

Analisa Karakteristik *Single Board Computer* sebagai *Streaming Video Server*

Muhammad Adlan Hawari^{*1}, Bambang Nurcahyo Prastowo²

¹Prodi Elektronika dan Instrumentasi, DIKE, FMIPA UGM, Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ^{*1}muhammad.adlan.h@mail.ugm.ac.id, ²prastowo@ugm.ac.id

Abstrak

Perkembangan pengolahan citra tidak hanya terfokus pada sensor kamera dan perangkat lunaknya, melainkan juga pada sarana penunjangnya. Salah satu penunjang yang sekarang banyak digunakan adalah kamera berbasis komunikasi IP atau yang lebih dikenal dengan IP Camera. Kendala pada penggunaan IP Camera adalah terbatasnya akses modifikasi yang ingin dilakukan oleh para pegiat pengolahan citra. Pada penelitian ini akan dibahas salah satu solusi untuk mengatasi masalah tersebut, yakni menggunakan Odroid XU 4 yang dihubungkan dengan kamera webcam sebagai streaming video server beserta pengujiannya pada jaringan LAN yang terhubung dengan 31 klien. Hasil penelitian menunjukkan bahwa frame rate video yang diterima klien bernilai rata - rata 21,46 fps dan packet loss yang terjadi rata –rata 1,20%. Hal ini menunjukkan bahwa Odroid XU4 mampu melakukan tugasnya dengan sangat baik sebagai streaming video server dalam jaringan LAN.

Kata kunci— *Streaming Video Server, Single Board Computer, LAN*

Abstract

Development in image processing is just not focused on the camera sensor and the software. Furthermore, the supporting tools has to be concerned. IP-based cameras are used widely today. Unfortunately, IP Camera has its own limitation. We cannot modificate the system as we want. This paper has introduced a system which has the same function of IP Camera but also can be modificate as we like. The system consists of Odroid XU4 and webcam as a streaming video server. Also, the test result is included in Local network which connected to 31 clients. The result shows that the video has 21,46 fps average on each client. The packet loss is only 1,20%. This means the system works properly and categorized as “very good”.

Keywords— *Streaming Video Server, Single Board Computer, LAN*

1. PENDAHULUAN

Kemajuan teknologi dalam bidang perdagangan semakin meningkat dengan maraknya penerapan teknologi oleh berbagai perusahaan penyedia barang dan jasa. Era ini disebut dengan industri 4.0, dimana hampir semua layanan pelanggan dapat dilakukan melalui piranti lunak aplikasi pada perangkat bergerak maupun melalui *website* [1]. Salah satu contoh teknologi paling modern pada bidang perdagangan dikembangkan oleh perusahaan Amazon yang disebut dengan *Amazon Go*. *Amazon Go* adalah sebuah swalayan yang memiliki sistem *cashierless store* dan *walk out shopping* dimana toko ini tidak memiliki kasir. Pembeli yang mengambil barang dapat

langsung meninggalkan toko ini. Pembayaran dilakukan secara otomatis setelah pembeli meninggalkan toko [2].

Teknologi pada *Amazon Go* terdiri dari *Computer vision*, *sensor fusion*, dan *deep machine learning*[3]. *Computer vision* pada sistem ini menggunakan banyak kamera sehingga seluruh objek dalam toko dapat dipantau. Citra hasil tangkapan kamera tersebut kemudian dikumpulkan dan direkonstruksi agar sistem dapat mengolah citra dengan pendekatan yang hampir mendekati keadaan nyata. Oleh karena itu, *computer vision* pada sistem *Amazon Go* menjadi inspirasi bagi para pengembang pengolahan citra untuk melakukan penelitian serupa dengan pendekatan yang berbeda, salah satunya dengan penggunaan *IP Camera* sebagai pengakuisisi citranya.

IP Camera terdiri dari satu kamera sebagai pengakuisisi citra dan sebuah SBC (*Single Board Computer*) sebagai servernya. Server pada *IP Camera* terkoneksi ke jaringan dengan cara yang serupa dengan perangkat elektronik lainnya, seperti laptop, tablet, dan *smartphone* [4]. Namun, *IP Camera* yang beredar di pasaran umumnya hanya dapat menampilkan gambar atau video menggunakan sistem yang disediakan oleh produsennya masing – masing. Para pengguna *IP Camera* tidak dapat mengakses langsung sistem di dalam *IP Camera* dan juga membutuhkan biaya yang lebih banyak, sedangkan pada kenyataannya, modifikasi sistem lambat laun akan dibutuhkan seiring berkembangnya teknologi [5].

