

# RANCANG BANGUN DAN ANALISA UNJUK KERJA SISTEM PENGAWASAN DENGAN FRAME RATE DINAMIS BERDASARKAN DETEKSI PERGERAKAN

S. I. Pella<sup>1</sup>

<sup>1</sup> Jurusan Teknik Elektro, Universitas Nusa Cendana,  
Jl AdiSucipto, Penfui, Kupang, Indonesia  
Email: s.i.pella@gmail.com

## Info Artikel

Histori Artikel:  
Diterima Mar 01, 2021  
Direvisi Mar 30, 2021  
Disetujui Apr 25, 2021

## ABSTRACT

Designing the Internet of Things (IoT) based surveillance systems is usually a trade of the frame rate in the video streaming to capture any movement in the frame and the bandwidth needed to stream the video. This research proposes a way to achieve bandwidth efficiency while still stream in a required frame rate by implementing video streaming with a dynamic frame rate based on movement detection. The system comprises surveillance devices; each consists of a Raspberry Pi single-board computer connected to a camera, and a central web server that mainly handles user authorization. The system was developed using Motion server, MySQL database engine and PHP programming language. A system measurement using Wireshark network analyser is used to evaluate key parameters of the system such as bandwidth requirement, round trip time delay and error packet. The result shows that using a system with a dynamic frame rate requires as less as 25% bandwidth compared to the system with fix framer rate.

**Keywords:** Surveillance System, IoT, Movement Detection, Raspberry Pi, QoS, Wireshark.

## ABSTRAK

Salah satu pertimbangan dalam merancang sistem pengawasan berbasis Internet of Things (IoT) adalah jumlah frame yang dikirim setiap detiknya harus cukup untuk mendeteksi setiap pergerakan yang terjadi dan bandwidth yang dibutuhkan untuk mentransmisi frame-frame tersebut. Penelitian ini bertujuan merancang sebuah sistem yang dapat mengirimkan video dengan kecepatan frame yang dinamis untuk efisiensi bandwidth. Sistem terdiri dari peralatan-peralatan pengawasan dan sebuah computer server. Setiap peralatan pengawasan terdiri dari computer single board Raspberry PI, terkoneksi dengan sebuah kamera dan menjalankan motion server untuk mengatur pengambilan gambar dan transmisi video. Komputer server bertujuan untuk otorisasi pengguna dan dibangun menggunakan bahasa program PHP dan mesin basis data MySQL. Pengukuran kinerja sistem dilakukan pada jaringan lokal dengan menggunakan Wireshark. Hasil pengujian menunjukkan kebutuhan bandwidth sistem dengan frame rate dinamis mencapai 25% dari kebutuhan bandwidth pada sistem dengan frame rate statis.

**Kata Kunci:** Sistem Pengawasan keamanan, IoT, Deteksi Pergerakan, Raspberry Pi, QoS, Wireshark

## Penulis Korespondensi:

S. I. Pella,  
Program Studi Teknik Elektro Fakultas Sains dan Teknik,  
Universitas Nusa Cendana,  
Jl. Adisucipto Penfui - Kupang.  
Email: s.i.pella@gmail.com

## 1 PENDAHULUAN

Sistem pengawasan dengan kamera pemantau sudah menjadi standar keamanan di berbagai lingkungan, seperti gedung perkantoran, asrama, sekolah, pusat bisnis, bahkan fasilitas masyarakatan [1, 2]. Pemanfaatan mikrokontroler dan *single board computer* yang sudah dilengkapi fasilitas Internet of Things (IoT) dan kemampuan pengolahan multimedia, menyebabkan hasil pemantauan kamera dan pengolahan video dapat diakses di mana saja tanpa pembatasan jarak secara fisik.

