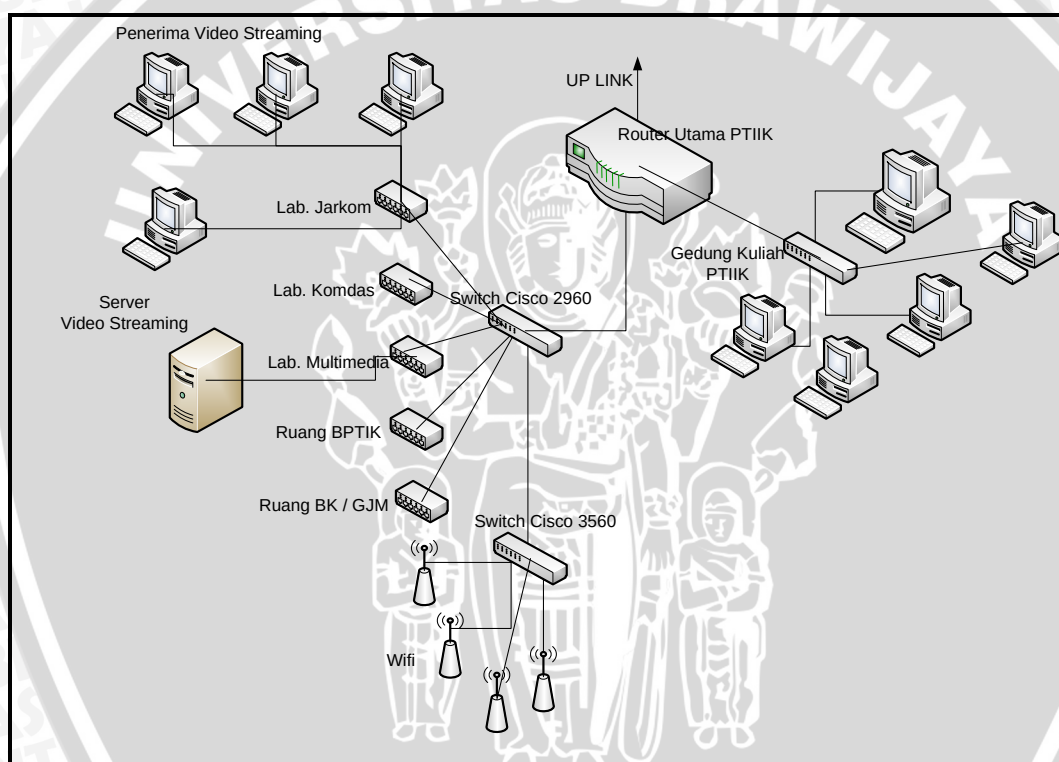
- **Lingkungan Perancangan dan Implementasi Sistem.**

Lingkungan perancangan dan implementasi sistem pada jaringan PTIIK adalah sebagai berikut :

- Terdapat 8 segmen yang berbeda di PTIIK yaitu lantai 1, lantai 2, Laboratorium Jararingan, BPTIK, Laboratorium Komunikasi data, Laboratorium Multimedia, Wifi PTIIK, dan GJM/BK
- Penyedia layanan streaming terhubung melalui kabel UTP dengan menerima IP dhcp di lingkungan PTIIK UB.
- Penerima terhubung melalui kabel UTP serta WiFi dengan menerima IP dhcp di lingkungan PTIIK UB.
- Teknologi yang digunakan untuk menghubungkan antara *server* dengan *client* adalah menggunakan *unicast* dan *multicast*.

- **Topologi Jaringan**

Topologi jaringan komputer yang digunakan dalam jaringan *streaming* pada implementasi merupakan topologi jaringan PTIIK yang telah ada sebelumnya. *Multicast* diimplementasikan ke dalam topologi jaringan yang sudah ada di PTIIK UB dianggap bahwa jaringan komputer yang ada merupakan jaringan dasar yang sudah terkoneksi dengan baik sehingga untuk jaringan *streaming* yang dibuat tinggal mengikuti dari jaringan yang sudah ada. Topologi PTIIK seperti pada gambar 4.4.



Gambar 4.4 Topologi Jaringan Streaming PTIIK

- **Teknologi Streaming yang Digunakan**

Teknologi *streaming* yang digunakan dalam penelitian ini adalah teknologi *unicast* dan *multicast*. *Unicast* merupakan transmisi informasi yang dilakukan oleh satu pengirim kepada satu penerima [SEP-09:02], lebih dikenal dengan transmisi *point to point*. Transmisi *multicast* merupakan transmisi dari satu pengirim ke banyak penerima [SEP-09:2]. Penggunaan *PIM Sparse Mode* dilandasi oleh karakteristik *PIM Sparse Mode* sesuai dengan kebutuhan *streaming*

video di PTIIK. Kebutuhan *streaming* di PTIIK berdasarkan adanya permintaan sewaktu-waktu dan bukan merupakan agenda rutin. Dalam penelitian ini penulis menggunakan teknologi *multicast* sebagai bahan analisa, karena dianggap efisien untuk melakukan *streaming* video secara simultan.

4.2 Perancangan

4.2.1 Perancangan Infrastruktur

Seperti yang telah dijelaskan pada bahasan sebelumnya dan berdasarkan pada topologi pada gambar 4.4 maka dapat diperoleh rincian infrastruktur yang digunakan dalam perancangan jaringan *streaming* ini. Rincian infrastruktur pada jaringan TIF sebelum mengalami perubahan topologi adalah sebagai berikut:

- Komputer *Server*
- Komputer *Client*
- Router
- Switch

4.2.2 Penggunaan Perangkat Lunak

Pada penggunaan perangkat lunak, dijelaskan perangkat lunak apa saja yang digunakan dalam penelitian mengenai video *streaming multicast*. Perangkat lunak tersebut meliputi VLC (VideoLAN Client) dan Wireshark. Kebutuhan perangkat lunak meliputi:

- Pemutar media sebagai pengirim video
- Pemutar media sebagai penerima video
- Perangkat lunak monitoring jaringan

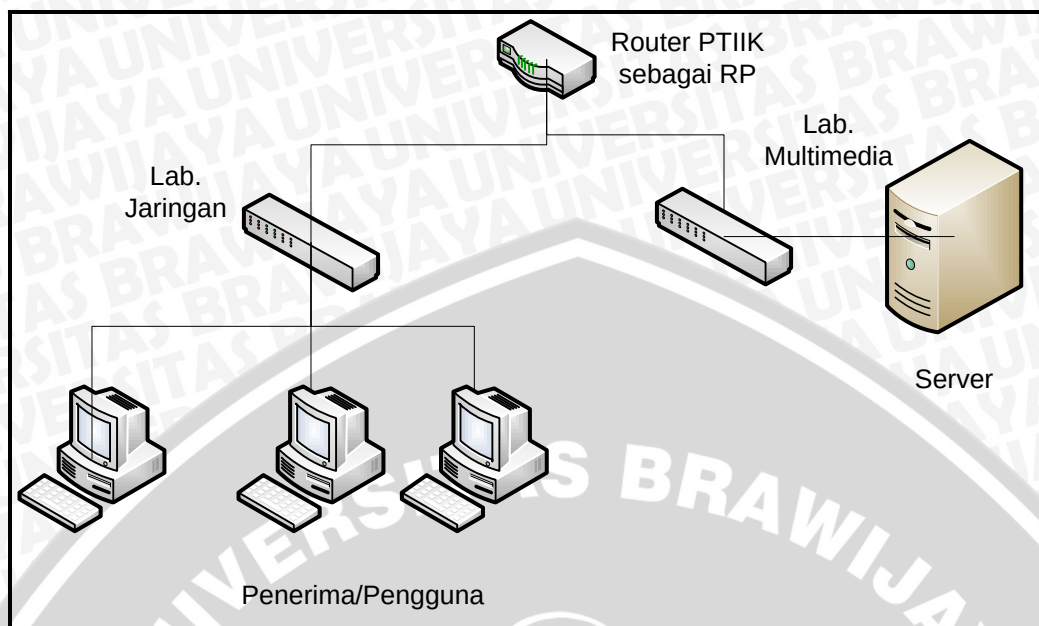
VLC mempunyai sistem untuk manajemen yaitu VLM (VideLAN Manager). VLM merupakan sebuah file berisi perintah pengoperasian VLC yang mana VLM dapat diakses dari komputer lain melalui antar muka telnet, web, atau remote desktop.

4.2.3 Perancangan Multicast

Pada *streaming video multicast* di PTIIK digunakan protokol routing IGMP dan PIM. Metode *multicast* pada *streaming video* di PTIIK menggunakan *sparse mode*. PTIIK menggunakan sebuah router untuk mengatur semua kebutuhan jaringan dimana router tersebut dijadikan *Rendezvous Point (RP)* pada jaringan PTIIK sebagai titik pertemuan seperti pada gambar 4.5. Pengguna/penerima akan melakukan permintaan bergabung dengan grup *multicast* yang akan di arahkan ke router PTIIK untuk menerima *streaming video multicast*. Setelah bergabung dengan grup *multicast* di RP maka router PTIIK akan meneruskan *streaming video multicast* dari server yang melakukan *streaming* menuju pengguna/penerima yang melakukan permintaan.

Teknologi yang digunakan untuk menghubungkan antara *server* dengan *client* adalah menggunakan IP group *multicast* yang dikonfigurasi pada router PTIIK yang dilewati untuk *streaming* secara UDP dan tabel *routing* untuk *streaming* secara TCP/IP. IP group *multicast* yang digunakan yaitu 224.1.1.1.

Server penyedia menggunakan VLC sebagai server *streaming* dengan sistem operasi Windows sedangkan *client* dapat menggunakan sistem operasi apapun dengan VLC sebagai pemutar penerima *streaming* atau pemutar lain dengan fungsi yang sama. Untuk kepentingan pengujian, sever diletakkan di Laboratorium Multimedia dengan pengguna/penerima berada di Laboratorium Jaringan



Gambar 4.5 Router PTIik sebagai RP

4.2.4 Proses Streaming Video

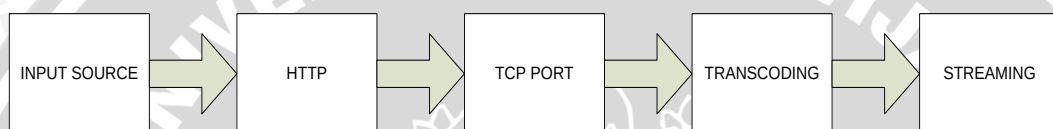
Proses *streaming* berawal dari server yang melakukan siaran *streaming*. Server tersebut melakukan *encoding* terhadap berkas media yang akan disiarkan. Setelah proses *encoding* selesai, berkas media di transmisikan ke tujuan, proses transmisi ini dapat menggunakan *unicast* maupun *multicast*. Setelah sampai di sisi penerima/pengguna, berkas media tersebut di *decode* oleh pemutar berkas media lalu ditampilkan. Alur proses *streaming* seperti pada bab 3 gambar 3.3.

Lapisan Transport bertanggung jawab atas pengiriman data *end-to-end* secara keseluruhan. Pada proses transmisi data, data disegmentasi ketika akan mengirim dan menyusun kembali potongan-potongan data tersebut ketika diterima. Setiap bagian data membutuhkan *header* yang akan ditambahkan pada lapisan Transport untuk menunjukkan jenis komunikasi terkait. Protokol lapisan Transport yaitu *Transfer Control Protocol* (TCP) dan *User Datagram Protocol* (UDP). Keduanya mempunyai fungsi spesifik yang diimplementasikan.

4.2.5 Transfer Control Protocol (TCP)

TCP merupakan sebuah koneksi yang berorientasi *protocol*. TCP menghilangkan *overhead* tambahan agar dapat berfungsi. Fungsi tambahan yang diberikan oleh TCP adalah pengiriman dengan perintah yang sama, pengiriman yang tangguh (*reliable*), dan kontrol aliran (*flow control*). Setiap bagian TCP memiliki 20 *byte* dari *overhead* pada proses *header* enkapsulasi data aplikasi *layer*, dimana setiap bagian UDP hanya memiliki 8 *bytes overhead* masing-masing protokol.

- **HTTP (Hyper Text Transfer Protocol)**



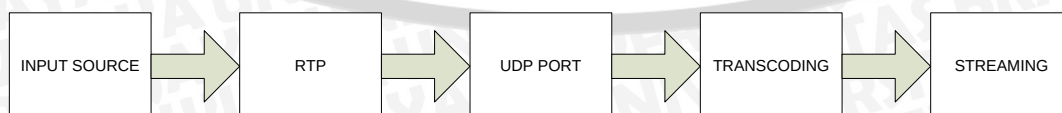
Gambar 4.6 Proses *Streaming* melalui HTTP

HTTP dapat digunakan sebagai protokol komunikasi *streaming* dan umumnya digunakan dengan cara *diembed* pada *browser*. Mekanisme *streaming* melalui protokol HTTP dapat dilihat pada gambar 4.6. Pengguna/penerima menerima siaran *streaming* dengan menggunakan *VLC player*.

4.4.6 User Datagram Protokol (UDP)

UDP sangat sederhana, bersifat *connectionless* dimana tidak terjadi proses negosiasi (*handshaking*) antara *host* yang bertukar informasi. UDP memberikan keunggulan dalam menyediakan pengiriman data yang sedikit *overhead*. Setiap bagian komunikasi dari UDP disebut datagrams.

- **RTP (Real-Time Transport Protocol)**



Gambar 4.7 Proses *Streaming* melalui RTP

RTP dirancang untuk arus data yang *end-to-end* atau *transfer real-time*. Protokol ini menyediakan fasilitas untuk kompensasi *jitter* dan deteksi keluar dari kedatangan urutan data, yang umum selama transmisi pada jaringan IP. RTP mendukung transfer data ke beberapa tujuan melalui IP *multicast*. RTP dianggap sebagai standar utama untuk audio/video dalam jaringan transportasi IP yang terkait format *payload*. Hal tersebut tidak terdapat pada protokol UDP biasa (UDP Legacy) meski sama sama mendukung *multicast*. Protokol MMS mempunyai kelebihan yang didukung oleh Windos Media Player sehingga kurang maksimal apabila digunakan oleh pemutar selain Windows Media Player [MMS-11]



BAB V

IMPLEMENTASI

Streaming video menggunakan *multicast* yang telah dirancang pada Bab IV akan diimplementasikan pada tahap ini. Implementasi mengacu pada perancangan yaitu infrastruktur, perangkat keras, dan perangkat lunak pendukung. Terdapat dua tahap yang akan dilakukan dalam implementasi ini, yaitu topologi dan sistem. Topologi menjelaskan implementasi *multicast* pada jaringan yang telah ada di PTIIK sedangkan sistem menjelaskan bagaimana streaming video menggunakan *multicast* di PTIIK. Tahapan implementasi seperti pada bab 3 gambar 3.4. Instalasi dan pengaturan *multicast* terlampir.

5.1 Spesifikasi Sistem

5.1.1 Perangkat Keras

Spesifikasi perangkat keras mengacu pada perancangan infrastruktur. Berikut merupakan spesifikasi perangkat keras yang digunakan dalam implementasi *streaming* video *multicast*.

- Server

Pada dasarnya semua PC yang terhubung dalam jaringan PTIIK dapat menjadi server yang melakukan *streaming*. Untuk pengujian PC yang digunakan sebagai *server* untuk *men-streaming*-kan video terletak di Laboratorium Multimedia dengan IP dhcp dari segmen network id 172.21.4.192. Server menggunakan Windows XP dengan VLC versi 2.0.2.

- Komputer *Client*

Pada dasarnya semua PC yang terhubung dalam jaringan PTIIK dapat menjadi pengguna/penerima siaran *streaming* video yang dilakukan server. Komputer terhubung dengan menerima IP dhcp di lingkungan PTIIK. Untuk kepentingan pengujian, komputer pengguna/penerima disimulasikan berada di Laboratorium Jaringan.

- Router mikrotik Mikrobits
- Switch

5.1.2 Penggunaan Perangkat Lunak

Perangkat lunak yang digunakan mengacu pada penggunaan perangkat lunak di bab perancangan. Perangkat lunak yang digunakan untuk keperluan *streaming video multicast* beserta pengujiannya adalah VLC dan *wireshark*.

- **VLC**

VLC dipergunakan sebagai pusat untuk mengendalikan video yang akan di-*streaming*-kan atau diterima oleh penerima/pengguna. VLC dipasang di server yang berada di lab multimedia yang bertugas untuk melakukan *streaming*. VLC juga dipasang di sisi penerima/pengguna untuk menerima siaran *streaming* dari server dengan terhubung pada jaringan PTIIK. VLC untuk berbagai sistem operasi dapat di unduh pada alamat <http://www.videolan.org/vlc/> kemudian di install.

