

Sistem Pemantauan Ruangan Dengan Server Raspberry Pi

Egrit Nurcahyo Wijatsongko*¹, Agfianto Eko Putra², Bambang Nurcahyo Prastowo³

¹ Prodi Elektronika dan Instrumentasi, FMIPA, UGM Yogyakarta

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

e-mail: *¹egrit.nur@gmail.com, ²agfi68@gmail.com, ³prastowo@ugm.ac.id

Abstrak

Pemantauan secara terus menerus akan membebani storage pada server, karena ukuran berkas video hasil rekaman akan sangat besar. Salah satu cara untuk mengatasi masalah tersebut adalah dengan membuat agar webcam hanya merekam pada saat-saat dibutuhkan saja. Contoh 'saat yang dibutuhkan' ini adalah saat seseorang memasuki ruangan. Karena itu dibutuhkan sebuah sistem pemantauan yang dapat melakukan deteksi gerakan pada suatu ruangan.

Dalam penelitian ini, masalah tersebut coba diselesaikan dengan merancang bangun sistem pemantau ruangan berbasis Komputer mini (Raspberry Pi). Sistem pemantauan ini dibuat menggunakan program motion. Program motion menangani deteksi gerak dan streaming, selain itu sistem juga dibuat agar dapat mengirimkan notifikasi saat terjadi gerakan melalui email. Untuk alasan keamanan dan backup data, sistem juga akan mengunggah video hasil rekaman ke Google Drive. Sistem terdiri dari 1 server dan 1 klien yang saling berhubungan menggunakan jaringan WLAN.

Sistem diuji kinerjanya dengan mengamati parameter frame rate video hasil rekaman. Sistem juga diuji kemampuan deteksi geraknya pada intensitas cahaya yang berbeda-beda. Dari pengujian didapatkan bahwa kondisi terbaik untuk sistem adalah pada framesize 320x240 dengan intensitas cahaya lebih dari 8 lux.

Kata kunci— Motion, deteksi gerak, Raspberry Pi, email, Google Drive

Abstract

Continuous record will overload storage, due to big size recorded video files. One of the solution to this problem is to make system only record at necessary time. The example of the "necessary time" is when someone enter the room. Because of that, surveillance system which can detect motion in the room is needed.

In this research, mentioned problem was tried to be solved by design and build a mini computer (Raspberry Pi) based room surveillance system. This surveillance system is made with Motion program. Motion program handle motion detection and streaming. Besides that, system also made so it can send notification via email, when motion is detected. For security and back up data reasons, system will also upload recorded video to Google Drive. System consists of one server and one client that are connected using WLAN network.

The System performance was tested by observing recorded video frame rates as parameter. The motion detection of the system was also tested on various light intensity. From the test, got that the best condition for the system is at frame size 320x240 with light intensity more than 8 lux.

Keywords— Motion, motion detection, Raspberry Pi, email, Google Drive

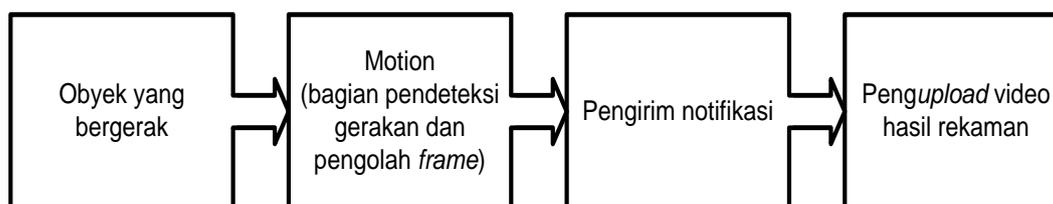
1. PENDAHULUAN

Pemantauan secara terus menerus seperti pada CCTV dipusat perbelanjaan atau perkantoran, akan membebani *storage* pada server, karena ukuran file video hasil rekaman akan sangat besar. Salah satu cara untuk mengatasi masalah tersebut dengan membuat agar webcam hanya merekam pada saat-saat yang dibutuhkan saja. Salah satu contoh 'saat yang dibutuhkan' ini adalah saat seseorang memasuki ruangan. Karena itu dibutuhkan sebuah sistem pemantauan yang dapat melakukan deteksi gerakan pada suatu ruangan.

Oleh karena itu untuk membuat sistem pengawasan tersebut dibutuhkan sebuah server yang dapat melakukan pengolahan data video digital sekaligus mengirimkan peringatan ke email pemilik. Untuk memenuhi kebutuhan tersebut bisa digunakan *Single Board Computer* (SBC) sebagai server dari sistem pemantauan tersebut. Salah satu contoh SBC yang dapat digunakan untuk membangun sistem pemantauan ini adalah Raspberry pi. Penelitian ini berbeda dengan penelitian sebelumnya dari sisi *hardware*nya, penelitian sebelumnya yang sejenis menggunakan PC sebagai server sistem pemantauannya[1]. Selain PC penelitian lain menggunakan mini PC, sebagai server, salah satu contoh mini pc yang digunakan adalah Mini PC Zotac Zbox Nano AD10[2].

