

Sistem Video Streaming dengan Server Mini Personal Computer (Mini Pc) pada Jaringan Ad-Hoc

Dwindawan Holandrio*¹, Raden Sumiharto², Bakhtiar Alldino Ardi Sumbodo³

¹Prodi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

^{2,3}Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *¹dwindawan.holandrio@gmail.com, ²r_sumiharto@ugm.ac.id, ³alldino.as@gmail.com

Abstrak

Telah dibuat sistem video streaming dengan mengimplementasikan metode socket programming (pemrograman soket) yang menggunakan protokol TCP/IP dan OpenCV (Open Computer Vision) sebagai library pemrograman untuk menangani I/O video. Sistem terdiri dari 1 server yaitu Mini PC Zotac Zbox Nano AD10 dan 1 klien yang saling berhubungan menggunakan Wi-Fi dengan mode akses Ad-Hoc.

Hasil dari penelitian ini adalah sebuah sistem video streaming yang diwujudkan dalam bentuk perangkat lunak yang ditanamkan pada Mini PC sebagai server dan laptop sebagai klien. Video yang dikirimkan dibedakan menjadi 2 yaitu video dengan format RGB (warna) dan format grayscale. Sistem diuji kinerjanya dengan mengamati parameter frame rate video yang ditampilkan oleh klien. Dari pengujian dengan 2 variasi waktu transfer dan resolusi berukuran 640 x 480 pixel didapatkan frame rate terbaik untuk jarak 15 meter video dengan format RGB adalah 0,7 fps, sedangkan untuk video grayscale didapatkan frame rate terbaik 6,56 fps. Kemudian untuk jarak tempuh maksimal yaitu 60 meter didapatkan nilai frame rate 0,04 fps untuk video RGB dan 0,69 fps untuk video grayscale.

Kata kunci—video streaming, sokcet programming, Mini PC, OpenCV

Abstract

A video streaming system has been made that implemented socket programming method which used TCP/IP protokol and OpenCV as a programming library for handling I/O video. The system consists of one server using Mini PC Zotac Zbox Nano AD10 and one client that are connected using the Wi-Fi Ad-Ho wireless mode.

The result of this research is a video streaming system which is realized in a software that been embedded in the Mini PC as the server and in notebook as a client. The system was tested by observing the performance parameters of the video frame rate that displayed by the client. The video can be transmitted into two formats, RGB (color) format and grayscale format. System performance was tested by observing the frame rate of video that being displayed by the client. From testing with two variations of the transfer time, the best frame rate was obtained at 15 meter that is 0,7 fps using RGB format, while for grayscale video frame rate obtained is 6,56 fps. While for a maximum distance of 60 meters, frame rate obtained is 0,04 fps for RGB video and 0,69 for grayscale video

Keywords—video streaming, socket programming, Mini PC, OpenCV

1. PENDAHULUAN

Sekarang ini teknologi sudah berkembang pesat, termasuk didalamnya teknologi *audio* dan *video*. Salah satu pemanfaatan perkembangan teknologi *audio* dan *video* adalah *video streaming*. *Video streaming* dapat diartikan sebagai sebuah teknologi yang memperbolehkan *user* untuk dapat menjalankan file *video* tanpa harus menyimpan ke dalam memori penyimpanan. Tujuannya yaitu *server* menyimpan dan *client* melakukan *request* terhadap *file media stream* yang ingin di jalankan oleh *client*. Didalam proses pengiriman, *file* berlangsung dari sebuah *server* ke *client* melalui jaringan lokal ataupun internet. Dan *file* yang dikirimkan tersebut berupa paket *time stamped* atau yang biasa disebut sebagai *stream media file* [1].