Beberapa penelitian pendahuluan sudah dilakukan dengan memanfaatkan pengolahan multimedia dan kecerdasan buatan untuk mengembangkan sistem pengawasan keamanan berbasis IoT. Penelitian pada [3], menggunakan algoritma *deep learning*, untuk mendeteksi wajah yang masuk dalam basis data daftar pencarian orang (DPO) dan melaporkannya secara online. Penelitian pada [4-6], mengembangkan sistem pengawasan keamanan dengan memanfaatkan sensor pergerakan. Penelitian yang lain menggunakan *threshold* perbedaan piksel antara *frame* yang berurutan untuk mendeteksi pergerakan, seperti pada [7, 8]. Setiap penelitian diatas, menggunakan deteksi pergerakan sebagai dasar melakukan aksi tertentu seperti mengirimkan laporan [4, 5, 7] atau melakukan perekaman data [6].

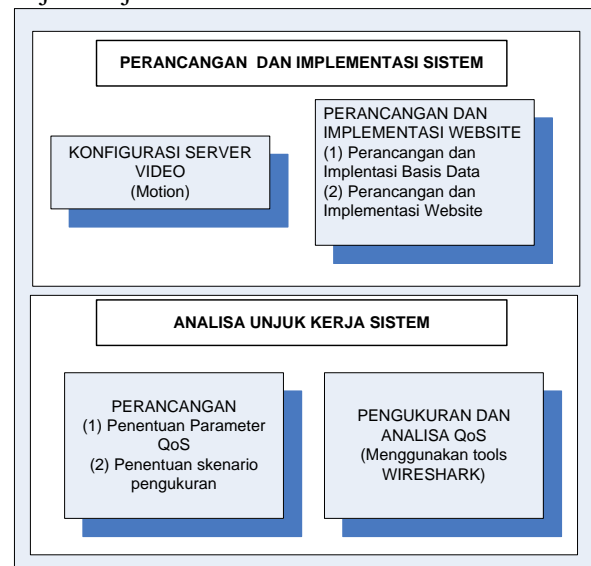
Pada perancangan sistem IoT, pengukuran dan analisa kinerja sistem pada jaringan computer merupakan bagian yang esensial. Penelitian [9], membandingkan unjuk kerja protocol TCP dan UDP pada sistem IoT pendeteksi asap rokok. Penelitian [10] menganalisa unjuk kerja protocol UDP pada beberapa aplikasi multimedia seperti VoIP dan *Video Conference*, menggunakan program simulasi NS-2. Penelitian pada [11, 12] mengukur kualitas pelayanan video streaming pada jaringan lokal. Diantara penelitian-penelitian yang disebutkan diatas, tidak ada yang menganalisa unjuk kerja sistem IoT berbasis multimedia. Penelitian [9] tidak melibatkan data multimedia, penelitian [11, 12] tidak mengukur sistem IoT dan penelitian [10] menggunakan simulasi bukan pengukuran sistem sebenarnya.

Penelitian ini memberikan dua kontribusi utama. Pertama, berbeda dengan penelitian-penelitian sebelumnya, penelitian ini menggunakan deteksi pergerakan sebagai dasar menentukan jumlah *frame* per detik (*frame rate*) yang dikirimkan

oleh peralatan keamanan (*surveillance devices*). Ketika tidak ada pergerakan terdeteksi, *surveillance dev* mengirimkan video dengan *frame rate* minimal. *Frame rate* akan meningkat sampai *frame rate* maksimal seiring dengan meningkatnya jumlah pergerakan yang terdeteksi. Hal ini bertujuan untuk mencapai efisiensi penggunaan bandwidth. Selain itu, penelitian ini menyajikan pengukuran dan analisa perbandingan unjuk kerja sistem pengawasan berbasis IoT dengan *frame rate* statis dan *frame rate* dinamis

## 2 METODE PENELITIAN

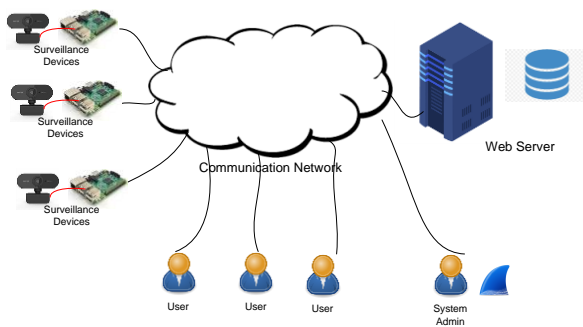
Blok diagram kerja pada penelitian ini dapat dilihat pada Gambar 1. Secara umum, penelitian ini dapat dibagi menjadi 2 tahap, yaitu perancangan dan implementasi sistem dan analisa unjuk kerja sistem.