Pada sisi server, ketika VLC telah terinstal di komputer server terdapat beberapa tahap untuk melakukan *streaming*. Tahap pertama klik tab media kemudian pilih *stream* (ctrl+s), setelah itu klik tombol add untuk memilih video yang akan di-*streaming*-kan lalu klik tombol *stream* kemudian akan timbul url dari file video yang akan di-*streaming*-kan. Klik *next* kemudian pilih metode *streaming* yang digunakan lalu klik *add*. Apabila menggunakan HTTP cukup klik *add* saja sedangkan bila menggunakan RTP masukkan alamat IP dari group *multicast* yaitu 224.1.1.1 beserta port yang digunakan. *Active transcoding* dinonaktifkan kemudian klik *next*. Aktifkan *stream all elementary stream* kemudian tentukan TTL (time-to-live) dari paket data yang akan di-*streaming*-kan setelah itu klik *stream*.

Pada sisi *client*, ketika VLC telah terinstal di komputer server terdapat beberapa tahap untuk menerima siaran *streaming*. Tahap pertama klik tab media kemudian pilih *open network stream* (ctrl+n), setelah itu masukkan protokol, alamat IP, dan port yang disesuaikan dengan server.

Manajemen *streaming* video dapat menggunakan VLM (VideLAN Manager). VLM merupakan sebuah file berisi perintah pengoperasian VLC. Pada linux perintah VLM berada pada `vlm.conf` sedangkan windows perlu membuat file sendiri dengan ekstensi `.vlm`.

VLM dapat dibuat lebih dari satu sehingga untuk kondisi tertentu kita dapat mengaktifkan VLM mana yang diinginkan. Kegunaan VLM antara lain dapat menyusun *VOD (Video On Demand)* atau siaran *broadcast* dan dapat menyusun penjadwalan dari siaran. Berikut merupakan contoh konfigurasi VLM untuk melakukan *streaming*.

```
new siaran broadcast enabled

setup siaran input
"file:///C:/Documents%20and%20Settings/Njom/My%20Documents/Downloads/Video/CLIP/Katy%20Perry%20-%20Part%20Of%20Me%20-%20YouTube.flv"

setup siaran output
#rtp{dst=224.1.1.1,port=5004,mux=ts,ttl=10}

setup siaran option file-caching=300
```

- **Wireshark**

Wireshark merupakan perangkat monitoring jaringan. Pada penelitian ini *wireshark* digunakan untuk menangkap data yang ada di jaringan agar dapat dianalisa terutama untuk menemukan nilai *delay*, *jitter*, *throughput*, dan *packet loss*. *Wireshark* melakukan pemindaian pada *ethernet* lalu menangkap data yang dapat dipergunakan untuk mengetahui *throughput*, *delay*, *jitter*, dan *packet loss*.

Wireshark pasang di server agar dapat mengetahui *throughput* dari server sebagai pengirim video. *Wireshark* juga dipasang di sisi penerima/pengguna agar dapat mengetahui keterlambatan data dan parameter kualitas layanan lainnya. *Wireshark* dijalankan ketika komunikasi data berlangsung. *Wireshark* menangkap segala data yang melintas pada *ethernet* komputer. Untuk memilah paket data yang diinginkan terdapat fitur filter pada *wireshark*. Misalnya untuk memilah

paket data yang menggunakan RTP dari alamat IP group *multicast* dengan nomor *frame* tertentu digunakan konfigurasi:.

```
ip.addr == 224.1.1.1 and udp and frame.number > 5000
```

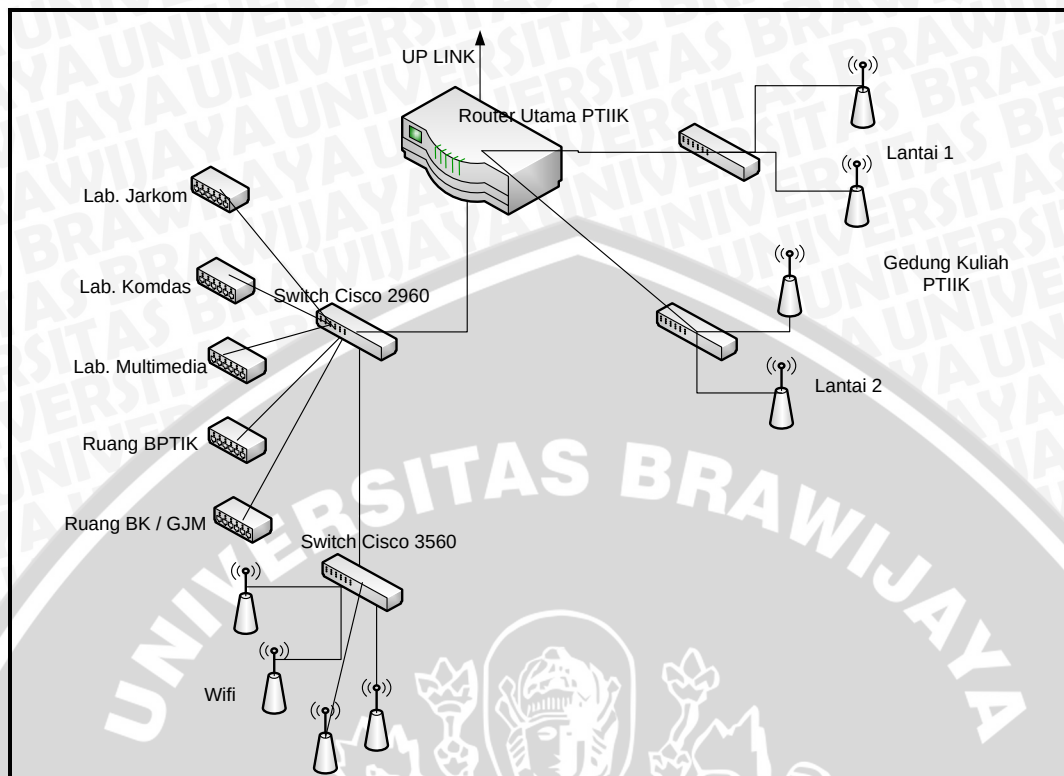
5.2 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan streaming video multicast adalah sebagai berikut:

1. Protokol jaringan multicast yang digunakan adalah IGMPv2 dan PIM Sparse Mode.
2. Router yang digunakan pada tahap perancangan, implementasi, dan pengujian adalah Mikrobits Ainos 2071
3. *Rendezvous Point* diletakkan pada router PTIIK Mikrobits Ainos 2071

5.3 Topologi

Dalam implementasi *multicast* dilakukan dalam 2 tahap yaitu instalasi paket *multicast* pada Router Mikrotik dan konfigurasi *multicast* pada Router Mikrotik. Paket *multicast* yang dipasang adalah paket terpisah yang tidak disertakan dalam paket sistem operasi (RouterOS) Mikrotik. Instalasi paket *multicast* dapat dilihat pada lampiran 1. Router PTIIK akan menjadi akar dari domain jaringan *multicast* dimana permintaan dan pengiriman data akan bertemu pada router PTIIK.



Gambar 5.1 Topologi jaringan PTIIK

5.4 Sistem Multicast

Pada streaming video *multicast* di PTIIK digunakan protokol routing IGMP dan PIM. Metode *multicast* pada streaming video di PTIIK menggunakan *sparse mode*. Router PTIIK dijadikan *Rendezvous Point* (RP) pada jaringan PTIIK sebagai titik pertemuan seperti pada gambar 5.2. Konfigurasi *multicast* selengkapnya ada pada lampiran 2, beberapa konfigurasi *multicast* misalnya:

```

routing pim interface add

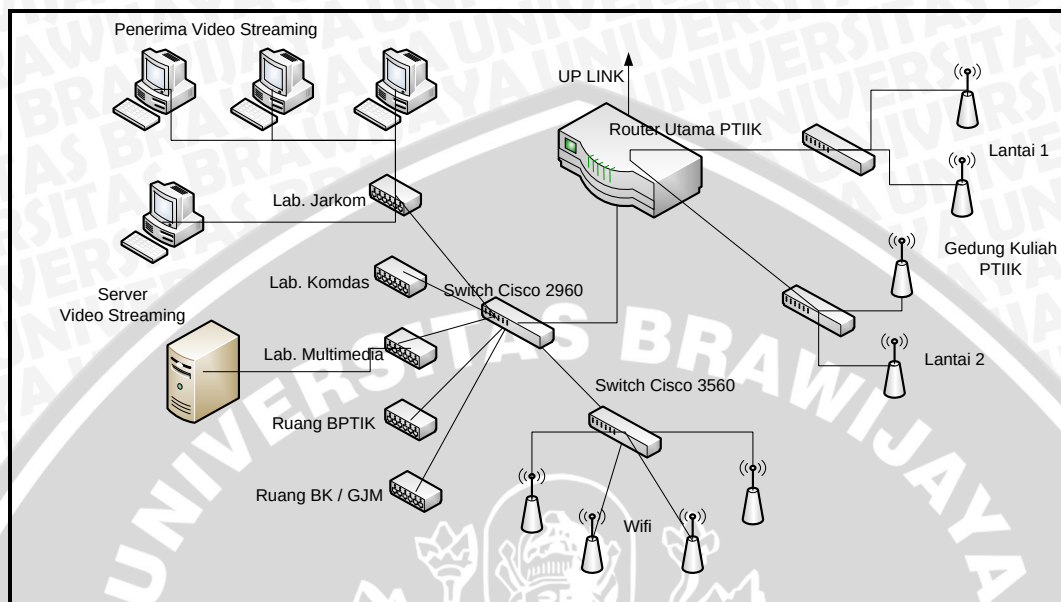
routing pim interface p

routing pim rp add address=172.21.4.1

```

Pengguna/penerima akan melakukan permintaan bergabung dengan grup *multicast* 224.1.1.1 yang akan di arahkan ke router PTIIK untuk menerima *streaming video multicast*. Setelah bergabung dengan grup *multicast* di RP maka

router PTIIK akan meneruskan *streaming* video *multicast* dari server yang melakukan *streaming* menuju pengguna/penerima yang melakukan permintaan.



Gambar 5.2 Desain topologi *multicast*

Pengujian nantinya dilakukan pada jam sibuk di PTIIK, untuk menunjang hal tersebut diperlukan monitoring terhadap lalu lintas network pada jaringan PTIIK tempat dilakukan pengujian agar dapat diketahui saat dimana jam sibuk terjadi. Monitoring tersebut dilakukan dengan cara melihat *throughput* dari jaringan PTIIK pada router PTIIK lalu dicari saat dimana *throughput* tersebut mempunyai rata-rata nilai yang paling besar. Beberapa konfigurasi router untuk memonitoring *throughput* adalah sebagai berikut:

```
tool graphing set store-every=5min

tool graphing print

tool graphing interface add interface=ether1

tool graphing interface add interface=ether2

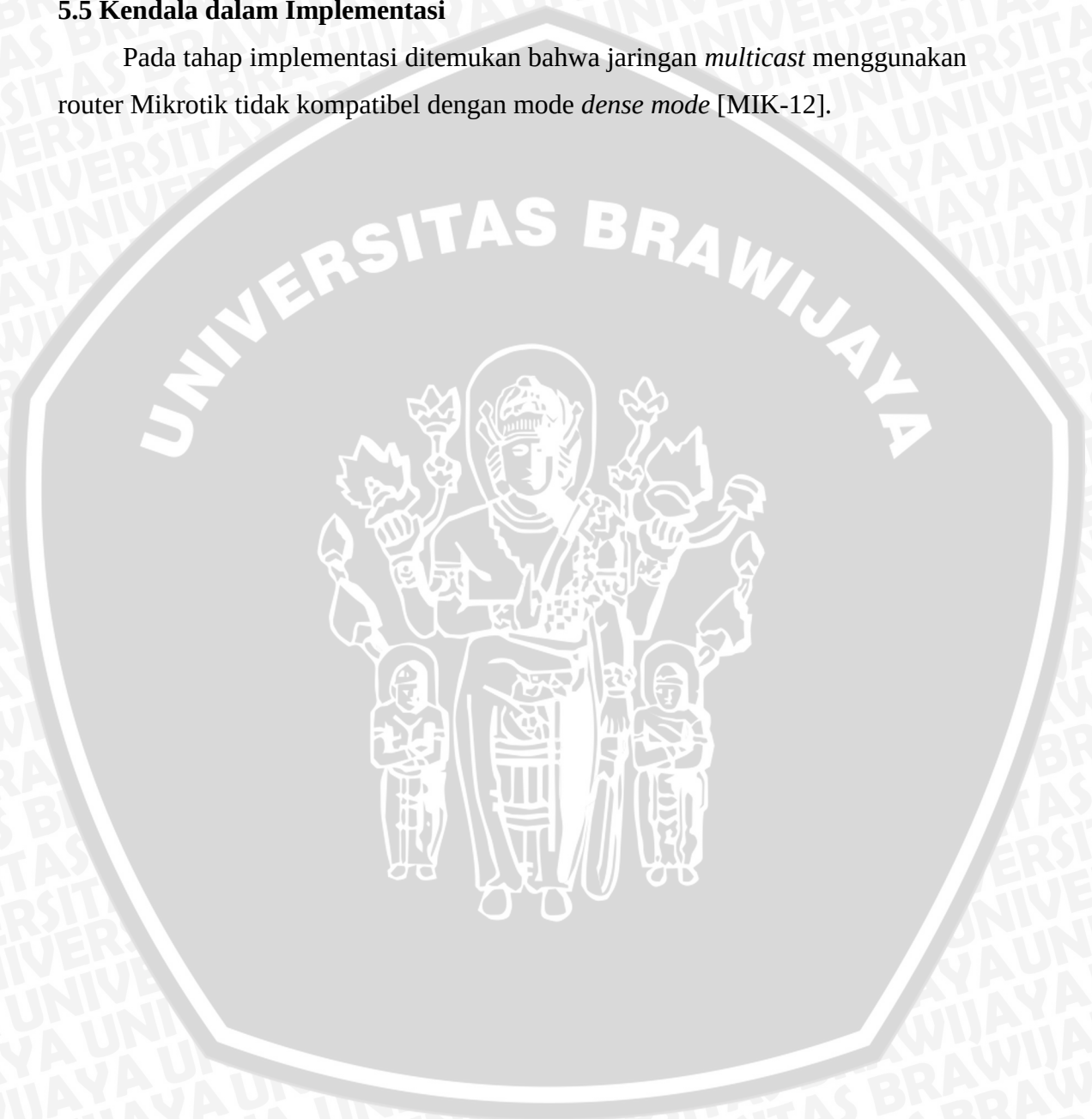
tool graphing interface print
```

Beberapa konfigurasi router untuk memonitoring *throughput* dalam bentuk grafik selengkapnya dapat dilihat pada lampiran 10. Untuk melihat hasil

dari konfigurasi tersebut dapat melalui browser yang mengakses alamat IP router PTIIK. Dalam hal ini misalnya <http://172.21.4.1/graphs/>, grafik tersebut terbagi menjadi 4 kategori dengan satuan harian, mingguan, bulanan, dan tahunan.

5.5 Kendala dalam Implementasi

Pada tahap implementasi ditemukan bahwa jaringan *multicast* menggunakan router Mikrotik tidak kompatibel dengan mode *dense mode* [MIK-12].



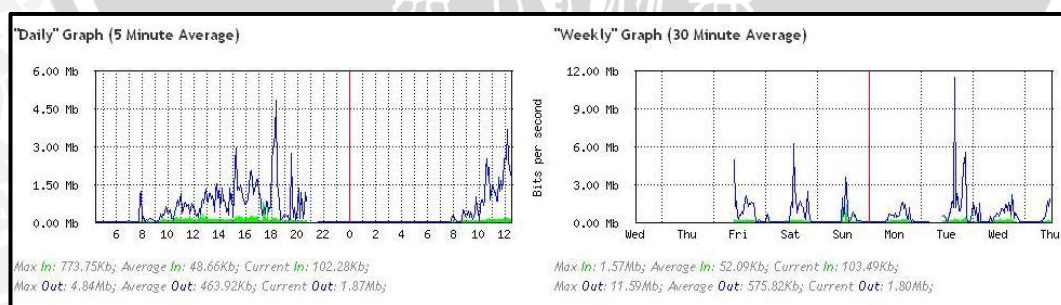
BAB VI

PENGUJIAN DAN ANALISIS

Sistem *multicast* untuk *streaming* yang telah diimplementasikan pada Bab 5 akan diuji dan dianalisis pada bab ini. Pengujian dan analisis sistem perlu dilakukan untuk mengetahui apakah sistem yang telah dirancang dan diimplementasikan telah bekerja sesuai dengan tujuan sistem. Pengujian dan analisis dilakukan secara bertahap sesuai urutan permasalahan yang terdapat pada bagian rumusan masalah.