2. METODE PENELITIAN

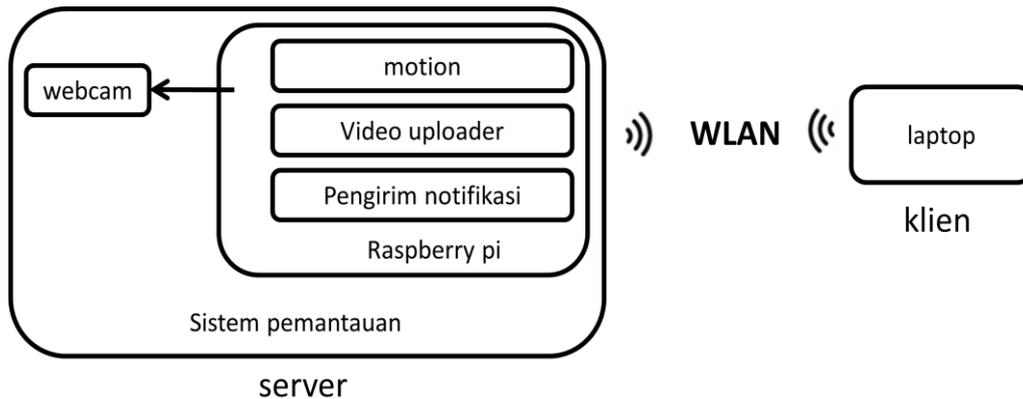
2.1 Rancangan Penelitian



Gambar 1 Diagram blok sistem untuk pendeteksian gerak

Sinyal video akan diproses di Raspberry Pi, gerakan pada video dideteksi dengan program bernama Motion[3], Motion mendeteksi gerakan dengan memantau sinyal video dari satu atau lebih kamera dan dapat mengetahui jika suatu bagian dari gambar telah berubah. Motion juga digunakan untuk menampilkan *video streaming* di *web browser*, atau *media player* yang mendukung *video streaming*. Seperti yang ditunjukkan pada Gambar 1, setelah terdeteksi gerakan maka sistem akan mengambil citra dan mengirimnya bersama notifikasi ke *email* pengguna. Lalu sistem merekam kondisi ruangan hingga tidak terdeteksi gerakan, setelah selesai merekam maka sistem akan mengunggah video tersebut ke Google Drive agar dapat diakses kembali sewaktu-waktu. Setelah selesai mengunggah, maka file video dalam *storage* Raspberry Pi akan segera dihapus agar tidak menumpuk di *storage*.

Perangkat keras yang dipakai dalam sistem ini berupa webcam, dan *single board computer* Raspberry Pi sebagai server, serta 1 unit PC sebagai klien. Webcam yang digunakan adalah Logitech C170 yang memiliki kualitas video maksimal 30 fps. *Single board computer* yang digunakan adalah Raspberry pi model B rev 1, komunikasi antara Raspberry pi (server) ke unit PC (klien) menggunakan WLAN. Webcam menangkap Gambar yang akan diproses oleh motion, dan dikirim ke klien melalui Wi-Fi dongle yang sudah menghubungkan Raspberry Pi dengan klien. Skema Rancangan perangkat keras tersebut ditunjukkan pada Gambar 2.



Gambar 2 Rancangan perangkat keras secara keseluruhan

2. 2 Implementasi Software

Rancangan perangkat lunak dalam sistem ini ada dua bagian, yang pertama bagian sistem monitoring yang terdapat di dalam server Rapsberry Pi, disini terjadi pengolahan sinyal video untuk mendeteksi gerakan, juga pemberian notifikasi ke pengguna. Yang kedua, bagian video streaming, ini adalah bagian dimana dilakukan pengolahan sinyal yang dikirim oleh server agar bisa ditampilkan di *web browser*, keduanya tergabung dalam satu sistem. Salah satu sistem operasi yang dapat digunakan di raspberry pi adalah sistem operasi Occidentalis 2.0 yang digunakan dalam penelitian ini. Occidentalis 2.0 berbasis Linux debian, Occidentalis sendiri merupakan modifikasi dari sistem operasi Raspbian, Linux Debian yang ditujukan untuk Raspberry pi, modifikasi dilakukan sehingga Occidentalis menjadi versi ringan dari Raspbian.

2. 2.1 Pendeteksian Gerakan Pada Motion

Gerakan pada video dideteksi oleh Motion dengan membandingkan *frame* acuan dan *frame* baru, jika terjadi perubahan yang melebihi *threshold* yang ditentukan, maka dapat disimpulkan terdapat gerakan yang tertangkap video. Secara *default*, *threshold* yang digunakan Motion adalah 1500 pixel, artinya perubahan pixel yang terjadi harus melebihi 1500 pixel agar dapat dideteksi sebagai gerakan. Pengiriman notifikasi dilakukan oleh program lain yang eksekusi nya dikendalikan oleh Motion, misalnya saat Motion mendeteksi gerakan maka motion juga akan memberi perintah kepada program lain untuk mengirimkan notifikasi, saat sudah selesai merekam, program lain akan tereksekusi untuk mengunggah video tersebut ke Google drive.

