

ANALISIS PERFORMANSI RTMP *LIVE STREAMING SERVER* BERBASIS *RASPBERRY PI* UNTUK *VIDEO SURVEILLANCE SYSTEM*

PERFORMANCE ANALYSIS RTMP LIVE STREAMING SERVER BASED RASPBERRY PI FOR VIDEO SURVEILLANCE SYSTEM

Fadli Ihsan Winarto¹, Budhi Irawan², Randy Erfa Saputra³

Prodi S1 Sistem Komputer, Fakultas Teknik, Universitas Telkom

¹winartofadliihsan@students.telkomuniversity.ac.id

²budhiirawan@telkomuniversity.ac.id, ³resaputra@telkomuniversity.ac.id

Abstrak

Live streaming memerlukan protokol *streaming* untuk mengirimkan *packet data*. Protokol tersebut memiliki fungsi menghubungkan *client* dengan server untuk saling berkomunikasi dan bertukar informasi. Banyak macam- macam protokol yang digunakan untuk *live streaming* salah satunya adalah *Real Time Messaging Protokol* atau RTMP.

Pada tugas akhir ini penulis akan menganalisis penggunaan protokol RTMP sebagai protokol *live streaming server* pada *video surveillance system*. *Video surveillance system* ini diimplementasikan menggunakan sebuah *embedded sytem* yaitu *raspberry pi*. *Raspberry pi* akan dijadikan *live streaming server* dan *web server*. Protokol RTMP akan dianalisis performansi dan kelayakan untuk dijadikan *video surveillance system* berdasarkan paramater parameter QoS.

Berdasarkan pengujian pada tugas akhir ini, protokol RTMP memiliki nilai *throughput* terbesar 0,24 Mbps lebih kecil dibandingkan dengan protokol RTSP. Sedangkan untuk nilai *delay*, protokol RTMP memiliki nilai terbesar 126,93 ms lebih kecil dibandingkan dengan nilai protokol RTSP. Sehingga protokol RTMP masih layak untuk dijadikan sebagai protokol *live streaming server* pada *video surveillance system*. Selain itu, *Raspberry pi* sebagai media *live streaming server* memiliki performansi yang cukup stabil.

Kata Kunci : *Video surveillance system*, *raspberry pi*, *live streaming server*, RTMP

Abstract

Live streaming requires a streaming protocol to transmit packet data. The protocol has the function of connecting the user to the server to communicate with each other and exchange information. There are a lot of kinds of protocols used for live streaming, one of them that often used is the Real Time Messaging Protocol or RTMP.

In this final project, the author will analyze the use RTMP protocol as live streaming server protocol on the video surveillance system. Video surveillance system is implemented using an embedded sytem that is raspberry pi. Raspberry pi will be a live streaming server and a web server. RTMP protocol will be analyzed performance and eligibility to serve as a video surveillance system based on QOS parameters.

Based on the test in this final project, RTMP protocol has the largest throughput of 0,24 Mbps compared with the RTSP protocol. As for the value of delay, RTMP protocol has the greatest value of 126.93 ms smaller than the value of RTSP protocol. So the RTMP protocol is still eligible to serve as a live streaming server protocol on a video surveillance system. In addition, the Raspberry pi as live streaming media server that the performance is stable.

Keywords: *Video surveillance system*, *raspberry pi*, *live streaming server*, RTMP

1. Pendahuluan

Perkembangan *video surveillance system* saat ini sangatlah penting. *Video surveillance system* telah sangat luas berkembang dan banyak digunakan di beberapa lokasi. *Video surveillance system* banyak dibutuhkan masyarakat sebagai sistem *monitoring* untuk keamanan.^[5] *Video surveillance system* ini bersifat *live streaming*. Dengan kebutuhan *video surveillance system* yang semakin meningkat setiap waktu maka diperlukan juga peningkatan kualitas layanan, performansi dan kehandalan *video streaming* itu sendiri yang bertugas menjalankan *video surveillance system*. *Video surveillance system* ini menggunakan sebuah protokol yang membawa paket-paket data *streaming*. Oleh karena itu, diperlukan protokol yang memiliki kehandalan dengan performa bagus dalam memproses *video streaming* pada *video surveillance system*. Salah satu contoh protokol yang umumnya digunakan untuk *streaming* yaitu *Real Time Messaging Protokol* (RTMP). RTMP merupakan protokol berbasis *streaming* yang

telah banyak digunakan.^[10] Selain itu protokol RTMP merupakan protokol yang didesain untuk performansi tinggi dalam transmisi audio, video dan data.^[8] Protokol tersebut nantinya yang akan mengontrol dan mengirimkan paket data *streaming* dari *server* ke *client*. *Client* sebagai pengguna *video surveillance system* yang akan menjalankan *video streaming* dari kamera yang terintegrasi dengan perangkat yang ada pada *surveillance system*.

