

## SISTEM PEMANTAUAN RUANGAN JARAK JAUH VIA INTERNET BERBASIS RASPBERRY PI 2 TERINTEGRASI DENGAN GOOGLE DRIVE DAN LINE

Juli Arianes<sup>1</sup>, Dr. Ir. Ponco Siwindarto, M.Eng.Sc.<sup>2</sup>, Dr.Eng. Panca Mudjirahardjo, S.T., M.T.<sup>3</sup>.

Teknik Elektro Universitas Brawijaya

Jalan M.T Haryono No.167 Malang 65145 Indonesia

Email: juli.arianes@gmail.com

Abstrak – Salah satu teknologi yang membuktikan berkembangnya teknologi komunikasi untuk kebutuhan manusia adalah kamera untuk melakukan pengintaian atau dikenal dengan CCTV (*Closed Circuit TeleVision*). Saat ini sudah banyak penggunaan kamera CCTV terutama untuk daerah dengan tingkat mobilitas tinggi seperti perkantoran, sekolah, jalan raya, dll. Namun dalam hal ini penggunaannya tidak efektif untuk daerah dengan tingkat mobilitas rendah seperti rumah yang ditinggalkan pemiliknya ataupun di ruang/gudang penyimpanan karena pada CCTV analog akan tetap merekam selama 24 jam tanpa henti dan hal ini tentu akan membebani ruang penyimpanan pada perangkat DVR (*data video recorder*). Penyimpanan hasil video rekaman pada DVR pun belum cukup aman, karena DVR tersebut dapat mengakibatkan kerusakan dan akan menyebabkan hilangnya video hasil rekaman. Untuk melakukan *monitoring* dengan melakukan video *streaming* dari jarak jauh, jaringan internet yang digunakan harus memiliki IP Publik dan biaya aktivasinya sangat mahal. Dari permasalahan tersebut alat ini dapat membuat ruangan dapat dipantau dari jarak jauh melalui internet yang tersedia pada *domain julishome.ngrok.io* dengan *Web server* yang digunakan adalah *apache*. Sistem dapat mendeteksi gerakan dan merekam menggunakan paket perangkat lunak *motion* yang berfungsi menganalisa gambar yang ditangkap oleh kamera dengan kondisi intensitas ruangan minimal sebesar 5 lux. Selain itu saat mendeteksi gerakan sistem akan mengirim pesan melalui aplikasi *LINE* ke pemilik rumah dengan waktu pengiriman rata-rata selama 5.15 s. Hasil rekaman akan disimpan di *MicroSD* dan jika ada video baru maka akan diunggah di *Google Drive*. Video akan diunggah oleh paket perangkat lunak *Gdrive* yang akan melakukan sinkronisasi *folder* pada *Google Drive* ke *folder* lokal *MicroSD* setiap 1 menit sekali. alarm dapat dikendalikan dari jarak jauh. Pemograman *web* menggunakan *php* yang akan memberikan perintah ke Pin GPIO Raspberry Pi melalui *script python*. Pin GPIO terhubung dengan rangkaian *Driver* yang memiliki karakteristik tegangan  $\leq 1$  V untuk membunyikan alarm. Sistem secara keseluruhan memiliki *error* sebesar 0% dalam mengirim pesan *LINE* dan mengunggah video ke *Google Drive*. Hasil pengujian Sistem memiliki performa yang baik.

**Kata Kunci** – *Raspberry Pi 2, Kamera, Motion, Ngrok, Google Drive, LINE*

*Abstract – One of technology proved that communication technology have been improved for human needed is spy camera or we known as CCTV(Closed Circuit Television). There are many high mobility resident such as office, school, highway etc. use CCTV camera. However, it isn't effective for a low mobility resident such as the owner leave his/her home, room or warehouse because for analog CCTV will record during 24 hours, so it makes the DVR (Data Video Recorder) will be stressed. Data video at DVR isn't secure due to DVR can be broken then the data will be lost. For doing monitoring a room from far aplace, the network must have public IP that the cost so expensive. To overcome this problem, this system is able to monitor a room from far place through internet at domain julishme.ngrok.io with apache as web server. System is able to detect a motion and record as video with software motion that can analyze a picture captured by camera with light intencity in the room minimum of 5 lux. Then when detect a motion, system will send a LINE's message to user with average time is 5 s. the result of video recording will be saved in microSD and if there is any new video it will be upload to Goolge Drive. Video uploaded by software Gdrive that can synchronize a folder in Google Drive with in local folder MicroSD every 1 minute. Alarm is able to control by user from far place throught internet. Web programining language using php that can call and ecexute python script to access GPIO Pin Raspberry Pi. GPIO Pin is connected to driver circuit with voltage characteristic  $\leq 1$  V for ringing an alarm. For sending a LINE's message and uploading video to Goolge Drive system has total error is 0 %. The experimental result shows that system have good performance.*

**Keywords** – *Raspberry Pi 2, Camera, Motion, Ngrok, Google Drive, LINE*

### PENDAHULUAN

Salah satu teknologi yang membuktikan berkembangnya teknologi komunikasi untuk kebutuhan manusia adalah kamera yang berfungsi untuk melakukan pengintaian atau dikenal dengan CCTV (*Closed Circuit TeleVision*). Sistem CCTV ini dapat menampilkan video *streaming* ke monitor PC (*Personal Computer*)

dan hasil dari rekaman video tersebut disimpan pada sebuah perangkat yaitu DVR (*Digital Video Recorder*). CCTV berguna dalam kebutuhan manusia akan rasa aman, karena kedudukan rekaman video tindak pidana (Rekaman CCTV) dalam pembuktian sistem pidana umum adalah sebagai alat bukti petunjuk (Pasal 188 Kitab Undang undang Hukum Acara Pidana).

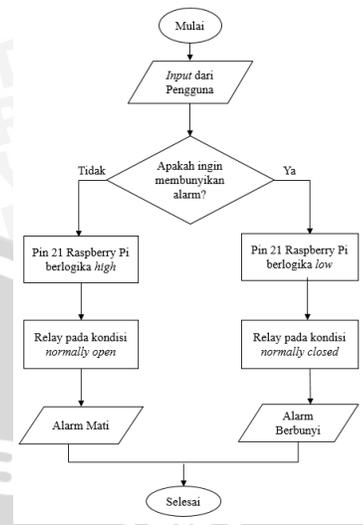
<sup>1</sup> Juli Arianes adalah mahasiswa Teknik Elektro Universitas Brawijaya, Malang, Indonesia

<sup>2</sup> Dr. Ir. Ponco Siwindarto, M.Eng.Sc adalah dosen Teknik Elektro Universitas Brawijaya, Malang, Indonesia

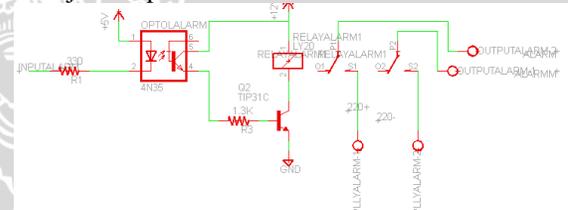
<sup>3</sup> Dr.Eng. Panca Mudjirahardjo, S.T., M.T adalah dosen Teknik Elektro Universitas Brawijaya, Malang, Indonesia

Saat ini sudah banyak penggunaan kamera CCTV terutama untuk daerah dengan tingkat mobilitas tinggi seperti perkantoran, sekolah, jalan raya, dll. Namun dalam hal ini penggunaannya tidak efektif untuk daerah dengan tingkat mobilitas rendah seperti rumah yang ditinggalkan pemiliknya ataupun di ruang/gudang penyimpanan karena pada CCTV analog akan tetap merekam selama 24 jam tanpa henti dan hal ini tentu akan membebani ruang penyimpanan pada perangkat DVR. Penyimpanan hasil video rekaman pada DVR pun belum cukup aman, karena DVR tersebut dapat mengalami kerusakan dan akan menyebabkan hilangnya video hasil rekaman. Untuk melakukan *monitoring* dari jarak jauh jaringan internet yang digunakan harus memiliki IP Publik yang biaya aktivasinya sangat mahal.