Salah satu cara untuk mengatasi keterbatasan tersebut adalah dengan menyusun sistem yang menyerupai sebuah *IP Camera*. Sehingga, para pengembang teknologi di bidang pengolahan citra dapat memaksimalkan *software* pengolah citra sesuai kehendak. Sistem yang menyerupai *IP Camera* dapat dibentuk menggunakan dua komponen utama, yakni kamera dan SBC. Jenis kamera yang paling mudah ditemukan dan dioperasikan adalah kamera jenis *webcam* [6]. Pemilihan SBC sebagai server merupakan hal yang penting terutama pada pengolahan video. Pergantian *frame* pada sebuah video bergantung pada kemampuan prosesor dari SBC yang digunakan. Semakin cepat prosesor, maka pergantian *frame* akan semakin cepat sehingga video yang dihasilkan semakin bagus [7]. Selain kemampuan pada pergantian *frame*, SBC yang digunakan juga harus mampu menangani permintaan pengiriman video yang diakses oleh klien. Dapat dikatakan bahwa SBC berperan sebagai *streaming* video server. Dalam hal ini, kemampuan server menangani klien diperlukan ketika adanya dua atau lebih proses pengolahan video yang hendak dilakukan oleh dua atau lebih komputer yang berbeda dan diharuskan mengakses video dari sebuah sumber video atau *browser* yang sama secara bersamaan [8].

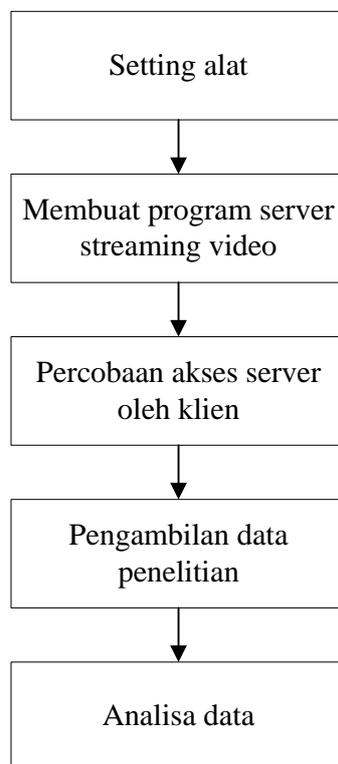
Pada penelitian ini, sistem yang menyerupai *IP Camera* dibentuk menggunakan Odroid XU4 sebagai server dan dua buah *webcam* berukuran 640 x 480 piksel. Pemilihan Odroid XU4 sebagai server dianggap sangat cocok karena memiliki spesifikasi yang mumpuni untuk sebuah SBC [9]. Sedangkan penggunaan dua kamera untuk menggambarkan modifikasi yang dilakukan pengembang pengolah video. Kemudian akan dilakukan analisis mengenai karakteristik SBC sebagai *streaming* video server berdasarkan sistem yang dibuat. Analisis akan mengacu pada parameter *Quality of Service* (QoS) yakni *packet loss* dan *frame rate* yang diterima oleh klien yang mengakses server [10].

2. METODE PENELITIAN

2.1 Analisa Penelitian

Pada penelitian ini system yang akan dirancang dan dibangun merupakan sistem untuk menjadikan *Single Board Computer* sebagai server yang akan melakukan *streaming video* dan