Secara umum, pengujian dilakukan untuk dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang dibutuhkan oleh sistem dan pengguna. Pada pengujian ini akan dilakukan beberapa pengujian yaitu pengujian dasar jaringan, pengujian *throughput*, pengujian parameter *throughput*, *delay*, *jitter*, dan *packet loss*, serta pengujian resolusi. Tahapan pengujian seperti pada bab 3 gambar 3.5.

Pengujian dilakukan pada jam sibuk di PTIIK dimana setelah diamati, jam sibuk terjadi pada jam 11.00 hingga 15.00 dengan kondisi puncak pada jam 12.00 hingga jam 14.00. Data tersebut didapat dengan mengaktifkan fasilitas router untuk memonitoring lalu lintas data yang melewati router seperti pada gambar 6.1. Pengujian dilakukan sebanyak 30 kali di tiap masing-masing pengujian untuk menunjang validitas data statistik dari hasil pengujian.



Gambar 6.1 Lalu lintas data PTIIK

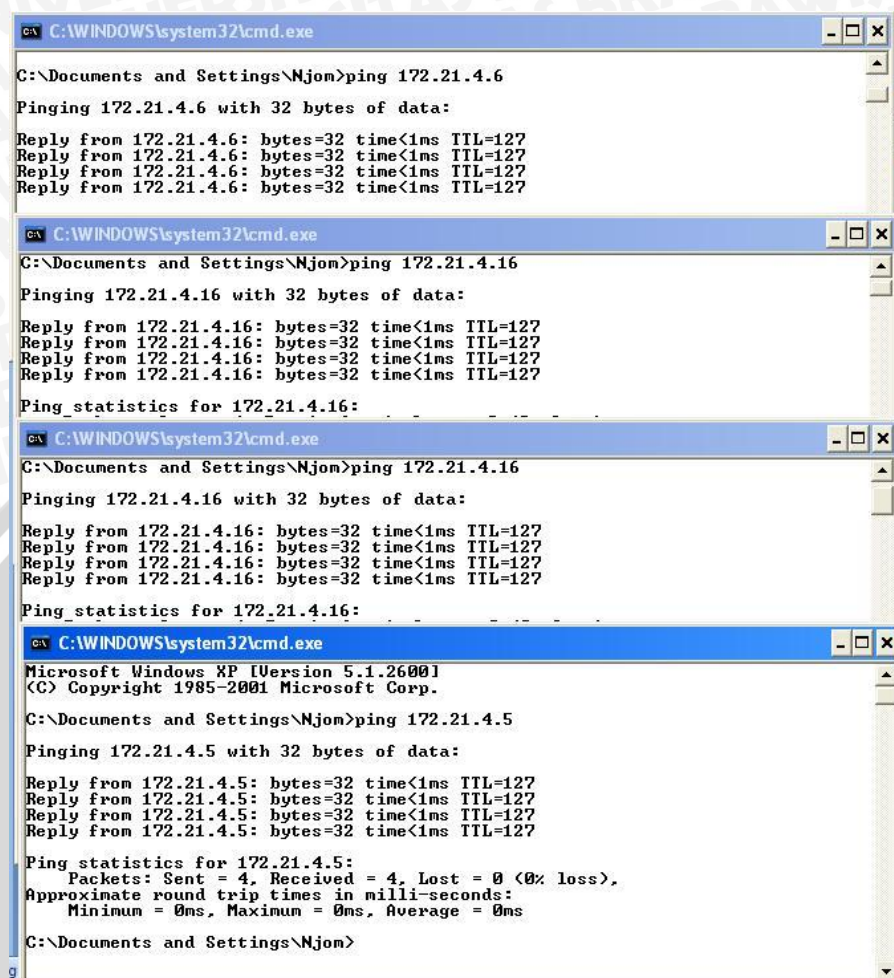
6.1. Pengujian Dasar Jaringan

Pengujian dasar jaringan pada skripsi ini dilakukan agar dapat menunjukkan bahwa jaringan dan *server* telah mampu bekerja dan terkoneksi dengan perangkat keras lain sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Pengujian dasar jaringan dimaksudkan untuk mengetahui apakah implementasi telah berhasil dilakukan dengan benar.

Untuk menguji *server*, dilakukan pengujian dengan melakukan tes ping dari *server* ke calon penerima/pengguna *streaming*. Setelah itu dilakukan *streaming* video dari *server* kepada penerima/pengguna melalui metode *unicast* menggunakan protokol HTTP dan *multicast* menggunakan protokol RTP. Pada pengujian ini dilakukan tes ping terhadap komputer-komputer yang terlibat kemudian diuji dengan melakukan *streaming* menggunakan HTTP setelah itu diuji dengan melakukan *streaming* menggunakan RTP. Tahapan pengujian dasar jaringan seperti pada bab 3 gambar 3.6.

- Tes ping

Pada tahap ini dilakukan tes ping dari *server* ke calon penerima/pengguna *streaming*. Apabila pada tes ping mendapat status reply maka jaringan telah terhubung. Hal tersebut ditunjukkan oleh gambar 6.2.

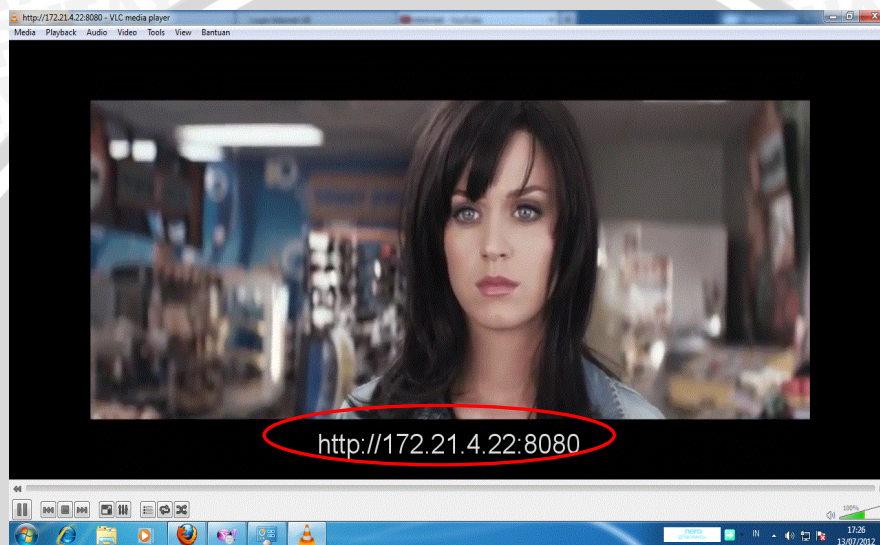


Gambar 6.2 Tes ping pada beberapa penerima

- Unicast

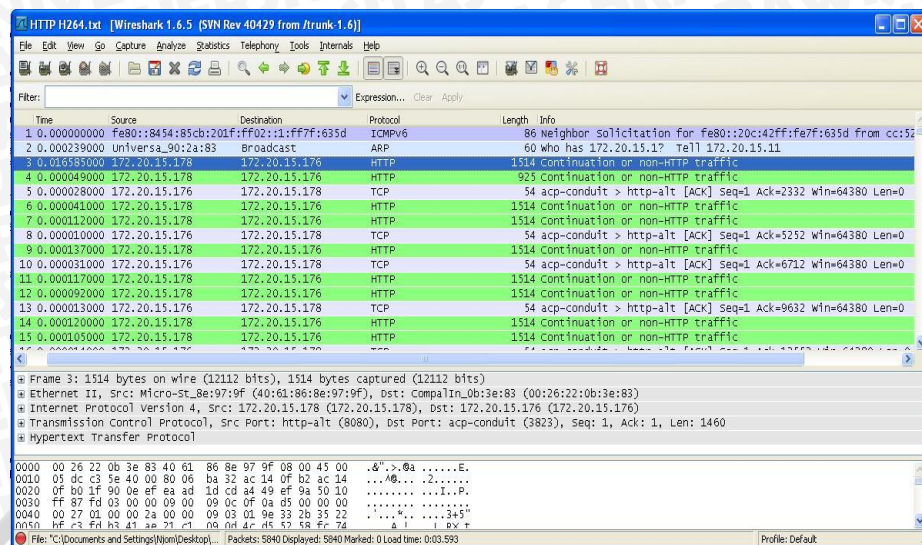
Pada tahap ini dilakukan pengujian server dengan melakukan *streaming* menggunakan *unicast* melalui HTTP. Tahap pertama klik tab media kemudian pilih *stream* (ctrl+s), setelah itu klik tombol add untuk memilih video yang akan di-*streaming*-kan lalu klik tombol *stream* kemudian akan timbul url dari file video yang akan di-*streaming*-kan. Klik *next* kemudian pilih metode streaming yang digunakan yaitu HTTP lalu klik *add* beserta port yang digunakan dengan konfigurasi awal port 8080. *Active transcoding* dinonaktifkan kemudian klik *next*. Aktifkan *stream all elementary stream* kemudian tentukan TTL (time-to-live) dari paket data yang akan di-*streaming*-kan setelah itu klik *stream*..

Penerima dapat menerima *streaming* video melalui VLC dengan membuka *network stream* dan mengisi alamat IP beserta port pengirim data untuk menerima data yang berupa video. Apabila sukses maka akan muncul tampilan video yang sedang diputar seperti pada gambar 6.3 ketika diuji pada topologi jaringan PTIIK. Keterangan yang berada dalam lingkaran berwarna merah menunjukkan bahwa video sukses diputar dengan alamat 172.21.4.22 menggunakan port 8080.



Gambar 6.3 Video berjalan dengan menggunakan metode *unicast* pada jaringan PTIIK

Apabila paket data ditangkap dengan *Wireshark* maka akan muncul keterangan dari aliran data *unicast* yang sedang mengalir memasuki antarmuka komputer. Hal tersebut ditunjukkan oleh gambar 6.4.

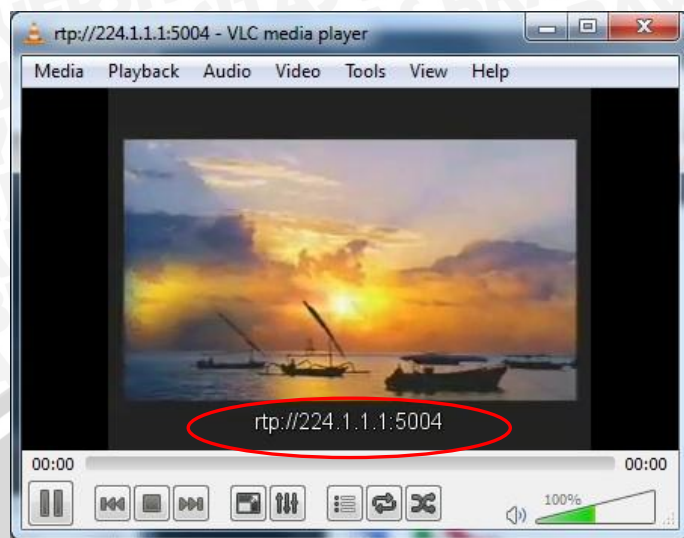


Gambar 6.4 Penangkapan data *unicast* menggunakan Wireshark

- **Multicast**

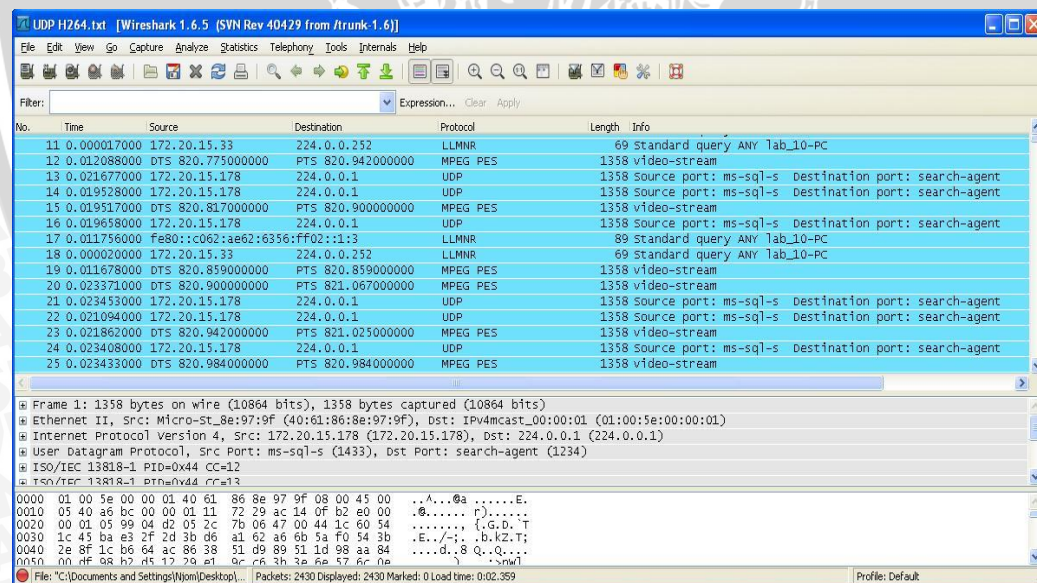
Pada tahap ini dilakukan pengujian *server* dengan melakukan *streaming* menggunakan *multicast* melalui RTP. Langkah-langkah melakukan *streaming* sama dengan *unicast* hingga saat memilih metode *streaming* yang digunakan yaitu RTP lalu klik *add* beserta port yang digunakan dengan konfigurasi awal port 5004. *Active transcoding* dinonaktifkan kemudian klik *next*. Aktifkan *stream all elementary stream* kemudian tentukan TTL (time-to-live) dari paket data yang akan di-*streaming*-kan setelah itu klik *stream*.

Penerima dapat menerima *streaming* video melalui VLC dengan membuka network stream lalu memasukkan alamat group *multicast* yang telah dibuat. Apabila sukses maka akan muncul tampilan video yang sedang diputar pada pemutar video seperti pada gambar 6.5. Keterangan yang berada dalam lingkaran berwarna merah menunjukkan bahwa video sukses diputar dengan alamat 172.21.4.22 menggunakan port 8080



Gambar 6.5 Video berjalan dengan menggunakan metode *multicast* pada jaringan PTIIK

Apabila paket data ditangkap dengan *Wireshark* maka akan muncul keterangan dari aliran data *multicast* yang sedang mengalir memasuki antarmuka komputer. Hal tersebut ditunjukkan oleh gambar 6.6.



Gambar 6.6 Penangkapan data *multicast* menggunakan *Wireshark*

- Analisa Pengujian Dasar Jaringan

Berhasilnya melakukan tes ping dan *streaming* secara *unicast* maupun *multicast* menunjukkan bahwa jaringan dan *server* telah mampu bekerja dan terkoneksi dengan perangkat keras lain sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Dengan hasil tersebut maka dapat dilakukan pengujian berikutnya yaitu pengujian *throughput*.

6.2. Pengujian Throughput

Pengujian *throughput* dilakukan setelah pengujian dasar jaringan telah sukses. Pengujian *throughput* pada skripsi ini dilakukan agar dapat mengetahui kondisi *throughput* pada beberapa skenario tertentu. Skenario tersebut yaitu dengan melakukan *streaming* menggunakan *unicast* yang diwakili oleh HTTP dan *multicast* yang diwakili oleh RTP secara bergantian pada 1, 2, 4 dan 40 penerima sekaligus masing-masing 30 kali pengujian. Langkah-langkah *streaming* yang dilakukan sama seperti pada pengujian dasar jaringan bagian *unicast* dan *multicast* kemudian pada server paket data yang keluar ditangkap menggunakan *wireshark*. Filter data yang digunakan pada *wireshark* dapat dilihat pada lampiran 11 mengenai filter *wireshark*. Tahapan pengujian *throughput* seperti pada bab 3 gambar 3.7.