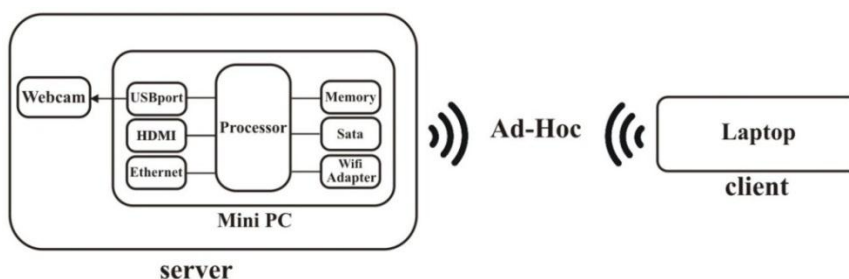
Pemanfaatan *video streaming* pada jaringan nirkabel (*wireless*) untuk keperluan khusus misalnya pada UAV (*Unmanned Aerial Vehicle*) atau pesawat tanpa awak biasanya dimanfaatkan untuk pengintaian atau pemantauan situasi pada suatu area. Hasil rekaman gambar oleh kamera dikirimkan dengan komunikasi *video transmitter* dan diterima oleh *receiver* yang ada pada *ground station* untuk kemudian ditampilkan dalam bentuk *video streaming* secara *real time*. Sehingga *ground station* bisa mendapat informasi tentang daerah yang dipantau.

Akan tetapi saat ini kebanyakan perangkat yang digunakan masih belum bisa melakukan pengolahan data digital. Padahal aplikasi *video streaming* pada UAV sebenarnya bisa dikembangkan lebih jauh lagi, jika sudah menggunakan perangkat yang mampu melakukan pengolahan data digital. Misalnya untuk pemantauan area sawah dengan *image processing* yang bisa mendeteksi petak area persawahan yang berwarna kuning, atau bisa juga dimanfaatkan dalam pencarian korban bencana melalui proses deteksi tubuh manusia. Tentu saja, hal itu tidak bisa dicapai jika pengiriman video masih menggunakan *videotransmitter-receiver* biasa. Oleh karena itu diperlukan adanya sistem *video streaming* pada UAV menggunakan sebuah server yang mampu melakukan pengolahan data video digital. Untuk bisa memenuhi kebutuhan tersebut bisa digunakan *Mini Personal Computer* (Mini PC) sebagai server dari sistem *video streaming*.

2. METODE PENELITIAN

2.1 Rancangan Sistem

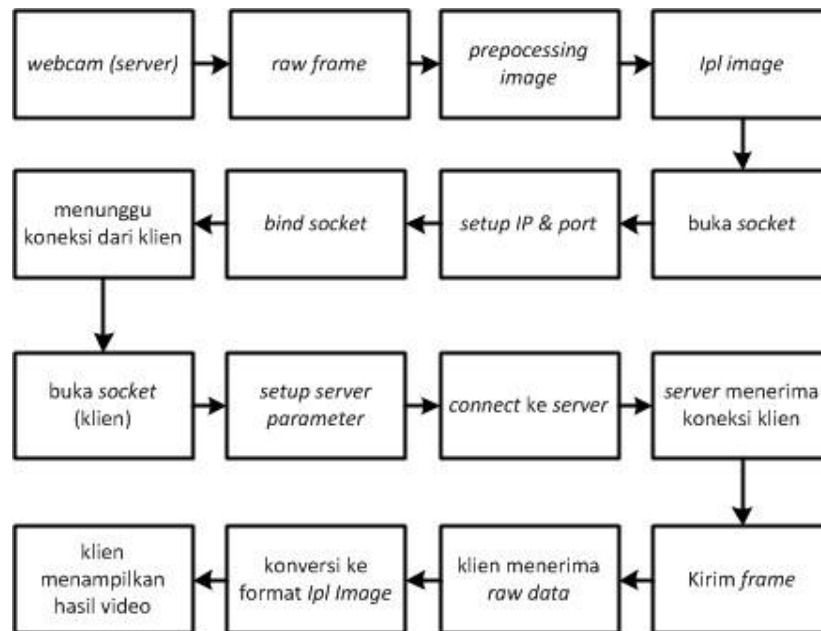
Sistem yang akan dibangun adalah sistem *video streaming* dengan metode *socket programming* (pemrograman socket) yang menggunakan protokol TCP/IP untuk proses interaksi antara server dan klien-nya. Sedangkan untuk penanganan video I/O antara lain mengakses *webcam*, membuat *window* untuk tampilan *video*, mengambil masukan gambar dari hasil *capture* dengan *webcam* digunakan *OpenCV library*. Sistem *video streaming* yang akan dibangun terdiri dari 1 server dan 1 klien yang saling berhubungan menggunakan komunikasi Wi-Fi dengan mode akses Ad-Hoc. Rancangan perangkat keras sistem ditunjukkan pada Gambar 1.