2. 2. 2 Video streaming

Untuk menampilkan *video streaming* pada *web browser*, digunakan program Motion, Motion menggunakan pemrograman berbasis protokol web untuk melakukan *video streaming*. *Video stream* yang dihasilkan Motion berformat mjpeg, atau motion jpeg, artinya *video stream* yang dihasilkan Motion sebenarnya adalah citra JPG yang di *stream* melalui protocol HTTP[4]. Untuk menampilkan *video streaming* pengguna cukup mengubah konfigurasi yang sudah disiapkan Motion. Pengguna cukup mengeset port yang akan digunakan untuk menampilkan *video streaming*, setelah itu tinggal menuliskannya pada *web browser*. Alamat web yang dituliskan berupa port yang sudah dialokasikan untuk webcam, contoh <http://192.168.0.110:8081> dimana 192.168.0.110 adalah alamat ip dari server (Raspberry pi) sedangkan 8081 adalah adalah port yang sudah kita sediakan untuk webcam.

2. 2.3 Implementasi Pemberi Notifikasi

Notifikasi diberikan oleh sistem ke pengguna atau klien melalui *email*, *email* yang digunakan adalah layanan *email* milik Google yaitu Gmail. Notifikasi yang dikirimkan berupa pemberitahuan bahwa terdeteksi gerakan di tempat yang diawasi, disertai dengan citra yang

ditangkap saat terdeteksi gerakan, citra ini dikirimkan sebagai lampiran pada *email* notifikasi, namun citra tersebut sebelumnya harus diubah ke format base 64 agar dapat dikirimkan sebagai *attachment* melalui *email*. Meng-*encode* ke base64 berarti mengubah file binary menjadi ASCII, sehingga dapat dibaca oleh *email*. *Email* dikirimkan melalui protokol SMTP[5].

2. 2.4 Implementasi Pengupload video

Saat terdeteksi gerakan motion akan mulai merekam hingga tidak terdeteksi gerakan selama waktu yang ditentukan. Hasil rekaman tersebut akan diupload ke Google Drive oleh sistem. Hal ini dilakukan dengan alasan keamanan dan *backup* data, semisal ruangan yang diawasi dimasuki oleh pencuri, kemudian pencuri menyadari dirinya telah direkam dan merusak sistem. Pengguna masih memiliki bukti berupa video yang tersimpan di Google drive. Penguploadan video dilakukan dengan API (*Application programming interface*) yang sudah disediakan Google untuk Google Drive.

2.3 Implementasi Hardware

Raspberry PI sebagai server bertugas untuk merekam gambar dengan *webcam*, untuk kemudian dikirimkan dengan komunikasi WLAN sebagai *video streaming* ke laptop klien dengan bantuan program Motion yang sudah diinstallkan di server. Raspberry pi dan laptop klien sama-sama menggunakan *wireless adapter* yang terintegrasi 802.11n. Pada Gambar 3 diperlihatkan gambar keseluruhan perangkat keras untuk server.



Gambar 3 Perangkat keras untuk server

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Sistem Secara Keseluruhan

Pengujian sistem ini dilakukan untuk menjawab salah satu tujuan penelitian yaitu, Merancang dan membuat sistem pemantauan menggunakan server Raspberry Pi yang dapat secara otomatis merekam dan memberi peringatan melalui *email* jika terdeteksi gerakan di tempat yang dipantau. Keberhasilan pengujian sistem secara keseluruhan menunjukkan bahwa Raspberry Pi dapat digunakan sebagai server sistem pemantauan.

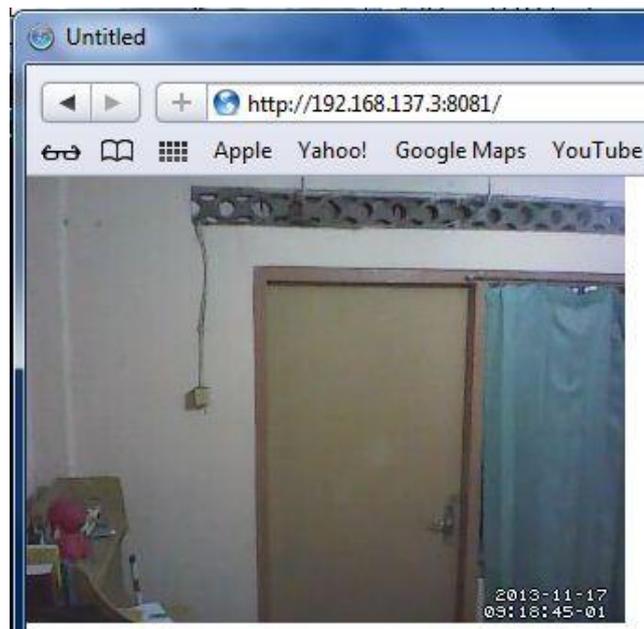
Pertama-tama sistem harus dapat berjalan begitu Raspberry Pi dinyalakan, hal ini ditujukan untuk memudahkan pengguna dalam mengoperasikan sistem. Untuk mengeksekusi