Gambar 1 Blok Kerja Penelitian

### 2.1 Disain Sistem Pengawasan

Sistem pengawasan yang dibangun pada penelitian ini terdiri dari sejumlah *surveillance devices* dan sebuah *web server*, seperti ditunjukkan pada Gambar 2. Sebuah *surveillance device* terdiri dari *single board computer* yang terhubung dengan camera. Perangkat lunak yang digunakan untuk mengatur pengambilan video oleh kamera dan *video streaming* adalah Motion, yang memungkinkan berbagai konfigurasi sesuai kebutuhan sistem. pengguna mengakses *surveillance devices* melalui webserver yang bertugas mengatur otorisasi pengguna. Pengukuran unjuk kerja sistem pada penelitian ini dilakukan pada sisi pengguna (client) menggunakan perangkat lunak analisa jaringan Wireshark.



Gambar 2 Diagram Blok

Properti yang digunakan untuk pengiriman video pada setiap *surveillance devices* dapat dilihat pada Tabel 1.

Tabel 1 Properti Pengiriman Video

Properti	Nilai
Resolusi	320 x 240
Frame rate	1 s/d 5
Threshold	1000 pikses
Transport Protokol	TCP
Socket Port	8088

Seperti yang tertera di Tabel 1, untuk mendapatkan efisiensi penggunaan *bandwidth*, sistem menggunakan *frame rate* yang dinamis antara 1 dan 5 *frame* per detik. Ketika tidak ada pergerakan yang terdeteksi (jumlah piksel yang berbeda antara dua buah *frame* berurutan dibawah *threshold*), maka video akan dikirimkan dengan *frame rate* minimal (1 *frame* per detik). Pada saat perbedaan piksel antara dua buah *frame* berurutan diatas nilai *threshold*, kecepatan pengiriman *frame I* meningkat, sebanding dengan jumlah piksel yang berubah, sampai mencapai maksimal *frame rate*, yaitu 5 *frame* per detik.

Protokol transport yang digunakan adalah *Transport Control Protocol* (TCP) yang memberikan garansi reliabilitas dengan pengiriman kembali frame ang diterima dalam keadaan rusak. Socket yang digunakan untuk *streaming* video dilekatkan pada port 8081.

Webserver berperan otorisasi akses *surveillance devices* oleh pengguna dan memungkinkan pengaksesan lebih dari satu *surveillance devices* pada saat yang bersamaan. Website yang digunakan pada sistem ini dibangun menggunakan bahasa pemrograman PHP dan basis data MySQL.

### 2.2 Perancangan Analisa Unjuk Kerja Sistem

Analisa unjuk kerja sistem meliputi tiga parameter *quality of service*, yaitu :

- Kebutuhan *bandwidth*, didefinisikan sebagai volum data yang terkirim dari *surveillance device(s)* ke computer pengguna per satuan waktu.
- Round trip time* (RTT), didefinisikan selang waktu antara pengiriman *acknowledgement* (ACK), sebuah paket sampai penerimaan paket selanjutnya.
- Packet error*, didefinisikan sebagai paket yang diterima dalam keadaan rusak

Skenario pengukuran QoSs sistem terdiri dari

- Pengukuran QoS dengan *frame rate* statis, dengan *frame rate* bervariasi
- Pengukuran QoS sistem dengan *frame rate* dinamis berdasarkan deteksi pergerakan dengan *frame rate* maksimal bervariasi
- Pengukuran QoS sistem dengan jumlah *surveillance devices* bervariasi.