Gambar 1. Rancangan perangkat keras sistem

Server bertugas untuk menangkap gambar dengan USBwebcam dan membuka koneksi untuk klien, agar klien bisa terhubung dengan server. Setelah server menerima koneksi dari klien, hasil perekaman gambar akan langsung dikirimkan ke klien. . Pada sisi klien, data yang

diterima langsung dikonversikan kembali ke format Ipl Image yang dikenali oleh OpenCV agar bisa ditampilkan dalam bentuk *video streaming*. Pada Gambar 2 akan ditunjukkan diagram blok proses secara keseluruhan.



Gambar 2. Blok diagram proses keseluruhan sistem

2. 2. Implementasi Sistem

2. 2.1 Implementasi Perangkat Keras

Implementasi sistem dalam penelitian ini menggunakan perangkat keras yaitu *Webcam* Logitech C170, Mini PC Zotac AD10 AMD E-350 Dual Core 1,6 GHz dan Laptop ASUS A43S Intel Core i3-2310M 2,1 GHz. Implementasi perangkat keras pada server ditunjukkan pada Gambar 3.



Gambar 3. Implementasi perangkat keras pada server

2. 2.2 Implementasi perangkat lunak

Perangkat lunak sistem dibuat untuk membangun server pada Mini PC dan klien pada laptop yang menggunakan sistem operasi Ubuntu. Pembuatan program dilakukan menggunakan antarmuka Geany dengan bahasa pemrograman C. Bahasa C dipilih sebagai *tool* untuk membangun server dan klien dikarenakan bahasa C dalam pemrogramannya yang cukup sederhana dan mudah diimplementasikan dalam pemrograman socket. Pemrograman perangkat

lunak untuk server dan klien ini sekaligus dibuat untuk antar mukanya menggunakan *library* HighGUI dari OpenCV versi 2.3.1. OpenCV *library* dipilih dengan alasan lebih mudah dalam pembuatan antarmuka dan mempunyai fungsi-fungsi yang dibutuhkan sistem.