motion saat startup, telah dibuat sebuah program yang akan berjalan saat Raspberry Pi start up, yaitu motionboot. Program ini akan mengeksekusi motion, sehingga secara tidak langsung motion juga akan berjalan begitu Raspberry Pi start up. Selain mengeksekusi motion, motionboot juga akan menghapus semua file berekstensi “.jpg” di direktori /etc/motion, direktori tempat menyimpan citra dan video hasil rekaman motion. Pada saat terdeteksi gerakan, motion akan mulai merekam dan semua *frame* yang terekam disimpan sebagai citra berekstensi “.jpg” di folder tersebut, sehingga setiap kali perekaman disimpan pula banyak citra yang bila dibiarkan lama kelamaan akan memenuhi *storage* dari raspberry pi, karena itu semua file citra tersebut perlu dihapus setiap kali *start up*. Untuk file video, sengaja tidak dihapus karena dapat dihapus setelah di *upload*, selain itu juga untuk menanggulangi saat jaringan internet mati, sehingga walaupun tidak berhasil di *upload*, file video masih tersisa di *storage*. Potongan program pada motionboot yang dieksekusi saat *booting* dapat dilihat pada Gambar 4.

```
case "$1" in
start)
echo "Starting motion"
#perintah yang dijalankan saat startup
sudo rm /etc/motion/*.jpg
sudo motion
;;
```

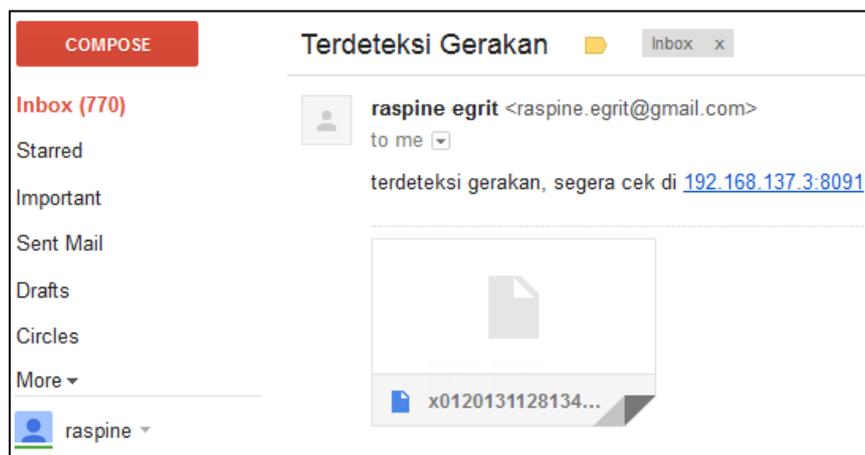
Gambar 4 Potongan program yang dieksekusi saat booting.

Untuk mengecek keberhasilan program motionboot, tinggal mengecek video *streaming* melalui *Web browser*, bila video *streaming* tertampil, berarti program motionboot berjalan dengan baik. Untuk melihat *stream* dari motion, masukkan alamat 192.168.137.3:8081 pada *browser* disisi klien, maka akan tertampil *stream* seperti pada Gambar 5. *Web browser* yang disarankan adalah *Web browser* Apple Safari, karena Apple Safari dapat menampilkan video *streaming* dengan baik. *Web browser* Mozilla Firefox, juga dapat menampilkan video *streaming* di awal namun lama kelamaan *browser* akan *error*, sedangkan update terbaru dari Google Chrome tidak dapat menampilkan video *streaming* sama sekali.



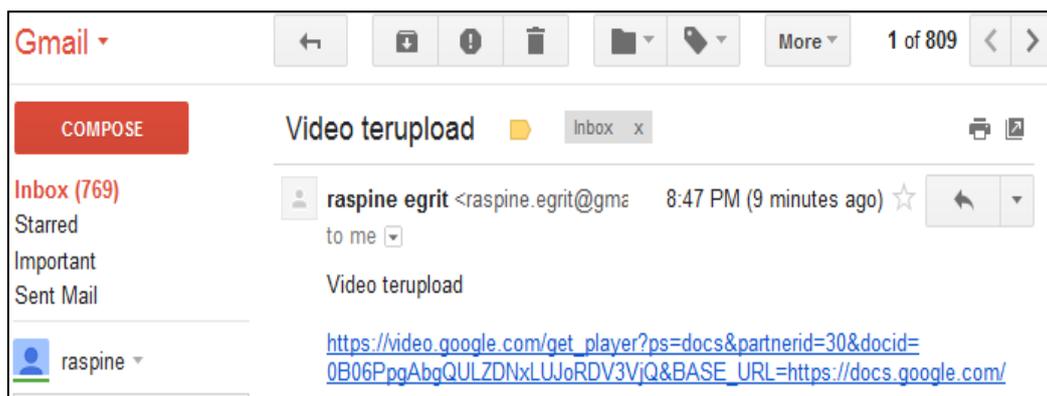
Gambar 5 Tampilan *streaming* pada *browser* Apple Safari