Dilihat dari permasalahan diatas maka dalam penelitian ini akan membahas mengenai analisis performansi protokol RTMP *live streaming server* untuk *video surveillance system* yang terintegrasi dengan *raspberry pi camera module*. Perangkat yang digunakan adalah sebuah *embedded system* yaitu *raspberry pi 2* yang akan dimanfaatkan sebagai *live streaming server* dan web server. Sehingga akan mendapatkan hasil dari performansi dan kelayakan protokol RTMP yang diimplementasikan pada *live streaming server*.

Penelitian ini memiliki beberapa rumusan masalah sebagai berikut:

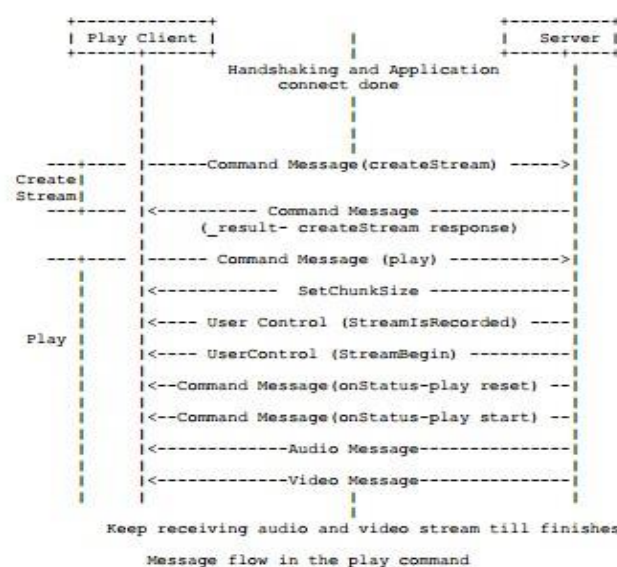
1. Bagaimana mengimplementasikan protokol RTMP sebagai *live streaming server* menggunakan *embedded system*?
2. Bagaimana performansi protokol RTMP sebagai *live streaming server* untuk *video surveillance system* serta perbandingan dengan protokol RTSP ?
3. Bagaimana kehandalan *embedded system* sebagai *live streaming server* untuk *video surveillance system* dilihat berdasarkan nilai CPU *usage* ?

Adapun tujuan yang ingin dicapai dari penelitian ini adalah, untuk mengetahui pengimplementasian protokol RTMP sebagai *live streaming server* menggunakan *embedded system*, untuk mengetahui hasil analisis performansi protokol RTMP sebagai *live streaming server* pada *video surveillance system* dan perbandingan dengan protokol lain, dan juga untuk mengetahui kehandalan *embedded system* dalam menjalankan *live streaming server* untuk keperluan *video surveillance system*.

2. Dasar Teori

2.1 RTMP

Real time messaging protocol atau RTMP merupakan sebuah protokol *proprietary* yang dikembangkan oleh adobe system untuk *streaming* audio, video dan data melalui *flash player* ataupun *flash media player*. RTMP merupakan contoh tradisional *streaming protocol*, RTMP salah satu versi dari sekian banyak *streaming protocol* untuk web. Protokol RTMP sebagai protokol yang bersifat *stateful* yaitu protokol yang menjaga komunikasi hubungan antara *client* sampai waktu aksesnya terputus. Pada RTMP dirancang dengan performa yang tinggi untuk mentransmisikan data video dan audio antara adobe flash.^[8] Protokol RTMP ini memiliki banyak variasi seperti RTMPS, RTMPE dan RTMPT. RTMPS merupakan RTMP melalui TSL /SSL koneksi, RTMPE RTMP yang dienkripsi dengan mekanisme keamanan, sementara RTMPT merupakan RTMP yang dikemas dalam HTTP permintaan untuk melintasi *firewall*.^[11] Protokol RTMP yang digunakan untuk *streaming* memiliki dua struktur penting dalam RTMP tersebut yaitu *message* dan *chunk* ^[10]