Berdasarkan masalah diatas maka pada skripsi ini adalah dirancang sistem pemantauan berbasis Raspberry Pi yang khusus untuk daerah dengan tingkat mobilitas rendah. Dengan menggunakan Raspberry Pi, monitoring dapat dilakukan dari jarak jauh dengan menggunakan *port forwarding*, dan hanya akan merekam apabila terjadi pergerakan yang terdeteksi pada kamera Raspberry Pi, kemudian menyimpan hasil video rekaman tersebut ke *Google Drive* Selain itu untuk ketika mendeteksi gerakan maka alat bertindak sebagai *server* yang mengirim pesan pengguna melalui *LINE*. Jika hasil dari monitoring terdeteksi aktivitas yang mencurigakan, maka sistem akan membunyikan alarm yang dikendalikan dari jarak jauh.



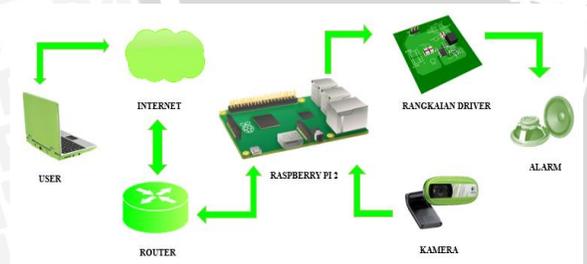
**Gambar 2.** Diagram Alir Kerja Driver Alarm  
 Untuk perancangan elektrik digunakan software EAGLE untuk membuat layout PCB (*Printed Circuit Board*). Skematik Rangkaian Driver ditunjukkan pada Gambar 3.



**Gambar 3.** Skematik Rangkaian Driver  
**B. Konfigurasi Jaringan Internet dan Web Server**

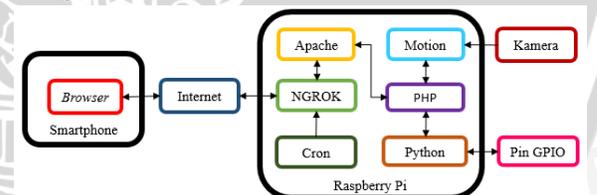
**I. PERANCANGAN DAN PEMBUATAN ALAT**

Perancangan alat dibagi menjadi beberapa bagian. Blok diagram sistem keseluruhan ditunjukkan dalam Gambar 1.



**Gambar 1.** Blok diagram sistem keseluruhan.  
**A. Perancangan Rangkaian Driver Alarm**

Perancangan elektrik meliputi perancangan dan pembuatan *Driver* alarm sesuai dengan diagram alir kerja *Driver* alarm pada Gambar 2.



**Gambar 4.** Blok Diagram Sistem Pemantauan Jarak Jauh

Blok diagram pada Gambar 4. menunjukkan pengguna menggunakan *smartphone* yang telah terinstall *browser* mengakses *web server* melalui domain yang telah dibuat yang merupakan hasil dari *port forwarding* paket perangkat lunak *NGROK*. *Web server* yang digunakan yaitu dengan paket perangkat lunak *APACHE*. *Web server* akan menampilkan halaman *streaming* kamera dan halaman akses kendali alarm. Konfigurasi perangkat lunak yang akan dilakukan meliputi:

**1. Konfigurasi alamat IP pada alat.**

a. Untuk melakukan konfigurasi *IP static* pada alat diketikkan *command* pada terminal seperti berikut:

```
root@raspberrypi:/home/pi# nano /etc/network/interfaces
```

Berikut adalah *script interfaces* yang digunakan pada Raspberry Pi yang digunakan.

```
GNU nano 2.2.6 File: /etc/network/interfaces

interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback
iface eth0 inet static
address 10.26.53.93
netmask 255.255.254.0
network 10.26.52.0
broadcast 10.26.53.255
gateway 10.26.52.1
iface default inet dhcp
```

b. Kemudian diketikkan *command* berikut:

```
root@raspberrypi:/home/pi# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:6a:c0:31
          inet addr:10.26.53.93  Bcast:10.26.53.255  Mask:255.255.254.0
          inet6 addr: fe80::ce4c:162f:4c47:91d8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Hasil menunjukkan bahwa pada eth0 telah terkonfigurasi *IP static*.

### 1. Konfigurasi Kamera USB

a. Kamera web dihubungkan ke port USB Raspberry Pi 2. Selanjutnya diketikkan *command* berikut:

```
pi@raspberrypi:~$ lsusb
Bus 001 Device 005: ID 046d:082b Logitech, Inc.
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wi
reless Adapter [Realtek RTL8188CUS]
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
```

Terlihat Kamera telah berhasil terhubung ini menunjukkan bahwa *Driver* untuk perangkat kamera jenis Logitech C170 telah tersedia pada sistem operasi yang digunakan.

### 2. Konfigurasi Motion

a. Pada terminal diketikkan *command* berikut untuk menginstal paket perangkat lunak *motion*.

```
root@raspberrypi:/home/pi# apt-get install motion
```

b. Dibuka file konfigurasi *Motion* dengan diketikkan *command* berikut:

```
root@raspberrypi:/home/pi# nano /etc/motion/motion.conf
```

c. Diubah *script file motion.conf* menjadi seperti yang ditunjukkan pada Tabel 1.

Tabel 1. *Script motion.conf*

| Script                      | Fungsi   |
|-----------------------------|--|
| <i>Daemon=ON</i>            | Memungkinkan <i>Motion</i> untuk berjalan.             |
| <i>stream_localhost off</i> | Memungkinkan output dilihat menggunakan jaringan luar. |
| <i>input -1</i>             | 1 berarti Input video yang digunakan yaitu USB kamera  |
| <i>width 640</i>            | lebar gambar (pixels).                                 |
| <i>height 480</i>           | Tinggi gambar (pixels).                                |
| <i>framerate 100</i>        | Jumlah <i>frame</i> maksimum                           |

|                          |   |
|--------------------------|---|
| <i>threshold 1500</i>    | Nilai <i>Threshold</i> piksel yang berubah.     |
| <i>stream_port 8081</i>  | Listen <i>Streaming</i> pada port 8081          |
| <i>stream_maxrate 24</i> | <i>framerate</i> maksimum saat <i>streaming</i> |

### 3. Konfigurasi Web server

a. Diketikkan *command* berikut untuk menginstall paket perangkat lunak *Apache2*.