Langkah selanjutnya adalah mengecek pengirim notifikasi, notifikasi akan dikirimkan sesaat setelah terdeteksi gerakan, notifikasi dikirimkan ke *email* pengguna, memberitahukan bahwa telah terjadi gerakan disertai dengan link untuk video *streaming* dan *Attachment* berupa citra yang diambil saat terdeteksi gerakan. Isi dari *email* notifikasi terlihat pada Gambar 6. pada gambar tersebut juga terlihat bahwa *email* tersebut membawa *Attachment*. *Attachment* tersebut berisi citra pada saat terdeteksi gerakan, namun sayangnya citra tersebut tidak dapat dilihat secara *preview*, jadi pengguna harus *download* citra tersebut agar dapat melihat isinya.

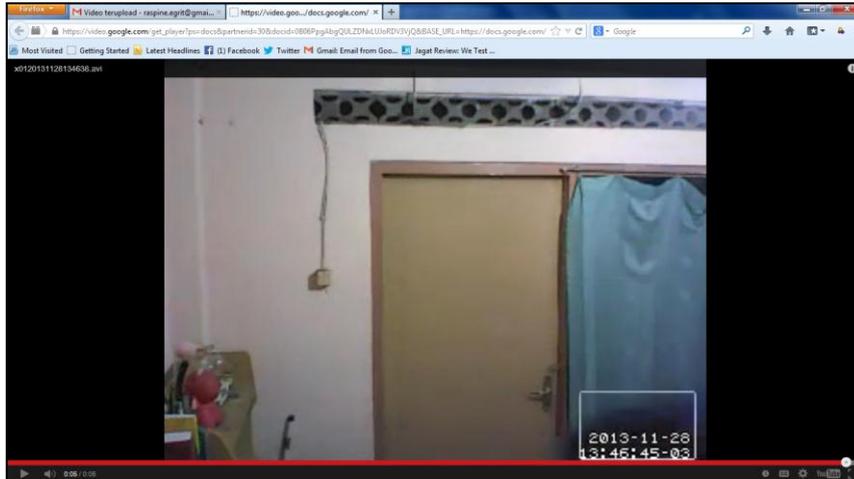


Gambar 6 Tampilan isi dari notifikasi terdeteksi gerakan

Langkah berikutnya adalah mengecek *upload* video hasil rekaman motion saat terdeteksi gerakan. Video hasil rekaman tersebut akan *upload* ke folder motion yang terdapat di Google Drive, setelahnya sistem akan dapat mengirim *email* ke pengguna untuk memberitahukan bahwa video telah selesai *upload*. Isi dari *email* pemberitahuan bahwa video telah *upload* dapat dilihat pada Gambar 7. Pada gambar tersebut terlihat juga *link* untuk video yang baru saja *upload*, jika *link* tersebut diklik, *browser* akan langsung menampilkan video yang bersangkutan, seperti yang terlihat pada Gambar 8. Dari pengujian ini dapat disimpulkan bahwa Raspberry Pi dapat digunakan sebagai server pada sistem pemantauan.



Gambar 7 Tampilan isi *email* pemberitahuan bahwa video telah *upload*



Gambar 8 Tampilan pada browser saat link pada email diklik

3.2 Pengujian Frame size

Komponen *frame size* (resolusi) menunjukkan besar bidang tampilan video yang ditampilkan pada layar. Pengujian *frame size* bertujuan untuk mengetahui ukuran *frame size* yang dapat diterima oleh sistem, dan menghasilkan video yang baik. Ukuran *frame size* sangat bergantung pada *software* maupun *hardware* yang digunakan oleh sistem. Hasil pengujian *frame size* ini dapat dilihat pada Tabel 1.

Tabel 1 Hasil pengujian *frame size*

no	<i>frame size</i>	keterangan
1	320x240	<i>frame size</i> diterima dan berjalan dengan baik
2	400x300	error, tinggi <i>frame</i> (300) bukan modulo 16
3	640x480	<i>frame size</i> diterima dan berjalan dengan baik
4	640x352	<i>frame size</i> diterima dan berjalan dengan baik
5	800x600	error, tinggi <i>frame</i> (600) bukan modulo 16
6	960x720	resolusi dikonvert kembali ke 640x480
7	1280x960	resolusi dikonvert kembali ke 640x480
8	1024x600	error, tinggi <i>frame</i> (600) bukan modulo 16