Secara keseluruhan terdapat 18 kondisi (K1-K9) yang digunakan dalam pengukuran dan analisa QoS dapat dilihat pada Tabel 2.

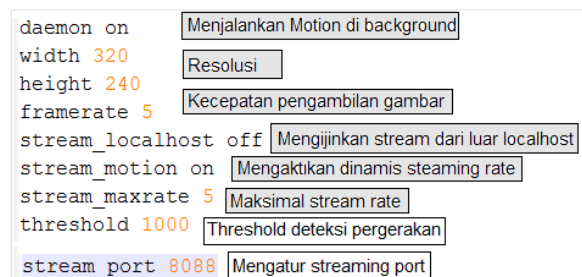
Tabel 2 Kondisi Pengukuran dan Analisa QoS

		Jumlah <i>Surveillance Devices</i>		
		1	2	3
Static frame rate	1 fps	K1		
	3 fps	K2	K7	
	Max 2 fps	K3		
Dynamic Frame rate	Max 3 fps	K4	K8	K9
	Max 4 fps	K5		
	Max 5 fps	K6		

## 3 HASIL DAN PEMBAHASAN

### 3.1 Implementasi Sistem Pengawasan

Bagian pertama dari implementasi sistem adalah konfigurasi perangkat lunak motion untuk perekaman dan pengiriman video. Konfigurasi yang digunakan dapat dilihat pada Gambar 3.



Gambar 3 Konfigurasi Server Motion.

Bagian utama dari konfigurasi server adalah mengaktifkan akses dari luar localhost,

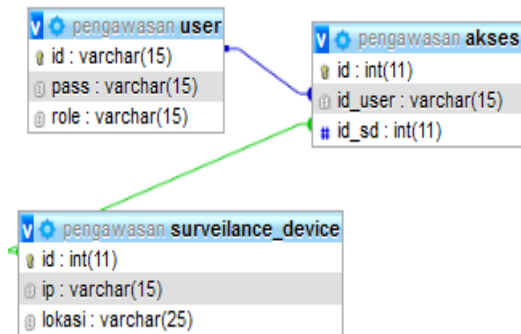
mengaktifkan dinamis streaming rate, mengatur maksimal frame yang dikirimkan per detik dan mengatur port yang dilekatkan pada socket. Setelah konfigurasi motion server pada *surveillance devices*, bagian kedua adalah implementasi basis data dan halaman website. Basis data dibangun menggunakan *database engine* MySQL dan halaman web dibangun dengan bahasa pemrograman PHP. Implementasi basis data dapat dilihat pada Gambar 4, dan implementasi halaman akses *surveillance devices* dapat dilihat pada Gambar 5.

**3.2 Analisa Unjuk Kerja Sistem.**

Analisa unjuk kerja sistem dilakukan dengan bantuan perangkat lunak *network sniffer* dan *analyzer* Wireshark dan dilakukan di jaringan lokal. Untuk memastikan, wireshark merekam paket yang benar, digunakan filter IP address *surveillance devices* dan port *streaming*.

Gambar 6 menunjukkan rangkuman perekaman paket dari server ke komputer pengguna pada kondisi *frame rate* statis (K1, K2) dan *frame rate* dinamis (K4), dengan durasi perekaman 60 detik. Perekaman dilakukan dengan dua macam pergerakan, yaitu terdeteksi pergerakan dengan total waktu 12 detik (K4, A) dan terdeteksi pergerakan dengan total waktu 60 detik (K4,B). Dari gambar terlihat, pada pengiriman video dengan *frame rate* statis kebutuhan penggunaan *bandwidth* meningkat tiga kali lipat dari 61Kbps (a) ke 180 kbps (b) pada saat *frame rate* meningkat dari 1 frame per detik ke 3 frame per detik. Pada pengiriman video dengan *frame rate* dinamis, dengan maksimal *frame rate* 3 frame per detik, peningkatan kebutuhan *bandwidth* hanya sekitar 25% untuk durasi pergerakan 20% dari durasi

rekaman, dan 40% untuk durasi pergerakan sama dengan durasi rekaman.