```
root@raspberrypi:/etc/bind# sudo apt-get install apache2
```

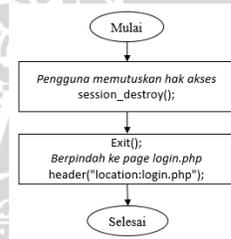
b. Kemudian cek status *web server* dengan mengetikkan *command* berikut pada terminal.

```
root@raspberrypi:/var/www/html# /etc/init.d/apache2 status
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2)
   Active: active (running) since Fri 2016-06-17 12:40:53 W
 Process: 5596 ExecStop=/etc/init.d/apache2 stop (code=exi
 Process: 5621 ExecStart=/etc/init.d/apache2 start (code=
   CGroup: /system.slice/apache2.service
          └─5637 /usr/sbin/apache2 -k start
          └─5640 /usr/sbin/apache2 -k start
          └─5641 /usr/sbin/apache2 -k start
          └─5642 /usr/sbin/apache2 -k start
```

Terlihat bahwa *Web server* telah berjalan.

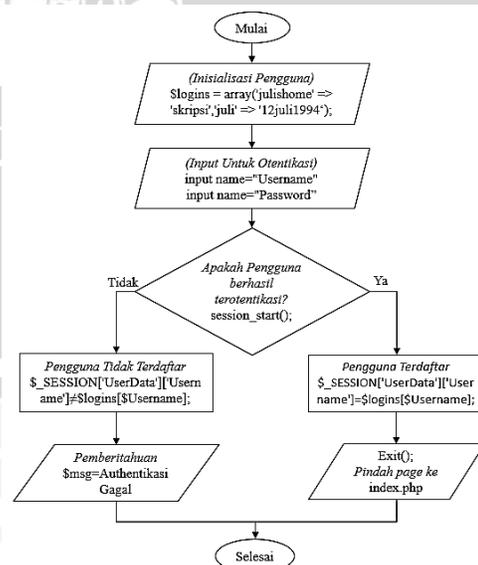
d. Kemudian dibuat program dengan bahasa pemrograman *php*. *Script* program yang dibuat berdasarkan diagram alir berikut:

Diagram alir *script* program *logout.php* ditunjukkan pada Gambar 5.



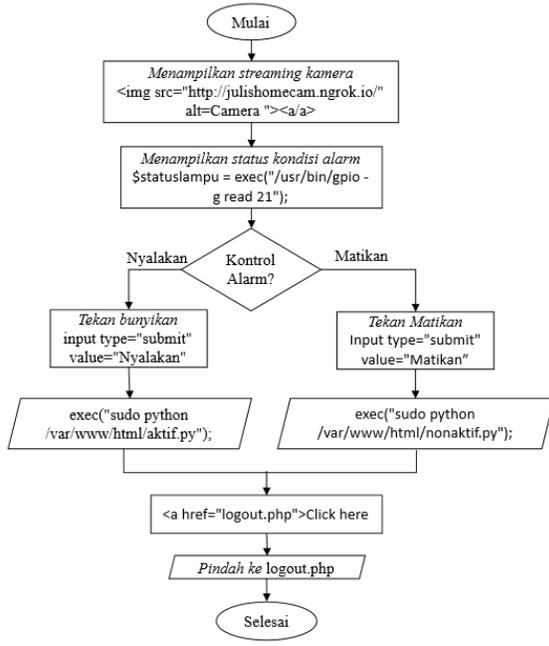
Gambar 5. Diagram alir *logout.php*

Diagram alir *Script* program *login.php* ditunjukkan pada Gambar 6.



Gambar 6. Diagram alir *login.php*

Diagram alir *script* program *index.php* ditunjukkan pada Gambar 7.



Gambar 7. Diagram alir *index.php*

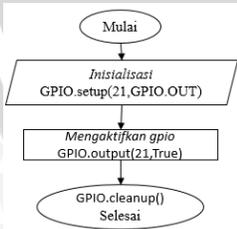
#### 4. Konfigurasi Python

a. Di-*install* paket perangkat lunak *Python* dengan diketikkan *command* berikut:

```
root@raspberrypi:/home/pi# apt-get install python
```

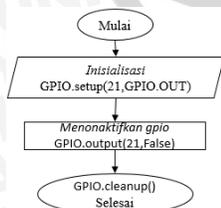
b. Dibuat file program *python*. Program yang akan dibuat ada 2 yaitu *aktif.py* dan *nonaktif.py*. *script python* dibuat berdasarkan diagram alir, diagram alir masing - masing program adalah sebagai berikut:

Diagram alir *script* program *aktif.py* ditunjukkan pada Gambar 8.



Gambar 8. Diagram alir *aktif.py*

Diagram alir *script* program *nonaktif.py* ditunjukkan pada Gambar 9.



Gambar 9. Diagram alir *nonaktif.py*

#### 5. Konfigurasi Cron

a. Di-*install* Paket Perangkat Lunak *Cron* dengan diketikkan *command* berikut:

```
root@raspberrypi:/home/pi# apt-get install cron
```

Paket perangkat lunak *Cron* akan digunakan pada sistem untuk menjadwalkan perintah yang harus dieksekusi.

#### 6. Konfigurasi NGROK

a. Di-*download* Paket perangkat lunak *NGROK* dengan *command* berikut:

```
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-arm.zip
```

b. Jika telah selesai maka dilakukan *unzip file* tersebut dengan *command* berikut:

```
/home/pi/ngrok2# unzip ngrok-stable-linux-arm.zip
```

c. Selanjutnya diketikkan *file config* pada path */root/.ngrok2/ngrok.yml* sebagai berikut:

```
pi@raspberrypi:~
GNU nano 2.2.6
authtoken: 4Bkf4tPCeMbf56UZWXxB2_4f9Qj1DRADyqJgJY19wjM
tunnels:
first-app:
  addr: 80
  proto: http
  subdomain: julishome
second-app:
  addr: 8081
  proto: http
  subdomain: julishomecam
```

d. Di-*install* paket perangkat lunak *NGROK* pada *Cron* dengan *Command* berikut:

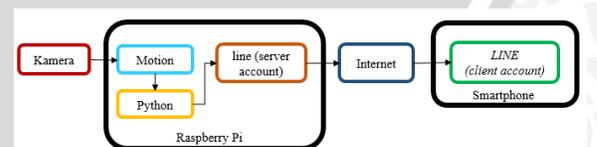
```
root@raspberrypi:/home/pi# crontab -e
```

e. Selanjutnya diketikkan *Script* berikut untuk memasang perintah di *Cron*.