Dari hasil pengujian terlihat bahwa tidak semua *frame size* dapat diterima oleh sistem, *frame size* yang dapat diterima antara lain 320x240, 640x480 dan 640x352, ketiga *frame size* inilah yang nantinya akan dipakai dan diujikan. Sedangkan *frame size* seperti 400x300, 800x600 dan 1024x600, tidak dapat diterima oleh sistem karena komponen tinggi *frame* -nya bukan modulo 16. Ini sesuai dengan ketentuan dari motion yang mengharuskan komponen tinggi dan lebar dari sebuah *frame* adalah bilangan dengan kelipatan 16, agar sesuai dengan algoritma pendeteksian gerak. Sedangkan *frame size* yang lain seperti 960x720 dan 1280x960 tidak dapat diterima dan akhirnya dikonvert kembali ke ukuran 640x480, ini menunjukkan bahwa nilai *frame size* maksimal yang dapat diterima oleh sistem adalah 640x480. Keterbatasan ini mungkin disebabkan oleh *driver* atau *hardware*nya, dalam hal ini Raspberry Pi. Nilai 640x480 sendiri ditentukan oleh *driver*, Motion memberikan perintah kepada *driver* untuk mengecek nilai *frame size* yang dimasukkan oleh user dengan perintah `xioctl(s->fd,`

VIDIOC_TRY_FMT, &s->fmt) jika nilai *frame size* di dukung oleh *driver* maka nilai *frame size* tersebut akan digunakan, jika tidak maka *driver* akan mencari nilai *frame size* yang sesuai.

3.3 Pengujian Pengaruh *Frame size* Terhadap *Frame rate*

Pada bagian ini akan dibahas pengaruh dari *frame size* yang digunakan terhadap *frame rate* pada video yang dihasilkan. *Frame rate* 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. Pada pengujian ini digunakan 3 *frame size* yang akan dibandingkan *frame rate* videonya. Pengujian dilakukan pada intensitas cahaya maksimum yaitu 15 lux. Pada Tabel 2 terlihat perbedaan pada *frame rate* tidak terlalu besar, walau *frame size* berubah. Selisih antara *frame rate* pada *frame size* 320x240 dengan *frame rate* pada *frame size* 640x352 hanya 0.2 fps begitu juga selisih antara *frame rate* pada *frame size* 640x352 dengan *frame rate* pada *frame size* 640x480. Dari Tabel 2 tersebut dapat ditarik kesimpulan bahwa pengaruh perubahan *frame size* terhadap *frame rate* tidak terlalu signifikan. *Frame rate* pada Tabel 2 adalah *frame rate* maksimum yang dapat digunakan oleh sistem. Secara *default frame rate* yang digunakan sistem adalah *frame rate* yang didukung oleh *Webcam*, dalam hal ini 24-30 fps, namun *frame rate* ini tidak langsung digunakan begitu saja, pertama-tama sistem akan menyuruh *driver* untuk mengecek *frame rate* ini, dengan perintah `xioctl(s->fd, VIDIOC_S_PARM, setfps)`, jika ternyata *frame rate* masukan tidak didukung oleh sistem maka *driver* akan memberikan nilai *framerate* yang dapat didukung oleh sistem. Keterbatasan ini dapat disebabkan oleh *hardware* (raspberry pi), atau *drivernya*.

Tabel 2 Hasil pengujian pengaruh *frame size* terhadap *frame rate*

no	intensitas cahaya (lux)	<i>frame rate</i>		
		320x240	640x480	640x352
1	15	6,96	7,93	6,95
2	15	6,97	5,95	6,96
3	15	6,97	5,95	5,96
4	15	6,98	6,94	6,94
5	15	6,96	5,95	6,96
	rata rata =	6,968	6,544	6,754

3.4 Pengujian Pengaruh *Frame size* Terhadap *Bitrate*

Frame size yang besar membutuhkan pengolahan data yang lebih lama, karena data yang diolah semakin banyak. Pada pengujian pengaruh *frame size* terhadap *bitrate* ini ditunjukkan *bitrate* yang digunakan pada setiap *frame size*. Pada Tabel 3 diperlihatkan nilai *bitrate* yang berbeda-beda pada setiap *frame size* yang diujikan, semakin besar *frame size* -nya semakin besar *bitratanya*.

Bitrate yang ditampilkan adalah *bitrate* pada video hasil perekaman, sehingga video telah terkompres dan membuat *bitrate* video yang dibutuhkan lebih kecil daripada *bitrate* video sebelum dikompres (*raw video*). Walaupun telah terkompres perbedaan *bitrate* pada video dengan *frame size* yang berbeda tetap terlihat signifikan dan cukup memberikan gambaran tentang pengaruh *framesize* terhadap *bitrate*. Pada video dengan *frame size* besar dibutuhkan waktu yang lebih lama untuk diolah, seperti yang telah dibahas pada subbab 3.3 tentang pengaruh *frame size* terhadap *frame rate*, diketahui bahwa nilai *frame rate* kurang lebih sama,

walaupun *frame size* nya berbeda-beda, hal ini dikarenakan nilai *bitrate* yang semakin membesar saat *frame size*nya juga membesar, sehingga pengolahan datanya bisa menjadi lebih cepat dan menghasilkan *frame rate* yang tetap. Ini menunjukkan sistem dapat mengimbangi kebutuhan *bitrate* yang semakin besar seiring semakin besarnya *frame size* yang digunakan.

