



ANALISIS PERBANDINGAN *REAL TIME STREAMING PROTOCOL* (RTSP) DAN *HYPertext TRANSFER PROTOCOL* (HTTP) PADA LAYANAN *LIVE VIDEO STREAMING*

Ady Nopaldi Rombe^{*1}, LM. Fid Aksara², La Surimi³

^{*1,2,3}Jurusan Teknik Informatika, Fakultas Teknik Universitas Halu Oleo, Kendari
e-mail : ^{*1}adynopaldi@gmail.com, ²fidaksara@uho.ac.id, ³lasurimi@uho.ac.id

Abstrak

Media *streaming* merupakan teknologi yang memungkinkan distribusi data audio, video, dan multimedia secara *real time* melalui internet. Salah satu kategori dari media *streaming* yaitu video *streaming*. Video *streaming* merupakan pengiriman media digital berupa video, suara dan data agar dapat diterima secara terus-menerus. *Real Time Streaming Protocol* (RTSP) merupakan protokol yang ada pada level aplikasi yang berfungsi untuk mengontrol mengirim data secara *real time*. RTSP menggunakan dan menyediakan kerangka kerja yang *extensible* untuk mengontrol pengiriman data *on-demand* seperti audio dan video.

HTTP (*Hypertext Transfer Protocol*) adalah sebuah protokol untuk meminta dan menjawab antara *client* dan *server*. Sebuah *client* HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke *port* yang tertentu ditempat yang jauh (biasanya *port* 80). Berdasarkan hasil yang diperoleh dari semua percobaan maka diketahui protokol HTTP tidak direkomendasikan sebagai media *streaming* melalui internet karena berdasarkan analisis pada semua parameter yang diujikan memiliki *Delay*, *Packet Loss* yang sangat tinggi dan *Throughput*-nya yang kecil, sedangkan RTSP dari hasil analisis pada semua yang diujikan memiliki nilai QoS yang terendah.

Kata Kunci—Video *Live Streaming*, *Real Time Streaming Protocol* (RTSP), *Hypertext Transfer Protocol* (HTTP)

Abstract

Streaming media is a technology that allows the distribution of audio, video and multimedia data in real time via the internet. One category of streaming media is video streaming. Video streaming is sending digital media in the form of video, sound, and data so that it can be received continuously. Real Time Streaming Protocol (RTSP) is a protocol at the application level that functions to control sending data in real time. RTSP uses and provides an extensible framework to control on-demand data transmissions such as audio and video.

HTTP (Hypertext Transfer Protocol) is a protocol for requesting and answering between client and server. An HTTP client such as a web browser usually starts a request by making a TCP/IP connection to a specific port in a remote place (usually port 80). Based on the results obtained from all experiments, it is known that the HTTP protocol is not recommended as a streaming media via the internet because based on the analysis of all tested parameters it has a very high Delay, Packet Loss, and small Throughput, whereas the RTSP from the results of the analysis on all that has been tested has the lowest QoS value.

Keywords—Video *Live Streaming*, *Real Time Streaming Protocol* (RTSP), *Hypertext Transfer Protocol* (HTTP)



1. PENDAHULUAN

Perkembangan teknologi akan kebutuhan layanan internet semakin meningkat. Semakin hari pihak penyedia jasa jaringan internet berusaha untuk mengembangkan layanan layanan yang diberikan kepada konsumen. Sebagai contoh sekarang telah ada layanan seperti video *streaming* yang akan membutuhkan ukuran data yang semakin besar membuat infrastruktur dan layanan yang tersedia semakin terbatas. Dengan kondisi tersebut, tentunya pihak konsumen pun menginginkan performansi *Quality of Service* (QoS) yang sesuai dengan keinginan saat mengakses layanan-layanan tersebut terutama video *streaming*. Karena video *streaming* merupakan sebuah layanan yang bersifat *real time* dengan penggunaan *bandwidth* yang lebar untuk mendapatkan performansi QoS yang baik [1].