Gambar 1 Basic Flow Protokol RTMP Stream Video

2.2 Raspberry pi

Raspberry pi merupakan sebuah *embedded system* yang memiliki ukuran sebesar kartu kredit yang digunakan untuk pemrograman sederhana. Raspberry pi memiliki OS berbasis linux. Raspberry pi menggunakan chip Broadcom BCM2835 yang mencakup 700 MHZ ARM 11 yang merupakan otak dari *central processing unit* (processor).^[4] Raspberry pi merupakan komputer mini yang juga sangat layak dijadikan sebuah *web server* terutama untuk aplikasi *web server* seperti apache, lightpd dan nginx.^[5]

2.3 Quality of Service

QoS merupakan kemampuan suatu jaringan untuk memberikan layanan yang lebih baik pada trafik data tertentu pada berbagai jenis platform teknologi. Beberapa parameter QoS adalah *delay*, *jitter*, *packet loss*, dan *throughput*^[3]

a. *Throughput*

Throughput merupakan kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Header* dalam paket data mengurangi nilai ini.. Nilai *throughput* didapat dari jumlah packet data yang diterima lalu dibagi dengan lama pengamatan.^{[3][14]}

$$\frac{\text{[Paket Data]}}{\text{[Waktu]}} = \text{[Throughput]} \quad (1)$$

b. *Delay*

Delay adalah total waktu yang dibutuhkan paket untuk menempuh jarak dari pengirim ke penerima . *Delay* yang paling sering dialami oleh *traffic* adalah *delay* transmisi.^{[3][14]}

$$\frac{\text{[Jarak]}}{\text{[Kecepatan]}} = \text{[Delay Transmisi]} \quad (2)$$

c. *Jitter*

Jitter merupakan variasi kedatangan paket, hal ini diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan paket.^{[3][14]}

d. *Packet loss*

Packet loss merupakan kegagalan transmisi paket mencapai tujuannya atau jumlah paket yang hilang. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinan, diantaranya yaitu:

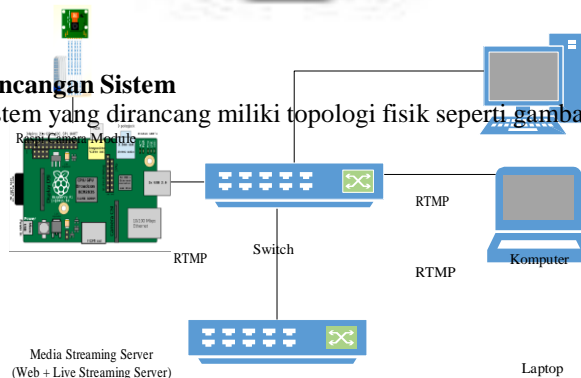
1. Terjadinya *overload trafik* didalam jaringan
2. Tabrakan (*congestion*) dalam jaringan
3. *Error* yang terjadi pada media fisik
4. Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena *overflow* yang terjadi pada *buffer*.^{[3][14]}

$$\frac{\text{[Paket yang Diterima]}}{\text{[Paket yang Dikirim]}} = \text{[Loss Rate]} \quad (3)$$

3. Perancangan Sistem

3.1. Gambaran Umum Perancangan Sistem

Secara sistematis system yang dirancang memiliki topologi fisik seperti gambar dibawah ini:



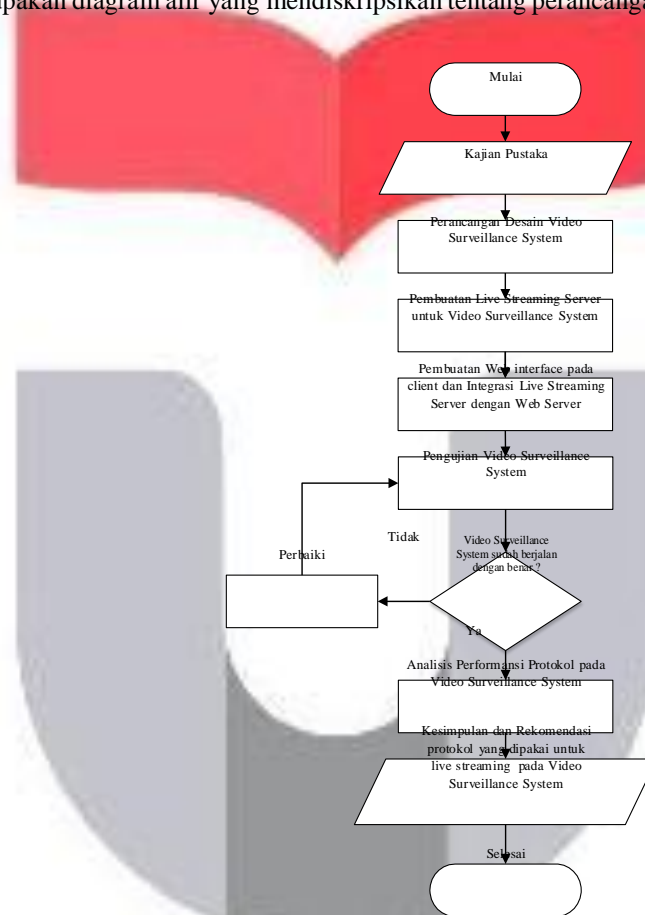
Gambar 2 Topologi Video Surveillance System

Alur proses *video surveillance system* dari *server* sampai ke *client* sesuai dengan gambar 3.1 diatas diantaranya:

1. *Raspberry pi* sebagai *live streaming server* yang terintegrasi dengan kamera video yaitu *raspberry pi camera module* dan *web server* sebagai *interface* yang akan melayani *client* yang mengakses *video surveillance system*.
2. *Client* akan terkoneksi dengan *live streaming server raspberry pi* dan akan menerima data *video* melalui LAN atau jaringan lokal.
3. Kamera video akan *capture* video secara *real time* dan akan mengirimkan video ke *raspberry pi*
4. RTMP sebagai protokol berbasis TCP akan bekerja untuk mengirim paket data *stream video surveillance system* pada *client* dan juga berfungsi sebagai pengontrol aliran data video dari *live streaming server* yang terintegrasi dengan kamera.

3.2 Diagram Alir Perancangan Sistem

Berikut ini merupakan diagram alir yang mendiskripsikan tentang perancangan dan realisasi sistem.



Gambar 3 Diagram Alir Perancangan *Video Surveillance System*

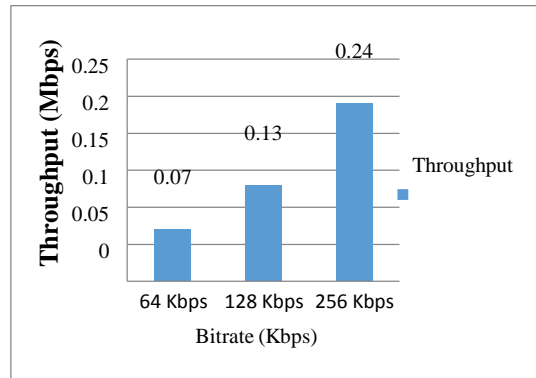
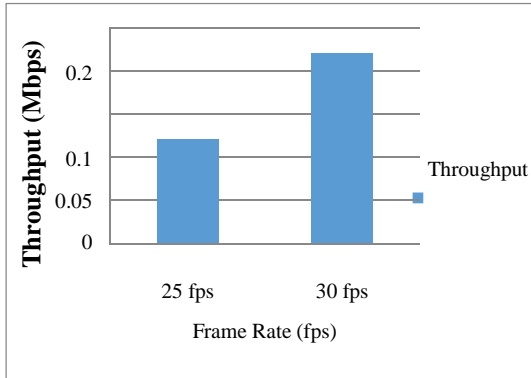
4. Pengujian dan Analisis

Pada tahap pengujian ini ditampilkan hasil pengukuran dari QoS dari sistem dan performansi *embedded system* yaitu *raspberry pi*.