Hasil pengujian *throughput* setelah diolah maka didapatkan data seperti pada tabel 6.1 dan 6.2. (pengujian dilakukan pada tanggal 2, 3, dan 4 oktober 2012 pada jam sibuk dengan resolusi video 850x480)

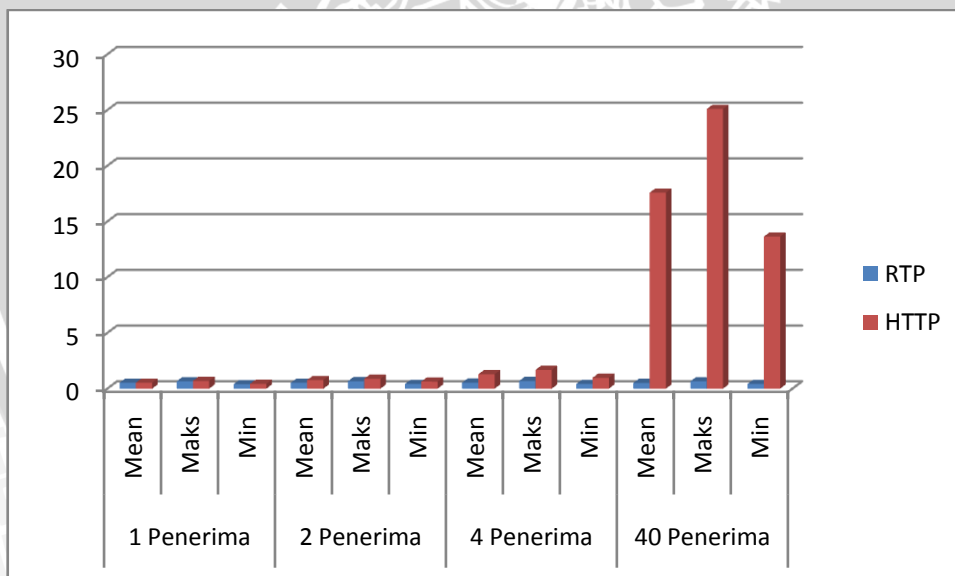
Tabel 6.1 *Throughput* pada jaringan PTIIK untuk 1 dan 2 Penerima

	1 Penerima (MBit/sec)			2 Penerima (MBit/sec)		
	Mean	Maks	Min	Mean	Maks	Min
RTP	0.528267	0.655	0.402	0.5293	0.669	0.418
HTTP	0.5329	0.694	0.429	0.7705	0.885	0.631

Tabel 6.2 *Throughput* pada jaringan PTIIK untuk 4 dan 40 Penerima

	4 Penerima (MBit/sec)			40 Penerima (MBit/sec)		
	Mean	Maks	Min	Mean	Maks	Min
RTP	0.536633	0.688	0.42	0.511833	0.655	0.418
HTTP	1.291	1.682	0.972	17.58733	25.093	13.642

Perbandingan *throughput* antar protokol dapat dilihat pada grafik *throughput* gambar 6.7. Dalam grafik tersebut *streaming* video menggunakan metode *multicast* menggunakan protokol RTP nilai *throughput* cenderung lebih kecil. Hal ini disebabkan karena *datagram* dari UDP lebih pendek daripada TCP. Apabila *throughput* dibandingkan dengan jumlah penerima siaran *streaming*, protokol RTP tidak mengalami kenaikan yang besar. Hal tersebut disebabkan karena *multicast* digunakan saat melakukan *streaming* melalui protokol RTP. Dengan demikian siaran *streaming* menggunakan *multicast* ke banyak penerima dapat dikatakan tidak terlalu membebani kondisi jaringan.

**Gambar 6.7** Grafik hasil pengujian *Throughput*

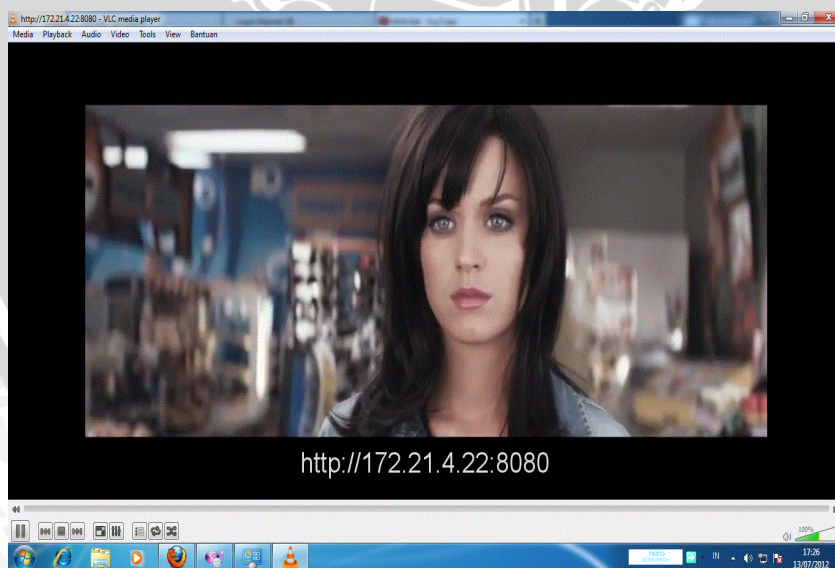
6.3. Pengujian Delay, Jitter

Pengujian *delay* dan *jitter* pada skripsi ini dilakukan agar dapat mengetahui kondisi pengiriman data pada jaringan saat dilakukan *streaming*. Pada pengujian *delay* dan *jitter*, dilakukan penangkapan paket data menggunakan *wireshark* sehingga dapat diketahui *delay* dan *jitter* dari paket data tersebut.

Pada pengujian *delay* dan *jitter* dilakukan dua kali pengujian yaitu menggunakan HTTP dan RTP. Pengujian tersebut dilakukan sebanyak 30 kali pada 1, 2, 4 dan 40 penerima. Tahapan pengujian *delay* dan *jitter* seperti pada bab 3 gambar 3.8.

6.3.1 HTTP

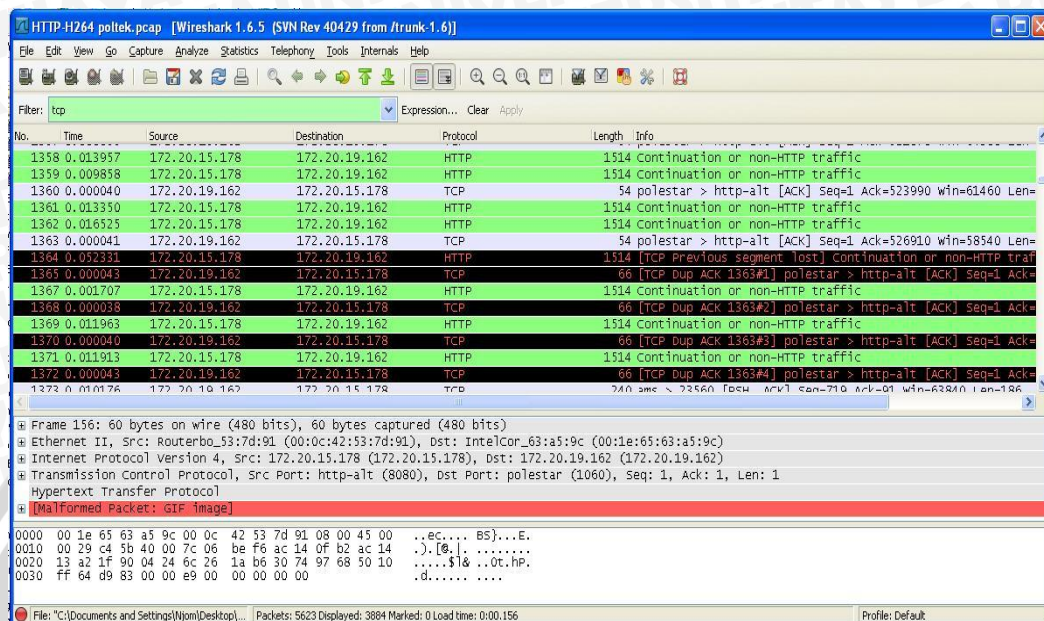
Pada tahap ini akan diuji *streaming* video menggunakan protokol HTTP. Langkah-langkah *streaming* yang dilakukan sama seperti pada pengujian dasar jaringan bagian *unicast* kemudian paket data yang masuk ditangkap menggunakan *wireshark* pada sisi penerima. Filter data yang digunakan pada *wireshark* dapat dilihat pada lampiran 11 mengenai filter *wireshark*. Video sukses berjalan seperti yang ditunjukkan pada gambar 6.8 di jaringan PTIIK,



Gambar 6.8 Video berjalan menggunakan protokol HTTP pada port 8080

Paket data tersebut dianalisa menggunakan *wireshark* seperti gambar 6.9 hingga didapatkan parameter yang dibutuhkan yaitu *delay* dan *jitter*. Data tersebut

tersaji pada tabel 6.3 dan 6.4 (pengujian dilakukan pada tanggal 2, 3, dan 4 oktober 2012 pada jam sibuk dengan resolusi video 850x480).



Gambar 6.9 Penangkapan data video HTTP

Tabel 6.3 Pengujian HTTP jaringan PTIIK untuk 1 dan 2 Penerima

	1 Penerima			2 Penerima		
HTTP	Mean	Maks	Min	Mean	Maks	Min
Throughput (Mbit/sec)	0.533	0.694	0.429	0.771	0.885	0.631
Jitter (msec)	9.34	34.99	0.135	15.05	39.984	0.047
Delay (msec)	34.25	43.45	15.4	38.11	44.835	32.05

Tabel 6.4 Pengujian HTTP jaringan PTIIK untuk 4 dan 40 Penerima

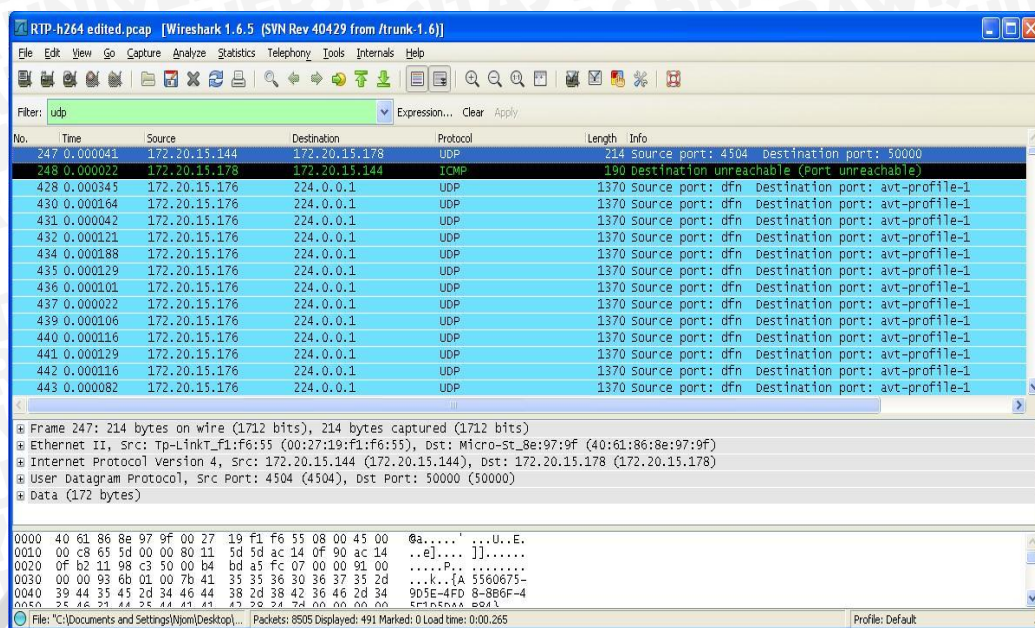
	4 Penerima			40 Penerima		
HTTP	Mean	Maks	Min	Mean	Maks	Min
Throughput (Mbit/sec)	1.291	1.682	0.972	17.59	25.093	13.64
Jitter (msec)	17.61	40	0.024	15.57	41.984	0.047
Delay (msec)	39.71	43.45	32.31	37.14	46.448	18.97

6.3.2 RTP

Pada tahap ini akan diuji *streaming* video menggunakan protokol RTP. Langkah-langkah *streaming* yang dilakukan sama seperti pada pengujian dasar jaringan bagian *multicast* kemudian paket data yang masuk ditangkap menggunakan *wireshark* pada sisi penerima. Filter data yang digunakan pada *wireshark* dapat dilihat pada lampiran 11 mengenai filter *wireshark*. Setelah sukses berjalan dengan indikasi video muncul pada player seperti pada gambar 6.10. Paket data tersebut dianalisa seperti gambar 6.11 hingga didapatkan parameter yang dibutuhkan yaitu *delay* dan *jitter*.



Gambar 6.10 Video berjalan menggunakan protokol RTP pada port 5004



Gambar 6.11 Penangkapan data video RTP dan analisa

Dari data kasar yang direkam, data tersebut diolah menjadi data yang lebih memberi informasi. Data dari pengujian RTP tersebut dengan beberapa penerima tersaji pada tabel 6.5 dan 6.6 (pengujian dilakukan pada tanggal 2, 3, dan 4 oktober 2012 pada jam sibuk dengan resolusi video 850x480).

Tabel 6.5 Pengujian RTP jaringan PTIIK untuk 1 dan 2 Penerima

	1 Penerima			2 Penerima		
RTP	Mean	Maks	Min	Mean	Maks	Min
Throughput (Mbit/sec)	0.528	0.655	0.402	0.529	0.669	0.418
Jitter (msec)	8.003	21.07	0.776	5.002	20.819	0.538
Delay (msec)	19.32	24.11	11.22	20.39	24.731	16.32

Tabel 6.6 Pengujian RTP jaringan PTIIK untuk 4 dan 40 Penerima

	4 Penerima			40 Penerima		
RTP	Mean	Maks	Min	Mean	Maks	Min
Throughput (Mbit/sec)	0.537	0.688	0.42	0.512	0.655	0.418
Jitter (msec)	3.83	14.27	0.811	7.055	21.067	0.638
Delay (msec)	20.56	27.1	16.32	19.38	28.097	15.93

6.2.3 Analisa Pengujian Delay, Jitter

Setelah selesai melakukan pengujian, data diolah dan digabungkan sehingga didapat perbandingan *delay* dan *jitter* antara HTTP dengan RTP. Tabel 6.7 dan 6.8 menunjukkan perbandingan *delay* sedangkan tabel 6.9 dan 10 menunjukkan perbandingan *jitter*.

Tabel 6.7 Tabel *delay* RTP dan HTTP untuk 1 dan 2 Penerima

	1 (msec)			2 (msec)		
Delay	Mean	Maks	Min	Mean	Maks	Min
RTP	19.32	24.11	11.22	20.39	24.731	16.32
HTTP	34.25	43.45	15.4	38.11	44.835	32.05

Tabel 6.8 Tabel *delay* RTP dan HTTP untuk 4 dan 40 Penerima

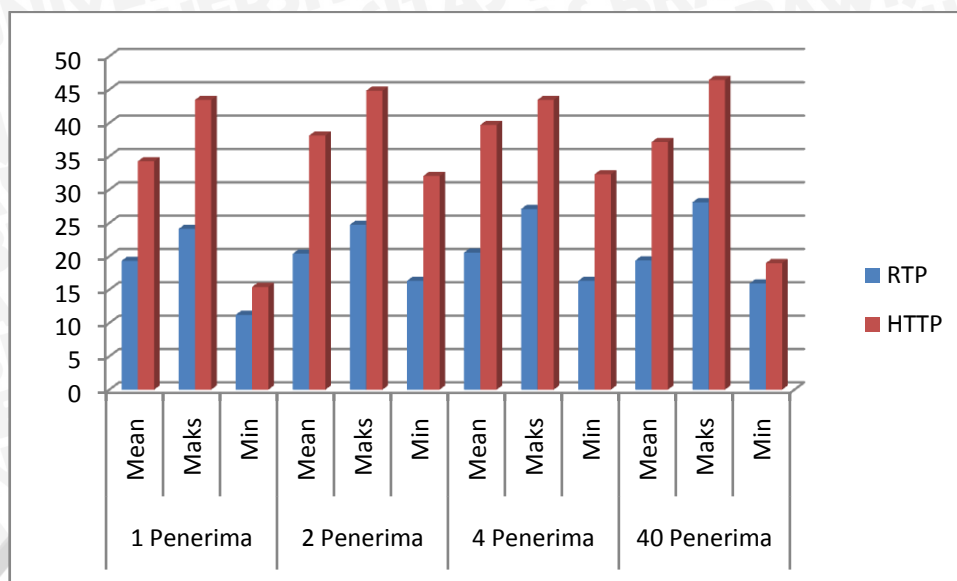
	4 (msec)			40 (msec)		
Delay	Mean	Maks	Min	Mean	Maks	Min
RTP	20.56	27.1	16.32	19.38	28.097	15.93
HTTP	39.71	43.45	32.31	37.14	46.448	18.97