3. HASIL DAN PEMBAHASAN

3. 1 Pengujian Frame Size

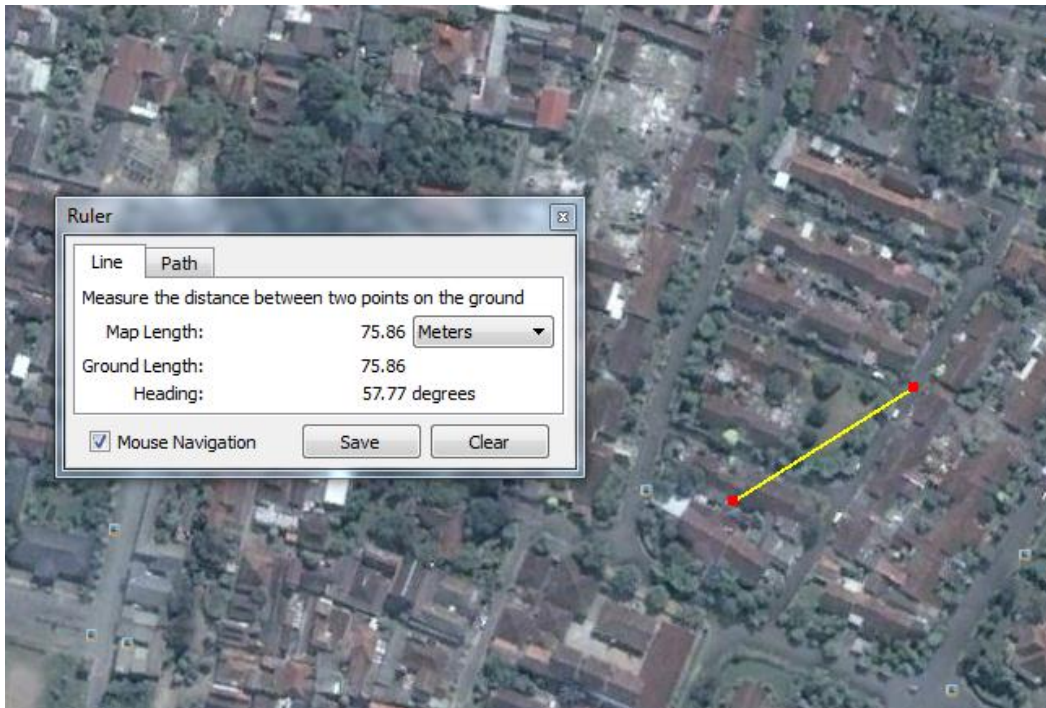
Dalam program yang dirancang, ukuran frame pada klien dapat diubah berdasarkan input yang diketikkan di terminal pada saat program akan dieksekusi. Argumen ketiga pada saat eksekusi program akan menjadi lebar frame dan argumen keempat menjadi tinggi frame. Pada bagian ini akan dilakukan pengujian terhadap klien dengan beberapa variasi ukuran frame dan kemudian diamati hasilnya. Variasi ukuran frame yang diujikan merupakan standar untuk *display video* pada kebanyakan komputer. Hasil dari pengujian ditunjukkan pada Tabel 1.

Tabel 1. Hasil pengujian frame size video RGB dan grayscale

No	Aspect Ratio	Width (pixel)	Height (pixel)	Keterangan
1	4 : 3	640	480	Gambar tertampil dengan baik dan stabil
2	4 : 3	800	600	<i>Error</i> , gambar naik turun dan terbagi menjadi 10 frame
3	17 : 10	1024	600	<i>Error</i> , gambar terbagi menjadi 16 frame
4	4 : 3	1024	768	<i>Error</i> , gambar terbagi menjadi 24 frame
5	4 : 3	1152	864	<i>Error</i> , gambar terbagi menjadi 27 frame
6	16 : 9	1280	720	Gambar tertampil dengan baik, 6 frame
7	5 : 3	1280	768	Gambar tertampil dengan baik tetapi naik turun, 6 frame
8	16 : 10	1280	800	Gambar tertampil dengan baik, naik turun, 6 frame
9	~ 16 : 9	1360	768	<i>Error</i> , naik turun
10	~ 16 : 9	1366	768	<i>Error</i> , naik turun

3. 2 Pengujian Jarak

Pengujian jarak dilakukan untuk mengetahui jarak jangkauan sistem *video streaming* pada jaringan Ad-hoc. Parameter yang diamati adalah *frame rate* yang dihasilkan oleh klien dari variasi jarak antara server dan klien. *Framerate* adalah jumlah bingkai gambar atau *frame* yang ditunjukkan setiap detik dalam membuat gambar bergerak, diwujudkan dalam satuan *frames per second* (fps). Makin tinggi angka fps-nya, semakin halus gambarnya. Pengujian dilakukan dengan menjalankan sistem pada mode akses Ad-hoc di sekitaran daerah UGM. Dalam hal ini, sistem diuji dengan kondisi server terhalang bangunan dan klien bebas bangunan. Lokasi pengujian sistem ditunjukkan pada Gambar 4, sedangkan untuk hasil pengujiannya ditunjukkan pada Tabel 2.