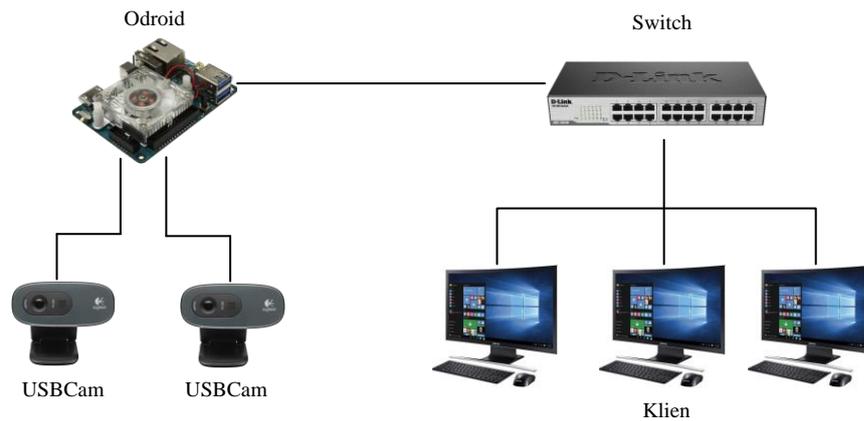
dapat diakses oleh klien yang mengaksesnya. *Interface* yang digunakan pada sistem ini adalah TCP/IP. Beberapa langkah yang dilakukan dalam penelitian ini yaitu, setting alat – alat yang dibutuhkan dalam penelitian baik perangkat lunak maupun perangkat keras. Setelah setting alat dilakukan, langkah selanjutnya adalah membuat program server *streaming* video yang akan diterapkan pada SBC yang digunakan, yakni Odroid XU4. Setelah program selesai dibuat, dilakukan percobaan untuk menguji program. Jika program sudah mampu melakukan *streaming* video, maka langkah selanjutnya adalah pengambilan data. Pengambilan data dilakukan menggunakan beberapa komputer yang berperan sebagai klien dan secara bersamaan mengakses *streaming* video yang disajikan oleh SBC. Lalu lintas data antara klien dan server akan direkam menggunakan *tcpdump*. Pengambilan data selesai ketika server tidak mampu lagi menangani klien yang ditandai dengan *hang* pada server. Setelah data diperoleh, data tersebut akan dianalisa menggunakan software Wireshark. Secara keseluruhan, blok diagram penelitian ini dapat dilihat pada Gambar 1.



Gambar 0 Blok diagram penelitian

2. 2 Rancangan Sistem

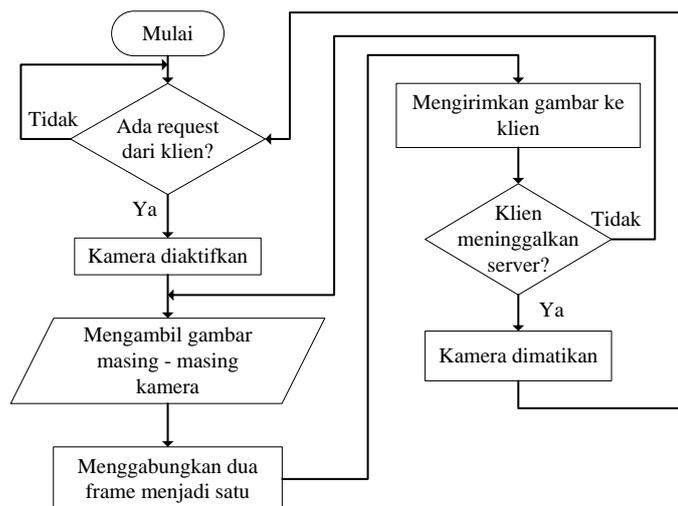
Penelitian ini menggunakan SBC dengan sistem operasi *Ubuntu* 16.04, bahasa pemrograman *python* 3.6.5, *library openCv* 3.4.2, *flask* sebagai *framework* pembuat server, *tcpdump* sebagai perekam lalu lintas data antara klien server dan *Wireshark* sebagai *software* pembaca data yang diperoleh. Perangkat keras yang digunakan berupa Odroid XU4 sebagai server, dua USB kamera untuk menangkap citra, Hub sebagai penghubung server dan klien dan 31 komputer yang tersedia sebagai klien. Kedua kamera dihubungkan ke port USB pada odroid. Odroid dihubungkan ke hub melalui kabel Ethernet. Setelah perangkat keras terpasang, langkah selanjutnya adalah menentukan alamat IP dalam mode *static*. Mode *static* digunakan agar odroid memperoleh alamat IP yang tidak berubah meskipun terjadi perubahan pada pengaturan koneksi. Rancangan perangkat keras dapat dilihat seperti pada Gambar 2.