Tabel 6.9 Tabel *jitter* RTP dan HTTP untuk 1 dan 2 Penerima

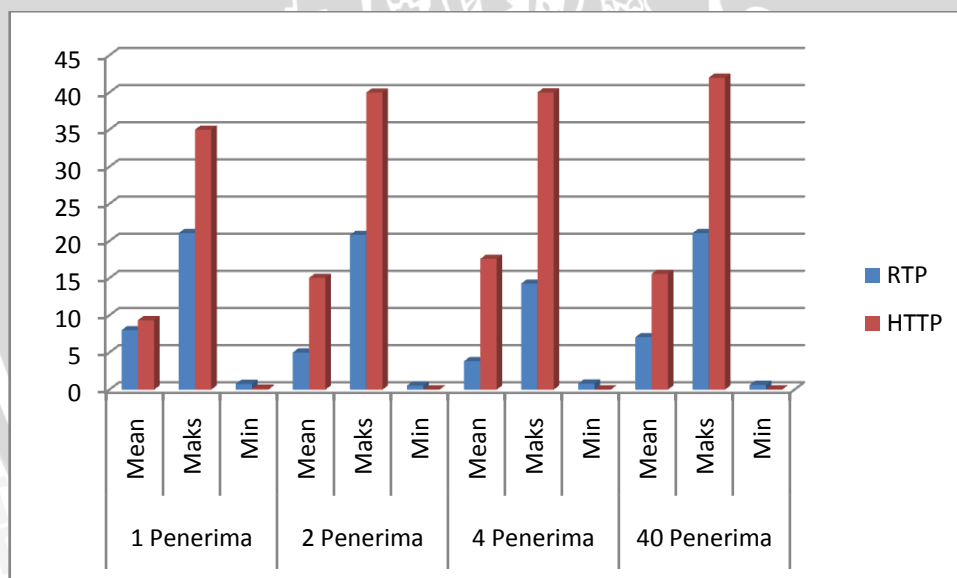
	1 (msec)			2 (msec)		
Jitter	Mean	Maks	Min	Mean	Maks	Min
RTP	8.003	21.07	0.776	5.002	20.819	0.538
HTTP	9.34	34.99	0.135	15.05	39.984	0.047

Tabel 6.10 Tabel *jitter* RTP dan HTTP untuk 4 dan 40 Penerima

	4 (msec)			40 (msec)		
Jitter	Mean	Maks	Min	Mean	Maks	Min
RTP	3.83	14.27	0.811	7.055	21.067	0.638
HTTP	17.61	40	0.024	15.57	41.984	0.047



Gambar 6.12 Grafik hasil pengujian *delay*



Gambar 6.13 Grafik hasil pengujian *jitter*

Pada gambar 6.12, grafik tersebut menunjukkan bahwa *delay* yang terjadi pada HTTP lebih besar daripada RTP. *Delay* tersebut tidak bertambah secara signifikan apabila jumlah penerima bertambah sehingga cenderung tetap. Perbedaan besar *delay* antara RTP dengan HTTP disebabkan karena perbedaan *header* TCP dengan UDP, dimana ukuran *header* TCP lebih panjang daripada

UDP. Menurut rekomendasi pada tabel 6.7 dan 6.8, *delay* yang terjadi baik RTP maupun HTTP masih dalam golongan yang dapat di terima oleh alat penerima kebanyakan.

Pada gambar 6.13, grafik tersebut menunjukkan bahwa dengan penerima *streaming* video berjumlah 1, 2, 4 dan 40 nilai *jitter* HTTP lebih lebih tinggi dari RTP. Hal ini menandakan kualitas video yang di-*streaming*-kan dengan RTP lebih bagus daripada HTTP. Apabila jumlah penerima meningkat, *bandwidth* yang dibutuhkan HTTP juga meningkat. Telah terbukti pada pengujian sebelumnya yaitu pengujian *throughput*. Dengan demikian apabila terdapat banyak penerima, akan lebih efisien menggunakan RTP dengan metode *multicast*. Kualitas video lebih bagus dan tidak mengalami peningkatan *bandwidth* yang dibutuhkan apabila jumlah penerima meningkat. Meskipun demikian baik HTTP maupun RTP mempunyai kualitas video juga masih tergolong bagus menurut standar yang ada pada bab 2 tabel 2.1 dan 2.2 saat di ujikan di PTIIK.

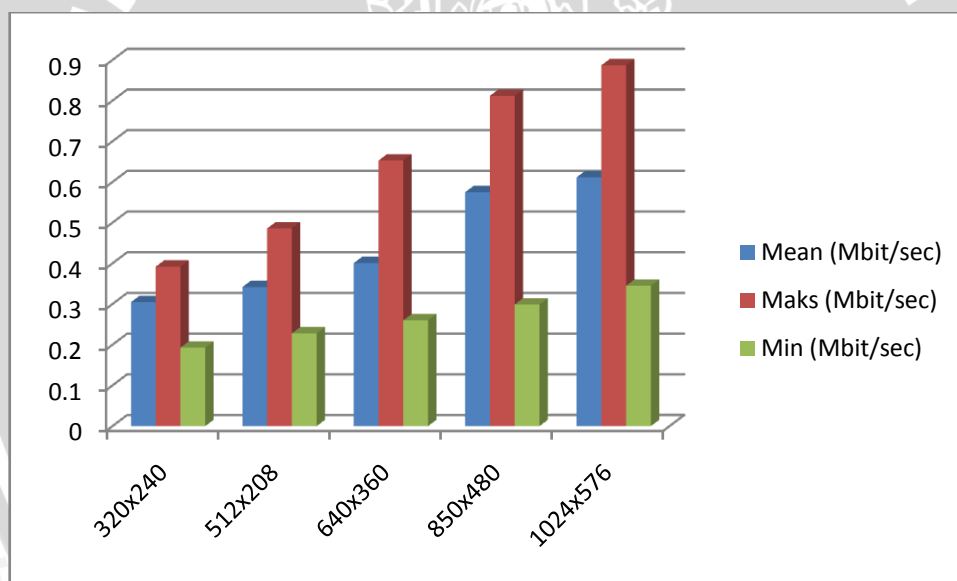
6.4. Pengujian Resolusi pada WiFi

Pengujian resolusi dimaksudkan untuk menguji bagaimanakah kondisi data apabila dilakukan *streaming* menggunakan *multicast* dengan resolusi video yang berbeda-beda melalui perangkat WiFi. Pengujian dilakukan dengan *streaming multicast* menggunakan protokol RTP dari *server* ke penerima. Langkah-langkah *streaming* yang dilakukan sama seperti pada pengujian dasar jaringan bagian *multicast* kemudian paket data yang masuk ditangkap menggunakan *wireshark* pada sisi penerima. Sedangkan untuk *throughput*, *wireshark* digunakan di sisi server untuk menangkap paket data yang keluar. Filter data yang digunakan pada *wireshark* dapat dilihat pada lampiran 11 mengenai filter *wireshark*. Paket data ditangkap dengan *wireshark* kemudian dianalisa berdasarkan *throughput*, *delay*, *jitter*, dan *packet loss*. Pengujian dilakukan sebanyak 30 kali. Tahapan pengujian resolusi seperti pada bab 3 gambar 3.9.

Pada pengujian resolusi dilakukan pengujian *streaming* video dengan 5 resolusi video yang berbeda melalui WiFi. Resolusi tersebut yaitu 320x240px, 512x208, 640x360, 850x480, 1024x576. Setelah selesai melakukan pengujian, data digabungkan sehingga didapat perbandingan antar resolusi yang diujikan.

Tabel 6.11 Hasil pengujian *throughput* dengan berbagai resolusi

Resolusi	Mean (Mbit/sec)	Maks (Mbit/sec)	Min (Mbit/sec)
320x240	0.304733333	0.392	0.193
512x208	0.3418	0.486	0.228
640x360	0.400933333	0.653	0.26
850x480	0.575066667	0.812	0.299
1024x576	0.611533333	0.887	0.345



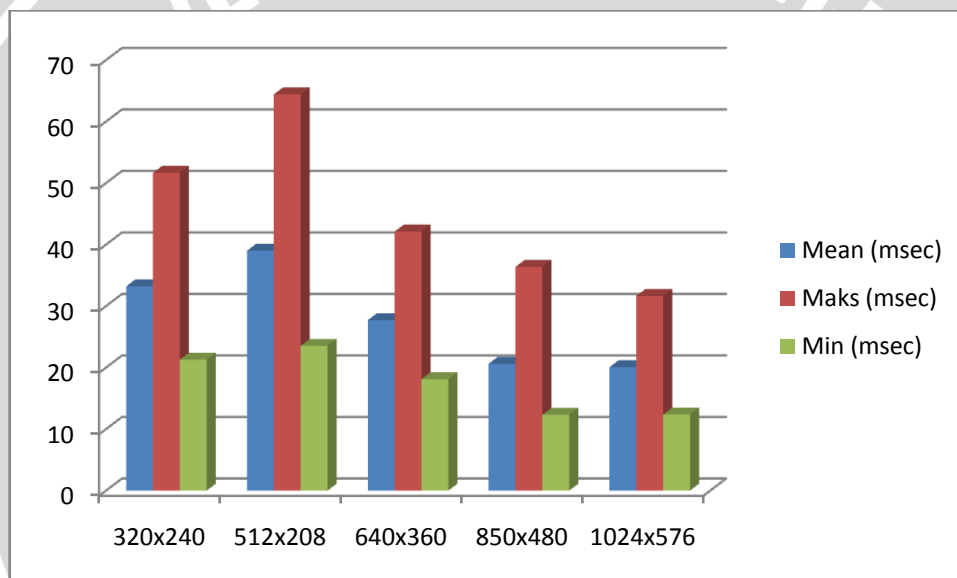
Gambar 6.14 Grafik hasil pengujian *Throughput* dengan berbagai resolusi

Perbandingan *throughput* antar protokol dapat dilihat pada grafik *throughput* gambar 6.14. Dalam grafik tersebut *throughput streaming* video menggunakan metode *multicast* meningkat seiring meningkatnya resolusi video yang di-*streaming*-kan. Besarnya *delay* dan *jitter* mengalami peningkatan apabila dibandingkan dengan pengujian menggunakan kabel UTP.

Kualitas dari video ditinjau dari *delay* dan *jitter* seperti ditunjukkan pada gambar 6.15 dan 6.16 tergolong bagus menurut standar pada tabel 6.12 dan 6.13 dengan nilai *delay* dibawah 150 msec dan nilai *jitter* dibawah 75 msec.

Tabel 6.12 Tabel *delay* dari berbagai resolusi

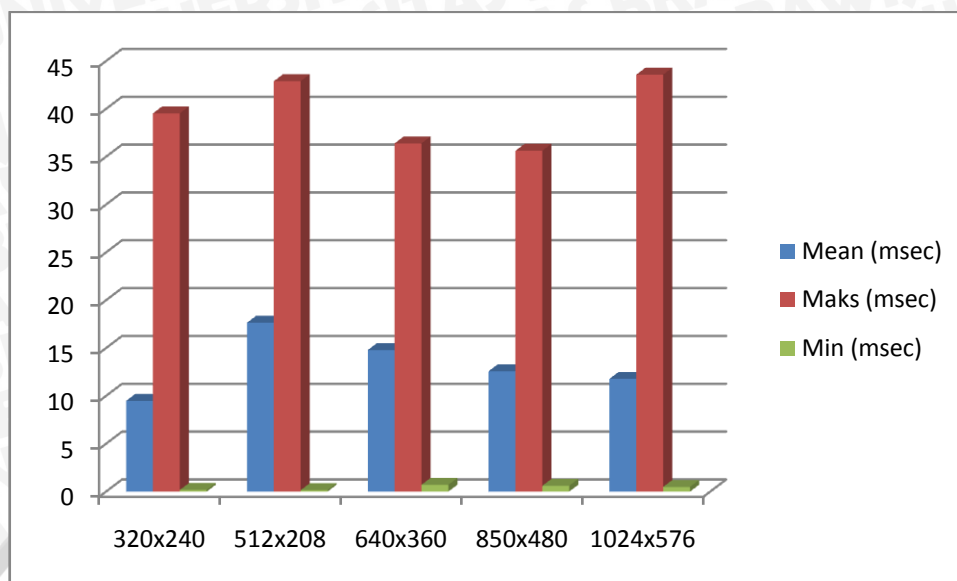
Resolusi	Mean (msec)	Maks (msec)	Min (msec)
320x240	33.20550703	51.66666667	21.2959459
512x208	38.98932343	64.43150685	23.5336891
640x360	27.70621706	42.12867647	18.1258065
850x480	20.63041516	36.37037037	12.331058
1024x576	20.06227871	31.65644172	12.3573883



Gambar 6.15 Grafik hasil pengujian *delay* dengan berbagai resolusi

Tabel 6.13 Tabel *jitter* dari berbagai resolusi

Resolusi	Mean (msec)	Maks (msec)	Min (msec)
320x240	9.486540833	39.5383	0.17155
512x208	17.64633617	42.894235	0.148
640x360	14.79882167	36.399305	0.71355
850x480	12.583643	35.63105	0.6069
1024x576	11.7856745	43.58145	0.4879



Gambar 6.16 Grafik hasil pengujian *jitter* dengan berbagai resolusi

Pada jaringan berbasis *wireless* di PTIIK, paket data mengalami kehilangan yang besar. Hal ini disebabkan karena jaringan berbasis *wireless* dapat mengalami penurunan kualitas sinyal sehingga kurang reliabel. Faktor lain yang dapat mempengaruhi yaitu jumlah pengguna jaringan *wireless* tersebut. Apabila pengguna jaringan *wireless* tersebut cenderung meningkat, maka *packet loss* yang terjadi akan ikut meningkat.

Tabel 6.14 Tabel *packet loss* dari berbagai resolusi

Resolusi	Mean (%)	Maks (%)	Min (%)
320x240	13.58	32.8	0.7
512x208	24.12	51.8	3.5
640x360	37.74	57.7	1.1
850x480	45.37333	66.4	9.7
1024x576	56.26	79.9	10

Packet loss yang tinggi menyebabkan video yang di-*streaming*-kan akan terlihat pecah-pecah atau bahkan berhenti sesaat. Berdasarkan standar pada bab 2 tabel 2.3, *streaming* video menggunakan resolusi 320x240 px masuk dalam kategori sedang dan *streaming* video menggunakan resolusi 512x208 px masuk

dalam kategori jelek, sedangkan sisanya mengalami *packet loss* yang terlalu besar. Berbeda halnya apabila melakukan *streaming* menggunakan jaringan berbasis kabel UTP. Melalui kabel UTP, *streaming multicast* di PTIIK tidak mengalami *packet loss*.



BAB VII

PENUTUP

Penutup berisi kesimpulan dari analisis dan implementasi sistem serta saran untuk pengembangan sistem.

7.1 Kesimpulan

Dari analisis dan implementasi jaringan multicast untuk *streaming* video, dapat disimpulkan bahwa :

1. Penyampaian konten multimedia di lingkungan kampus PTIIK dapat dilakukan dengan *streaming* video menggunakan *multicast*.
2. Penyampaian konten multimedia menggunakan *multicast* dapat dilakukan dan diterima dimanapun selama terhubung di jaringan PTIIK menggunakan protokol routing PIM (*Protocol Independent Multicasting*) dengan mode SM (*Sparse Mode*) serta IGMPv2 dimana router PTIIK menjadi RP (*Rendevous Point*) dari topologi *multicast*.
3. Hasil dari pengujian throughput menunjukkan bahwa untuk keperluan *streaming* video dengan kondisi jumlah penerima banyak, penggunaan protokol *multicast* lebih efisien dibandingkan dengan protokol *unicast* yaitu *throughput* menggunakan *multicast* dengan 4 penerima sebesar 0.536633 Mbit/sec sedangkan *unicast* sebesar 1.291 Mbit/sec.
4. Hasil dari pengujian *delay* dan *jitter* menunjukkan bahwa kualitas video yang di-*streaming*-kan menggunakan *multicast* lebih baik daripada *unicast* karena nilai *delay* dan *jitter*-nya lebih kecil. *Delay* dan *jitter multicast* pada 4 penerima sebesar 20.557 msec dan 3.8299 msec sedangkan *unicast* sebesar 39.706 msec dan 17.607 msec, namun keduanya masih tergolong bagus menurut standar ITU dan Telkom.
5. Hasil dari pengujian *resolusi* menunjukkan bahwa video yang distreamingkan melalui jaringan wifi mengalami *packet loss* yang besar. Dengan menggunakan jaringan wifi, resolusi video yang dapat si-*streaming*-kan adalah 320x240 px dan 512x208 px. Sedangkan *streaming*

video menggunakan jaringan berbasis kabel UTP tidak mengalami kehilangan data. Hal ini disebabkan karena pada jaringan WiFi mengalami penurunan kualitas sinyal.

7.2 Saran

Saran dalam pengembangan **streaming video multicast** adalah.