```
@reboot /home/pi/ngrok2/ngrok start --all
```

*Script* diatas untuk menjalankan *file config* perangkat lunak *NGROK*.

#### C. Konfigurasi Pendeteksi Gerakan dan Integrasi dengan LINE



Gambar 10. Blok Diagram Sistem Pengirim Pesan LINE

Blok diagram pada Gambar 10. menunjukkan *Motion* akan mendeteksi jika terjadi gerakan kemudian akan menjalankan program *python* yang terintegrasi dengan *LINE* sebagai *library LINE*. *LINE* akan mengirim pesan kepada pengguna melalui internet. Konfigurasi perangkat lunak yang meliputi:

### 1. Konfigurasi LINE

a. Di-Download dan Di-install paket perangkat lunak LINE dengan diketikkan *command* berikut:

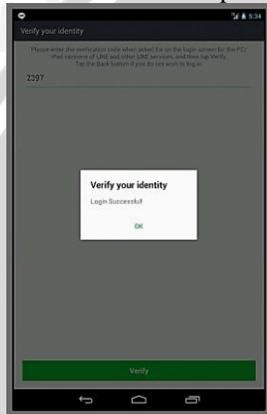
```
git clone https://github.com/carpedm20/LINE.git
```

b. Setelah berhasil di-download, Selanjutnya akan dilakukan otentikasi untuk akun yang akan digunakan menjadi *server* LINE. Pada terminal diketikkan *script* berikut:

```
root@raspberrypi:~/LINE# python
Python 2.7.9 (default, Mar 8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from line import LineClient
>>> client = LineClient("*****", "*****")
Enter PinCode '2397' to your mobile phone in 2 minutes
```

Terlihat pada terminal setelah memasukkan *email* dan *password* maka akan muncul *PinCode* yang harus dimasukkan.

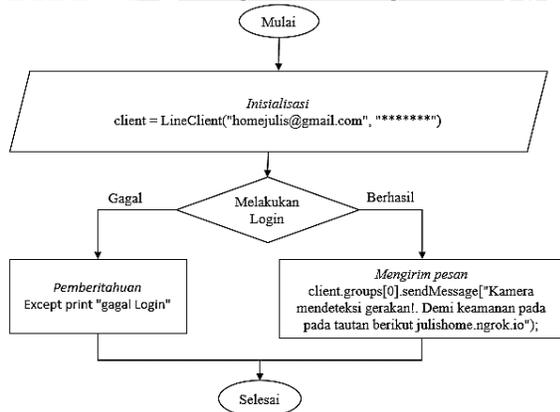
c. Pada akun LINE akan terlihat seperti Gambar 11.



Gambar 11. Verifikasi Akun LINE

Setelah memasukkan *pincode* pada akun LINE *server* maka terlihat pada gambar akun *server* paket perangkat lunak LINE telah terverifikasi.

d. Kemudian dibuat *script* python untuk mengirim pesan dari akun *server* ke akun klien. *Script* yang dibuat berdasarkan diagram alir sebagai berikut:



Gambar 12. Diagram Alir sendLINE.py

Berdasarkan diagram alir yang ditunjukkan pada Gambar 12. maka pesan akan dikirimkan ke grup LINE ke 0. Grup LINE ke 0 adalah grup LINE keluarga penulis sehingga kemungkinan pesan tidak

terbaca semakin kecil jika dibandingkan dikirim ke *personal message*.

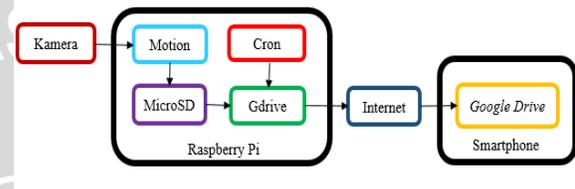
### 2. Konfigurasi Motion

agar pesan terkirim ketika mendeteksi gerakan maka dilakukan penambahan *script* file konfigurasi pada Motion pada direktori `/etc/motion/motion.conf`. Kemudian ditambahkan pada baris berikut:

```
on_event_start python /home/pi/LINE/sendline.py
```

*Script* ini menunjukkan ketika terjadi gerakan didepan kamera maka akan mengeksekusi perintah mengirim pesan LINE.

### D. Konfigurasi Penangkap Gerakan dan Integrasi Sistem dengan Google Drive



Gambar 13. Blok Diagram Sistem Penangkap Gerakan dan Pengunggah Video

Blok diagram pada Gambar 13. menunjukkan Apabila ada gerakan maka *software* MOTION akan merekam dan menyimpannya ke MicroSD. GDRIVE melakukan pengunggahan video setiap 1 menit sekali yang diatur pada paket perangkat lunak *cron*, dengan metode sinkronisasi *folder online* pada Google Drive ke *folder* lokal hasil rekaman pada Raspberry Pi maka dengan demikian dapat menghindari gagal *upload* jika koneksi internet sedang bermasalah.

Konfigurasi perangkat lunak meliputi:

### 1. Konfigurasi Motion

a. Dibuka file *motion.conf* dan ditambahkan *script* berikut:

```
target_dir /usr/local/bin/motion
```

*Script* diatas menunjukkan video hasil rekaman akan tersimpan pada lokasi `/usr/local/bin/motion`

### 1. Konfigurasi Perangkat Lunak GDRIVE

a. Di-Download paket perangkat lunak Gdrive dengan perintah berikut:

```
wget https://docs.google.com/uc?id=0B3X9G1R6EmbnVXNLanp4ZFRRbz
```

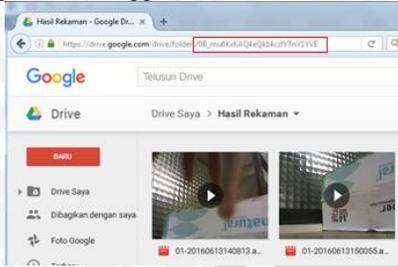
b. Dilakukan otentikasi akun Google Drive server dengan mengetikkan perintah berikut:

```
root@raspberrypi:~/usr/local/bin# drive -c foo list
Authentication needed
Go to the following url in your browser:
https://accounts.google.com/o/oauth2/auth?access_type=
13Awq13Aoauth13A2.013Aocb1response_type=code1scope=htt
Enter verification code: 4/Sa9-1WYSLnQPK56sxssJ199Ykwc
```

Terlihat pada terminal bahwa untuk melakukan oentikasi akun diharuskan membuka *url* yang tertera pada terminal. Setelah dibuka pada *browser* maka akan mendapatkan kode verifikasi. Jika kode

verifikasi berhasil dimasukkan maka akun Google Drive server siap digunakan.

c. Selanjutnya dibuat folder pada akun Google Drive, folder ini nantinya akan digunakan sebagai tempat file yang telah terunggah.



Gambar 14. Folder Google Drive

Setelah folder dibuat catat id folder yang ditunjukkan pada kotak merah pada Gambar 14.

## 2. Konfigurasi Script CRON

a. Ditambahkan task baru dengan diketikkan script berikut:

```
*/* * * * * /usr/local/bin/gdrive -c /usr/local/bin/foo sync upload
/usr/local/bin/motion/ OB_mu6KxKAQ4eQk14cz1YtN1Y1YVE
```

Script diatas menunjukkan bahwa setiap 1 menit dilakukan sinkronasi folder online Google Drive dengan folder lokal hasil rekaman dengan lokasi /usr/local/motion.

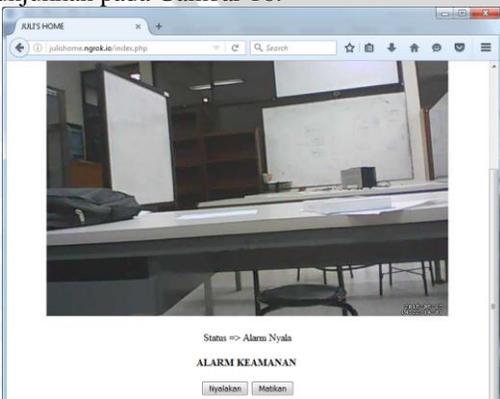
## III. PENGUJIAN DAN ANALISIS

### A. Jaringan dan Web Server Pada Sistem

Tampilan halaman autentikasi pada komputer klien ditunjukkan pada Gambar 15.



Gambar 15. Halaman Otentikasi Pengguna tampilan halaman streaming dan akses kendali alarm ditunjukkan pada Gambar 16.



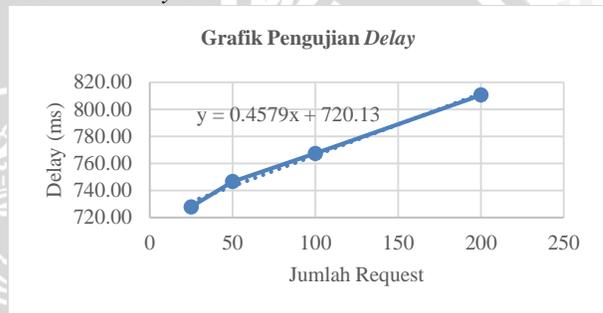
Gambar 16. Halaman Streaming Kamera

## 1. Pengujian Delay Jaringan

Tabel 2. Pengujian Delay

| Jumlah Request | Delay (ms) Pengujian ke- |        |        | Delay (ms) Rata-rata |
|----------------|--------------------------|--------|--------|----------------------|
|                | 1                        | 2      | 3      |                      |
| 25             | 721.5                    | 727.59 | 734.24 | 727.78               |
| 50             | 727.78                   | 774.02 | 737.63 | 746.48               |
| 100            | 756.75                   | 775.61 | 769.77 | 767.38               |
| 200            | 816.41                   | 864.89 | 750.45 | 810.58               |

Dari Tabel 2. didapatkan rata – rata delay jaringan saat jumlah request sebesar 25 besar delay 727.78 ms. Pada jumlah request sebesar 50 besar delay 746.48 ms. Pada jumlah request sebesar 100 besar delay 767.38 ms. Pada jumlah request sebesar 200 besar delay 810.58 ms.



Gambar 17. Grafik Pengujian Delay

Dari Gambar 17. didapatkan Semakin banyak jumlah request, maka semakin besar delay. kategori latensi diatas rata-rata delay memiliki performansi jelek. Dari persamaan regresi linier menunjukkan setiap jumlah Request bertambah 1 maka Delay naik sebesar 0.4579 ms.

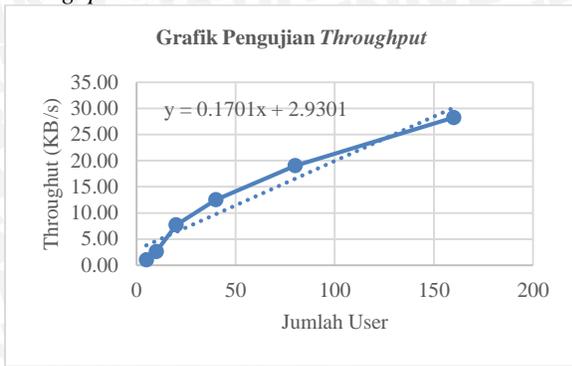
## 2. Pengujian Throughput Jaringan

Tabel 3. Pengujian Throughput

| Jumlah User | Throughput (kB/s) Pengujian ke- |       |       | Throughput (kB/s) Rata-rata |
|-------------|---------------------------------|-------|-------|-----------------------------|
|             | 1                               | 2     | 3     |                             |
| 5           | 0.92                            | 1.06  | 1.18  | 1.05                        |
| 10          | 2.3                             | 2.34  | 3.18  | 2.61                        |
| 20          | 8.05                            | 7.9   | 7.22  | 7.72                        |
| 40          | 10.6                            | 12.99 | 13.93 | 12.51                       |
| 80          | 18.94                           | 15.37 | 22.76 | 19.02                       |
| 160         | 23.06                           | 26.58 | 35.1  | 28.25                       |

Dari Tabel 3. didapatkan rata – rata throughput jaringan saat jumlah user sebanyak 5 besar throughput 1.05 kB/s. Pada jumlah user sebanyak 10 besar throughput 2.61 kB/s. Pada jumlah user sebanyak 20 besar throughput 7.72 kB/s. Pada jumlah user sebanyak 40 besar throughput 12.51 kB/s. Pada jumlah user sebanyak 80 besar throughput

19.02 kB/s. Pada jumlah *user* sebanyak 160 besar *throughput* 28.25 kB/s



**Gambar 18.** Grafik Pengujian Troughput

Dari Gambar 18. didapatkan Semakin banyak jumlah *user*, maka semakin besar *throughput*. Dari persamaan regresi linier menunjukkan setiap *user* bertambah 1 maka *Throughput* naik sebesar 0.1701 kB/s.

**3. Pengujian Packet loss Jaringan**

**Tabel 4.** Pengujian *Packet loss*

| Jumlah Request | Packet loss (%) Pengujian ke- |      |      | Packet loss (%) Rata-rata |
|----------------|-------------------------------|------|------|---------------------------|
|                | 1                             | 2    | 3    |                           |
| 25             | 0.00                          | 0.00 | 0.00 | 0                         |
| 50             | 0.00                          | 0.00 | 0.00 | 0                         |
| 100            | 0.00                          | 0.00 | 0.00 | 0                         |
| 200            | 0.00                          | 0.00 | 0.00 | 0                         |

Dari Tabel 4. didapatkan rata – rata *packet loss* saat jumlah *request* sebanyak 25 sampai 200 *packet loss* yang terjadi adalah 0%. Berdasarkan pada tabel kategori *Packet loss* maka rata-rata *packet loss* adalah sangat loss.

**4. Pengujian Fungsionalitas Web Server**

**Tabel 5.** Pengujian Fungsionalitas Web Server

| Jenis Browser     | Tindakan                      |                               |
|-------------------|-------------------------------|-------------------------------|
|                   | Menampilkan Halaman Streaming | Mengakses Tombol Kendali Alam |
| Mozilla Firefox   | v                             | v                             |
| Opera Mini        | x                             | v                             |
| UCBrowser         | x                             | v                             |
| Internet Explorer | x                             | v                             |
| Chrome            | v                             | v                             |

Keterangan:  
v = Berhasil  
x = Gagal

Dari Tabel 5. menunjukkan bahwa semua *browser* dapat mrngakses tombol kedali alarm. Dari

tabe *browser* yang dapat menampilkan halaman *streaming* kamera yaitu *browser* jenis Mozilla Firefox dan Chrome. dari pengujian ini dapat disimpulkan rekomendasi *browser* yang digunakan pada sistem yaitu jenis Mozilla firefox dan Chromes.

