

## QUALITY OF SERVICE (QoS) TEKNOLOGI STREAMING UNTUK APLIKASI SURVEILLANCE

Abdusy Syarif, Afrius Setiawan, Achmad Kodar

Teknik Informatika, Universitas Mercu Buana – Jakarta

Email : [abdusyarif@mercubuana.ac.id](mailto:abdusyarif@mercubuana.ac.id), [afrius2008@yahoo.com](mailto:afrius2008@yahoo.com), [akodar@yahoo.com](mailto:akodar@yahoo.com)

### Abstraksi

Salah satu aplikasi dari teknologi streaming adalah sistem monitoring atau surveillance system. Dan terdapat beberapa faktor yang dapat mempengaruhi kualitas teknologi streaming, mulai dari perangkat keras sampai kepada perangkat lunak. Apalagi data yang digunakan adalah data multimedia seperti suara (audio) dan gambar bergerak (video). Belum lagi aspek komunikasi dan komputasi yang menyaratkan adanya kualitas atau jaminan layanan data multimedia.

Protokol RTP adalah salah satu protokol yang bertanggung jawab dalam melakukan proses pengiriman dan RTCP yang melakukan kontrol terhadap statistik pengiriman tersebut. Dalam penelitian ini akan diuji beberapa parameter dengan melakukan pengukuran kinerja atau Quality of Service (QoS) dari protokol RTP, RTCP, failed transmission, local collision, dan remote collision.

Hasil penelitian ini menunjukkan bahwa kualitas yang diberikan oleh protokol RTP dan RTCP baik pengujian dengan metode unicast maupun dengan multicast menunjukkan rata-rata prosentase keberhasilan diatas 95%.

**Keywords:** Streaming, Real-time Transport Protokol, Real Time Control Protokol

### 1. PENDAHULUAN

Kampus Universitas Mercu Buana sering terjadi beberapa permasalahan pada ruangan perkuliahan dan ruangan laboratorium, diantaranya penggunaan lokal/ruangan perkuliahan yang tidak optimal, fanalisme pada dinding ruangan kuliah, dan pada ruangan laboratorium sering terjadi kehilangan beberapa perangkat komputer. Semua itu seharusnya bisa dihindari dengan menggunakan alat monitoring setiap ruangan.

Belum adanya teknologi streaming yang diterapkan pada kampus Universitas Mercu Buana mendorong kami untuk menerapkan dan melakukan penelitian terhadap kualitas layanan teknologi streaming, yang bertujuan untuk membuktikan kelayakan teknologi streaming pada kampus Universitas Mercu Buana sebelum diterapkan.

Adapun tujuan penelitian ini sebagai berikut

1. Untuk menajagi beberapa Parameter QoS pada teknologi streaming,
2. Menerapkan dan membuktikan kelayakan teknologi streaming pada lingkungan Kampus Universitas Mercu Buana.

Ruang lingkup penelitian ini hanya mengukur kualitas layanan (*quality of service*) dari penerapan teknologi streaming menggunakan beberapa parameter yaitu : RTP dan RTCP, failed transmission, local collision, remote collision.

### 2. LITERATUR

Pada bagian ini kami akan memberikan seputar kualitas layanan teknologi streaming.

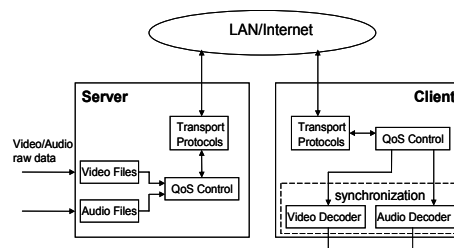
Streaming adalah teknologi transmisi pengiriman data, video atau audio secara *real time / pre-recorded* dari *sender* pada *receiver*. Ide dari

streaming adalah membagi data, *encoding*, mengirimkan melalui jaringan dan saat bagian data tiba pada *client* maka dilakukan *decoding* serta membaca data.