Gambar 4 Implementasi Basis Data

**Pemantauan Keamanan**



Gambar 5 Halaman Website Akses Surveillance Devuces

Measurement	Captured	Displayed
Packets	415	382 (92.0%)
Time span, s	61.933	54.823
Average pps	6.7	7.0
Average packet size, B	1052	1099
Bytes	436648	419638 (96.1%)
Average bytes/s	7050	7654
Average bits/s	56k	61k

(a)

Measurement	Captured	Displayed
Packets	1260	1260 (100.0%)
Time span, s	61.362	61.362
Average pps	20.5	20.5
Average packet size, B	1101	1101
Bytes	1386894	1386894 (100.0%)
Average bytes/s	22k	22k
Average bits/s	180k	180k

(b)

Measurement	Captured	Displayed
Packets	586	586 (100.0%)
Time span, s	61.271	61.271
Average pps	9.6	9.6
Average packet size, B	985	985
Bytes	576993	576993 (100.0%)
Average bytes/s	9417	9417
Average bits/s	75k	75k

(c)

Measurement	Captured	Displayed
Packets	739	739 (100.0%)
Time span, s	61.721	61.721
Average pps	12.0	12.0
Average packet size, B	883	883
Bytes	652305	652305 (100.0%)
Average bytes/s	10k	10k
Average bits/s	84k	84k

(d)

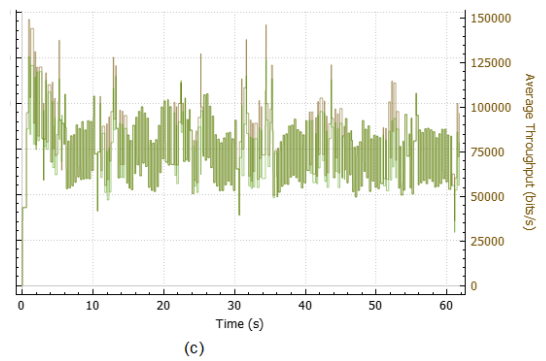
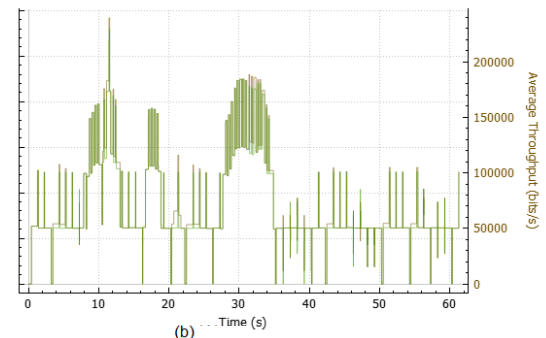
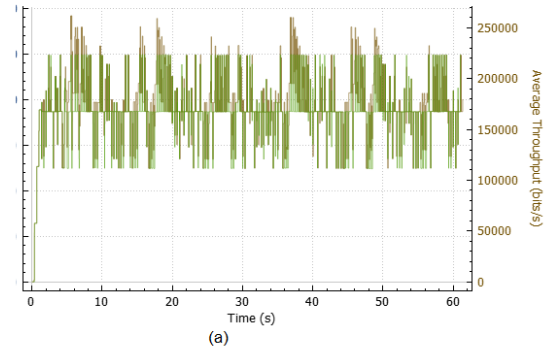
Gambar 6 Rangkuman Perekaman Data Video Streaming untuk (a) K1, (b) K2, (c) K4,A, (d) K4,B