Gambar 4. Lokasi pengujian sistem

Tabel 2. Hasil pengukuran jarak maksimal

No	Jarak (m)	Keterangan
1	15	Klien terhubung server
2	30	Klien terhubung server
3	45	Klien terhubung server
4	60	Klien terhubung server
5	> 60	Klien sudah tidak bisa menangkap sinyal dari server

Dari pengujian dapat dilihat bahwa jarak tempuh maksimal sistem video streaming yang dibuat ketika diuji dengan mode akses Ad-hoc adalah 60 meter. Jika lebih dari 60 meter, klien tidak bisa lagi menangkap sinyal wireless Ad-hoc dari server. Selain variasi jarak, pengujian juga dilakukan dengan dua variasi waktu transfer, yaitu dua menit dan empat menit. Pengujian pertama untuk video format RGB dengan waktu transfer 2 menit didapatkan hasil rata-rata *frame rate* yang ditunjukkan pada Tabel 3, sedangkan pengujian dengan waktu transfer 4 menit hasilnya ditunjukkan pada Tabel 4.

Tabel 3. Hasil pengujian jarak dengan waktu transfer 2 menit untuk video format RGB

No	Jarak (m)	Banyak data	Rata-rata <i>frame rate</i> (fps)
1	15	41	0,63
2	30	35	0,21
3	45	22	0,12
4	60	11	0,07

Tabel 4. Hasil pengujian jarak dengan waktu transfer 4 menit untuk video format RGB

No	Jarak (m)	Banyak data	Rata-rata <i>frame rate</i> (fps)
1	15	120	0,7
2	30	99	0,36
3	45	34	0,12
4	60	15	0,04

Pengujian kedua dilakukan dengan mengubah format video yang dikirimkan dari RGB menjadi format *grayscale*. Pengujian dengan waktu transfer 2 menit hasil *frame rate*-nya ditunjukkan pada Tabel 5 dan untuk waktu transfer 4 menit hasilnya ditunjukkan pada Tabel 6.

Tabel 5. Hasil pengujian jarak dengan waktu transfer 4 menit untuk video format grayscale

No	Jarak (m)	Banyak data	Rata-rata <i>frame rate</i> (fps)
1	15	507	6,50
2	30	343	3,24
3	45	47	0,39
4	60	24	0,17

Tabel 5. Hasil pengujian jarak dengan waktu transfer 4 menit untuk video format grayscale

No	Jarak (m)	Banyak data	Rata-rata <i>frame rate</i> (fps)
1	15	511	6,56
2	30	511	3,49
3	45	166	0,71
4	60	175	0,69

Dari hasil pengujian dapat disimpulkan perbedaan format video yang dikirimkan mempengaruhi nilai *frame rate* yang dihasilkan. *Frame rate* untuk video RGB lebih kecil daripada video *grayscale*. Hal tersebut dikarenakan format *grayscale* yang hanya memiliki 1 *channel*, sehingga waktu yang dibutuhkan program untuk menerima raw data dan mengkonversinya ke dalam format Ipl Image tentu akan lebih cepat jika dibandingkan format RGB yang memiliki 3 *channel* (R, G dan B).

Tabel hasil pengujian juga menunjukkan bahwa jarak dan waktu transfer berpengaruh terhadap *frame rate*. Semakin jauh jarak jangkauan, maka *frame rate* akan semakin kecil. Hal tersebut terjadi karena semakin jauh jarak penerima (klien) terhadap pemancar (server), level daya pemancar (server) akan semakin melemah. Penurunan kuat sinyal tersebut mengakibatkan kinerja jaringan menurun [2]. Selain itu kondisi server yang terhalang dinding bangunan juga mempengaruhi hasil pengujian.