Ciri – ciri aplikasi Streaming : distribusi audio, video dan multimedia pada jaringan secara *real time* atau *on demand*, transfer media data digital oleh *server* dan diterima oleh *client* sebagai *real time stream* simultan, *client* tidak perlu menunggu keseluruhan data di *download* karena *server* mengirimkan data yang diperlukan setiap waktu tertentu. Sehingga memungkinkan *client* menjalankan file *content* seketika dengan periode *buffer* pendek [10]

### 3. PROTOKOL DATA STREAMING DAN JAVA MEDIA FRAMEWORK

Arsitektur streaming (lihat gambar 1) adalah aliran paket data yang berisi konten media tertentu yang dibangkitkan oleh *streaming media server*. Salah satu keuntungannya adalah tidak ada data yang disimpan permanen pada *client* setelah data yang diterima dimainkan. Sehingga, *user* tidak dapat menyimpan atau meneruskan file yang telah ia terima kepada user lain [3].

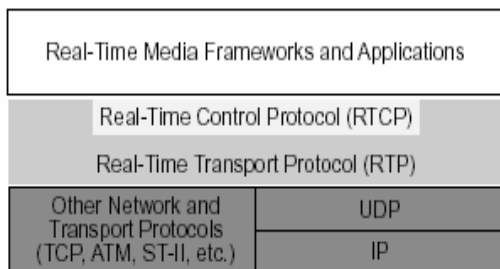


Gambar 1. Arsitektur dan komponen streaming

Hingga saat ini setidaknya terdapat tiga jenis format streaming yang banyak digunakan pada situs-situs Internet. Format tersebut adalah keluaran Real Media (.rm/.ra/.ram), Windows Media (.asf/.wmf/.asx) dan Quick Time (.mov) masing-masing format tersebut memiliki kekurangan dan kelebihan sendiri-sendiri

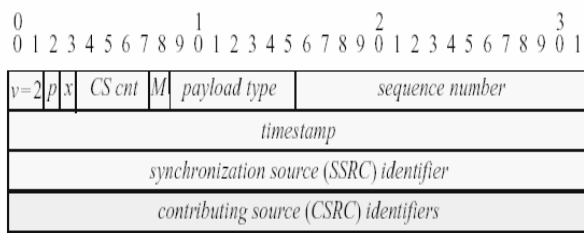
### 3.1. RTP (Real-time Transport Protocol)

RTP Adalah protokol yang dibuat untuk mengkompensasi jitter dan desequencing yang terjadi pada jaringan IP. RTP dapat digunakan untuk beberapa macam data stream yang realtime seperti data suara dan data video dalam layanan jaringan multicast atau unicast [2]. Seperti terlihat pada gambar 2 dan 3, arsitektur RTP berisi informasi tipe data yang di kirim, timestamps yang digunakan untuk pengaturan waktu suara percakapan terdengar seperti sebagaimana diucapkan, dan *sequence-number* yang digunakan untuk pengurutan paket data dan mendeteksi adanya paket yang hilang.



Gambar 2. Arsitektur RTP [6]

RTP tidak memiliki standar port TCP atau UDP ketika ingin berkomunikasi. Meskipun tidak terdapat standar yang diberikan. RTP umumnya dikonfigurasi menggunakan port antara 16834-32767. RTP dapat membawa banyak data dengan karakteristik yang *real time*, diantaranya audio dan video interaktif.



Gambar 3. Paket Header RTP [6]

### 3.2. RTCP (Real Time Control Protocol)

Merupakan suatu protocol yang biasanya digunakan bersama-sama dengan RTP. RTCP digunakan untuk mengirimkan paket control setiap terminal yang berpartisipasi pada percakapan yang digunakan sebagai informasi untuk kualitas transmisi pada jaringan [2]. Terdapat dua komponen penting pada paket RTCP, yang pertama adalah sender report yang berisikan informasi banyaknya

data yang dikirimkan, pengecekan timestamp pada header RTP dan memastikan bahwa datanya tepat dengan timestamp-nya. Elemen yang kedua adalah receiver report yang dikirimkan oleh penerima panggilan. Receiver report berisi informasi mengenai jumlah paket yang hilang selama sesi percakapan, menampilkan timestamp terakhir dan delay sejak pengiriman sender report yang terakhir.

Fungsi paling utama RTCP yaitu menyediakan *feedback* terhadap kualitas layanan (quality of services) yang disediakan oleh RTP yang merupakan peran dari RTP sebagai protokol transport. RTCP mengumpulkan statistik pada media komunikasi dan informasi seperti bytes send, packets sent, lost packets, jitter, feedback dan round trip delay.