Real Time Streaming Protocol (RTSP) merupakan protokol yang ada pada level aplikasi yang berfungsi untuk mengontrol mengirim data secara *real time*. RTSP menggunakan dan menyediakan kerangka kerja yang *extensible* untuk mengontrol pengiriman data *on-demand* seperti audio dan video. Data yang berasal dari pengirim dapat mencakup *live* data dan data yang disimpan. Protokol ini berfungsi untuk mengendalikan beberapa pengiriman data seperti *User Datagram Protokol* (UDP), *Multicast UDP* dan *Transmission Control Protokol* TCP [2].

HTTP (*Hypertext Transfer Protocol*) adalah sebuah protokol untuk meminta dan menjawab antara *client* dan *server*. Sebuah *client* HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke *port* yang tertentu ditempat yang jauh (biasanya *port* 80). Sebuah *server* HTTP yang mendengarkan di *port* tersebut menuggu *client* mengirim kode permintaan yang akan meminta halaman yang sudah ditentukan [3].

Media *streaming* merupakan teknologi yang memungkinkan distribusi data audio, video dan multimedia secara *real time* melalui internet. Salah satu kategori dari media *streaming* yaitu video *streaming*. Video *streaming* merupakan pengiriman media digital berupa video, suara dan data agar dapat diterima secara terus-menerus. Menurut Lippens, diawal tahun 2000 terjadi kenaikan

drastis pada *bandwidth* jaringan internet. Bersamaan dengan pengompresian media yang lebih baik dan sistem komputer pribadi yang lebih kuat, pengiriman media menjadi mungkin terjadi. Video *streaming* dirancang agar pengguna bisa melakukan siaran langsung dari suatu perangkat.

Berdasarkan latar belakang yang telah diuraikan, maka Penulis mengambil topik penelitian dengan judul "Analisis Perbandingan *Real Time Streaming Protocol* (RTSP) dan *Hypertext Transfer Protocol* (HTTP) pada Layanan *Live Video Streaming*".

2. METODE PENELITIAN

2.1 *Real Time Streaming Protocol* (RTSP)

RTSP merupakan sebuah protokol level aplikasi untuk kontrol atas pengiriman data dengan sifat *real time*. RTSP menyediakan kerangka *extensible* untuk mengaktifkan kendali pada pengiriman data *real time*, seperti *audio* dan *video*, dengan menggunakan *Transmission Control Protokol* (TCP) atau *User Data Protokol* (UDP) [4].

RTSP sangat banyak digunakan oleh industri pengembang teknologi *streaming* media, hingga saat ini untuk media *player* pada *handphone* dan *smartphone* telah terintegrasi dengan protokol RTSP. Media *player* pada *handphone* dan *smartphone* yang terintegrasi dengan protokol RTSP seperti pada *vendor handphone* dan *smartphone*, baik berbasis Java J2ME, Symbian maupun Android yang telah bergabung dengan *project Helix Player DNA* yang merupakan produk yang dikembangkan oleh *Real Networks* telah mampu memutar *streaming* media dengan format RA, RV, RM, H264/AAC, H263/AMR, MPEG-4/AAC dan MPEG-4/AMR secara langsung. Protokol yang dikembangkan pertama kali oleh *Real Networks* adalah protokol standard untuk pendistribusian *streaming* media. *Port default* yang digunakan adalah 544 [5].

2.2 *Hypertext Transfer Protocol* (HTTP)

Hypertext Transfer Protocol atau yang lebih dikenal dengan singkatan HTTP adalah suatu protokol yang digunakan untuk akses antara *client* dan *server* pada jaringan komputer, khususnya internet, atau dapat diartikan juga sebagai sebuah protokol pada internet yang digunakan sebagai metode dalam

mentransfer halaman *World Wide Web* (www). Dalam menerima data dari *server* umumnya *client* menggunakan *web browser*.