Gambar 2 Rancangan Perangkat Keras

2.3 Rancangan Program

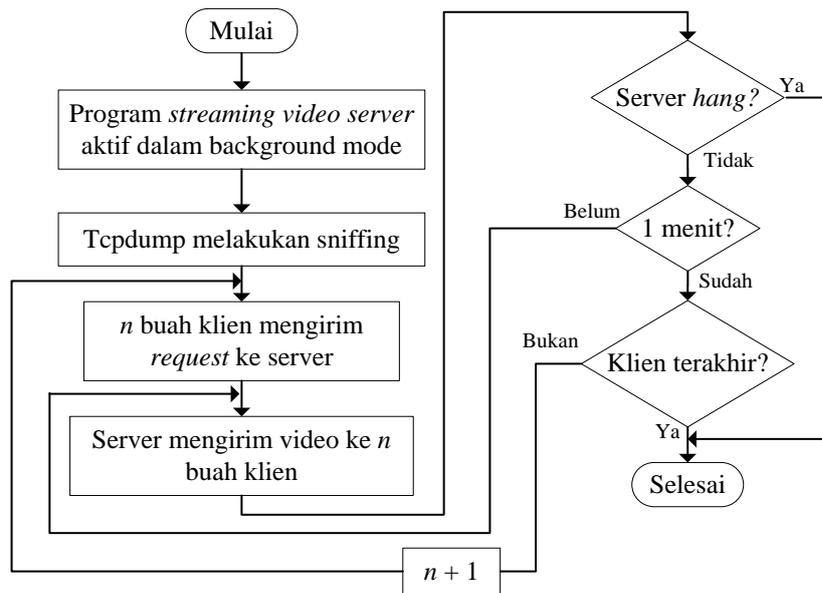
Program *streaming video* yang akan diimplementasikan pada server menggunakan *microframework* flask. Flask dianggap sangat cocok digunakan sebagai framework untuk membuat program webserver karena kemudahannya dan keandalannya untuk diterapkan pada *Single Board Computer*. Flask merupakan framework yang menggunakan bahasa pemrograman python. Gambar 3 adalah diagram alir program yang akan dibuat.



Gambar 3 Diagram alir sistem

Mula – mula, sistem akan mendeteksi adanya *request* klien. Hal ini bertujuan untuk mencegah pemborosan penggunaan daya. Apabila ada *request* dari klien, maka kamera diaktifkan. Dua buah kamera akan mengambil gambar. Gambar dari masing – masing kamera kemudian digabung menjadi satu *frame*. *Frame* tersebut kemudian dikirim ke klien. Proses pengambilan gambar, penggabungan, dan pengiriman frame ke klien akan terus dilakukan selama klien masih terhubung dengan server. Apabila klien memutuskan koneksi atau meninggalkan server, maka kamera akan dimatikan.

Setelah program server dibuat, langkah selanjutnya adalah percobaan akses server oleh satu klien. Langkah ini perlu dilakukan untuk menguji program pada server. Jika server belum dapat melakukan *streaming video*, maka perlu dilakukan perbaikan pada programnya. Namun, apabila program sudah mampu menyajikan gambar ke klien, maka sistem siap untuk melakukan tahap pengambilan data. Pengambilan data dilakukan dengan menggunakan 33 komputer sebagai klien. Gambar 4 merupakan gambaran diagram alir rancangan pengambilan data.



Gambar 4 Diagram Alir Pengambilan Data

Mula – mula, program *streaming video server* dijalankan dalam mode *background* agar odroid dapat menjalankan program lainnya, yakni *tcpdump* untuk melakukan *sniffing* lalu lintas data. Setelah program *streaming video server* dan *tcpdump* dijalankan, langkah selanjutnya adalah klien satu persatu mengakses server, dimulai dengan satu buah klien. Klien mengetikkan alamat IP dan port server pada web browser. Setelah satu menit, dilakukan penambahan satu klien. Klien terus menerus ditambahkan untuk mengetahui seberapa banyak klien yang dapat ditangani oleh server. Apabila server mengalami *hang*, maka pengambilan data dihentikan dan dianggap selesai. Lalu lintas data yang terekam oleh *tcpdump* inilah yang kemudian akan dilakukan analisa lebih lanjut. Pada proses pengambilan data, data yang diperoleh adalah data lalu lintas data atau paket yang terjadi antara server dengan klien. Data ini memiliki ekstensi *.pcap* sehingga dibutuhkan perangkat lunak *wireshark* untuk menampilkan dan mengolah data tersebut. Komponen – komponen yang dianalisa dari data tersebut adalah *frame rate* dan *packet loss*.