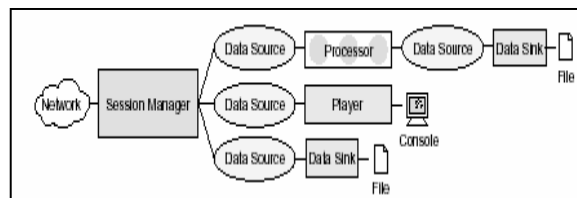
### 3.3. RTP/RTCP Pada Java Media Framework

Java Media Framework (JMF) merupakan salah satu *interface* aplikasi multimedia yang dapat bekerja baik pada sistem operasi Microsoft Windows maupun Linux sehingga dalam penerapannya tidak memerlukan penyesuaian yang begitu banyak untuk dapat dioperasikan pada sistem operasi tersebut. Salah satu bentuk multimedia yang dapat dioperasikan dengan JMF adalah *webcam*.

Beberapa fungsi JMF, yaitu :

- Dapat digunakan untuk berbagai file multimedia pada Java Applet atau aplikasi. Format yang mendukung antara lain .au, .avi, .midi, .mpeg, .qt dan .wav.
- Play media streaming dari internet
- Capture audio dan video dengan mikropon dan kamera video kemudian menyimpan data tersebut kedalam format yang mendukungnya.
- Mengirimkan audio dan video secara *realtime* ke Internet atau Intranet.
- Dapat digunakan untuk pemrograman penyiaran radio atau televisi secara langsung.

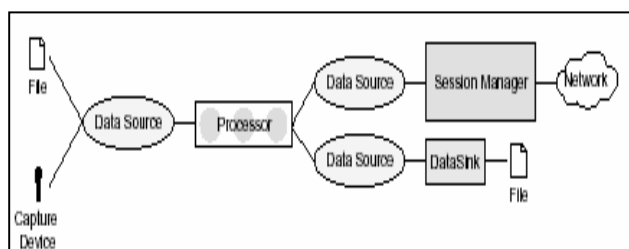
Mekanisme RTP Receive dapat dilihat pada gambar 4 berikut.



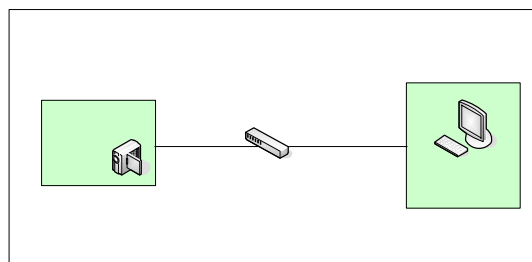
Gambar 4. RTP Reception [6]

Pada RTP Reception ini, RTP digunakan untuk mengirimkan hasil capture atau menyimpan media stream dalam network. Hasil paket stream dapat dimainkan secara lokal atau disimpan dalam file, ataupun keduanya (dimainkan secara lokal dan disimpan dalam file).

Mekanisme RTP Transmission dapat dilihat pada gambar 5.



Gambar 5. RTP Transmission [6]



Gambar 6. Skenario Unicast

Kita dapat melakukan pengiriman paket stream dengan menggunakan RTP session yang terpisah dari macam-macam tipe media (capture device).

**Keterangan :**

*Capture device :*

Dapat berupa mikropon, video capture (webcam), atau sumber lainnya yang dapat meneruskan atau menyediakan time-based data media.

*Processor :*

Tipe khusus dari Player JMF yang dapat menyediakan control bagaimana data media diproses sebelum dimainkan.

*Data Source :*

Objek yang akan diimplementasikan pada interface *Datasource* untuk di enkapsulasi pada media lokasi dan protokol serta software yang digunakan untuk mengirimkan media.

*Data Sink :*

Objek yang diimplementasikan pada interface *Datasink* untuk membaca isi media dari *Datasource* dan mengembalikan media ke tujuan.

*Session Manager :*

Pada JMF, Session Manager digunakan untuk mengkoordinasi sesion RTP. Sesion manager menjaga partisipasi sesion dan stream yang telah ditransmisikan.