Tabel 3 Hasil pengujian pengaruh *frame size* terhadap *bitrate*

no	intensitas cahaya (lux)	<i>bitrate</i> (kbit/s)		
		320x240	640x480	640x352
1	15	402	533	529
2	15	416	536	518
3	15	412	529	521
4	15	399	534	531
5	15	452	517	518
	rata rata =	416,2	529,8	523,4

3.5 Pengujian Pengaruh *Frame size* Terhadap Deteksi Gerak

Pada bagian ini dibahas tentang pengaruh *frame size* terhadap deteksi gerak. Pendeteksian gerakan pada motion dilakukan dengan membandingkan *frame* terbaru dengan *frame* acuan, bila ada perbedaan maka akan dideteksi sebagai gerakan. Semakin besar *frame size* videonya maka semakin lama motion harus mengolah sebuah *frame* untuk mendeteksi ada tidaknya gerakan dalam *frame* tersebut. Semakin lama sebuah *frame* diolah oleh motion maka akan semakin banyak *frame* yang tidak diikutsertakan pada rekaman dan menyebabkan durasi video hasil perekaman lebih pendek dari seharusnya. Hal ini dibuktikan dengan hasil pengujian yang ditunjukkan pada Tabel 4, dimana rata rata durasi video hasil rekaman berbeda-beda pada tiap *frame size* yang diujikan. Pada saat pengujian penguji membuat gerakan untuk direkam oleh motion selama 30 detik.

Tabel 4 Hasil Pengujian pengaruh *frame size* terhadap durasi video

no	intensitas cahaya (lux)	durasi (detik)		
		320x240	640x352	640x480
1	15	40	20	15
2	15	39	21	19
3	15	39	23	20
4	15	40	19	16
5	15	21	23	20
	rata rata =	35,8	21,2	18

3.6 Pengujian Pengaruh Intensitas Cahaya Terhadap Deteksi Gerak

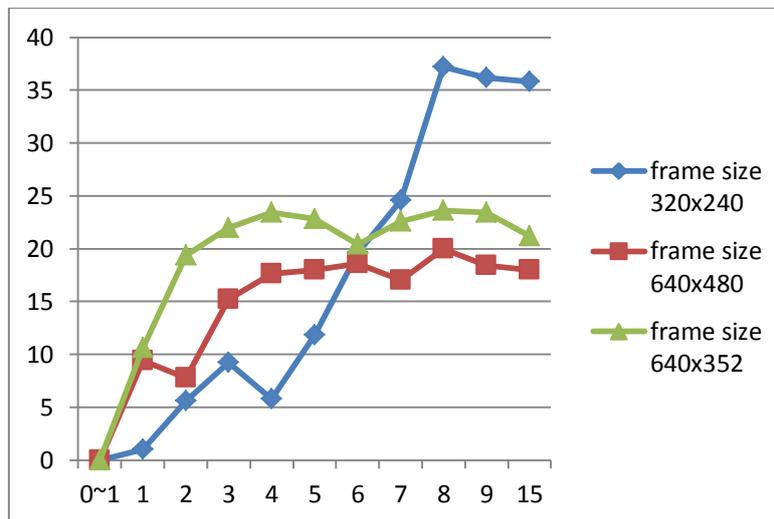
Pada pengujian ini kita dapat melihat pengaruh dari intensitas cahaya terhadap deteksi gerak dengan melihat durasi dari video yang dihasilkan. Pada pengujian ini penguji menggunakan *stopwatch* agar sistem merekam gerakan selama 30 detik. Namun pada video hasil rekaman, durasi video berbeda-beda, seperti yang terlihat pada Tabel 5, karena sistem

tidak dapat mendeteksi gerakan dengan sempurna sehingga *frame* yang tidak terdeteksi gerakan tidak dimasukkan dalam video.

Tabel 5 Hasil pengujian pengaruh *frame size* terhadap durasi video pada intensitas yang bervariasi

no	intensitas cahaya(lux)	Durasi(detik)		
		320x240	640x480	640x352
1	0~1	0	0	0
2	1	1	9,4	10,6
3	2	5,6	7,8	19,4
4	3	9,2	15,2	22
5	4	5,8	17,6	23,4
6	5	11,8	18	22,8
7	6	19,6	18,6	20,4
8	7	24,6	17	22,6
9	8	37,2	20	23,6
10	9	36,2	18,4	23,4
11	15	35,8	18	21,2

Pada resolusi 320x240 dengan intensitas cahaya maksimum yaitu 15 lux, rata-rata durasi video yang dihasilkan adalah 35,8 detik, karena *frame* acuan yang belum terupdate sehingga masih ada obyek didalam *frame* acuan padahal pada kenyataannya sudah tidak ada, bila terdeteksi gerakan maka *frame* tersebut akan disambung dengan *frame* sebelumnya, bila tidak terdeteksi gerakan selama 10 detik maka *event* tersebut dianggap selesai. Pada Tabel 5 terlihat durasi video semakin meningkat bila intensitas cahaya semakin tinggi, ini menunjukkan bahwa semakin tinggi intensitas cahayanya semakin baik pendeteksian gerakannya.