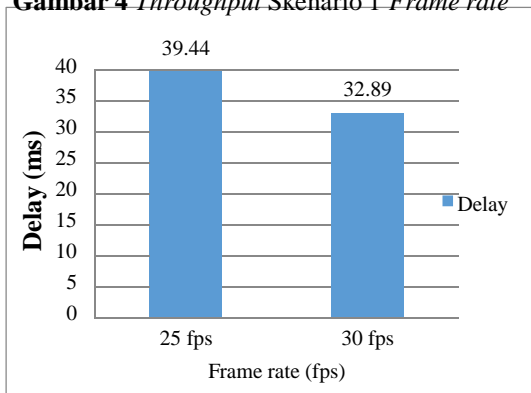
- a. Skenario 1 (pengaruh *frame rate* dan *bitrate*)
- b. Skenario 2 (pengaruh *background traffic* pada *bitrate*)
- c. Skenario 3 (pengaruh *background traffic* pada *frame rate*)

4.1 Hasil pengujian QoS

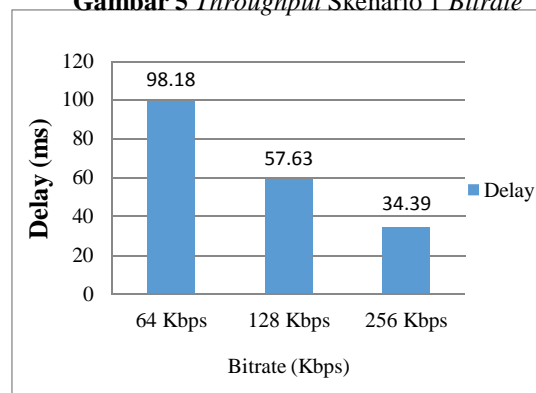
- Skenario 1



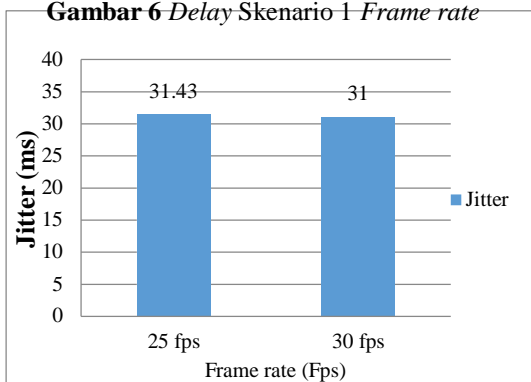
Gambar 4 Throughput Skenario 1 Frame rate