**4. IMPLEMENTASI DAN PENGUJIAN**

Pada tahap pengujian, kami menggunakan 2 skenario ;

(1). Unicast

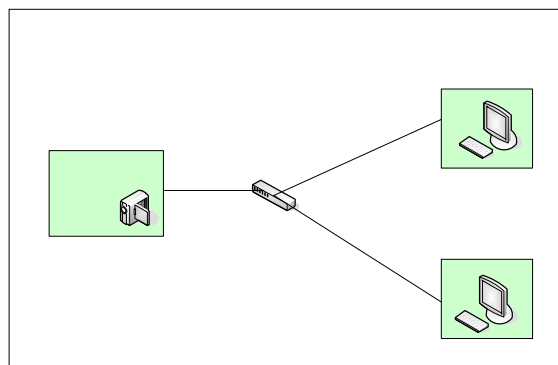
Seperti terlihat pada gambar 6, hanya terdapat 1 buah komputer yang bertindak sebagai RTP server dan 1 RTP client atau kegiatan monitoring Lab C-121 akan dilakukan oleh komputer yang berada pada ruang dosen.

*Penjelasan :*

Pada ruangan Lab C.121 akan dipasang sebuah webcam pada satu komputer di ruangan tersebut. Komputer tersebut akan bertindak sebagai RTP Server / RTP Transmission yang nantinya akan melakukan transmisi kepada client, sedangkan yang bertindak sebagai RTP Client / RTP Reception adalah komputer yang berada pada ruangan dosen. Perangkat lunak yang akan digunakan yaitu JMF (Java media framework) yang diinstal pada komputer di ruangan Lab C-121 serta pada ruangan dosen. JMF ini digunakan untuk melakukan Open Session RTP dari RTP client yang akan diarahkan pada RTP server serta RTP server yang membuat session tersebut.

(2) Multicast

Bersifat *One-to-many*, hanya pada client tertentu.



Gambar 7. Skenario Multicast

*Penjelasan :*

Hampir sama seperti skenario 1, hanya saja pada skenario 2 ini terdapat 1 komputer lagi yang akan melakukan monitoring, dimana akan melakukan pengujian dengan menggunakan network yang berbeda, serta sistem operasi yang berbeda pula, yaitu client 1 (Windows XP), dan client 2 (Linux Mandrake 10.2).

**4.1. Implementasi Skenario 1**

Setelah melakukan Instalasi Software JMF (Java media framework) pengujian siap dilakukan , pada PC1 (Sistem Operasi windows XP) akan melakukan RTP Trasnmission pada JMF, kemudian pada PC2 (Sistem Operasi windows XP) akan

melakukan Open RTP session dengan mengarah ke alamat RTP sesion yang dibuka dan alamat portnya.

### RTP Server

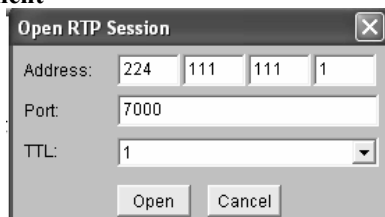


Gambar 8. RTP Transmission

Gambar 8 merupakan menu control device pada RTP server yang akan melakukan Transmission kepada RTP Client .

Gambar 9 merupakan menu untuk open RTP session, dimana address merupakan alamat dari RTP server yang melakukan Transmission, dan PORT 7000 port yang digunakan untuk melakukan komunikasi, dan TTL (time to live) diatur pada 1 detik (second).

### RTP Client



Gambar 9. Open RTP session

## 4.2. Implementasi Skenario 2

Pada pengujian skenario 2 (multicast) persiapan yang dilakukan sama seperti pada skenario 1 hanya saja pada pengujian skenario 2 melakukan monitoring dari RTP client dengan network yang berbeda. Dimana terdapat dua PC yang melakukan monitoring, PC yang pertama menggunakan sistem operasi windows XP profesional dan PC yang kedua menggunakan Sistem operasi Linux Mandrake 10.2.

## 4.3. Hasil Pengujian Skenario 1

Hasil Pengujian Skenario 1, melakukan pengukuran selama 10 menit dan 20 menit.