1. Untuk pengembangan lebih lanjut sistem video streaming ini dapat dikembangkan untuk *live streaming*.
2. Router akan mendahulukan transfer *multicast* daripada *unicast* sehingga pada penggunaan *multicast* diharapkan memperhitungkan besar file yang akan ditransfer karena pada umumnya kabel yang terpasang pada topologi adalah jenis 100Mb/s.
3. Pada jaringan WiFi mengalami *packet loss* yang besar, untuk memaksimalkan dapat digunakan media lain seperti kabel UTP.
4. Untuk pengembangan lebih lanjut, perlu diadakan penelitian *multicast* dari PTIIK ke luar lingkungan kampus, sehingga pihak luar dapat menerima siaran streaming dari PTIIK.

Daftar Pustaka

- [ABR-10] Abror, Ahmad Afis, M.Zen Samsono Hadi, Idris Winarno. 2010. *Rancang Bangun Dan Analisa QoS Audio Dan Video Streaming pada Jaringan MPLS VPN*. Jurusan Teknik Telekomunikasi Politeknik Negeri Surabaya.
- [ANO-11] Azhary, Wan Luptan. 2006. *Rancang Bangun Intranet Video Live Streaming untuk Proses Belajar Mengajar*. Akses dari: <http://www.docstoc.com/docs/37709216/PROYEK-AKHIR-RANCANG-BANGUN-INTRANET-VIDEO-LIVE-STREAMING-UNTUK>. Tanggal akses: 11-07-2011.
- [AST-11] VLC Official Site. 2011. *Streaming How To New*. Diakses dari : <http://wiki.videolan.org>. Tanggal Akses: 22-07-2011
- [AVI-05] McNell, David A. 2005. *Pro-AV: Inside Media Streaming*. Special Technologies Group of Newcomb & Boyd
- [DAR-11] Kusuma , Darmawan Surya, Kodrat Iman Satoto. 2011. *Implementasi Sistem CCTV Live Streaming Untuk Memantau Kinerja Proyek PT Pembangunan Perumahan (Persero)*. Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro.
- [DIK-11] Diktat TI Telkom. 2011. *Bab Xvi Quality Of Service (Qos) Dan Pengukurannya*
- [ERI-07] Setton, Eric, Bern Girod. 2007. *Peer-to-Peer Video Streaming*. New York: Springer.
- [FLA-11] FLAC Official Site. 2011. *What Is Flac*. Diakses dari : <http://www.flac.org>. Tanggal Akses: 31-10-2011
- [FOU-11] Fourcc Official Site. 2011. *Codec*. Diakses dari : <http://www.fourcc.org>. Tanggal Akses: 1-11-2011

- [FRA-02] Kozamernlk, Franc. 2002. *Media Streaming Over Internet*. EBU Technical Department
- [GAT-09] Santoso, Gatot, Adhi Susanto, Marshal Budi Wardani. 2009. *Perancangan Konten M-Learning Dengan Sistem Live Multimedia Berbasis Selular*. Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Sains & Teknologi AKPRIND Yogyakarta.
- [HAR-08] Harte, Peter, Lawrence. 2008. *Introduction to Data Multicasting*. Fuquay-Varina: Althos Publishing
- [ITU-95] Anonim. 1995. *ITU-T Recommendation E.800*.
- [JAN-10] Chen,Jane. 2010. *Third Party or Open Source Software Integration*. ACTi
- [MIK-12] Mikrotik Official Site. 2012. *Manual:Multicast Detailed Example*. Diakses dari : <http://www.wiki.mikrotik.com> Tanggal Akses: 28-5-2012
- [MMS-11] Anonim. 2011. *Windows Media HTTP Streaming Protocol Specification*. Microsoft Corporation.
- [RTSP-98] H. Schulzrinne, A. Rao, R. Lanphier. 1998. *Real Time Streaming Protocol (RTSP)*. IETF (Internet Engineering Task Force) draft.
- [RTP-03] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. 1998. *RTP: A Transport Protocol for Real-Time Applications*. IETF (Internet Engineering Task Force) draft.
- [SAS-07] Sastra, Nyoman Putra, I Mada Oka Widyantara. 2007. *Video Streaming MPEG-4 Pada Jaringan Wireless IP 802.11 b*. Jurusan Teknik Elektro Universitas Udayana.
- [SEP-09] Septima, Uzma. 2009. *Video Steaming dengan Video LAN Project*. Jurusan Teknik Elektro Politeknik Negeri Padang.
- [SMD-02] Topic, Michael. 2002. *Streaming Media Demystified*. New York: Mc. Graw Hill.

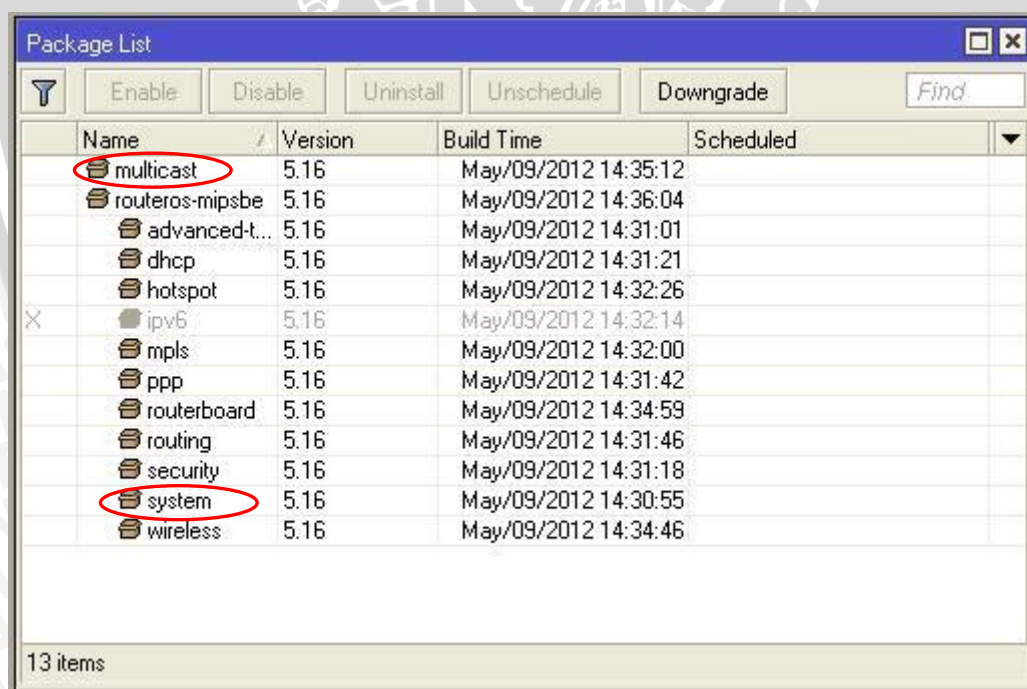
- [SOI-08] Asaeda,Hitoshi. 2008. *Toward IP Multicast Deployment*. School of Media and Governance Keio University. SOI ASIA
- [SOM-11] Sommerville, Ian. 2011. *Software Engineering Ninth Edition*. Boston: Addison-Wesley
- [THI-07] Ramalingam,Thiya, Scott Firestone, Steve Fry. 2007. *Voice and Video Conferencing Fundamentals*. Indianapolis: Cisco Press.
- [TRI-08] Hens, Francisco J, Jose M. Caballero. 2008. *Triple Play: Building The Converged Network for IP, VOIP, and IPTV*. London: John Wiley & Sons Ltd.
- [VAS-05] Austerberry, David. 2005. *The Technology of Video & Audio Streaming*. -2nd ed. Oxford: Focal Press.
- [WES-00] Westerink, Peter, Lisa Amini, Sundas Veliah, Will Belknap. 2000. *A Live Intranet Distance Learning System Using MPEG-4 Over RTP/RTSP*. Howthorne: IBM T.J Watson Research Center
- [WIR-11] Wireshark Official Site. 2011. *Wireshark*. Diakses dari : <http://www.wireshark.org>. Tanggal Akses: 29-07-2011

LAMPIRAN

Lampiran 1 Instalasi Paket Multicast Mikrotik

Pada Router Mikrotik v3.x Routing-test dan paket multicast tidak kompatibel. Pada kasus tersebut ketika keduanya hadir salah satu dari mereka akan dinonaktifkan. Mulai dari v5.15 telah kompatibel dengan *multicast* sehingga kita dapat mengaktifkan mode multicast pada *access point*. Mode ini menggunakan *unicast* penerima alamat MAC untuk meneruskan lalu lintas *multicast*.

Paket yang dibutuhkan untuk melakukan *multicast* adalah sistem dan *multicast* seperti pada **Gambar 1**. Paket *multicast* dapat diunduh secara terpisah. Paket tersebut dapat ditransfer melalui Winbox dengan metode *drag-and-drop* atau dapat ditransfer melalui FTP dengan tujuan IP dari *router* yang akan ditambahkan paket *multicast*. Setelah transfer selesai, *Routerboard* harus di restart terlebih dahulu untuk penginstalan.



Name	Version	Build Time	Scheduled
multicast	5.16	May/09/2012 14:35:12	
routeros-mipsbe	5.16	May/09/2012 14:36:04	
advanced-t...	5.16	May/09/2012 14:31:01	
dhcp	5.16	May/09/2012 14:31:21	
hotspot	5.16	May/09/2012 14:32:26	
ipv6	5.16	May/09/2012 14:32:14	
mpls	5.16	May/09/2012 14:32:00	
ppp	5.16	May/09/2012 14:31:42	
routerboard	5.16	May/09/2012 14:34:59	
routing	5.16	May/09/2012 14:31:46	
security	5.16	May/09/2012 14:31:18	
system	5.16	May/09/2012 14:30:55	
wireless	5.16	May/09/2012 14:34:46	

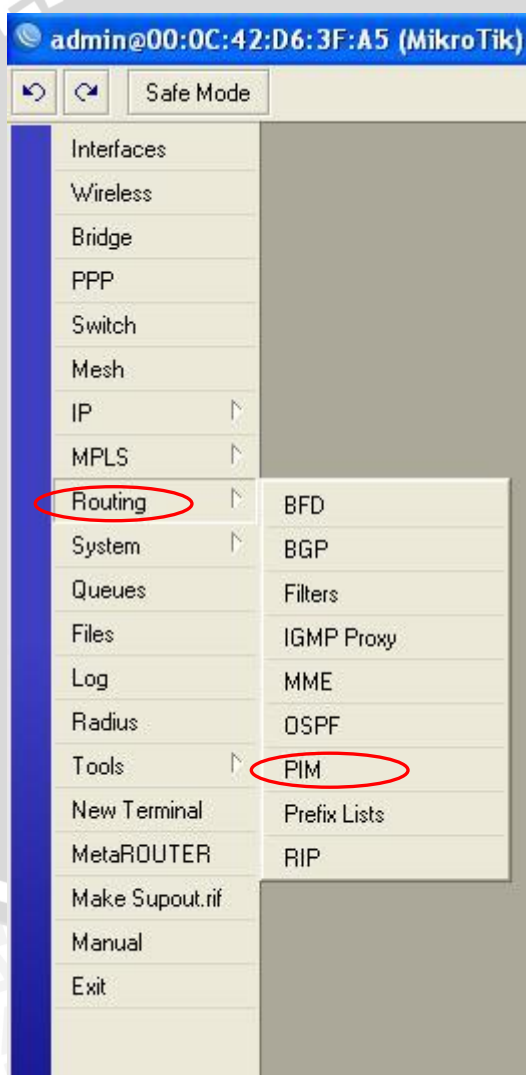
13 items

Gambar 1 Daftar paket pada *router Mikrotik*

Lampiran 2 Konfigurasi Multicast Mikrotik

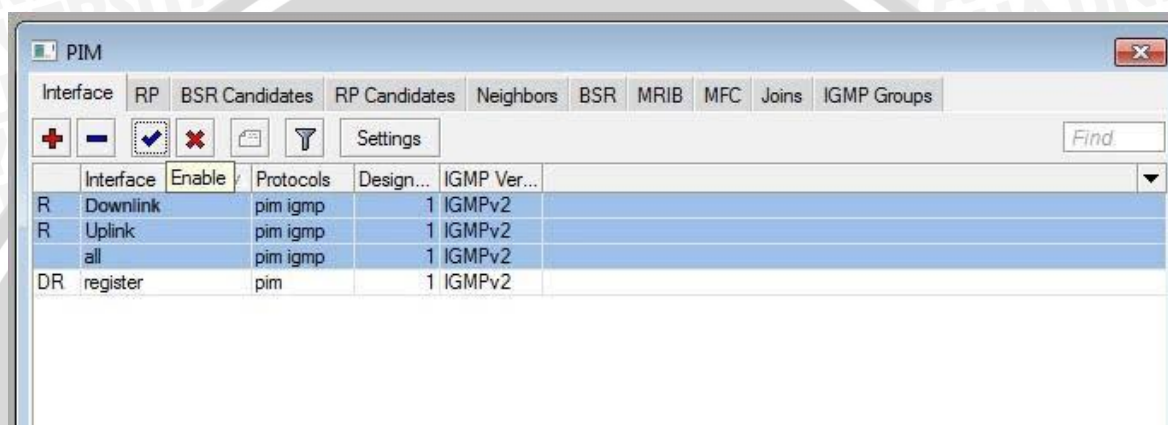
Terdapat 2 poin penting dalam konfigurasi *multicast* yaitu pembuatan *rendezvous point* (RP) dan routing *multicast*. Setelah paket *multicast* terpasang, maka akan muncul tab baru pada kolom Menu baris *Routing*. Bila di klik akan tampak tambahan menu baru yaitu PIM seperti yang terlihat pada **Gambar 2**.

Mikrotik mendukung protokol routing PIM-SM dimana PIM merupakan *Protocol Independent Multicast* dimana protokol routing tersebut berdiri sendiri, tidak terikat terhadap routing *unicast*. dan SM merupakan *sparse-mode*, lawan dari *dense-mode* yang mana *sparse-mode* mengontrol pesan memastikan penerima paket merupakan *subnet* yang diminta untuk menerimanya [MIK-12].

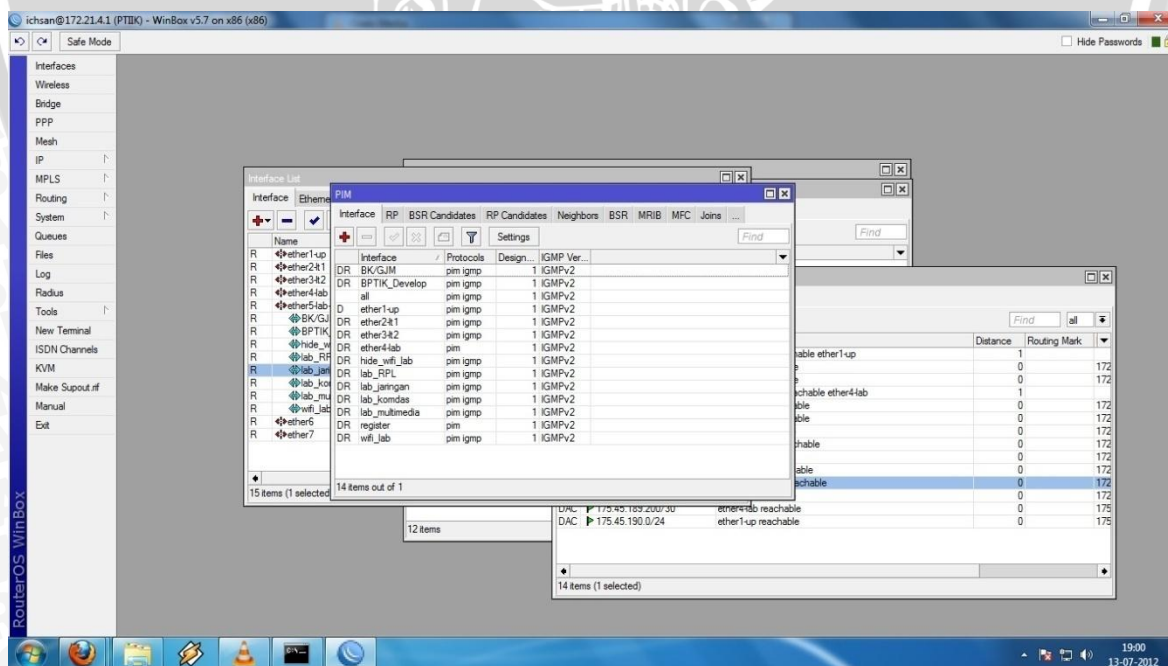


Gambar 2 Menu baru *PIM*

Apabila menu baru tersebut akan muncul jendela baru untuk melakukan konfigurasi PIM seperti yang diperlihatkan pada **Gambar 3**. Konfigurasi Mikrotik pada router menggunakan RP yang diletakkan pada Mikrobits pada PTIIK. Konfigurasi tersebut seperti pada **Gambar 4** dilakukan konfigurasi pada PIM untuk mendaftarkan antarmuka yang akan berhubungan melalui *multicast* seperti pada gambar dibawah