3. 3 Pengujian Video bitrate

Dalam sistem ini perlu diketahui berapa jumlah data yang dikirim dari server ke klien, maka dalam pengujian kali ini akan dihitung berapa *video bitrate* untuk setiap variasi *frame size* yang telah diujikan. *Video bitrate* dihitung dengan rumus seperti yang ditunjukkan pada persamaan 1. Sedangkan hasil pengujinannya dapat dilihat pada Tabel 7 dan Tabel 8.

$$\text{Video bitrate} = \text{resolusi} \times \text{kedalaman warna (depth)} \times \text{frame rate} \quad (1)$$

Tabel 7. Hasil pengujian video bitrate format RGB

No	Aspect Ratio	Width (pixel)	Height (pixel)	Rata-rata <i>frame rate</i> (fps)	<i>Video bitrate</i>
1	4 : 3	640	480	2,4	17, 6 Mbit/s
2	4 : 3	800	600	1,4	16,13 Mbit/s
3	17 : 10	1024	600	1	14,7 Mbit/s
4	4 : 3	1024	768	0,8	15,1 Mbit/s
5	4 : 3	1152	864	0,68	16,2 Mbit/s
6	16 : 9	1280	720	0,68	15,04 Mbit/s
7	5 : 3	1280	768	0,65	16,04 Mbit/s
8	16 : 10	1280	800	0,6	14,7 Mbit/s
9	~ 16 : 9	1360	768	0,6	15,04 Mbit/s
10	~ 16 : 9	1366	768	0,58	14,6 Mbit/s

Tabel 8. Hasil pengujian video bitrate format grayscale

No	Aspect Ratio	Width (pixel)	Height (pixel)	Rata-rata frame rate (fps)	Video bitrate
1	4 : 3	640	480	4,3	10,6 Mbit/s
2	4 : 3	800	600	3,6	13,8 Mbit/s
3	17 : 10	1024	600	3,1	15,2 Mbit/s
4	4 : 3	1024	768	2,4	15,1 Mbit/s
5	4 : 3	1152	864	2,09	16,6 Mbit/s
6	16 : 9	1280	720	2,01	14,8 Mbit/s
7	5 : 3	1280	768	1,99	15,6 Mbit/s
8	16 : 10	1280	800	1,96	16,1 Mbit/s
9	~ 16 : 9	1360	768	1,78	14,9 Mbit/s
10	~ 16 : 9	1366	768	1,74	14,6 Mbit/s

Dari kedua tabel di atas dapat dilihat, perbedaan nilai *frame rate* dipengaruhi oleh perbedaan resolusi video. Nilai *frame rate* semakin kecil setiap bertambahnya resolusi video. Dapat disimpulkan hal ini dikarenakan video dengan resolusi yang kecil membutuhkan total paket stream yang lebih sedikit jika dibandingkan dengan video dengan resolusi yang besar [3]. Sehingga pada resolusi besar, klien akan membutuhkan waktu yang lebih lama dalam menerima dan mengolah data dari server.

3. Pengujian Throughput

Untuk mengukur kualitas layanan sebuah jaringan dapat dilakukan dengan menghitung *throughput* dari jaringan tersebut. Dengan pengujian ini akan dihitung *throughput* untuk menunjukkan kemampuan sebenarnya jaringan Ad-Hoc dalam melakukan pengiriman data. Perhitungan nilai *throughput* ditunjukkan pada persamaan 2 dan hasil perhitungan *throughput* ditunjukkan pada Tabel 9.

$$\text{Throughput} = \text{Paket yang diterima} / \text{waktu transfer} \quad (2)$$

Tabel 9. Hasil pengujian Throughput

Resolusi (pixel)	Format video	Jarak (meter)	Frame rate (fps)	Video bitrate	Throughput
640 x 480	RGB	15	0,7	5,16 Mbp/s	1,29 Mbit/s
		60	0,04	0,29 Mbp/s	0,07 Mbit/s
	Grayscale	15	6,56	16,1 Mbp/s	4,03 Mbit/s
		60	0,69	1,7 Mbp/s	0,43 Mbit/s

Nilai *throughput* tersebut menunjukkan bahwa sebenarnya dari *bandwidth* yang tersedia yaitu 54 Mbit/s, ternyata maksimal yang digunakan oleh sistem hanya sebesar 4,03 Mbit/s. Selain itu dapat diamati bahwa *bitrate* berbanding lurus dengan *throughput* yang dihasilkan. Semakin besar *throughput*-nya, semakin baik pula kemampuan jaringan dalam mentransmisikan data [4].