Tabel 1. Hasil Pengujian Skenario 1

Parameter	Capture 10 menit (paket)	Capture 20 menit (paket)
RTP Transmission	10.176	20.415
RTP Receive	9.747	20.098
RTCP Transmission	138	254

SR packet receive	128	253
-------------------	-----	-----

Tabel 2. Prosentasi Keberhasilan Skenario 1

	Waktu Capture	Keberhasilan (%)	Kegagalan (%)
Server → client (RTP)	10 menit	95,78 %	4,22 %
	20 menit	98,45 %	1,55%
Server → client (RCTP)	10 menit	92, 75 %	7,25 %
	20 menit	99,61	0,39 %

Pada tabel 1 diatas adalah hasil statistik pengiriman dari RTP Server selama 10 menit, Total paket RTP yang dikirim 10.176 paket, total Bytes yang dikirim 5.997.032 bytes, total paket RTCP yang dikirim 138, Local collision 0, remote collision 0, dan failed transmission 0. Sedangkan selama 20 menit diatas adalah hasil statistik pengiriman dari RTP Server selama 20 menit, Total paket RTP yang dikirim 20.415 paket, total Bytes yang dikirim 11836185 bytes, Total paket RTCP yang dikirim 254, Local collision 0, remote collision 0, dan failed transmission 0.

Dan merupakan hasil statistik penerimaan pada RTP Client selama 10 menit, total paket RTP yang diterima 9.749, total paket bytes 57.360.881 bytes, Bad RTP packet 0, , Local collision 0, remote collision 0, dan failed transmission 0, Paket RTCP yang diterima 247 paket, SR paket yang diterima 128 paket.

Tabel 3. Hasil Pengujian Skenario 2

	Parameter	5 menit (paket)	10 menit (paket)	15 menit (paket)
PC 1	RTP Transmission	5.324	10.789	15.845
	RTCP Transmission	65	123	191
PC 2	RTP Receive	5.109	10.290	15.379
	SR RTCP Receive	63	122	190
PC 3	RTP Receive	5.106	10.540	15.615
	SR RTCP Receive	65	122	179

Tabel 4. Prosentase Hasil Pengujian Skenario 2

	Waktu Capture	Keberhasilan (%)	Kegagalan (%)
Server → Client 1 (RTP)	5 menit	95,96	4,04
	10 menit	95,34	4,66
	15 menit	97,06	2,94
Server → Client 2 (RTP)	5 menit	95,91	4,09
	10 menit	97,69	2,31
	15 menit	98,55	1,45
Server → Client 1 (RTCP)	5 menit	96,92	3,08
	10 menit	99,19	0,81
	15 menit	99,45	0,55
Server → Client 2 (RTCP)	5 menit	100	0
	10 menit	99,19	0,81
	15 menit	93,72	6,28

Dan juga merupakan hasil statistik penerimaan pada RTP Client selama 20 menit, Total paket RTP yang diterima 20.098, Total paket bytes 11.815.819 bytes, Bad RTP packets 0, Local collision 0, remote collision 0, dan failed transmission 0, Paket RTCP yang diterima 493 paket, SR paket yang diterima 253 paket.

Total paket RTP yang diterima oleh client 1 5.109 paket, total bytes yang diterima 2.913.290

bytes, Total Paket RTCP yang diterima 326 paket, dan SR (sending report) yang diterima 63 paket serta terjadi 1 local collision.

Total paket yang diterima oleh client 1 selama 10 menit, total Paket RTP yang diterima 10.290 paket, total bytes 5.835.615 bytes, local collision 1, paket RTCP yang diterima 626, SR paket 122 paket.

Total paket yang diterima oleh RTP client 1 selama 15 menit, Total Paket RTP yang diterima 15.379, Total Bytes 8.717.599 bytes, total Paket RTCP yang diterima 939, paket SR 190, dan terjadi 1 local collision.

Total paket yang diterima oleh client 2 selama 5 menit, total paket RTP 5.806 paket, 3.178.402 bytes, Total paket RTCP 131 paket, Total SR 65 paket.

Total paket yang diterima oleh client 2 selama 10 menit, total paket RTP 10.540 paket, 5.949.380 bytes, Total paket RTCP 245 paket, Total Paket SR 122.