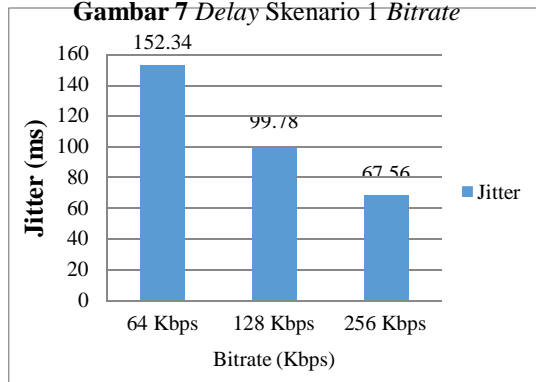
Gambar 5 Throughput Skenario 1 Bitrate



Gambar 6 Delay Skenario 1 Frame rate



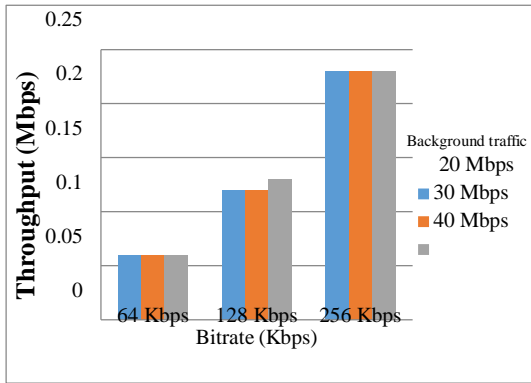
Gambar 7 Delay Skenario 1 Bitrate



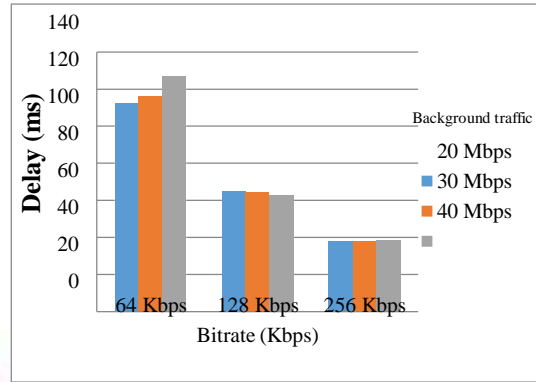
Gambar 8 Jitter Skenario 1 Frame rate

Gambar 9 Jitter Skenario 1 Bitrate

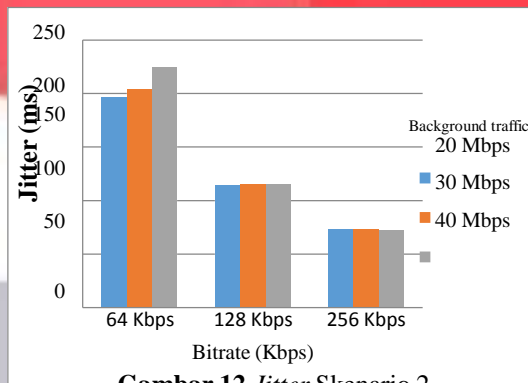
- Skenario 2



Gambar 10 Throughput Skenario 2

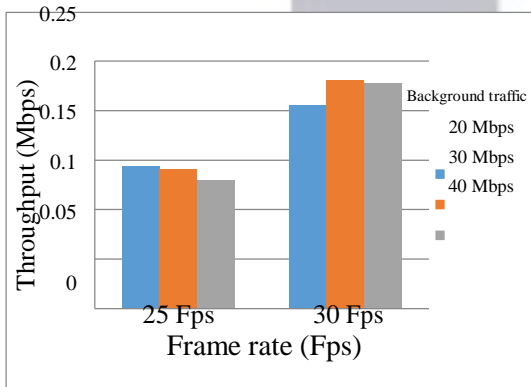


Gambar 11 Delay Skenario 2

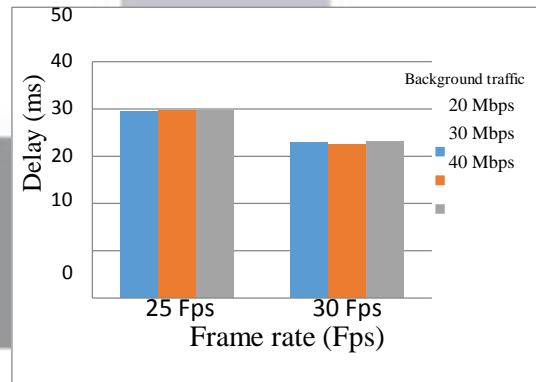


Gambar 12 Jitter Skenario 2

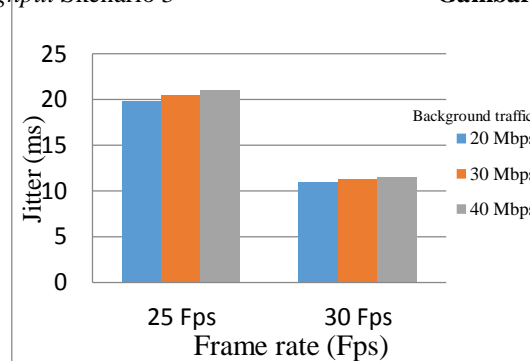
- Skenario 3



Gambar 13 Throughput Skenario 3



Gambar 14 Delay Skenario 3



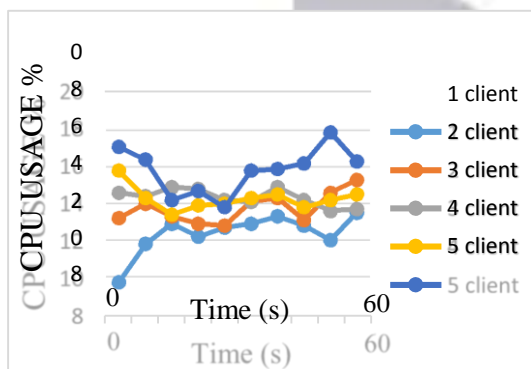
Gambar 15 Jitter Skenario 3

4.2 Analisis Pengujian QoS

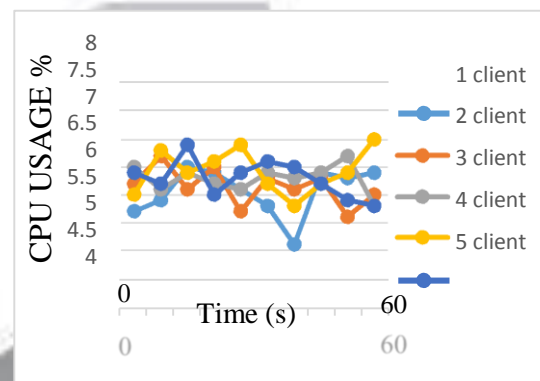
Hasil pengujian dari skenario 1, skenario 2 dan skenario 3 dapat dilihat nilai *throughput* akan selalu meningkat apabila nilai *frame rate* dan *bitrate* semakin besar. Dikarenakan semakin besar nilai *frame rate* atau nilai *bitrate* maka jumlah paket data juga semakin banyak sehingga nilai *throughput* juga akan semakin mengalami kenaikan yang cukup besar. Untuk pengujian pengaruh *background traffic* terhadap skenario uji 2 dan uji 3 yaitu *bitrate* dan *frame rate*, didapatkan hasil bahwa *background traffic* tidak terlalu mempengaruhi nilai *throughput*, *delay* dan *jitter*. Kecenderungan nilai hampir sama walaupun *background traffic*-nya berbeda-beda. Untuk nilai *delay* akan semakin menurun berdasarkan kenaikan dari *frame rate* dan *bitrate* dikarenakan jumlah data yang semakin banyak membuat nilai *delay* semakin turun karena data yang semakin banyak membuat nilai *delay* antar paket menjadi kecil. Begitu juga dengan nilai *jitter* yang juga akan semakin menurun berdasarkan kenaikan dari *frame rate* dan *bitrate*. Tetapi nilai *jitter* terkadang tidak stabil berdasarkan dari waktu kedatangan tiap paket yang berbeda beda.