**B. Pendeteksi Gerakan dan Integrasi dengan LINE Pada Sistem**

Hasil pengiriman pesan *LINE* pada *smartphone* klien ditunjukkan pada Gambar 19.



**Gambar 19.** Tampilan Pesan LINE

**1. Pengujian Intensitas Cahaya Ruangan**

**Tabel 6.** Pengujian Intensias Cahaya Ruangan

| Intensitas Cahaya (lux) | Gambar Tangkapan Kamera | Pengujian ke - |   |   |
|-------------------------|-------------------------|----------------|---|---|
|                         |                         | 1              | 2 | 3 |
| 0                       |                         | x              | x | x |
| 5                       |                         | v              | v | v |
| 10                      |                         | v              | v | v |
| 20                      |                         | v              | v | v |
| 40                      |                         | v              | v | v |
| 80                      |                         | v              | v | v |

keterangan:  
x = tidak mendeteksi gerakan  
v = mendeteksi gerakan

Dari Tabel 6. didapatkan saat kondisi intensitas cahaya ruangan sebesar 0 lux selama 3 kali pengujian obyek bergerak didepan kamera didapatkan hasil *motion* tidak dapat mendeteksi gerakan. Pada saat kondisi intensitas cahaya ruangan sebesar 5 - 80 lux selama 3 kali pengujian obyek bergerak didepan kamera didapatkan hasil *motion* mendeteksi adanya gerakan.

**2. Pengujian Waktu Pengiriman Pesan LINE**

**Tabel 7.** Pengujian Waktu Pengiriman Pesan *LINE*

| Pengujian ke-    | Pengiriman Pesan <i>LINE</i> | Waktu(s)    |
|------------------|------------------------------|-------------|
| 1                | v                            | 5.69        |
| 2                | v                            | 5.44        |
| 3                | v                            | 5.74        |
| 4                | v                            | 5.06        |
| 5                | v                            | 4.86        |
| 6                | v                            | 4.87        |
| 7                | v                            | 4.81        |
| 8                | v                            | 4.97        |
| 9                | v                            | 4.93        |
| 10               | v                            | 5.11        |
| <b>Rata-rata</b> |                              | <b>5.15</b> |
| <b>Error</b>     |                              | <b>0%</b>   |

keterangan:

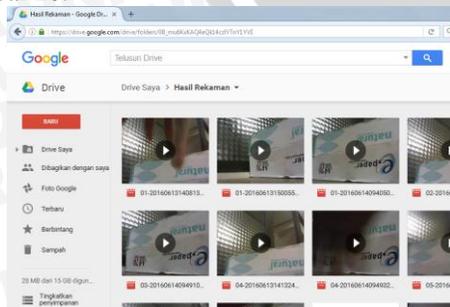
x = tidak terkirim

v = terkirim

Dari Tabel 7. didapatkan rata – rata waktu pengiriman pesan *LINE* sebesar 5.15 s. Hasil percobaan menunjukkan *error* 0%. Maka tingkat keberhasilan pengiriman pesan *LINE* saat terjadi gerakan adalah sebesar 100%. Hasil banyak hal yang mempengaruhi salah satunya adalah kestabilan koneksi internet pada Raspberry Pi.

**C. Penangkap Gerakan dan Integrasi Sistem dengan Google Drive**

Hasil video yang berhasil terunggah di *Google Drive* pada komputer klien ditunjukkan pada Gambar 20.



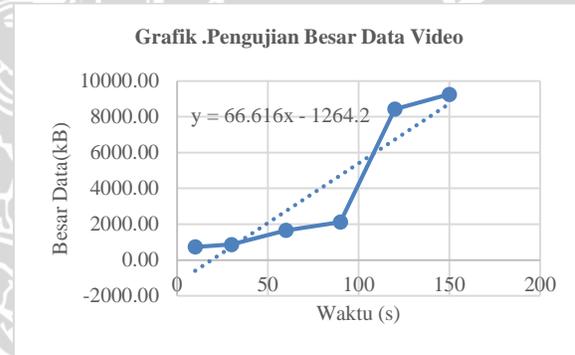
**Gambar 20.** Video Pada *Google Drive*

**1. Pengujian Besar Data Video**

**Tabel 8.** Pengujian Besar Data Video

| Pengujian ke- | Waktu Video terekam (s) | Besar Data(kB) |
|---------------|-------------------------|----------------|
| 1             | 10                      | 733.70         |
| 2             | 30                      | 866.25         |
| 3             | 60                      | 1661.74        |
| 4             | 90                      | 2117.96        |
| 5             | 120                     | 8427.45        |
| 6             | 150                     | 9251.46        |

Dari Tabel 8. didapatkan besar data ketika waktu rekaman video selama 10 s adalah 733.70 kB. Pada rekaman video selama 30 s besar data adalah 866.25 kB. Pada rekaman video selama 60 s besar data adalah 1661.74 kB. Pada rekaman video selama 90 s besar data adalah 2117.96 kB. Pada rekaman video selama 120 s besar data adalah 8427.45 kB. Pada rekaman video selama 150 s besar data adalah 9251.45 kB.



**Gambar 21.** Pengujian Besar Data Video

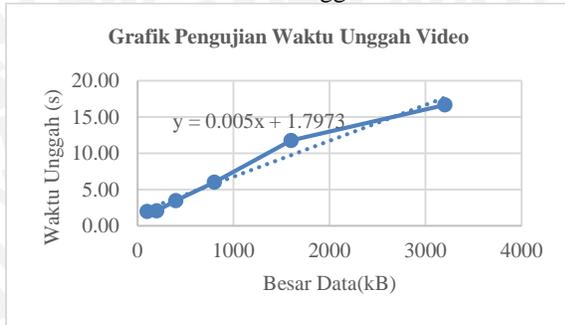
Dari Gambar 21. didapatkan Semakin lama waktu rekaman video, maka semakin besar data video yan tersimpan. Sebaliknya Semakin singkat waktu rekaman video, maka semakin kecil data video yang tersimpan. Dari persamaan regresi linier menunjukkan setiap waktu rekam bertambah 1 s maka besar data naik sebesar 66.616 kB.

**2. Pengujian Waktu Unggah Video**

**Tabel 9.** Pengujian Waktu Unggah Video

| Pengujian ke- | Besar Data (KB) | Waktu Unggah(s) |
|---------------|-----------------|-----------------|
| 1             | 100             | 2.01            |
| 2             | 200             | 2.07            |
| 3             | 400             | 3.49            |
| 4             | 800             | 6.01            |
| 5             | 1600            | 11.78           |
| 6             | 3200            | 16.67           |

Dari Tabel 9. didapatkan waktu unggah ketika besar data 100 KB adalah 2.01 s. Pada besar data 200 KB waktu unggah adalah 2.07 s. Pada besar data 400 KB waktu unggah adalah 3.49 s. Pada besar data 800 KB waktu unggah adalah 6.0 s. Pada besar data 1600 KB waktu unggah adalah 11.78 s. Pada besar data 3200 KB waktu unggah adalah 16.67 s.

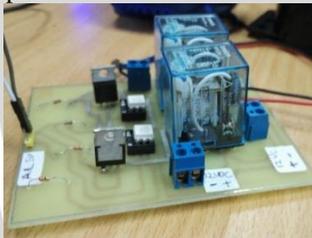


**Gambar 22.** Pengujian Waktu Unggah Video

Dari Gambar 22. didapatkan Semakin besar data rekaman video, maka semakin lama waktu unggah. Dari persamaan regresi linier menunjukkan setiap besar data bertambah 1 kB maka waktu unggah naik sebesar 0.005 s.

**D. Rangkaian Driver Alarm Pada Sistem**

Board Rangkaian Driver alarm hasil rancangan ditunjukkan pada Gambar 23.