HTTP ialah suatu protokol yang umumnya digunakan www untuk mengatur komunikasi *client* dan *server*. Jadi *client* mengirim permintaan kepada *server*, lalu *server* akan merespon dan mengirim data yang diminta oleh *client* dan menampilkannya [6].

2.3 Video Streaming

Video *streaming* adalah perangkat yang berfungsi sebagai penerima gambar dan suara. Sedangkan *streaming* adalah sebuah teknologi untuk memainkan *file* video atau audio secara langsung ataupun *recorder* dari sebuah mesin *server* (*web server*). Video *streaming* adalah salah satu cara untuk mengetahui informasi atau berita secara audio atau visual dari *server* secara langsung ketika terdapat permintaan *client*. Hal ini berbeda seperti menjalankan *file* yang telah selesai di-*download* dari komputer.

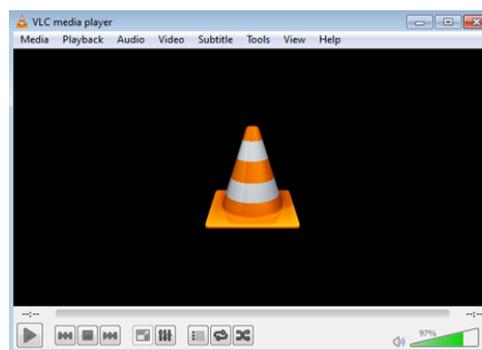
Teknologi *streaming* ini memungkinkan untuk menonton secara langsung dari komputer, tanpa perlu men-*download*, adapun kendala yang dihadapi dalam video *streaming* salah satunya adalah lambatnya kecepatan waktu yang dibutuhkan untuk video *buffer* yang lebih banyak membutuhkan waktu dibanding menonton *file* itu sendiri [7].

2.4 VLC Player

VLC *Player* adalah pemutar multimedia *powerfull* yang kaya fitur dan dapat memutar hampir semua jenis format *file* audio maupun video. Aplikasi VLC *Player* ini juga salah satu aplikasi multimedia favorit pengguna Linux karena kemampuannya yang dapat memutar hampir semua jenis *file* multimedia. Sejarah VLC *Player* ini berawal dari proyek akademik untuk menghadirkan *player* berbasis *client-server* yang mulai dikembangkan pada tahun 1996 oleh siswa di École Centrale Paris, Perancis.

Sejak tahun 2001, VLC telah menjadi proyek *open source* di bawah organisasi nirlaba videoLAN yang rilis di bawah lisensi GPL dan tersedia di hampir semua *platform*. Aplikasi ini memiliki kemampuan untuk *streaming* konten multimedia terenkripsi ke HDTV. Selain itu, aplikasi ini juga mampu mengambil *screenshot* dari video yang sedang diputar. VLC mendukung hampir semua

format *file* multimedia populer. Gambar 1 menunjukkan tampilan VLC *Player* [8].



Gambar 1 VLC *Player*

3. HASIL DAN PEMBAHASAN

Throughput adalah parameter yang menunjukkan jumlah *byte* rata-rata data yang dikirim dari satu *node* ke *node* lainnya persatuan detik. Pengujian *Throughput* dilakukan dengan melakukan *capture* terhadap paket TCP yang dikirim dari *server* ke *client* selama proses *streaming* berlangsung.

Dari hasil perhitungan *Throughput* selama proses *live video streaming* pada *client* 1 RTSP dan HTTP maka didapatkan data seperti pada Tabel 1.

Tabel 1 Hasil Perbandingan *Throughput* Video *Live Streaming* antara RTSP dan HTTP pada Percobaan 1.

<i>Client</i>	RTSP	HTTP
1	0.222 MBit/sec	0.199 MBit/sec
Rata-Rata	0.222 Mbit/sec	0.199 Mbit/sec

Dari hasil perhitungan *Throughput* selama proses *live video streaming* pada *client* 1, *client* 2 RTSP dan HTTP maka didapatkan data seperti pada Tabel 2.