Gambar 7, menunjukkan grafik distribusi kebutuhan bandwidth selama durasi perekaman. Pada sistem menggunakan *frame rate* statis dengan 3 *frame*/detik (a), kebutuhan *bandwidth* merata selama durasi perekaman (sekitar 180 kbps). Pada gambar (6(b)) deteksi pergerakan terjadi pada sekitar detik ke-10, ke -18 dan ke-30. Pada saat terjadi pergerakan terjadi peningkatan kebutuhan *bandwidth*. Pada saat tidak ada gerakan terdeteksi, sistem bekerja dengan *frame rate* minimal yaitu 1 *frame* perdetik. Pada Gambar 6(c), terjadi pergerakan selama durasi perekaman. Peningkatan *frame rate* sejalan dengan kecepatan terjadi pergerakan (peningkatan perubahan piksel pada dua buah *frame* yang berurutan). Peningkatan *bandwidth* yang terjadi tidak sebesar pada sistem dengan *frame rate* statis.

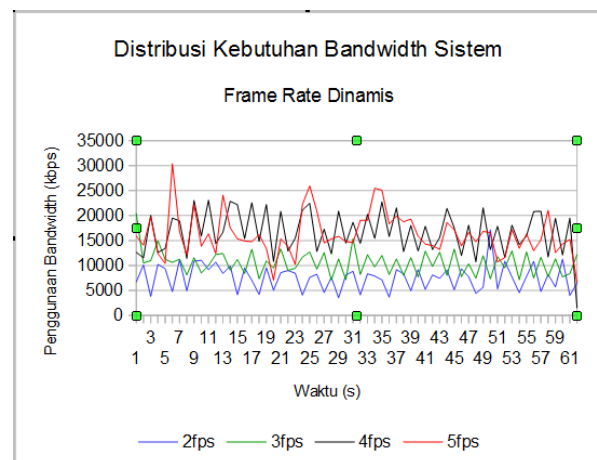
Gambar 7 dan 8 menunjukkan perubahan kebutuhan bandwidth dan *packet error* pada sistem pengawasan dengan *frame rate* dinamis ketika *frame rate* maksimal ditingkatkan. Untuk mendapatkan kebutuhan *bandwidth* maksimal, pengukuran dilakukan dengan kondisi selalu terjadi pergerakan.

Gambar 8 menunjukkan grafik distribusi kebutuhan bandwidth ketika kualitas video ditingkatkan dengan meningkatkan maksimal *frame rate* dalam keadaan selalu terdeteksi pergerakan selama durasi perekaman. Seperti diharapkan, penggunaan bandwidth bertambah ketika *frame rate* maksimal ditingkatkan. Poin yang menarik adalah, ketika *frame rate* maksimal ditingkatkan hing 5 *frame* per detik, pada keadaan selalu terdeteksi pergerakan, kebutuhan bandwidth rata-ratanya adalah tiga kali lipat dari bandwidth yang dibutuhkan untuk mentransfer video 3 *frame* per detik dengan *frame rate* statis. Dari grafik juga dapat dilihat bahwa kebutuhan bandwidth untuk sistem dengan maksimal *frame rate* 4 *frame* per detik dan sistem dengan maksimal *frame rate* 5 *frame* per detik hampir sama. Hal ini disebabkan dengan pengaturan kecepatan pengiriman *frame* dilakukan dengan otomatis, pengiriman *frame* tidak selalu mencapai *frame rate* maksimal, tetapi bergantung pada kecepatan pergerakan yang dideteksi.

Gambar 9 menunjukkan distribusi paket error (dalam bytes) untuk setiap maksimal *frame rate*. Paket error meningkat sejalan dengan peningkatan maksimal *frame rate*.