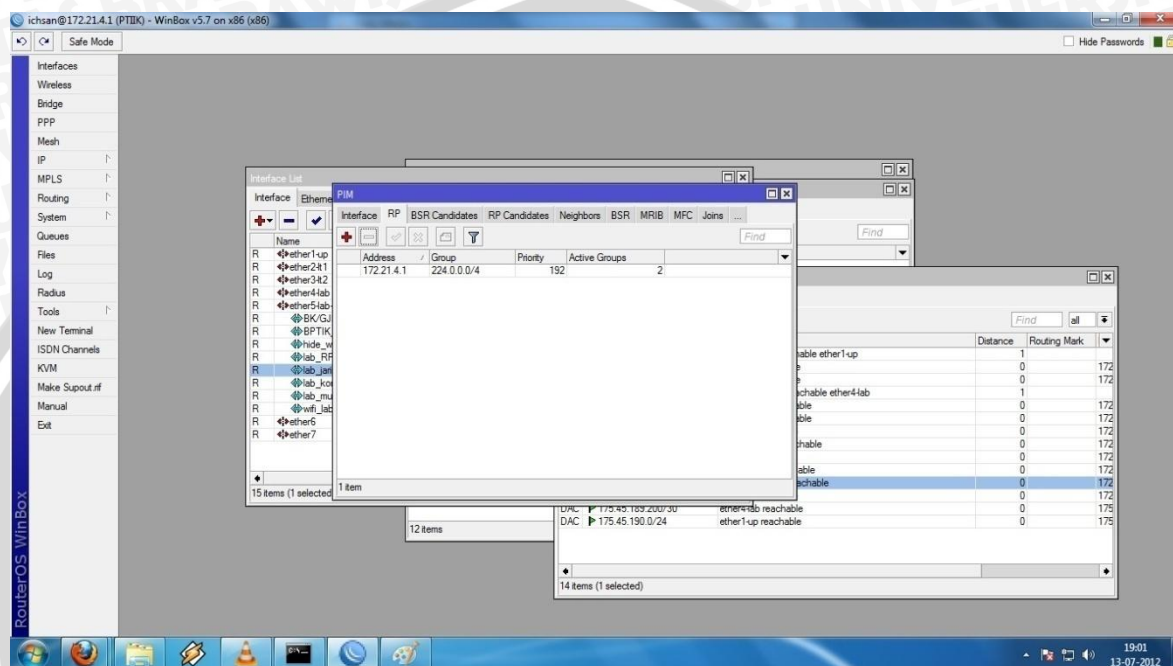


Gambar 3 Konfigurasi PIM

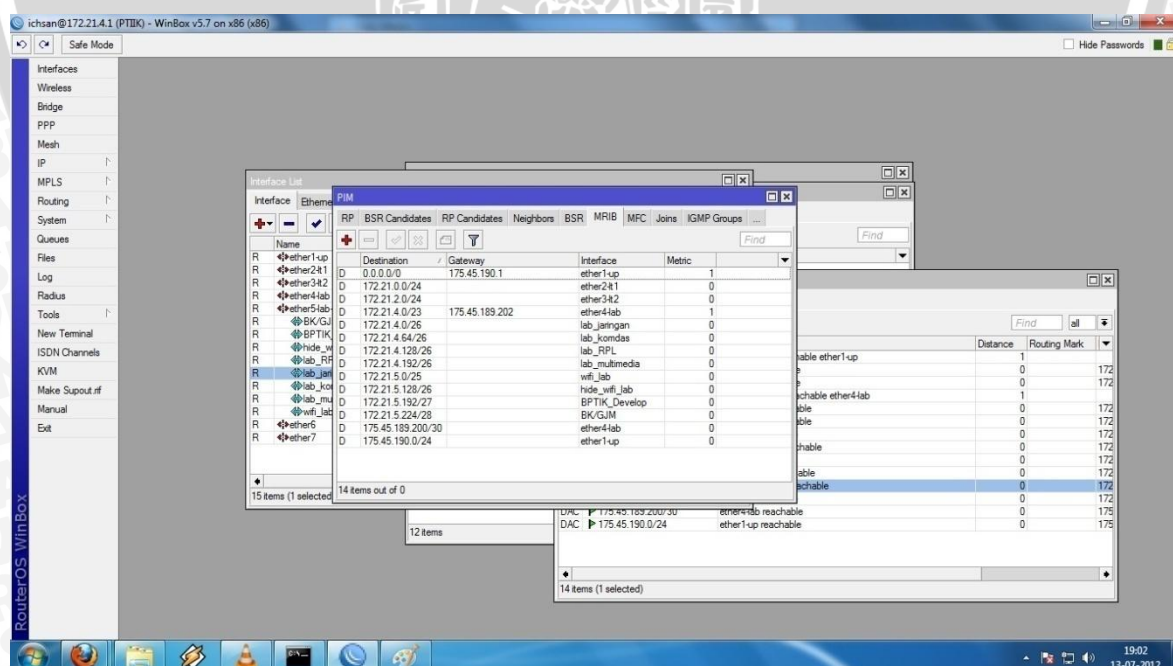


Gambar 4 Konfigurasi antarmuka PIM

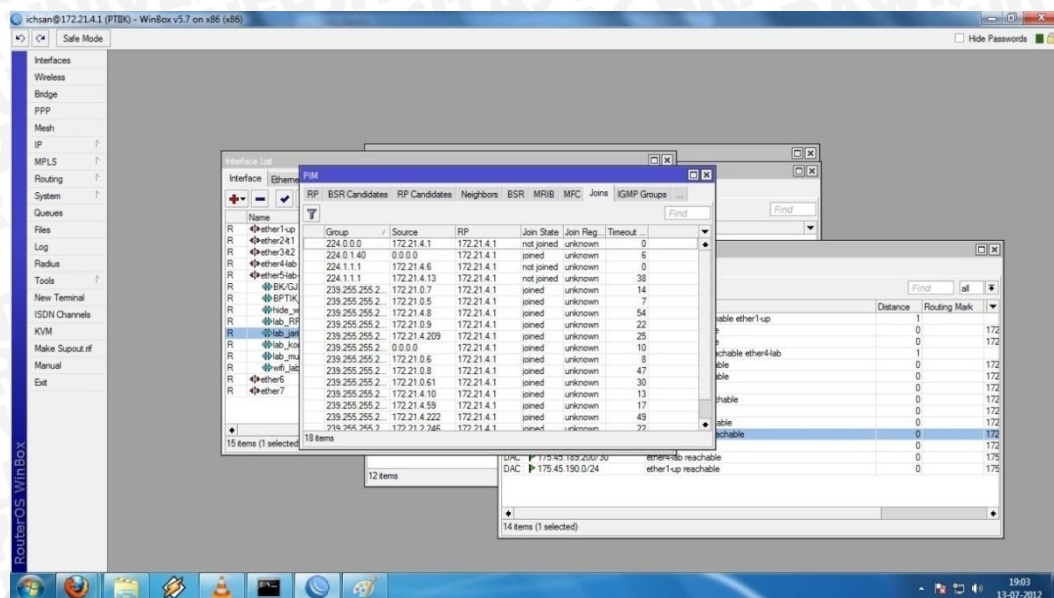
Setelah mendaftarkan antarmuka, berikutnya mendaftarkan IP untuk menjadi RP dari group *multicast* seperti pada **Gambar 5**. RP tersebut akan menjadi tujuan men-*streaming*-kan video dan menerima siaran *streaming* dari server. Karena pada PTIIK hanya menggunakan 1 router maka pada MRIB telah tedarftar secara otomatis seperti pada **Gambar 6**.



Gambar 5 IP Rendezvous Point

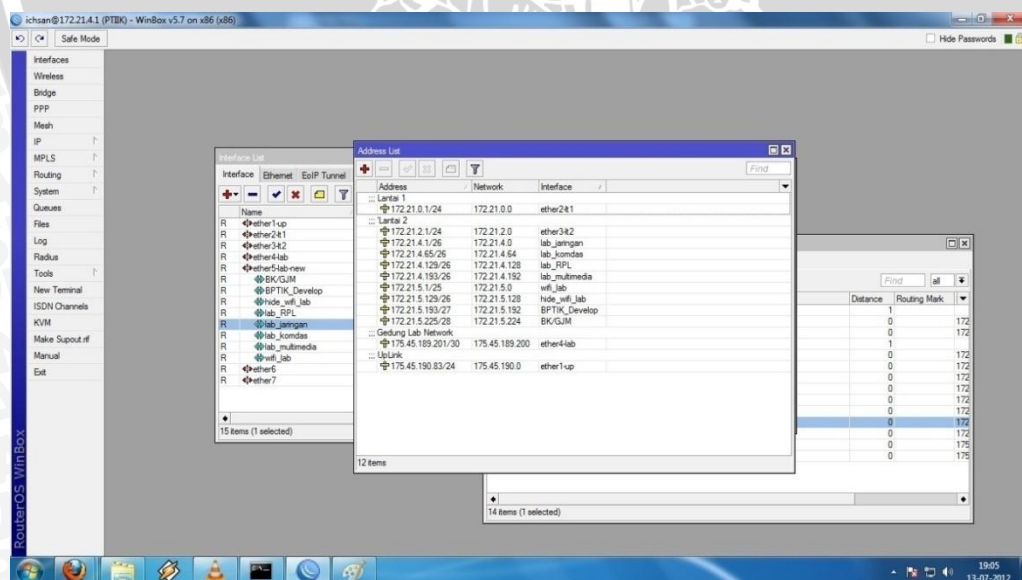


Gambar 6 Tabel routing MRIB



Gambar 7 Join state

Apabila kita melakukan *streaming multicast* dan ada yang join maka akan terlihat pada jendela joins seperti pada Gambar 7. Pada Gambar 7 terlihat alamat IP 224.1.1.1 yang merupakan IP group *multicast* yang menjadi tujuan dari penerima. Semua *penerima* yang bergabung pada group *multicast* akan tampak pada jendela joins. Pada Gambar 8 menunjukkan IP dari segmen mana saja yang dapat menerima siaran *multicast*.



Gambar 8 Daftar alamat IP yang terhubung pada Mikrobts

Lampiran 3 Data Throughput

THROUGHPUT (MBit/sec)							
HTTP (Penerima)				RTP (Penerima)			
1	2	4	40	1	2	4	40
0.479	0.838	1.682	25.093	0.505	0.559	0.492	0.505
0.45	0.642	1.109	14.482	0.551	0.521	0.669	0.551
0.534	0.792	1.157	16.338	0.536	0.561	0.642	0.536
0.539	0.726	1.519	21.145	0.487	0.655	0.505	0.47
0.655	0.885	0.977	20.977	0.418	0.554	0.548	0.418
0.49	0.823	1.482	20.838	0.556	0.514	0.452	0.556
0.436	0.807	1.338	14.338	0.565	0.588	0.553	0.521
0.573	0.879	1.293	14.293	0.452	0.569	0.488	0.561
0.453	0.682	1.332	16.332	0.593	0.539	0.441	0.441
0.541	0.631	1.172	13.642	0.467	0.526	0.482	0.482
0.479	0.838	1.682	14.792	0.402	0.432	0.505	0.559
0.475	0.642	1.109	15.726	0.619	0.669	0.551	0.521
0.649	0.792	1.157	21.885	0.642	0.642	0.536	0.561
0.614	0.726	1.519	13.807	0.505	0.505	0.47	0.553
0.655	0.885	1.093	23.879	0.548	0.548	0.478	0.488
0.449	0.823	1.482	18.682	0.452	0.452	0.556	0.441
0.436	0.807	1.338	17.631	0.553	0.553	0.565	0.482
0.673	0.879	1.145	14.838	0.488	0.488	0.42	0.553
0.453	0.682	1.265	16.642	0.441	0.441	0.593	0.488
0.541	0.631	1.172	14.172	0.482	0.482	0.467	0.441
0.479	0.838	1.682	16.655	0.559	0.505	0.559	0.482
0.457	0.642	1.109	18.449	0.521	0.551	0.521	0.655
0.646	0.792	1.157	16.436	0.561	0.536	0.561	0.554
0.694	0.726	1.519	15.673	0.655	0.47	0.655	0.514
0.605	0.885	1.377	18.453	0.554	0.418	0.554	0.47
0.429	0.823	1.482	18.541	0.514	0.556	0.514	0.418
0.436	0.807	1.338	15.338	0.588	0.565	0.688	0.556
0.673	0.879	1.399	21.399	0.569	0.42	0.569	0.565
0.453	0.682	0.972	19.972	0.539	0.593	0.539	0.42
0.541	0.631	1.172	17.172	0.526	0.467	0.526	0.593

Lampiran 4 Data Delay

DELAY (msec)							
HTTP (Penerima)				RTP (Penerima)			
1	2	4	40	1	2	4	40
0.039286	0.023961	0.039286	0.039286	0.021628	0.01953	0.027097	0.016322
0.019322	0.022455	0.044848	0.032051	0.019786	0.021053	0.016322	0.016986
0.034848	0.021221	0.039322	0.033584	0.02043	0.01944	0.016986	0.021563
0.035401	0.022263	0.043448	0.037997	0.023232	0.016736	0.021563	0.019896
0.041348	0.051698	0.039286	0.034824	0.016081	0.019679	0.019896	0.01934
0.023454	0.022051	0.039286	0.033355	0.01958	0.021222	0.024108	0.015926
0.024835	0.023584	0.032308	0.032263	0.01934	0.018561	0.019722	0.021222
0.042308	0.041111	0.020033	0.039961	0.015926	0.019183	0.022335	0.018561
0.020033	0.024824	0.023454	0.032051	0.018373	0.020316	0.024731	0.019183
0.041026	0.027539	0.024835	0.033584	0.023463	0.020738	0.02266	0.019183
0.039286	0.039286	0.043454	0.039286	0.01953	0.021628	0.01953	0.020316
0.039322	0.039322	0.039286	0.039936	0.021053	0.019786	0.021053	0.016736
0.034848	0.044848	0.034848	0.043448	0.01944	0.02043	0.01944	0.019679
0.043448	0.043448	0.044848	0.041348	0.016736	0.023232	0.016736	0.021222
0.041348	0.041348	0.020033	0.039961	0.019679	0.020081	0.019679	0.020316
0.01522	0.043454	0.022308	0.039936	0.011222	0.01958	0.021222	0.020738
0.024835	0.044835	0.044835	0.024835	0.018561	0.01934	0.018561	0.021628
0.022308	0.052308	0.020033	0.042308	0.019183	0.020926	0.019183	0.019786
0.043429	0.043429	0.020027	0.039286	0.020316	0.018373	0.020316	0.018373
0.041026	0.041026	0.041348	0.039322	0.020738	0.023463	0.020738	0.018373
0.039286	0.023961	0.039286	0.043454	0.017097	0.020097	0.021628	0.020316
0.039322	0.022455	0.039322	0.018974	0.016322	0.016322	0.019786	0.020738
0.015401	0.051221	0.044848	0.039322	0.016986	0.016986	0.02043	0.017097
0.023448	0.022263	0.043448	0.034848	0.021563	0.021563	0.021232	0.016322
0.041348	0.021698	0.041348	0.046448	0.019896	0.019896	0.021081	0.016986
0.043454	0.037051	0.043454	0.039936	0.024108	0.024108	0.01958	0.021628
0.018974	0.023584	0.044835	0.033355	0.019722	0.019722	0.01934	0.028097
0.032308	0.047511	0.062308	0.039286	0.022335	0.022335	0.021926	0.016322
0.043429	0.024824	0.043429	0.039961	0.014731	0.024731	0.018373	0.016986
0.041026	0.027539	0.041026	0.039936	0.02266	0.02266	0.021463	0.021563