Tabel 2 Hasil Perbandingan *Throughput* Video *Live Streaming* antara RTSP Dan HTTP pada Percobaan 2

<i>Client</i>	RTSP	HTTP
1	0.222 Mbit/sec	0.200 Mbit/sec
2	0.222 Mbit/sec	0.201 Mbit/sec
Rata-Rata	0.222 Mbit/sec	0.200 Mbit/sec

Dari hasil perhitungan *Delay* selama proses *live video streaming* pada *client* 1 RTSP dan HTTP maka didapatkan data seperti pada Tabel 3.

Tabel 3 Hasil Perbandingan *Delay Video Live Streaming* antara RTSP dan HTTP pada Percobaan 1

Client	RTSP	HTTP
1	15.3 millisecond	40.9 millisecond
Rata-Rata	15.3 millisecond	40.9 millisecond

Dari hasil perhitungan *Packet Loss* selama proses *live video streaming* pada *client* 1 RTSP dan HTTP maka didapatkan data seperti pada Tabel 4-6.

Tabel 4 Hasil Perbandingan *Delay Video Live Streaming* antara RTSP dan HTTP pada Percobaan 2

Client	RTSP	HTTP
1	15.3 millisecond	39.6 millisecond
2	15.3 millisecond	51.8 millisecond
Rata-Rata	15.3 millisecond	45.7 millisecond

Tabel 5 Hasil Perbandingan *Packet Loss Video Live Streaming* antara RTSP dan HTTP pada Percobaan 1

Client	RTSP	HTTP
1	4.76 %	43.73 %
Rata-Rata	4.76 %	43.73 %

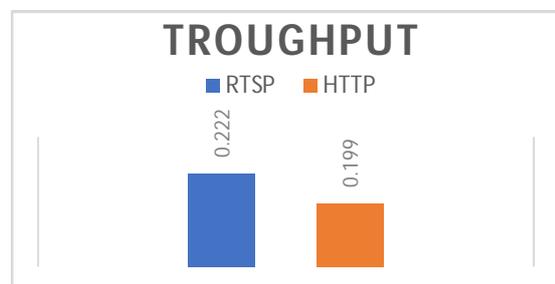
Tabel 6 Hasil Perbandingan *Packet Loss Video Live Streaming* antara RTSP dan HTTP pada Percobaan 2

Client	RTSP	HTTP
1	4.80 %	35.76 %
2	4.48 %	32.17 %
Rata-Rata	4.8 %	33.96 %

Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Throughput* pada saat *streaming* video, RTSP memiliki nilai yaitu 0,222 Mbit/sec. Dari analisis data statistiknya proses *live video streaming* RTSP memiliki banyak *Throughput*, karena RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming media*. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Throughput* pada *live video streaming* yaitu 0,199 Mbit/sec. Dari analisis data statistiknya proses *live video streaming*

HTTP memiliki sedikit *Throughput*, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*.

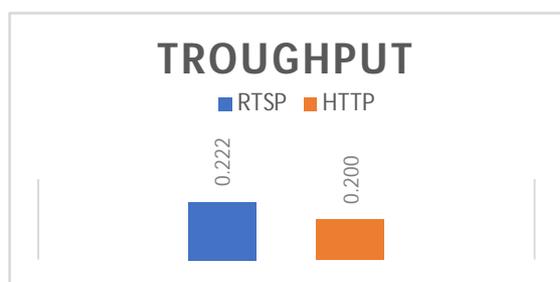
HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan, HTTP dijalankan pada UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket, sehingga RTSP memiliki *Throughput* paling tinggi dibandingkan HTTP. Gambar 2 menunjukkan grafik *Throughput* perbandingan RTSP dan HTTP pada Percobaan 1.