Total paket yang diterima oleh client 2 selama 15 menit, total paket RTP 15.615 paket, 8.832.821 bytes, Total paket RTCP 363 paket, Total Paket SR 179 paket.

## 5. ANALISIS HASIL PENGUJIAN

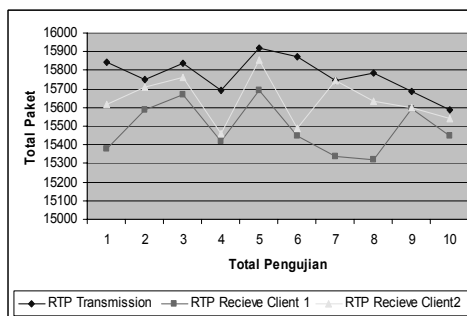
Berdasarkan data yang didapat selama pengujian maka kami menghitung rata-rata dan deviasi minimum dan maksimum dari masing-masing parameter dan membuat grafiknya.

Tabel 5. Rata-rata Pengujian Skenario 1

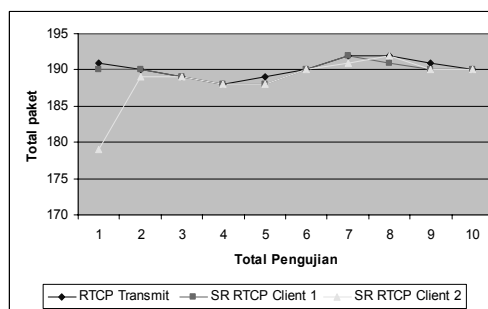
Node	Total Paket	Keberhasilan (%)	Kegagalan (%)	Deviasi Min	Deviasi Max
RTP Transmit	10.123	95,93	4,07	0,01	0,40
RTP Receive	9.711				
RTCP Receive	132				
SR Packet Receive	130	98,36	1,64	0,01	1,70

**Tabel 6.** Rata-rata Pengujian Skenario 2

Node	Total Paket	Keberhasilan (%)	Kegagalan (%)	Deviasi Min	Deviasi Max
RTP Transmit (Server)	10.520	98,31	1,69	0,01	0,20
RTP Receive (Client 1)	10.342				
RTP Receive (Client 2)	10.337	98,26	1,74	0,003	0,23
RTCP Transmission (server)	122	99,35	0,65	0,05	0,21
SR packet Receive (Client 1)	122				
SR packet Receive (Client 2)	122	99,34	0,66	0,05	0,21



**Gambar 12.** Grafik Jumlah Paket Data RTP selama 10 menit Pada Skenario 2



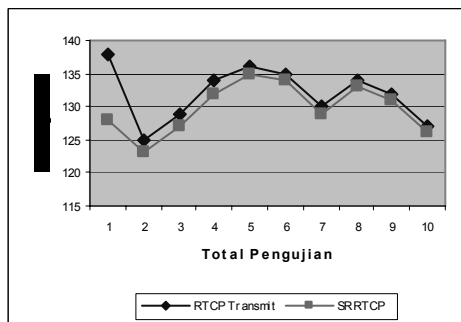
**Gambar 13.** Grafik Jumlah Paket Data RTP selama 20 menit Pada Skenario 2

Terlihat dari hasil paket yang dikirim dan paket yang diterima terdapat beberapa hasil data yang diperoleh antara RTP transmission maupun RTP receive tidak berbeda jauh gambar yang dimonitoring dari RTP client pun sama hasilnya dengan RTP server, monitoring berjalan dengan baik pada pengujian unicast ini. Failed transmission, local collision, remote collision tidak ada, berarti hasil pengiriman dapat diterima oleh RTP client dengan baik.

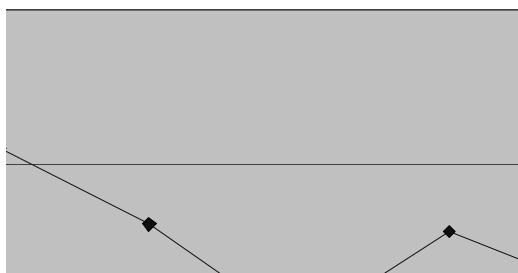
Pengujian dengan unicast dapat ditarik kesimpulan bahwa proses streaming berjalan dengan lancar, maksudnya tanpa ada paket-paket data yang gagal dikirim dan perbedaan antara paket data yang dikirimkan tidak jauh berbeda dengan paket data yang diterima oleh RTP Client.