3. 4 Pengujian Text mode server

Pada bagian ini akan dibahas pengujian sistem yang dijalankan dengan text mode server. Sistem operasi ubuntu memiliki fitur untuk mengubah mode grafis ke dalam mode text. Artinya, GUI (mode grafis) pada sisi server akan dihilangkan sehingga sistem operasi Ubuntu hanya akan memiliki tampilan hitam putih berbasis text atau text mode. Sistem akan dijalankan pada resolusi 640 x 480 dan hasil frame ratenya akan diamati dan dibandingkan dengan nilai frame rate yang dihasilkan ketika server dijalankan pada GUI mode. Hasil dari pengujian ditunjukkan pada Tabel 10.

Tabel 10. Hasil pengujian GUI vs text mode

Resolusi (pixel)	Server mode	Format video	Rata-rata <i>frame rate</i> (fps)
640 x 480	GUI	RGB	2,4
		Grayscale	6,7
	Text	RGB	2,36
		Grayscale	6,6

4. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

- Jarak maksimal yang bisa ditempuh oleh sistem pada mode jaringan Ad-Hoc dengan kondisi server terdapat halangan adalah 60 meter.
- Semakin jauh jarak jangkauan, nilai *frame rate* akan menurun. Hal tersebut terjadi karena penurunan kuat sinyal server yang mengakibatkan kinerja jaringan menurun.
- Nilai *frame rate* dapat dipengaruhi oleh resolusi dan format dari video tersebut (RGB atau grayscale). Video dengan resolusi yang kecil akan menghasilkan nilai *frame rate* yang kecil, dan video dengan format *grayscale* akan menghasilkan nilai *frame rate* yang lebih besar.
- Nilai *frame rate* terbaik yaitu 6,56 fps, didapat saat pengujian video format *grayscale* dengan resolusi 640 x 480 pixel pada jarak 15 meter.
- Nilai *throughput* terbesar yang didapat dari jaringan adalah 4,03 Mbit/s.
- Nilai *frame rate* yang dihasilkan pada text mode dan GUI mode tidak memiliki perbedaan yang signifikan. Artinya, perangkat lunak yang dibuat dengan OpenCV library tidak memakai banyak *resource* dari komputer.

5. SARAN

- Sistem ini masih sebatas diujikan dengan mode akses Ad-Hoc. Diharapkan penelitian selanjutnya dapat digunakan *access point* untuk menambah jarak jangkauan pengujian.
- Untuk meningkatkan performa jaringan diperlukan adanya QoS (*Quality of Service*) pada sisi server.
- Jika selanjutnya masih mengembangkan sistem video streaming dengan OpenCV *library*, maka diharapkan dapat menerapkan algoritma untuk kompresi video. Selain itu juga terdapat peluang lebih lanjut untuk mencari dan menggunakan *library* yang mendukung untuk *audio processing* dalam video streaming.
- Video dari hasil *streaming* sebaiknya dapat langsung disimpan di media penyimpanan.

DAFTAR PUSTAKA

- [1] Jayusman, 2009, Video streaming, <http://jayusman12.wordpress.com/?s=video+treaming&searchbutton=go!>, 14 Oktober 2012, diakses tanggal 14 Maret 2012.
- [2] Sari, I. P., 2010, Optimasi Penataan Sistem Wi-Fi di PENS-ITS Dengan Menggunakan Metode Monte Carlo, Seminar Proyek Akhir Jurusan Telekomunikasi, Politeknik Elektronika Negeri Surabaya, Surabaya.
- [3] Pratama, H. S, 2011, Rancang Bangun Terminal Access Multi-Source Streaming Server Dengan Java Media Framework, Teknik Telekomunikasi, Institut Teknologi Sepuluh November.
- [4] Abror, A. A, 2011, Rancang Bangun dan Analisa QOS Audio dan Video Streaming Pada Jaringan MPLS VPN.