Gambar 2 Grafik *Throughput* Perbandingan RTSP dan HTTP pada Percobaan 1

Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Throughput* pada saat *streaming* video, RTSP memiliki nilai yaitu 0,222 Mbit/sec. Dari analisis data statistiknya proses *live video streaming* RTSP memiliki banyak *Throughput*, karena RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming media*. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Throughput* pada *live video streaming* yaitu 0,201 Mbit/sec. Dari analisis data statistiknya proses *live video streaming* HTTP memiliki sedikit *Throughput*, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*. HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan

sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan. HTTP dijalankan pada UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket, sehingga RTSP memiliki *Throughput* paling tinggi dibandingkan HTTP. Grafik *Throughput* perbandingan RTSP dan HTTP pada Percobaan 2 ditunjukkan oleh Gambar 3.

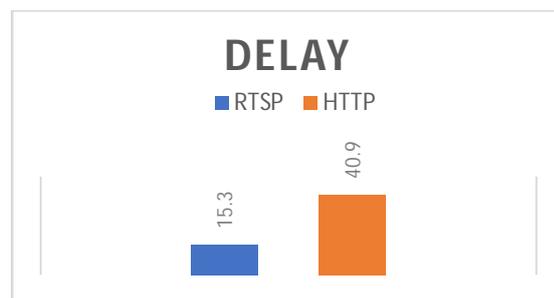


Gambar 3 Grafik *Throughput* Perbandingan RTSP dan HTTP pada Percobaan 2

Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Delay* pada saat *streaming* video, RTSP memiliki nilai yaitu 15,3 *millisecond*. Dari analisis data statistiknya proses *live video streaming* RTSP memiliki *Delay* kecil, karena RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming media*. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Delay* pada *live video streaming* yaitu 40,9 *millisecond*. Dari analisis data statistiknya proses *live video streaming* HTTP memiliki *Delay* lebih besar, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*.

HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan, HTTP dijalankan di atas UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket, sehingga RTSP memiliki *Delay* paling kecil dibandingkan HTTP. Gambar 4 menunjukkan

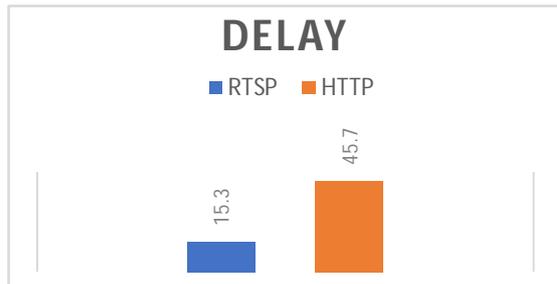
grafik *Delay* perbandingan RTSP dan HTTP pada Percobaan 1.



Gambar 4 Grafik *Delay* Perbandingan RTSP dan HTTP pada Percobaan 1

Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Delay* pada saat *streaming* video, RTSP memiliki nilai yaitu 15,3 *millisecond*. Dari analisis data statistiknya proses *live video streaming* RTSP memiliki *Delay* kecil, karena RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming media*. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Delay* pada *live video streaming* yaitu 45,7 *millisecond*. Dari analisis data statistiknya proses *live video streaming* HTTP memiliki *Delay* lebih besar, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*.

HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan, HTTP dijalankan pada UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket, sehingga RTSP memiliki *Delay* paling kecil dibandingkan HTTP. Gambar 5 menunjukkan grafik *Delay* perbandingan RTSP dan HTTP pada Percobaan 2.



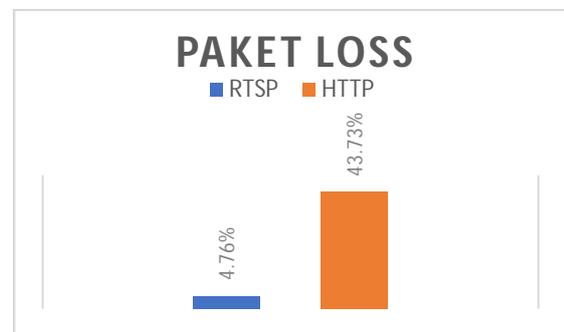
Gambar 5 Grafik Delay Perbandingan RTSP dan HTTP pada Percobaan 2

Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Packet Loss* pada saat *streaming* video, RTSP memiliki nilai yaitu 4,76%. Dari analisis data statistiknya proses *live video streaming* RTSP mempunyai *Packet Loss* paling kecil, karena RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming media*. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Packet Loss* pada *live video streaming* yaitu 43,73 %. Dari analisis data statistiknya proses *live video streaming* HTTP mempunyai *Packet Loss* sangat tinggi, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*.

HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan, HTTP dijalankan pada UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket, sehingga RTSP memiliki *Packet Loss* paling kecil dibandingkan HTTP. Gambar 6 menunjukkan grafik *Packet Loss* perbandingan RTSP dan HTTP pada Percobaan 1.

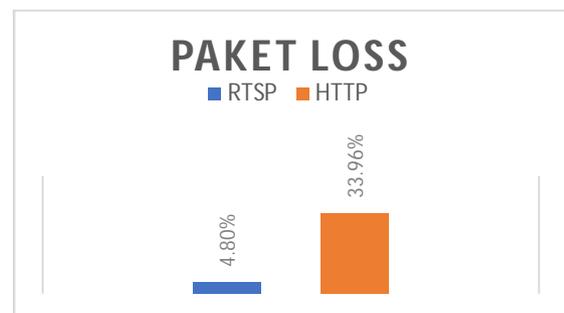
Berdasarkan hasil yang diperoleh dari Percobaan 1 menggunakan RTSP dan HTTP, maka diketahui nilai rata-rata *Packet Loss* pada saat *streaming* video, RTSP memiliki nilai yaitu 4,80 %. Dari analisis data statistiknya proses *live video streaming* RTSP mempunyai *Packet Loss* paling kecil, karena

RTSP merupakan protokol level aplikasi untuk kendali pengiriman data yang memiliki karakteristik *real time* seperti *streaming* media. RTSP bukan protokol pengiriman data yang aktual berupa *streaming* audio maupun video. RTSP sering disebut sebagai *remote control* jaringan untuk *streaming server*. RTSP dipakai atas protokol TCP untuk mengendalikan aliran data, sedangkan HTTP memiliki nilai *Packet Loss* pada *live video streaming* yaitu 33,96 %. Dari analisis data statistiknya proses *live video streaming* HTTP mempunyai *Packet Loss* sangat tinggi, karena HTTP dirancang bukan untuk *streaming* melainkan untuk menangani pengiriman data secara aktual yang berupa gambar atau *text*.



Gambar 6 Grafik *Packet Loss* Perbandingan RTSP dan HTTP pada Percobaan 1

HTTP merupakan sebuah *layer* aplikasi protokol yang dirancang dalam kerangka internet *protocol suite* yang membutuhkan sebuah lapisan dari *transport* yang dapat diandalkan jadi kualitas *streaming* tergantung koneksi jaringan, HTTP dijalankan pada UDP untuk *transport* data dan tidak ada jaminan dalam pengiriman paket. Sehingga RTSP memiliki *Packet Loss* paling kecil dibandingkan HTTP. Gambar 7 menunjukkan grafik *Packet Loss* perbandingan RTSP dan HTTP pada Percobaan 2.