Pengujian dengan skenario 2 (multicast) ini sangat mempengaruhi hasil dari streaming. Karena terdapat lebih dari 1 client sekaligus yang mengakses RTP session, tetapi pengujian disini tidak terlalu berpengaruh karena hanya terdapat 2 client saja yang mengakses atau melakukan monitoring. Apabila terdapat banyak client yang melakukan monitoring, maka hasil yang diperoleh akan lebih lambat daripada pengujian dengan hanya 2 client.

Dilihat dari perangkat lunak java media framework (jmf) dapat mendukung platform yang berbeda, maksudnya pada pengujian skenario 2 dari sisi RTP server sistem operasinya Windows XP sedangkan RTP client yang melakukan monitoring sistem operasi berbasis Linux Mandrake 10.2. disini



**Gambar 10.** Grafik Jumlah Paket Data RTP selama 10 menit Pada Skenario 1



**Gambar 11.** Grafik Jumlah Paket Data RTP selama 20 menit Pada Skenario 1

membuktikan bahwa Java Media Framework dapat digunakan oleh platform yang berbeda.

## 6. KESIMPULAN

Kesimpulan yang dapat ditarik dari penelitian ini adalah :

- 1) Dilihat dari perbandingan antara total paket yang dikirim dan total paket yang diterima secara keseluruhan (rasio keberhasilan antara 95,93% dan 99,35% ini dapat membuktikan teknologi streaming dapat diterapkan di lingkungan kampus Universitas Mercu Buana.
- 2) Parameter QOS yang diukur pada pengujian skenario 1, Failed Transmission, Local collision, remote collision tidak ada. Dengan demikian proses streaming dapat berjalan dengan lancar, sedangkan pada pengujian skenario 2 terjadi 1 local collision pada client 2 (Linux) yang menyebabkan hasil streaming terputus-putus pada client 2, namun tidak berjalan lama, hasilnya menjadi lebih baik kembali (tidak putus-putus).
- 3) Failed Transmission, Local collision, dan remote collision sangat mempengaruhi hasil streaming, terutama ketika terjadi perubahan gambar yang sangat cepat.

Adapun saran yang dapat diberikan adalah:

- 1) Dalam dunia streaming, disarankan menggunakan komputer dengan spesifikasi tinggi (kecepatan prosesor, VGA card, memory) karena akan mempengaruhi hasil dari streaming tersebut.
- 2) Untuk penelitian kedepan disarankan untuk menguji streaming audio dari JMF (Java Media Framework), serta efek terhadap RGB, resolusi, bit rate, frame rate, pada konfigurasi video device.
- 3) Menguji kompatibilitas JMF (java media framework) dengan tools lain yang mendukung proses streaming seperti VLC player.

## DAFTAR PUSTAKA

- [1] Drew Heywood. *Konsep dan Penerapan Microsoft TCP/IP*. Andi Yogyakarta 1996
- [2] Stefan A. Goor, *Experimental Performance Analysis of RTP-Based Approaches for Lowbitrate Transmission of MPEG-4 Video Content*. 2005
- [3] <http://www.cs.columbia.edu/~hgs/rtp/> (diakses 3 Desember 2007)
- [4] <http://www.id.wikipedia> (diakses 12 Desember 2007)
- [5] <http://www.BMRC.Berkeley.EDU/~larry> (diakses 17 Desember 2007)

- [6] <http://java.sun.com/products/java-media/jmf2.1.1/guide/> (diakses 2 Desember 2007)
- [7] <http://www.ilmukomputer.com> (diakses 12 Desember 2007)
- [8] <http://www.elektro.undip.ac.id/transmisi/jun05/agungbpjun05.pdf> (diakses 17 Desember 2007)
- [9] <http://www.ietf.org/internet-drafts/draft-ietfavt-rtp-new-04.txt> (diakses 17 Desember 2007)
- [10] <http://www.caip.rutgers.edu/TASSL/Thesis/RanginiThesis.pdf> (diakses 25 Januari 2008)