**Gambar 23.** Board Rangkaian Driver

**1. Pengujian Tegangan Input Rangkaian Driver**

**Tabel 10.** Pengujian Tegangan Input Rangkaian Driver

| Pengujian Ke- | Tegangan Masukkan (V) | Kondisi Alarm |
|---------------|-----------------------|---------------|
| 1             | 0                     | BUNYI         |
| 2             | 0.5                   | BUNYI         |
| 3             | 1                     | BUNYI         |
| 4             | 1.5                   | MATI          |
| 5             | 2                     | MATI          |
| 6             | 2.5                   | MATI          |
| 7             | 3                     | MATI          |
| 8             | 3.5                   | MATI          |
| 9             | 4                     | MATI          |
| 10            | 4.5                   | MATI          |
| 11            | 5                     | MATI          |

Dari Tabel 10. didapatkan saat rangkaian Driver diberi masukan 0 – 1 V kondisi alarm akan mati. Pada saat rangkaian Driver diberi masukan 1.5 – 5 V kondisi alarm mati.

**2. Pengujian Tegangan Keluaran Pin GPIO Raspberry Pi**

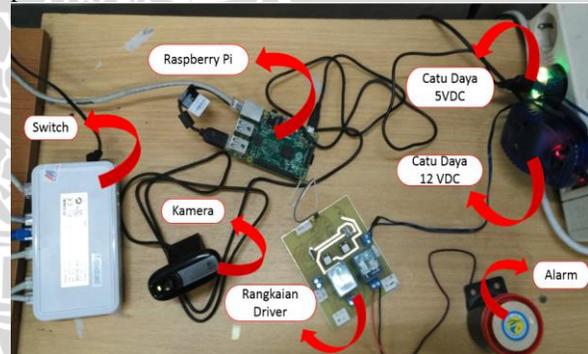
**Tabel 11.** Pengujian Tegangan Keluaran Pin GPIO

| Pengujian Ke-    | Perlakuan      |                   |
|------------------|----------------|-------------------|
|                  | Pin Diaktifkan | Pin Dinonaktifkan |
|                  | VOH (V)        | VOL (V)           |
| 1                | 3.25           | 0.0               |
| 2                | 3.25           | 0.0               |
| 3                | 3.25           | 0.0               |
| <b>Rata-rata</b> | <b>3.25</b>    | <b>0</b>          |

Dari Tabel 11. didapatkan rata-rata tegangan keluaran Pin saat diaktifkan sebesar 3.25. pada saat pin dinonaktifkan tegangan keluaran pin rata-rata sebesar 0 V. dari hasil ini maka tegangan saat pin diaktifkan cukup untuk mematikan alarm. Dan saat pin dinonaktifkan nilai tegangan akan menyebabkan relay pada *normally closed* sehingga alarm berbunyi.

**E. Performa Sistem Keseluruhan**

Antarmuka sistem keseluruhan ditunjukkan pada Gambar 24.



**Gambar 24.** Antarmuka Sistem Keseluruhan

**1. Pengujian Suhu CPU**

**Tabel 12.** Pengujian Suhu CPU

| Pengujian Ke-    | Hari ke- | Suhu (°C)    |
|------------------|----------|--------------|
| 1                | 1        | 42.8         |
| 2                | 2        | 43.3         |
| 3                | 3        | 40.1         |
| 4                | 4        | 40.6         |
| 5                | 5        | 42.8         |
| 6                | 6        | 40.1         |
| 7                | 7        | 42.8         |
| <b>Rata-rata</b> |          | <b>41.79</b> |

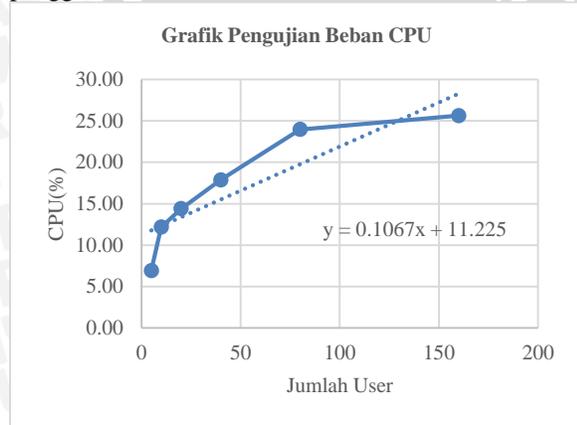
Dari Tabel 12. didapatkan rata – rata suhu CPU yang dilakukan pembacaan selama 7 hari berturut-turut adalah sebesar 41.79 °C. suhu masih bisa dikatakan normal, karena Raspberry Pi dapat bekerja normal selama 7 hari tanpa adanya kendala seperti restart tiba-tiba bahkan mati karena *overheat*. Peran kipas pendingin/*heatsink* sangat membantu untuk menjaga kondisi CPU Raspberry Pi.

**2. Pengujian Beban CPU**

**Tabel 13.** Pengujian Beban CPU

| Jumlah User | CPU (%)<br>Pengujian ke- |      |      | CPU (%)<br>Rata-rata |
|-------------|--------------------------|------|------|----------------------|
|             | 1                        | 2    | 3    |                      |
| 5           | 6.6                      | 7.5  | 6.7  | 6.93                 |
| 10          | 11.6                     | 13.3 | 11.6 | 12.17                |
| 20          | 14.1                     | 15.7 | 13.4 | 14.40                |
| 40          | 18.2                     | 15.7 | 19.7 | 17.87                |
| 80          | 23.8                     | 24.3 | 23.8 | 23.97                |
| 160         | 24.7                     | 26.3 | 25.9 | 25.63                |

Dari Tabel 13. didapatkan rata – rata penggunaan CPU jumlah *user* sebanyak 5 adalah 6.93 %. Pada jumlah *user* sebanyak 10 besar penggunaan CPU rata-rata adalah 12.17 %. Pada jumlah *user* sebanyak 20 besar penggunaan CPU rata-rata adalah 14.40 %. Pada jumlah *user* sebanyak 40 besar penggunaan CPU rata-rata adalah 17.87 %. Pada jumlah *user* sebanyak 80 besar penggunaan CPU rata-rata adalah 23.97 %. Pada jumlah *user* sebanyak 160 besar penggunaan CPU rata-rata adalah 25.63 %.



**Gambar 25.** Pengujian Beban CPU

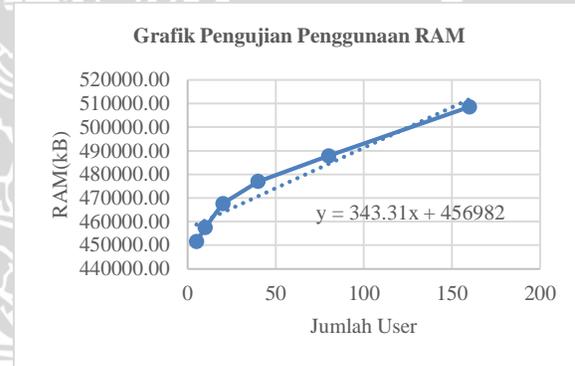
Dari Gambar 25. didapatkan Semakin banyak jumlah *user*, maka semakin besar penggunaan CPU. Sebaliknya Semakin sedikit jumlah *user*, maka semakin kecil penggunaan CPU. Dari persamaan regresi linier menunjukkan setiap jumlah *user* bertambah 1 maka CPU naik sebesar 0.1067 %.

**3. Pengujian Penggunaan RAM**

**Tabel 14.** Pengujian Penggunaan RAM

| Jumlah User | RAM (kB) Pengujian ke- |        |        | RAM(kB)<br>Rata-rata |
|-------------|------------------------|--------|--------|----------------------|
|             | 1                      | 2      | 3      |                      |
| 5           | 449752                 | 451264 | 453432 | 451482.67            |
| 10          | 456064                 | 457508 | 459024 | 457532.00            |
| 20          | 463284                 | 469396 | 470160 | 467613.33            |
| 40          | 472292                 | 478236 | 480424 | 476984.00            |
| 80          | 482900                 | 486884 | 493728 | 487837.33            |
| 160         | 512692                 | 512512 | 500548 | 508584.00            |