Gambar 7 Grafik *Packet Loss* Perbandingan RTSP dan HTTP pada Percobaan 2

4. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. *Throughput* pada HTTP pada *live video streaming* melalui 2 kali percobaan dengan *client 1, client 2*, terjadi perbedaan nilai *Throughput* yang sangat besar, dan dimana RTSP pada *live video streaming*, memiliki nilai *Throughput* sangat besar dibandingkan dengan HTTP. Berdasarkan hasil analisis menunjukkan bahwa *Throughput* pada HTTP memiliki *Throughput* yang sedikit, sedangkan pada RTSP memiliki *Throughput* yang sangat tinggi.
2. *Delay* pada HTTP pada *live video streaming* melalui 5 kali percobaan dengan *client 1, client 2*, terjadi perbedaan nilai *Delay* yang sangat besar, dan dimana RTSP pada *live video streaming*, memiliki nilai *Delay* sangat kecil dibandingkan dengan HTTP. Berdasarkan hasil analisis menunjukkan bahwa *Delay* pada HTTP memiliki *Delay* yang sangat tinggi, sedangkan pada RTSP memiliki *Delay* sangat kecil.
3. *Packet Loss* pada HTTP pada *live video streaming* melalui 5 kali percobaan dengan *client 1, client 2*, terjadi perbedaan nilai *Packet Loss* yang sangat besar, dan dimana RTSP pada *live video streaming*, memiliki nilai *Packet Loss* sangat kecil dibandingkan dengan HTTP. Berdasarkan hasil analisis menunjukkan bahwa *Packet Loss* pada HTTP memiliki *Packet Loss* sangat tinggi sedangkan RTSP memiliki *Packet Loss* yang sangat kecil.
4. Berdasarkan hasil yang diperoleh dari semua percobaan maka diketahui protokol HTTP tidak direkomendasikan sebagai media *streaming* melalui internet karena berdasarkan analisis pada semua parameter yang diujikan memiliki *Delay, Packet Loss* yang sangat tinggi dan *Throughput*-nya yang kecil, sedangkan RTSP dari hasil analisis pada semua yang diujikan memiliki nilai QoS yang terendah.

5. SARAN

Adapun saran dalam penelitian ini yaitu diharapkan internet *protocol* yang digunakan

pada penelitian ini dapat ditingkatkan lagi ke internet *protocol* versi barunya. Diharapkan dapat dikembangkan dengan menambahkan parameter QoS yang lain.

DAFTAR PUSTAKA

- [1] D. H. Janius, "Analisis QoS Video Streaming pada Jaringan Wireless Menggunakan Metode HTB (Hierarchical Token Bucket)," Universitas Islam Negeri Sultan Syarif Kasim Riau Pekanbaru, 2013.
- [2] I. Mashari, Y. Purwanto, and I. D. Irawati, "Analisis Performansi Protokol Routing OSPF dan ISIS pada Layanan Video Streaming," Universitas Telkom, 2010.
- [3] A. Nurhayati and M. E. Sihaloho, "Simulasi Perbandingan Protokol Routing OSPF dan ISIS Menggunakan GNS 3," *J. ICT*, Vol. 5, No. 8, pp. 35–39, 2014.
- [4] A. A. Abror, M. Z. S. Hadi, and I. Winarno, "Rancang Bangun dan Analisa QoS Audio dan Video Streaming pada Jaringan MPLS VPN," 2010. [Online]. Available: <http://repo.pens.ac.id/374/1/1113.pdf>. [Accessed: 04-May-2018].
- [5] A. B. R. Kumar, L. C. Reddy, P. S. Hiremath, and Naresh. S.S, "RTSP Audio and Video Streaming for QoS in Wireless Mobile Devices," *J. Comput. Sci.*, Vol. 8, No. 1, pp. 96–101, 2008.
- [6] M. Y. Hariyawan, M. Susantok, and R. Tampubolon, "Study Analisis QoS pada Jaringan Multimedia MPLS," *SNTIKI 3*, pp. 277–283, 2011.
- [7] N. Baldo, U. Horn, M. Kampmann and F. Hartung, "RTCP Feedback based Transmission Rate Control for 3G Wireless Multimedia Streaming," *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, Vol. 3, pp. 1817-1821, 2004.
- [8] A. R. Mukti, Y. N. Kunang, and Suyanto, "Analisis Kinerja Routing Protocol RIP pada Jaringan IPv4 dan

IPv6,” 2014. [Online]. Available:
http://eprints.binadarma.ac.id/2017/1/Jurnal_AAN_10142350.pdf. [Accessed:
24-May-2018].