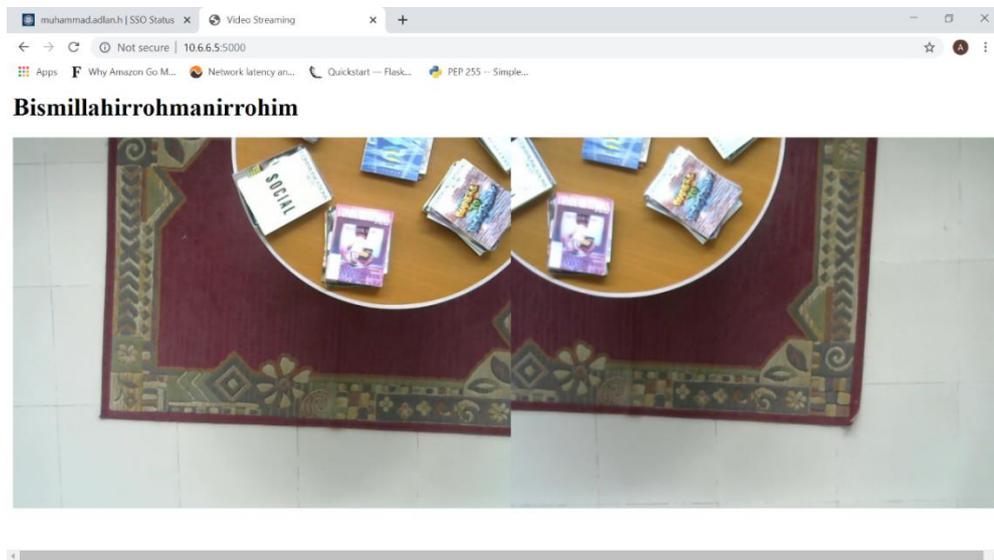
2. 3 Program Streaming Video Server

Program *streaming video server* dibuat dalam *framework flask* dengan bahasa *python*. Skrip program dibuat terlebih dahulu pada komputer. Setelah selesai dibuat, *file* skrip tersebut disalin ke Odroid. *File* skrip yang dibuat terdiri dari beberapa *file* yang disimpan dalam direktori tersusun. Tujuan dibuat susunan direktori adalah untuk memisahkan *file – file* yang tugasnya berbeda agar lebih mudah dalam melakukan *troubleshoot* jika terjadi masalah. *File* tersebut yaitu *file app.py* sebagai skrip utama yang menggunakan *framework flask* yang berfungsi untuk mengaktifkan Odroid sebagai server. Di dalam *file app.py* terdapat perintah untuk memanggil *class* yang berfungsi untuk mengaktifkan kamera. *Class* tersebut berada dalam *file camera.py*. Pengontrolan aktivasi kamera dan penampungan *frame* tangkapan kamera tertuang dalam *file base_camera.py*. Folder template berisi *index.html* yang merupakan laman utama apabila klien mengakses alamat IP server. Kedua ada *file camera.py* yang berisi perintah untuk mengaktifkan kamera, menangkap gambar, dan menggabungkan dua gambar dari masing – masing kamera menjadi sebuah *frame*. Masing – masing kamera menangkap gambar berukuran lebar 640 piksel dan tinggi 480 piksel. Kedua gambar tersebut disatukan menjadi sebuah *frame* berukuran dua kali lebar semula.

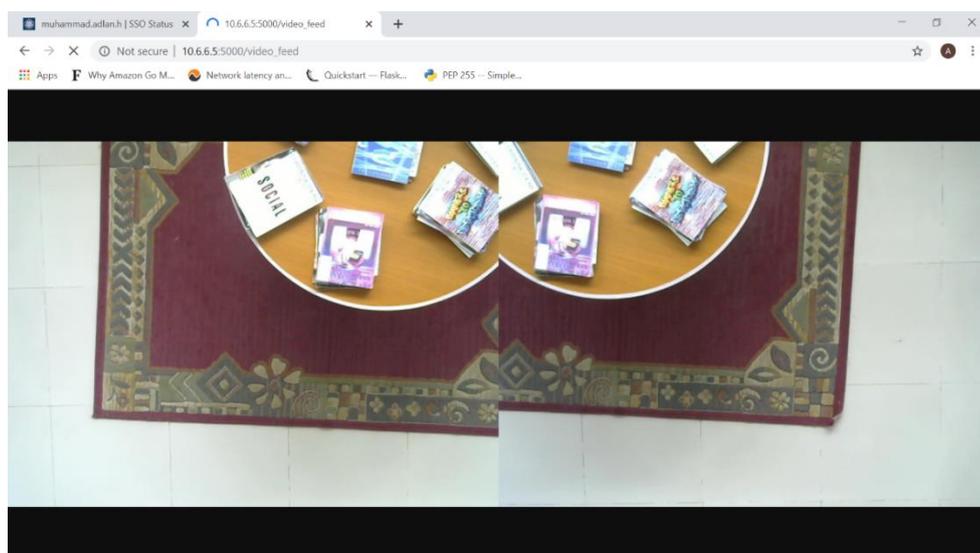
Ketiga ada *file base_camera.py* berisi perintah aktivasi kamera dan pengiriman *frame* ke tiap klien. Terdapat dua *class* dalam *base_camera.py*. *Class* tersebut adalah *class CameraEvent* dan *BaseCamera*. Kedua *class* ini berjalan secara konkuren. Dan terakhir ada skrip *index.html*