Gambar 9 Grafik hubungan antara intensitas cahaya (sumbu x) terhadap durasi video (sumbu y)

Pada pendeteksian gerakan oleh Motion, terdapat nilai *threshold* minimal yang harus dilewati agar selisih pada satu *frame* dengan *frame* acuan dapat dianggap sebagai gerakan. Nilai *default* dari *threshold* adalah 1500 pixel, maka dari itu selisih atau perubahan yang terdeteksi pada satu *frame* harus melebihi 1500 pixel agar dapat dianggap sebagai gerakan. Sehingga pada

intensitas cahaya yang sama, dengan *frame size* yang lebih besar akan lebih mudah mendeteksi gerakan karena perubahan pixelnya menjadi lebih banyak. Hal ini ditunjukkan pada Tabel 5 dimana pada intensitas cahaya yang rendah, seperti 1,2,3, dan 4 lux, *frame size* 640x480 dan 640x352 memiliki durasi video yang lebih panjang daripada durasi video pada *frame size* 320x240. Hal ini juga yang menyebabkan *frame size* 640x480 dan 640x352 mencapai durasi yang sama dengan durasi pada intensitas cahaya maksimal pada intensitas antara 3-4 lux atau dengan kata lain mencapai durasi maksimal pada intensitas cahaya lebih dari 4 lux. Sedangkan *frame size* 320x240 mulai mencapai durasi maksimal pada intensitas 8 lux seperti yang ditunjukkan pada grafik pada Gambar 9. Tercapainya kondisi durasi maksimal ditunjukkan dengan kecenderungan grafik membentuk garis horizontal yang dimulai pada intensitas cahaya tertentu, 8 lux untuk *frame size* 320x240, 3 lux untuk *frame size* 640x352, dan 4 lux untuk 640x480. Pada intensitas cahaya tersebut sistem mencapai kondisi durasi maksimal yang ditandai dengan tercapainya durasi video maksimal yaitu antara 20 hingga 25 detik pada *frame size* 640x352 dan antara 15 hingga 20 detik pada *frame size* 640x480. Dari pengujian ini didapatkan kesimpulan bahwa semakin besar *frame size* video dari sistem maka sistem tersebut akan semakin peka terhadap gerakan, namun semakin besar *frame size* videonya maka semakin lama waktu yang dibutuhkan sistem untuk mengolah satu *frame*, dan mengakibatkan semakin banyak *frame* yang tidak terolah oleh sistem dan tidak masuk kedalam video.

4. KESIMPULAN

Dari penelitian yang telah dilakukan, telah dibuat sebuah sistem pemantauan ruangan dengan server raspberry pi yang dapat secara otomatis merekam dan memberikan notifikasi jika terdeteksi gerakan, kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Untuk mendapatkan video tanpa ada *frame* yang hilang, *frame size* yang digunakan adalah 320x240 dengan intensitas cahaya minimal 8 lux
2. Pengaruh perubahan *frame size* terhadap *frame rate* tidak signifikan.
3. *Frame size* yang lebih besar menyebabkan motion lebih peka terhadap gerakan karena jumlah perubahan pixel yang terjadi menjadi lebih banyak daripada pada *frame size* yang lebih kecil.
4. Intensitas minimal untuk motion hingga dapat mendeteksi gerakan adalah 1 lux pada frame size 320x240, 640x352 dan 640x480.

5. SARAN

1. Sistem ini masih sebatas diujikan pada jaringan WLAN. Diharapkan penelitian selanjutnya dapat ditingkatkan sehingga video streaming dapat diakses melalui internet.
2. Jika selanjutnya masih mengembangkan sistem pemantauan dengan motion, maka diharapkan dapat menggunakan lebih dari satu webcam sehingga dapat mengetahui pengaruh dari jumlah webcam terhadap kemampuan raspberry pi sebagai sistem pemantauan.
3. Penelitian selanjutnya diharapkan menggunakan program selain motion, sehingga dapat dibandingkan dengan pengujian ini.

DAFTAR PUSTAKA

- [1] Ero, J, 2009, Sistem Monitoring Berbasis Live Video Streaming Dan Dilengkapi Notifikasi SMS, *Skripsi*, Ilmu Komputer FMIPA UGM, Yogyakarta.
- [2] Holandrio, D., 2012, Sistem Video Streaming Dengan Server Mini Personal Computer (Mini Pc) Pada Jaringan Ad-Hoc, *Skripsi*, Elektronika dan Instrumentasi FMIPA UGM, Yogyakarta.
- [3] Lavrsen, K., 2010, *Motion Guide for Motion version 3.2.12*, <http://www.lavrsen.dk/foswiki/bin/view/Motion/MotionGuide3x1x20>, diakses pada tanggal 26 Februari 2013
- [4] Al-Tayeeb, R., 2012, Motion JPEG Streaming Server, <http://www.codeproject.com/Articles/371955/Motion-JPEG-Streaming-Server>, diakses pada tanggal 1 Desember 2013
- [5] Novasandro, R., dan Kautsar, A. E., 2009, *Prinsip Kerja Protokol - Protokol Electronic Mail*, Jurusan Teknik Elektro Fakultas Teknik, Universitas Gadjah Mada