Dari Tabel 14. didapatkan rata – rata penggunaan RAM jumlah *user* sebanyak 5 adalah 451482.67 kB. Pada jumlah *user* sebanyak 10 besar penggunaan RAM rata-rata adalah 457532.00 kB. Pada jumlah *user* sebanyak 20 besar penggunaan RAM rata-rata adalah 467613.33 kB. Pada jumlah *user* sebanyak 40 besar penggunaan RAM rata-rata adalah 476984.00 kB. Pada jumlah *user* sebanyak 80 besar penggunaan RAM rata-rata adalah 487837.3 kB. Pada jumlah *user* sebanyak 160 besar penggunaan RAM rata-rata adalah 508584.00 kB.



**Gambar 26.** Pengujian Penggunaan RAM

Dari Gambar 26. didapatkan Semakin banyak jumlah *user*, maka semakin besar penggunaan RAM. Dari persamaan regresi linier menunjukkan setiap jumlah *user* bertambah 1 maka RAM naik sebesar 343.31 kB.

**4. Pengujian Fungsionalitas Kendali Alarm**

**Tabel 15.** Pengujian Fungsionalitas Kendali Alarm

| Perlakuan        | Waktu Eksekusi (s)<br>Pengujian ke- |       |       | Rata-rata (s) |
|------------------|-------------------------------------|-------|-------|---------------|
|                  | 1                                   | 2     | 3     |               |
| Alarm diaktifkan | 1.163                               | 1.66  | 1.168 | 1.33          |
| Alarm dimatikan  | 1.162                               | 1.166 | 1.172 | 1.17          |

Dari Tabel 15. didapatkan rata-rata waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah sebesar 1.33s. Sedangkan rata-rata waktu eksekusi untuk mematikan alarm 1.17 s.

5. Pengujian Fungsionalitas Sistem Keseluruhan

Tabel 16. Pengujian Fungsionalitas Sistem Keseluruhan

| Pengujian Ke- | Pesan LINE |       | Unggah Video |       |
|---------------|------------|-------|--------------|-------|
|               | Terkirim   | Tidak | Terunggah    | Gagal |
| 1             | v          |       | v            |       |
| 2             | v          |       | v            |       |
| 3             | v          |       | v            |       |
| 4             | v          |       | v            |       |
| 5             | v          |       | v            |       |
| 6             | v          |       | v            |       |
| 7             | v          |       | v            |       |
| 8             | v          |       | v            |       |
| 9             | v          |       | v            |       |
| 10            | v          |       | v            |       |
| Error         | 0%         |       | 0%           |       |

Keterangan:

v = hasil yang didapat.

Dari Tabel 16. didapatkan *error* yang terjadi dalam pengiriman pesan *LINE* selama 10 kali pengujian adalah sebesar 0 % maka keberhasilan mengirim pesan *LINE* adalah sebesar 100%. Selain itu didapatkan keberhasilan untuk mengunggah video ke *Google Drive* adalah sebesar 100%. Hasil tidak terlepas dari faktor kestabilan jaringan internet yang digunakan.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

- 1 Ruangannya dapat dipantau dari jarak jauh melalui internet yang tersedia pada domain *julishome.ngrok.io* melalui browser jenis *Mozilla firefox* atau *chrome*. *Web server* yang digunakan adalah *Apache*, dengan performansi *web server* berdasarkan *Packet Loss* hasil yang didapat sangat bagus, berdasarkan *Throughput* hasil yang didapat bagus dan berdasarkan *Delay* hasil yang didapat jelek. Sebagai *server* sistem memiliki performa untuk suhu tergolong normal dengan rata-rata sebesar 41.79 °C, Hasil pengujian menunjukkan beban penggunaan CPU kenaikan sebesar 0.1067%/user dan hasil pengujian beban RAM menunjukkan kenaikan 343.31 kB/user.
- 2 Sistem dapat mendeteksi gerakan menggunakan paket perangkat lunak *Motion* yang berfungsi menganalisa Gambar yang ditangkap oleh kamera dengan kondisi intensitas ruangan minimal sebesar 5 lux. Selain itu saat mendeteksi gerakan sistem mengirim pesan melalui aplikasi *LINE* ke pemilik rumah dengan waktu

pengiriman rata-rata selama 5.15 s. Hasil Pengujian menunjukkan *error* dalam mengirim pesan *LINE* sebesar 0 %.

- 3 Sistem dapat merekam saat terjadi gerakan yang tertangkap kamera. Hasil rekaman akan disimpan di *MicroSD* dan jika ada video baru maka akan diunggah di *Google Drive*. Video akan diunggah oleh paket perangkat lunak *GDrive* yang akan melakukan sinkronisasi *folder* pada *Google Drive* ke *folder* lokal *MicroSD* setiap 1 menit sekali. Hasil pengujian besar data menunjukkan kenaikan sebesar 66.616 kB/s dan waktu unggah video menunjukkan kenaikan sebesar 0.005 s/kB. Hasil Pengujian menunjukkan *error* dalam mengunggah video *Google drive* sebesar 0 %.
- 4 Alarm dapat dikendalikan dari jarak jauh melalui halaman *web* pada domain *julishomengrok.io*. Pemograman *web* menggunakan *PHP* yang akan memberikan perintah ke Pin *GPIO* raspberry melalui *script Python*. Pin *GPIO* terhubung dengan rangkaian *Driver* yang memiliki karakteristik tegangan masukan  $\leq 1$  V untuk membunyikan alarm.

B. Saran

1. Sistem ini dapat dikembangkan menjadi sebuah *home automation* yang dapat mengendalikan semua perangkat listrik dirumah dari jarak jauh.
2. Sistem ini dapat ditambahkan fitur untuk menganalisis hasil tangkapan kamera lebih jauh seperti mendeteksi besar ukuran obyek, membedakan benda dan manusia, membedakan warna dan sebagainya.
3. Sistem ini memungkinkan untuk menambah lebih dari 1 kamera sehingga bisa dilakukan pemantauan banyak ruang.

DAFTAR PUSTAKA

- [1] Ada, lady. 2015. *Introducing the Raspberry Pi 2 - Model B*. Adafruit Industries: Adafruit Learning System.
- [2] Carpedm20. 2014. *line documentation*. <http://github.io/line/genidex.html>. (diakses 27 April 2016)
- [3] Inconshreveable. 2015. *ngrok documentation*. <https://ngrok.com/docs>. (diakses 18 Februari 2016).
- [4] Prasmussen. 2016. *gdrive documentation*. <https://github.com/prasmussen/gdrive/blob/master/README.md>. (diakses 2 maret 2016).
- [5] Tabrani & Yarza Aprizal. 2014. Jurnal: *PERANCANGAN MONITORING JARAK JAUH MENGGUNAKAN RASPBERRY PI DAN WEBCAM BERBASIS INTERNET*. Palembang: STMIK PalComTech.