membentuk laman utama *streaming video web*. Laman ini terdiri dari judul dan video, dalam penelitian ini video yang dimaksud dibentuk dari *frame* yang terus menerus diperbarui. *Frame* tersebut berekstensi jpeg. Oleh karena itu, di bawah judul dituliskan *img tag*. *Source* *img* tersebut adalah url “*video_feed*”. Maka, walaupun klien mengakses laman utama web *streaming video*, isi proses tetap mengarahkan klien ke *routing “video_feed”* Skrip *index.html* diletakkan di dalam folder template untuk memisahkan skrip *html* dengan *python*.

Tahapan pengambilan data diawali dengan mempersiapkan 31 PC yang berada di Laboratorium Elektronika dan Instrumentasi Fakultas MIPA UGM. PC tersebut harus terhubung dengan server menggunakan jaringan LAN. Semua PC yang akan digunakan sebagai klien harus sudah terinstal web browser di dalamnya. Pada penelitian ini, web browser yang digunakan adalah Google Chrome. Semua PC dalam keadaan membuka halaman Google Chrome dan bersiap untuk mengakses *streaming video website* dengan url *10.6.6.5:5000/video_feed*. Halaman yang akan dibuka adalah halaman web pada *routing video_feed* agar memperoleh halaman yang hanya berisi video seperti ditunjukkan pada Gambar 5 dan Gambar 6.



Gambar 5 Laman Utama Website Streaming Video



Gambar 6 Laman “*video_feed*” pada Website Streaming Video

Langkah kedua adalah mengaktifkan *streaming video website* pada server. Koneksikan sebuah laptop ke Odroid menggunakan *Secure Shell*. Setelah berhasil login, buka terminal. Kemudian aktifkan website dalam mode *background*. Langkah ketiga adalah mengaktifkan *tcpdump* untuk melakukan *sniffing* dan merekam lalu lintas data antara server dengan klien. Perlu diketahui bahwa alasan dilakukannya *sniffing* sebelum aktivitas pengiriman data adalah agar seluruh paket benar – benar terdeteksi. *Tcpdump* harus di-*run* menggunakan superuser. *Sniffing* dilakukan pada alamat IP yang digunakan Odroid yakni 10.6.6.5 dan port yang digunakan sebagai website server yakni port 5000. Paket – paket data yang berhasil di-*sniff* harus disimpan untuk selanjutnya dianalisa. Data tersebut harus disimpan dengan ekstensi *.pcap*. Langkah keempat adalah pemeriksaan alamat IP masing – masing PC yang akan digunakan sebagai klien. Alamat IP klien digunakan untuk mempermudah proses analisa data. Masing – masing PC yang sudah diperiksa alamat IP nya kemudian dicatat. Alamat IP masing – masing PC dapat dilihat pada Tabel 1.

Tabel 1 Alamat IP masing-masing klien

No.	Alamat IP	No	Alamat IP	No.	Alamat IP	No.	Alamat IP
1	10.6.12.20	9	10.6.12.28	17	10.6.12.37	25	10.6.12.46
2	10.6.12.21	10	10.6.12.29	18	10.6.12.38	26	10.6.12.47
3	10.6.12.22	11	10.6.12.30	19	10.6.12.39	27	10.6.12.48
4	10.6.12.23	12	10.6.12.31	20	10.6.12.40	28	10.6.12.49
5	10.6.12.24	13	10.6.12.32	21	10.6.12.41	29	10.6.12.50
6	10.6.12.25	14	10.6.12.34	22	10.6.12.42	30	10.6.12.33
7	10.6.12.26	15	10.6.12.35	23	10.6.12.43	31	10.6.12.120
8	10.6.12.27	16	10.6.12.36	24	10.6.12.45		