Dari hasil pengujian 3 skenario diatas, dapat dilihat bahwa semua nilai *delay* yang dihasilkan tidak lebih besar dari 150 ms yang merupakan batas nilai *delay* kategori baik menurut ITU-T G.1010.^[15] Nilai *delay* untuk aplikasi video direkomendasikan < 150 ms. Nilai *delay* tertinggi dari hasil pengujian ini adalah 126,93 ms dan yang terendah adalah 32,52 ms. Sehingga *live streaming* RTMP ini masih masuk kategori layak untuk *live streaming*. Sementara nilai *jitter* tertinggi yaitu 225,18 ms dan yang terendah 10,98 ms Tidak terdapat *packet loss* dari semua skenario uji yang dilakukan, nilai *packet loss* 0 %. Hal ini dikarenakan protokol RTMP merupakan protokol berbasis TCP yang mempertahankan sebuah koneksi dan bersifat *connection oriented* sehingga paket yang dikirim dari sumber akan terjamin sampai ke *client*. Pada pebandingan performansi RTMP dengan RTSP, protokol RTSP lebih memiliki nilai *throughput* yang lebih tinggi dari pada RTMP dari seluruh scenario pengujian yang telah dilakukan. RTSP juga lebih memiliki kualitas gambar yang lebih baik dalam *live video streaming*.

4.3 Hasil Pengujian Performansi Embedded System



Gambar 15 CPU Usage berdasarkan Frame Rate



Gambar 16 CPU Usage berdasarkan Bitrate

4.4 Analisis Pengujian Performansi Embedded System

Hasil nilai CPU usage raspberry pi yang merupakan *embedded system live streaming server* masih cenderung stabil. Baik saat menjalankan *live streaming* berdasarkan *frame rate* maupun *bitrate*. Nilai CPU usage tidak terlalu mengalami peningkatan. Tidak terdapat perbedaan nilai CPU usage yang terlalu signifikan meskipun jumlah *client* yang mengakses raspberry pi bertambah. Dapat disimpulkan bahwa raspberry pi yang merupakan sebuah *embedded system* sangat baik dijadikan sebuah *live streaming sever* dikarenakan memiliki kestabilan dari sisi CPU usage.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil percobaan yang telah penulis lakukan dan analisa data yang ada dapat disimpulkan bahwa

1. Nilai *delay* yang dihasilkan dari seluruh skenario uji masih dibawah batas standar untuk video streaming berdasarkan ITU-T G.1010 yaitu masih dibawah 150 ms.^[15] Nilai *jitter* tertinggi dari 3 skenario uji adalah 225,18 ms sedangkan paling rendah adalah 10,98 ms.
2. Pengaruh *background traffic* tidak terlalu signifikan pada jaringan, sehingga *background traffic* tidak terlalu mempengaruhi penurunan nilai QoS pada jaringan *video surveillance system*.
3. Tidak terdapat *packet loss* dari semua pengujian sistem, sehingga protokol RTMP sangat baik digunakan implementasikan sebagai *live streaming server*.