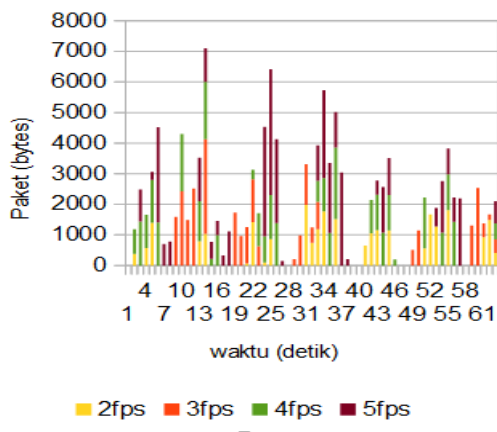


Gambar 7 Distribusi Kebutuhan Bandwidth selama Durasi Perekaman



Gambar 8 Distribusi Kebutuhan Bandwidth pada Sistem dengan *Frame Rate* Dinamis

Error Paket Sistem dengan Frame Rate Dinamis

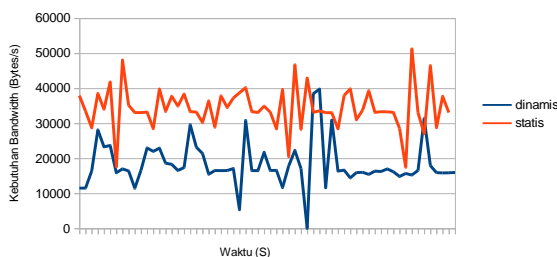


Gambar 9 Paket Error Sistem dengan Frame Rate Dinamis

Gambar 10 dan 11 menunjukkan perubahan kebutuhan bandwidth dan perubahan waktu RTT pada saat jumlah *surveillance devices* bertambah pada sistem dengan *frame rate* statis dan *frame rate* dinamis. Pengujian pada Gambar 9 dan 10 dilakukan dengan *frame rate* statis 3 frame per detik dan *frame rate* dinamis dengan maksimal *frame rate* 3 frame per detik.

Pada Gambar 10 dapat terlihat bahwa pada sistem dengan 2 *surveillance devices* secara umum membutuhkan *bandwidth* yang lebih besar daripada sistem dengan kondisi yang sama pada pengujian 1 *surveillance device*. Dari grafik dapat dilihat jika penggunaan *frame rate* dinamis menghasilkan efisiensi *bandwidth* sekitar 50% dari penggunaan *frame rate* statis

Kebutuhan Bandwidth 2 Pengguna

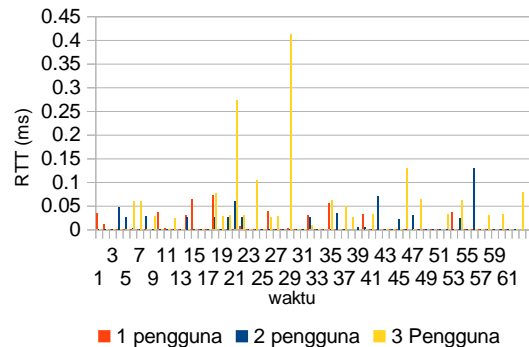


Gambar 10 Kebutuhan Bandwidth 2 Surveillance Devices

Gambar 11 menunjukkan grafik delay RTT pada sistem dengan 1, 2 dan 3 *surveillance devices*. Terlihat pada grafik bahwa sistem dengan lebih banyak *surveillance devices* memiliki RTT yang sedikit lebih besar daripada sistem dengan lebih sedikit *surveillance devices*. Perbedaan RTT tidak terlalu signifikan disebabkan oleh kondisi, yaitu kecepatan jaringan local (54 Mbps) jauh

lebih besar dari kebutuhan *bandwidth* sistem ( $\leq 400$  Kbps) dan nilai RTT dipengaruhi kesibukan jaringan local pada saat dilakukan pengujian.

Round Trip Time Delay Sistem dengan Frame Rate Dinamis



Gambar 11 Delay RTT pada Sistem dengan Jumlah Surveillance Devices Bervariasi

#### 4 KESIMPULAN

Penelitian ini berhasil merancang bangun sistem pengawasan berbasis IoT dengan *frame rate* dinamis. Hasil pengukuran kinerja sistem dilakukan pada jaringan local dengan menggunakan Wireshark. Hasil pengujian kebutuhan bandwidth menunjukkan bahwa kebutuhan sistem dengan *frame rate* dinamis berkisar antara 25%-50% dari sistem dengan *frame rate* statis untuk kualitas video dan jumlah *surveillance devices* yang sama. Pada sistem dengan *frame rate* dinamis, ketika nilai maksimal framer rate bertambah, kebutuhan bandwidth juga bertambah, namun laju pertambahan kebutuhan bandwidth dibawah laju pertambahan maksimal frame rate. Laju pengiriman paket dengan kesalahan bertambah seiring dengan pertambahan. Pada pengujian di jaringan local, pertambahan jumlah *surveillance devices* tidak berpengaruh secara signifikan terhadap laju pertambahan delay RTT.