Setelah dicatat, selanjutnya adalah pengaksesan klien ke *streaming video website*. Pengaksesan dimulai oleh klien nomor urut 1. Setelah satu menit, klien nomor 2 ikut mengakses website sehingga jumlah klien yang semula satu menjadi dua. Setelah satu menit, klien nomor 3 ikut mengakses website sehingga jumlah klien bertambah menjadi tiga. Penambahan jumlah klien setiap satu menit terus dilakukan. Pengambilan data akan dihentikan apabila server mengalami *hang* atau apabila klien terakhir sudah mengakses server. Dari total 31 klien, server mampu mengirimkan *streaming video* hingga ke klien terakhir.

3. HASIL DAN PEMBAHASAN

3. 1 Pengamatan Frame Rate yang Diterima

Pengambilan data dilakukan hingga klien terakhir, yakni klien ke 31. Semua klien dapat mengakses *streaming video website* yang dibuat dan menampilkan video yang disajikan. Data yang telah diperoleh kemudian dianalisa menggunakan perangkat lunak Wireshark. File yang berkstensi *.pcap* dibuka dengan perangkat Wireshark dan dengan menerapkan filter untuk memperoleh jumlah paket yang merupakan awalan sebuah *frame*. Seperti yang telah diterangkan sebelumnya bahwa tiap *frame* yang dikirim diawali dengan header protokol IP. Salah satu kata yang menjadi header tiap *frame* berbunyi “jpeg”. Selain dengan filter *frame*, filter yang diterapkan adalah waktu. Setiap penambahan klien dilakukan pada waktu yang berbeda – beda, namun dalam setiap satu menit. Setelah difilter, maka diperoleh paket – paket yang berisi header *frame*. Jumlah paket yang diperoleh setiap periode klien kemudian dikalkulasi dengan persamaan

berikut. Hasil akumulasi frame rate kemudian ditampilkan dalam Tabel 2 menggunakan persamaan (1).

$$Frame\ rate = \left(\frac{Jumlah\ paket\ header\ frame\ klien}{waktu} \right) \quad (1)$$

Tabel 2 Data Penelitian *Frame rate*

Jumlah Klien	Frame rate (fps)			Jumlah Klien	Frame rate (fps)		
	Min	Avg	Max		Min	Avg	Max
1	21,97	21,97	21,97	17	21,46	23,00	24,83
2	23,38	23,39	23,4	18	21,00	22,68	24,13
3	21,07	22,69	23,52	19	20,70	22,36	23,86
4	22,83	23,06	23,36	20	19,83	21,48	22,70
5	22,65	23,11	23,53	21	18,13	19,50	20,51
6	22,37	23,08	23,98	22	19,60	21,24	22,32
7	21,59	22,91	23,79	23	16,33	20,86	22,06
8	22,02	23,10	23,81	24	18,86	20,55	21,61
9	22,53	23,40	24,15	25	19,53	21,25	22,56
10	19,27	22,01	23,89	26	15,75	19,77	20,73
11	19,8	21,87	23,50	27	18,08	19,99	20,89
12	19,9	22,29	23,20	28	18,80	20,36	21,15
13	20,83	23,05	24,06	29	18,91	20,18	20,91
14	21,85	23,45	24,81	30	17,21	18,72	19,61
15	21,02	22,23	23,50	31	17,11	18,79	19,50
16	21,52	23,16	25,05				

Hasil penelitian menunjukkan adanya penurunan *frame rate* pada tiap penambahan 10 klien. Walau demikian, *frame rate* tetap mengalami kenaikan dan penurunan secara acak. Frame rate minimum yang diterima salah satu klien adalah 15,75 fps dan rata – rata keseluruhan adalah 21,46 fps. Dapat diperoleh hasil bahwa server mampu mengirimkan video ke 31 klien dengan frame rate yang masih dapat diterima mata manusia, yakni di atas 15 fps.

3. 2 Pengamatan Packet Loss

Setelah data *frame rate* diperoleh, selanjutnya adalah mengolah data untuk memperoleh *packet loss* untuk mengetahui seberapa banyak server mengalami gagal transmisi. File berekstensi .pcap tadi dibuka dengan Wireshark dan dilakukan filter packet loss. Cara memperoleh paket – paket yang merupakan *packet loss* adalah dengan melakukan filter untuk memperoleh paket *retransmission*. Paket tersebut adalah paket yang biasa dikirim oleh server apabila klien merespon bahwa mereka tidak mendapatkan paket yang dikirim server. Tabel 3 adalah hasil *packet loss* di tiap penambahan klien.