4. Protokol RTMP memiliki nilai *throughput* yang lebih kecil dibandingkan dengan protokol RTSP. Untuk nilai *delay* protokol RTMP memiliki nilai yang lebih kecil dari pada nilai RTSP sehingga RTMP lebih cepat dalam mengirimkan video sementara RTSP lebih baik dalam kualitas video selain itu *raspberry pi* sebagai *embedded system live streaming server* cukup memiliki performansi yang stabil dan sangat baik diimplementasikan untuk *video surveillance system*.

5.2 Saran

Saran yang bias diberikan untuk penelitian selanjutnya dengan topic yang sama adalah:

1. Penelitian tentang keamanan dari video surveillance system ini.
2. Pengembangan sistem *video surveillance system* yang terintegrasi dengan jaringan *internet*.
3. Pengembangan pembuatan aplikasi berbasis *mobile* yang terintegrasi dengan *video surveillance system* ini.

Daftar Pustaka

- [1] Parmar, H. Thornburgh, M. 2012. *RTMP Specification*.
- [2] Diharso, Adi Setio. *Analisa Perbandingan Performansi Video Streaming Antara Darwin Streaming Server dengan Red5*. Jurnal komputer dan komunikasi. 2013.
- [3] Diwi, Anggelina I, M. R Rumani dan Wahidah, Ida. *Analisis Kualitas Layanan Video Live streaming Pada Jaringan Lokal Universitas Telkom*. Jurnal Pusat Penelitian dan Pengembangan Sumber Daya dan Perangkat Pos dan Informatika. Volume 12. 2014. ISSN: 16930991.
- [4] Dawood, Rahmad. Fairuz Qiana, Said. Muchallil, Sayed. *Kelayakan Raspberry Pi sebagai Web Server: Perbandingan Kinerja Nginx, Apache, dan Lighttpd pada Platform Raspberry Pi*. Jurnal Rekayasa Elektrika Vol 11, No 1, April 2014. Halaman 25-29.
- [5] Kanzariya, Suni. Vora, Vishal. *Real Time Monitoring System Using Raspberry Pi*. *International journal of Advance Engineering Development*.
- [6] Fahmi, Muhammad Agli. T, Iwan Iwut. S, Iman Hedi. *Implementasi dan Analisis TV Streaming Menggunakan Protokol RTMP Pada Aplikasi Berbasis Web*.
- [7] Prasad, Sanjana. Mahalakshmi, P. Sunder, A John Clement dan Swathi, R. 2014. *Smart Surveillance Monitoring System Using Raspberry PI dan PIR Sensor*. *International Journal of computer and information Technology*, vol 5.
- [8] Guniganti, Ramesh Goud dan Ankam, Srikanth. 2012. *A Comparison of RTMP and HTTP Protocols with respect to Packet loss and Delay Variation based on Qoe*. Master Thesis Electrical Engineering.
- [9] Prasetya, Bayu Aditiya. 2008. *Pengaruh Video Bitrate dan Background Terhadap Kinerja Video Streamin pada Jaringan Wireless LAN*. Jurnal Departemen Ilmu Komputer IPB.
- [10] Lei, Xiuhua. Jiang, Xiuhua. Wang, Caihong. *Design and Implementation of Streaming Media Processing Software based on RTMP*. School of Information Engineering Communication University of China Beijing, China.
- [11] Modi, Darshan. *Quality Control in Video Streaming*. *International Research Journal of Engineering dan Technology (IRJET)*. Vol. 02 Issue: 06 | Sep-2015. ISSN: 2395-0072.
- [12] NginX Website. NginX Wiki's Documentation. (<https://www.nginx.com/resources/wiki/>, Diakses pada 12 Juni 2016, pukul 15.35)
- [13] Xiaohua, Lei. Jiang, Xiuhua. Wang, Caihong. 2013. *Design and Impelementation of a Real-Time Video Stream Analysis System Based on FFmpeg*.
- [14] Kurniawan, E. Sani, A. 2014. *Analisis kualitas real time video streaming terhadap bandwidth jaringan yang tersedia*. Vol. 9 No 2.
- [15] ITU-T Recommendation G.1010. (<https://www.itu.int/rec/T-REC-G.1010-200111-I/en>, Diakses pada 12 Juni 2016 pukul 10