#### DAFTAR PUSTAKA

- [1] F. M. Akbar, "ASRAMA INKLUSIF MAHASISWA BABARASI DEPOK SLEMAN DENGAN PENDEKATAN LIVING COMMUNAL SPACE," 2019.
- [2] R. Yanarsya, R. Nugroho, and A. K. Wahyuwibowo, "Strategi Penerapan Sistem Keamanan Pada Desain Lembaga Masyarakat Kelas II Di Surakarta," *ARSITEKTURA*, vol. 17, pp. 151-158.
- [3] H. F. Lami, S. Tena, B. H. Manafe, J. F. Bowakh, N. Nursalim, and S. Sudirman,

- "Rancang Bangun Sistem Pengenalan Wajah Daftar Pencarian Orang (Dpo) Berbasis Jaringan Saraf Tiruan," *Jurnal Media Elektro*, pp. 129-133, 2019.
- [4] M. I. Kurniawan, U. Sunarya, and R. Tulloh, "Internet of Things: Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 6, p. 1, 2018.
- [5] D. E. Kurniawan and S. Fani, "Perancangan sistem kamera pengawas berbasis perangkat bergerak menggunakan raspberry pi," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 3, 2017.
- [6] F. Gozali and N. Akbar, "PENDETEKSIAN DAN PEREKAMAN GERAKAN BENDA DALAM SISTEM PENGAWASAN KEAMANAN DENGAN MENGGUNAKAN RASPBERRY PI," *Jetri: Jurnal Ilmiah Teknik Elektro*, vol. 11, 2017.
- [7] A. Antonius, D. Triyanto, and I. Ruslianto, "Penerapan Pengolahan Citra Dengan Metode Adaptive Motion Detection Algorithm Pada Sistem Kamera Keamanan Dengan Push Notification Ke Smartphone Dengan Android," *Coding Jurnal Komputer dan Aplikasi*, vol. 3, 2015.
- [8] A. N. A. Thohari and R. D. Ramadhani, "Sistem Pengawasan Berbasis Deteksi Gerak Menggunakan Single Board Computer," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 8, pp. 1-6, 2019.
- [9] P. V. Romony, L. Sitanayah, and J. B. Sanger, "PERBANDINGAN QUALITY OF SERVICE PROTOKOL KOMUNIKASI DATA PADA SISTEM DETEKSI ASAP ROKOK BERBASIS INTERNET OF THINGS," *Jurnal Ilmiah Realtech*, vol. 16, pp. 19-24, 2020.
- [10] J. F.S. Adinata, A. Sukmaaji, "ANALISIS PERBANDINGAN UNJUK KERJA PROTOKOL UDP DENGAN DCCP MENGGUNAKAN DATA MULTIMEDIA," *Journal of Control and Network Systems*, vol. 3, pp. 86-93, 2014.
- [11] A. I. Diwi, R. R. Mangkudjaja, and I. Wahidah, "Analisis Kualitas Layanan Video Live Streaming pada Jaringan Lokal Universitas Telkom," *Buletin Pos dan Telekomunikasi*, vol. 12, pp. 207-216, 2014.
- [12] D. D. O. Siswoyo, R. Saptono, and N. Suharto, "PERFORMANSI APLIKASI VIDEO STREAMING MODE ON DEMAND PADA JARINGAN IPV6 DI POLITEKNIK NEGERI MALANG," *Jurnal Jartel: Jurnal Jaringan Telekomunikasi*, vol. 4, p. 86, 2017.