Tabel 3 Data Penelitian *Packet Loss*

Jumlah Klien	Packet Loss	Jumlah Klien	Packet Loss	Jumlah Klien	Packet Loss
1	0,18%	12	1,35%	23	1,22%
2	0,40%	13	1,39%	24	1,22%
3	0,89%	14	1,42%	25	1,17%
4	1,01%	15	1,41%	26	1,15%
5	1,16%	16	1,35%	27	1,14%
6	1,33%	17	1,33%	28	1,05%
7	1,29%	18	1,26%	29	0,99%
8	1,41%	19	1,29%	30	1,01%
9	1,42%	20	1,29%	31	0,98%
10	2,02%	21	1,22%		
11	1,63%	22	1,24%		

Dari hasil pengamatan, didapatkan nilai *packet loss* rata – rata adalah 1,20 % yang memiliki indeks 4, dan paling besar adalah ketika klien yang terhubung sebanyak 10, dengan nilai 2,02 % yang memiliki indeks 4. Hal ini menandakan bahwa Odroid masuk ke dalam kategori sangat bagus jika dilihat dari hasil pengamatan *packet loss*.

4. KESIMPULAN

Sebuah sistem *streaming video website* yang menyerupai IP Camera telah berhasil dibuat menggunakan perangkat keras Odroid XU4 dan dua buah kamera webcam dalam jaringan LAN. Server dapat diakses hingga 31 klien. Memberikan hasil pengamatan *frame rate* menunjukkan bahwa Odroid XU4 dapat menangani 31 klien dengan baik. Rata – rata *frame rate* yang diterima klien adalah 21,46 fps dan hasil pengamatan *packet loss* menunjukkan bahwa Odroid XU4 dapat menangani 31 klien dan masuk kategori sangat bagus, yakni dengan rata – rata 1,20%.

5. SARAN

Penelitian lebih lanjut diharapkan melakukan pengambilan data dengan klien sebanyak mungkin untuk mengetahui batas klien yang dapat ditangani server dan melakukan pengolahan citra yang lebih rumit untuk benar – benar menguji server.

DAFTAR PUSTAKA

- [1] L. Bassi, “Industry 4.0: Hope, hype or revolution?,” *RTSI 2017 - IEEE 3rd Int. Forum Res. Technol. Soc. Ind. Conf. Proc.*, 2017.
- [2] K. Wankhede, B. Wukkadada, and V. Nadar, “Just Walk-Out Technology and its Challenges: A Case of Amazon Go,” *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA*

- 2018, no. Icirca, pp. 254–257, 2018.
- [3] A. Polacco, “JBM-2401-04-full,” vol. 24, no. March, pp. 79–92, 2018.
 - [4] V. A. Akpan, “Configuration , Interfacing and Networking of Wireless IP-Based Camera for Real-Time Security Surveillance Systems Design,” vol. 8, no. 2, pp. 107–114, 2015.
 - [5] S. Widiyanto, Y. Karyanti, and D. T. Wardani, “Texture features extraction for indonesian macroscopic and microscopic beef digital images based on gray-level co-occurrence matrix,” *Adv. Sci. Lett.*, vol. 23, no. 3, pp. 2629–2632, 2017.
 - [6] S. Singh, P. Anap, Y. Bhaigade, and P. J. P. Chavan, “IP Camera Video Surveillance using Raspberry Pi,” *Ijarcce*, vol. 4, no. 2, pp. 326–328, 2015.
 - [7] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, “Need for Speed: A Benchmark for Higher Frame Rate Object Tracking,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 1134–1143, 2017.
 - [8] Cisco, “Stream Manager Video Surveillance Solutions Reference Network Design.”
 - [9] R. Wiska *et al.*, “Vehicle traffic monitoring using single camera and embedded systems,” *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACISIS 2016*, pp. 117–122, 2017.
 - [10] A. I. Diwi, R. R. Mangkudjaja, and I. Wahidah, “Analisis Kualitas Layanan Video Live Streaming pada Jaringan Lokal Universitas Telkom,” *Bul. Pos dan Telekomun.*, vol. 12, no. 3, p. 207, 2015.