Lampiran 5 Data Jitter

JITTER (msec)							
HTTP (Penerima)				RTP (Penerima)			
1	2	4	40	1	2	4	40
0.027997	0.000582	0.039823	8.3E-05	0.000976	0.000911	0.003963	0.001948
0.03499	0.009669	3E-05	9.7E-05	0.00174	0.01067	0.006252	0.003003
0.034491	4.9E-05	8.3E-05	0.039798	0.001998	0.002685	0.000989	0.001959
0.000135	0.039971	9.7E-05	4.7E-05	0.000776	0.01954	0.00145	0.00094
0.002312	0.039953	0.039798	0.041984	0.016774	0.000933	0.003903	0.000962
0.0077	0.039923	6.5E-05	0.039953	0.002309	0.001101	0.001948	0.005882
0.002958	0.039905	0.030073	0.000103	0.00094	0.010708	0.003003	0.000638
0.004816	7.8E-05	0.032642	0.001135	0.000962	0.001389	0.001959	0.003954
0.002356	4.7E-05	0.039798	0.002312	0.005882	0.002919	0.007812	0.000977
0.000247	0.039984	0.001081	0.0077	0.007812	0.000538	0.002467	0.002018
0.002621	0.039953	0.0378	0.039798	0.010232	0.003954	0.002007	0.000819
0.003228	0.000103	0.001089	0.039798	0.020033	0.000977	0.001558	0.009509
0.001437	0.000141	2.4E-05	0.037763	0.000898	0.002018	0.002713	0.001325
0.019135	0.003228	0.04	0.031152	0.019592	0.000819	0.014275	0.018339
0.0153	0.029913	0.039798	0.002849	0.021067	0.014275	0.007156	0.007702
0.003657	0.03993	0.004265	0.000575	0.01991	0.005378	0.000811	0.003963
0.02478	0.039958	0.039798	0.000233	0.010318	0.001911	0.001911	0.006252
0.02782	0.020003	0.039798	0.000288	0.009509	0.020819	0.001819	0.020819
0.029798	0.029023	0.037763	0.000467	0.001325	0.001911	0.000927	0.001911
0.003228	0.001598	0.031152	0.039958	0.018339	0.001819	0.007702	0.001819
0.004614	0.000785	0.002849	0.020003	0.016774	0.000538	0.003963	0.007812
0.002636	0.003228	0.029798	0.029023	0.002309	0.003954	0.006252	0.010232
0.001598	0.00086	0.000573	0.002621	0.00094	0.000977	0.007702	0.020033
0.000813	0.003159	0.029026	0.003228	0.000962	0.000933	0.001558	0.007812
0.003228	0.02615	0.000461	0.001437	0.005882	0.001101	0.003903	0.010232
0.001063	0.000575	0.000322	0.019135	0.007812	0.010708	0.007702	0.020819
0.000408	0.000233	0.000514	0.02782	0.010232	0.020819	0.001558	0.001911
0.000562	0.000288	0.000634	0.029798	0.020033	0.001911	0.003903	0.021067
0.003228	0.000467	0.008706	0.003228	0.00174	0.001819	0.001911	0.002713
0.013036	0.001771	0.00035	0.004614	0.001998	0.002018	0.001819	0.014275

Lampiran 6 Data Throughput WiFi

THROUGHPUT WIFI (Mbit/sec)				
320x240	512x208	640x360	850x480	1024x576
0.392	0.376	0.501	0.501	0.59
0.315	0.282	0.338	0.601	0.779
0.259	0.383	0.26	0.579	0.881
0.358	0.343	0.509	0.568	0.784
0.389	0.304	0.433	0.757	0.541
0.349	0.259	0.501	0.614	0.542
0.365	0.421	0.4	0.614	0.683
0.241	0.236	0.415	0.789	0.887
0.317	0.375	0.359	0.627	0.887
0.325	0.369	0.285	0.299	0.346
0.307	0.376	0.653	0.301	0.348
0.248	0.273	0.354	0.812	0.592
0.238	0.268	0.375	0.323	0.345
0.275	0.486	0.328	0.663	0.606
0.193	0.428	0.353	0.578	0.362
0.392	0.376	0.501	0.501	0.59
0.315	0.282	0.338	0.601	0.779
0.259	0.383	0.26	0.579	0.881
0.358	0.343	0.509	0.568	0.784
0.389	0.304	0.433	0.757	0.541
0.349	0.257	0.501	0.614	0.542
0.365	0.421	0.4	0.614	0.683
0.241	0.236	0.415	0.789	0.887
0.317	0.375	0.359	0.627	0.887
0.325	0.469	0.285	0.299	0.346
0.307	0.376	0.553	0.301	0.348
0.248	0.273	0.354	0.812	0.592
0.238	0.266	0.375	0.323	0.345
0.275	0.486	0.328	0.663	0.606
0.193	0.228	0.353	0.578	0.362

Lampiran 7 Data Delay WiFi

DELAY WIFI (msec)				
320x240	512x208	640x360	850x480	1024x576
0.021296	0.023534	0.021839	0.021426	0.018538
0.028102	0.038638	0.032297	0.017498	0.014046
0.035781	0.028458	0.042129	0.017716	0.012414
0.027838	0.031852	0.021481	0.019277	0.013947
0.027255	0.03582	0.024285	0.012746	0.019506
0.027574	0.043506	0.020739	0.017803	0.02023
0.025788	0.025893	0.025222	0.017788	0.016043
0.03634	0.046075	0.023927	0.012331	0.012359
0.030897	0.02908	0.029704	0.017405	0.012357
0.032556	0.064432	0.035878	0.036316	0.031656
0.034019	0.029019	0.018126	0.03637	0.031509
0.04192	0.039884	0.029285	0.013497	0.018498
0.042583	0.041892	0.028929	0.03392	0.03156
0.034469	0.058992	0.032617	0.016489	0.017989
0.051667	0.047766	0.029134	0.018874	0.030282
0.021296	0.023534	0.021839	0.021426	0.018538
0.028102	0.038638	0.032297	0.017498	0.014046
0.035781	0.028458	0.042129	0.017716	0.012414
0.027838	0.031852	0.021481	0.019277	0.013947
0.027255	0.03582	0.024285	0.012746	0.019506
0.027574	0.043506	0.020739	0.017803	0.02023
0.025788	0.025893	0.025222	0.017788	0.016043
0.03634	0.046075	0.023927	0.012331	0.012359
0.030897	0.02908	0.029704	0.017405	0.012357
0.032556	0.064432	0.035878	0.036316	0.031656
0.034019	0.029019	0.018126	0.03637	0.031509
0.04192	0.039884	0.029285	0.013497	0.018498
0.042583	0.041892	0.028929	0.03392	0.03156
0.034469	0.058992	0.032617	0.016489	0.017989
0.051667	0.047766	0.029134	0.018874	0.030282

Lampiran 8 Data Jitter WiFi

JITTER WIFI (msec)				
320x240	512x208	640x360	850x480	1024x576
0.001702	0.007932	0.000714	0.015446	0.0185
0.001699	0.018363	0.021934	0.018548	0.000488
0.014881	0.013827	0.028725	0.01136	0.002147
0.009917	0.024421	0.026691	0.010119	0.000725
0.000905	0.011227	0.015375	0.008457	0.000779
0.003292	0.009033	0.018967	0.017451	0.00063
0.00019	0.000148	0.00075	0.000607	0.008934
0.007116	0.03049	0.010318	0.029451	0.002071
0.00114	0.004409	0.008909	0.000692	0.027414
0.001674	0.019717	0.009532	0.001408	0.020631
0.000642	0.028142	0.001334	0.002362	0.00187
0.002591	0.03065	0.001018	0.02519	0.017349
0.000172	0.000686	0.019166	0.019376	0.010899
0.000274	0.01732	0.010085	0.011023	0.006569
0.039538	0.010787	0.018489	0.001746	0.000888
0.001343	0.024043	0.011952	0.001317	0.005256
0.013709	0.011484	0.036399	0.001451	0.000628
0.009593	0.024568	0.012855	0.000721	0.000902
0.015025	0.021401	0.014026	0.024554	0.005938
0.001767	0.007788	0.014073	0.023605	0.035095
0.006798	0.028387	0.006531	0.016794	0.022619
0.012913	0.006708	0.021745	0.032897	0.043581
0.008278	0.017965	0.022842	0.003418	0.003306
0.030955	0.022228	0.019164	0.004501	0.018497
0.034182	0.013043	0.004351	0.035631	0.036391
0.006014	0.011893	0.002836	0.024921	0.005107
0.031801	0.016852	0.025994	0.013805	0.008172
0.004198	0.042894	0.010598	0.004665	0.008255
0.007901	0.03205	0.022374	0.010426	0.033679
0.014385	0.020936	0.026217	0.005566	0.006253

Lampiran 9 Data Packet Loss WiFi

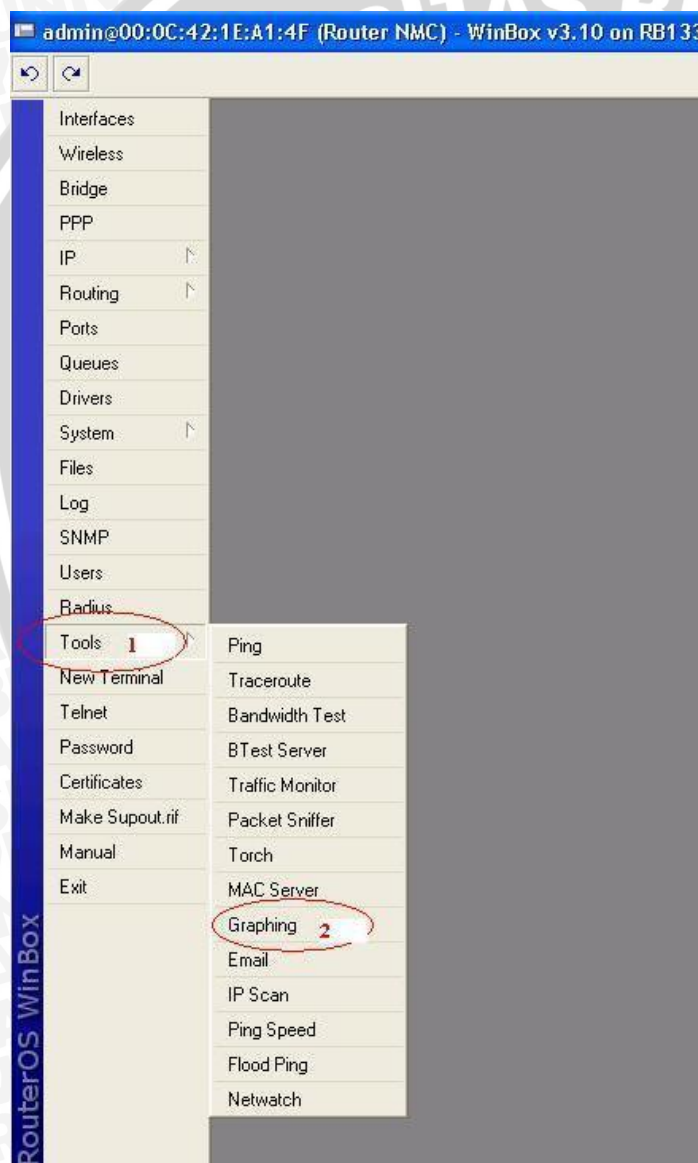
PACKET LOSS WIFI (%)				
320x240	512x208	640x360	850x480	1024x576
24.2	40.9	43.9	65.9	73.1
5.9	16.1	5.7	10	65.1
8.1	40.8	11.1	61	69
24.4	45.2	48.2	30	79.6
16.5	13.5	45.1	32	59.9
10	7.1	19	59	69.9
25.7	23.8	44.8	18	65.5
2.1	7	47.1	52.2	59.8
2.8	31.4	49.8	66.4	79.6
19.7	9	12	59	49.9
14.5	35.6	29.6	62.4	10
16.6	51.8	53.3	29.2	59.8
1.7	13.5	48.8	9.7	50
1.2	16.8	57.7	65.1	72.8
10.8	9.3	50	65.7	79.9
14.2	40.9	43.9	65.9	73.1
15.9	6.1	15.7	40	65.1
18.1	40.8	1.1	31	79
24.4	45.2	48.2	20	19.6
16.5	3.5	45.1	32	49.9
30	17.1	9	59	69.9
25.7	23.8	44.8	28	65.5
2.1	17	47.1	52.2	29.8
32.8	31.4	49.8	66.4	39.6
19.7	19	22	49	59.9
4.5	35.6	29.6	52.4	20
6.6	51.8	53.3	39.2	49.8
0.7	3.5	48.8	9.7	10
1.2	6.8	57.7	65.1	72.8
10.8	19.3	50	65.7	39.9

Lampiran 10 Konfigurasi Grafik PTIIK

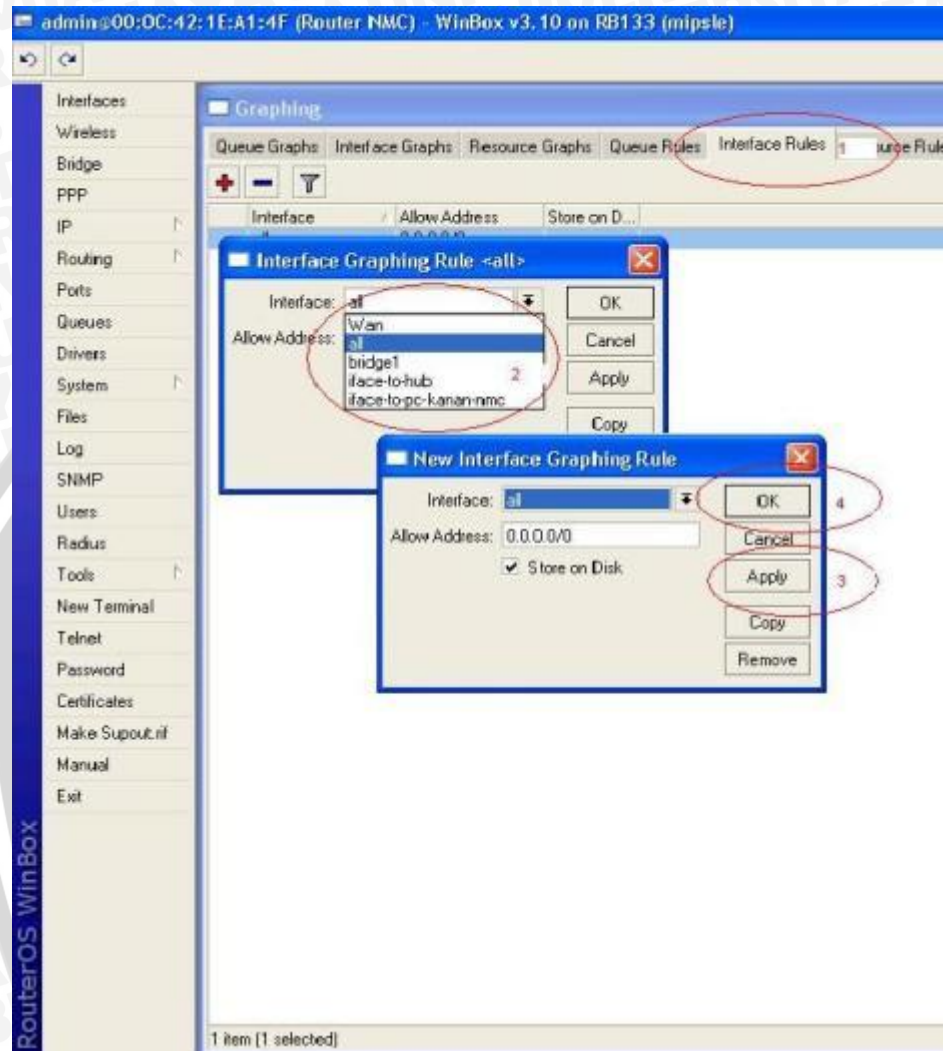
Monitoring terhadap lalu lintas network pada jaringan PTIIK tempat dilakukan pengujian diperlukan agar dapat diketahui saat dimana jam sibuk terjadi. Adapun langkahnya menggunakan console winbox sbb:

Login ke mikrotik sebagai akses full terlebih dahulu.

Tool > Graphing



Interface Rules > Pilih interface yang akan dibuatkan graph [untuk memudahkan pilih semuanya / all] > Apply > OK



Pembuatan selesai, untukcek bisa dibuka di browser ” <http://172.21.4.1/graphs/>”, pilih interface yang akan dimonitoring.

Lampiran 11 Filter Wireshark

- Filter pengujian yang menggunakan HTTP

```
ip.addr == 172.21.4.22 and http and frame.number > 5000
```

Nilai *frame number* menyesuaikan hasil penangkapan melalui *wireshark*

- Filter pengujian yang menggunakan RTP

```
ip.addr == 224.1.1.1 and udp and frame.number > 5000
```

Setelah proses filter selesai dilakukan, paket UDP di *decode* terlebih dahulu menjadi RTP. Pada tab *Analyze* pilih *Decode as* kemudian pilih RTP.

Nilai *frame number* sebesar 5000 pada filter diatas dapat berubah, menyesuaikan hasil penangkapan melalui *wireshark*.

