



TUGAS AKHIR - KI091391

**PENGAWASAN BATAS TERITORIAL PONDOK
PESANTREN MENGGUNAKAN TEKNOLOGI
RASPERRY PI**

**TIRTA TARUNA RAMADHAN
NRP 5110100701**

**Dosen Pembimbing I
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Dosen Pembimbing II
Hudan Studiawan, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014**



UNDERGRADUATE THESES - KI091391

TERRITORIAL LIMITS SUPERVISION BOARDING SCHOOLS USING TECHNOLOGY RASPBERRY PI

**TIRTA TARUNA RAMADHAN
NRP 5110100701**

**Supervisor I
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**Supervisor II
Hudan Studiawan, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2014**

PENGAWASAN BATAS TERITORIAL PONDOK PESANTREN MENGGUNAKAN TEKNOLOGI RASPERRY PI

Nama Mahasiswa : TIRTA TARUNA RAMADHAN
NRP : 5110100701
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Tohari Ahmad, S.Kom., MIT.,
Ph.D.
Dosen Pembimbing 2 : Hudan Studiawan, S.Kom.,
M.Kom.

ABSTRAK

Sesuai dengan perkembangan zaman, pendidikan agama dinilai masyarakat sebagai salah satu standar ilmu yang akhir akhir ini disejajarkan dengan pendidikan formal. Pesantren adalah sebagai lembaga yang menerapkan pendidikan agama terpadu, sedangkan ilmu umum seperti matematika, IPA, IPS, dan lain lain dipandang sebagai alternatif yang menjajikan untuk bentuk sekolah formal yang ada. Untuk mendukung pendidikan agama dan formal, pesantren memberikan pendidikan moral dan ilmu disiplin yang sangat bersinergi dengan pendidikan agama.

Namun menerapkan pendidikan moral dan kedisiplinan sulit diterapkan pada era sekarang, mengingat jumlah santri yang sangat banyak dan tidak sebanding dengan sumber daya ustad yang relatif sedikit, oleh karena itu diperlukannya alat bantu yang dapat membantu menerapkan kedisiplinan.

Salah satu pelanggaran kedisiplinan yang sering dilakukan oleh para santri ialah keluar dari pondok pesantren pada saat jam belajar pondok, oleh karena itu diperlukan alat

bantu modern untuk membantu para ustad mengawasi lingkungan pondok pesantren.

Dengan menggunakan teknologi Raspberry Pi yang sudah mulai berkembang pada era ini. Dengan cara memanfaatkan Raspberry Pi tersebut dengan beberapa alat sensor yang dapat digunakan untuk mengawasi para santri, sensor sensor tersebut ialah sensor PIR (passive infrared), sensor Ultrasonic dan ditambah dengan modul kamera. Tugas akhir ini diharapkan dapat membantu meningkatkan penerapan ilmu kedisiplinan di lingkungan pondok pesantren.

Kata kunci: Modul Kamera, Pondok Pesantren, Raspberry Pi, Sensor.

TERRITORIAL LIMITS SUPERVISION BOARDING SCHOOLS USING TECHNOLOGY RASPBERRY PI

Student's Name : TIRTA TARUNA RAMADHAN
Student's ID : 5110100701
Department : Teknik Informatika FTIF-ITS
First Advisor : Tohari Ahmad, S.Kom., MIT.,
Ph.D.
Second Advisor : Hudan Studiawan, S.Kom.,
M.Kom.

ABSTRACT

Lately, religious education is given equal treatment along with formal education. Pesantren (Islamic Boarding School) is regarded as one of the institution that implementing integrated religious education while at the same time also teaching formal education like science, math, social studies etc. Pesantren can be also treated like an alternative option from formal schools. In order to further giving moral and religious knowledge, pesantren is implementing moral knowledge and usually strict discipline which is mutually enforcing in religious character building.

However, implementing both moral knowledge discipline has proven to be difficult in this modern era. As we know that every year, santri (student in Islamic Boarding School) has been increasing in number. And also, due to the relatively scarce teaching resource made this problem even more difficult. Therefore, invention of a device which could solve this problem is considered important.

One of the most frequently indisciplinary act has ever done by santri is trespassing out from the Islamic Boarding School ground during lecture time. Therefore, a surveillance device which could help teachers monitoring santri and signalling teachers if any of santri trespassing outside the school grounds.

Raspberry Pi using technology that has begun to develop in this era. By way of utilizing the Raspberry Pi with several sensors that can be used to supervise the students, the sensor is a sensor PIR sensor (passive infrared), Ultrasonic sensor and coupled with a camera module. This final project is expected to help improve the application of environmental science discipline in the boarding school.

Keywords: Boarding School, Camera module, Raspberry Pi, Sensor.

LEMBAR PENGESAHAN
PENGAWASAN BATAS TERITORIAL PONDOK
PESANTREN MENGGUNAKAN TEKNOLOGI
RASPBERRY PI

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

TIRTA TARUNA RAMADHAN
NRP: 5110 100 701

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Tohari Ahmad, S.Kom., MIT., Ph.D.
NIP:197505252003121002 (Pembimbing 1)
2. Hudan Studiawan, S.Kom., M.Kom.
NIP:198705112012121003 (Pembimbing 2)

SURABAYA
JULI, 2014

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul ***“PENGAWASAN BATAS TERITORIAL PONDOK PESANTREN MENGGUNAKAN TEKNOLOGI RASPBERRY PI”***.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah melancarkan segala urusan dalam berbagai masalah yang saya hadapi.
2. Ayah, ibu, dan keempat adik saya Shakti, Ayu, Rizki, Intan yang telah memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Bapak Tohari Ahmad, S.Kom, MIT, Ph.D. selaku pembimbing I yang telah membantu dan membimbing penulis dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Bapak Hudan Studiawan, S.Kom, M.Kom. selaku pembimbing II yang telah memberikan motivasi, nasehat, bimbingan dan bantuan yang banyak kepada penulis dalam mengerjakan Tugas Akhir ini.

5. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS dan sekaligus sebagai dosen wali, Bapak Dr.Eng.Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Terima kasih atas bantuan senior senior teknik informatika atas masukan masukan pada Tugas Akhir ini mas Beni, mas Thamrin, mas Panji, mas Hari, mas Chika, mas Dading, mas Mahfud, mas Rozaq, mbak Laili, mas Kholil.
7. Teman-teman administrator laboratorium *Grid Computing* (GCL) dan teman teman senagkatan yang telah berbagi ilmu dan informasi dan motivasi dalam menyelesaikan Tugas Akhir ini Alif, Afif, Valent, Naufal, Rasyid, Zepri, Ubal, Khairy, Bobby, Syahrul, Raditya, Wiwid, Yanto, Iqbal, Arthur, Nabil, Ibam, Fahry, Nabil, Radite, Tora, Grezio, Afrian, Fajri dan teman-teman laboratorium AJK Baihaqi, Idham, Happy, Nandez, Rama, Puspa, Bowo.
8. Teman-teman bermain dota yang telah menemani selama empat tahun ini Ubal, Adri, Wintang, Geri, Samodro, Fahry, Valent, Afif, Iqbal, dan *coach* Zepri.
9. Teman-teman TC angkatan 2010 yang selalu menjaga kebersamaan, kakak-kakak angkatan 2007, 2008, dan 2009 serta adik-adik angkatan 2011 dan 2012 yang membuat penulis untuk selalu belajar.
10. Dan wanita pujaan hati yang telah menjadi penyemangat dan inspirasi penulis, semoga kita dipertemukan pada waktu yang tepat.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juli 2014

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
<i>Abstrak</i>	ix
Abstract	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Manfaat	2
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan Tugas Akhir	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Raspberry Pi.....	7
2.2 HC-SR501 Passive Infrared Sensor (PIR)	9
2.3 Modul Kamera Raspberry Pi NoIR.....	11
2.4 Ranging Detector Mod Distance Sensor HC-SR04	12
2.5 Raspbian OS.....	13
2.6 SMTP (<i>Simple Mail Transfer Protocol</i>).....	14
2.7 <i>Socket</i> GPIO Raspberry Pi	15
2.8 Open CV	17
2.9 SAR (<i>System Activity Report</i>)	18
BAB III PERANCANGAN PERANGKAT LUNAK DAN KERAS.....	19
3.1 Deskripsi Sistem Secara Umum.....	19
3.2 Arsitektur Umum Sistem.....	20
3.3 Perancangan Perangkat Keras	21
3.4 Perancangan Antarmuka	23
3.5 Diagram Alir Aplikasi.....	23
3.6 Perancangan Diagram Alir Data <i>Level 0</i>	25

BAB IV IMPLEMENTASI.....	27
4.1 Lingkungan Implementasi.....	27
4.1.1 Lingkungan Implementasi Perangkat Keras	27
4.1.2 Lingkungan Implementasi Perangkat Lunak	27
4.2 Implementasi Perangkat Keras	28
4.3 Implementasi Perangkat Lunak	31
4.3.1 Implementasi sensor.....	31
4.3.2 Implementasi kamera	33
4.3.3 Implementasi SMTP	34
BAB V PENGUJIAN DAN EVALUASI.....	37
5.1 Lingkungan Uji Coba	37
5.2 Uji Coba Fungsionalitas Sistem	38
5.2.1 Uji Coba Fungsionalitas Pendeteksian Objek Bergerak dengan Sensor PIR	38
5.2.2 Uji Coba Fungsionalitas Pendeteksian Letak Objek dengan Sensor Deteksi Jarak	39
5.2.3 Pengujian Perangkat Sistem Utama	40
5.2.4 Skenario Uji Coba 1	40
5.2.5 Skenario Uji Coba 2	42
5.2.6 Skenario Uji Coba 3	44
5.3 Uji Coba Performa	46
5.3.1 Uji Coba Penggunaan CPU Raspberry Pi	46
5.3.2 Uji Coba Penggunaan Memori Raspberry Pi	48
BAB VI KESIMPULAN DAN SARAN	51
6.1 Kesimpulan.....	51
6.2 Saran.....	51
DAFTAR PUSTAKA.....	53
53	
LAMPIRAN	55
A. Implementasi pada Raspberry Pi	55
B. Kode Program	59
BIODATA PENULIS.....	65

DAFTAR TABEL

Tabel 5.1 Uji Coba Fungsionalitas Sensor PIR Berdasarkan Jarak	38
Tabel 5.2 Uji Coba Fungsionalitas Sensor Deteksi Jarak.....	39
Tabel 5.3 Hasil Uji Coba 1	40
Tabel 5.4 Hasil Uji Coba 2.....	42
Tabel 5.5 Hasil Uji Coba 3.....	44
Tabel 5.6 Tabel Penggunaan CPU	47
Tabel 5.7 Tabel Pemakaian Memori	49

DAFTAR GAMBAR

Gambar 2.1 Raspberry Pi	8
Gambar 2.2 Fitur Raspberry Pi.....	8
Gambar 2.3 Sensor HC-SR501 PIR	10
Gambar 2.4 Bagian bagian PIR [3]	11
Gambar 2.5 Modul Kamera Raspberry Pi NoIR	12
Gambar 2.6 Sensor HC-SR04	12
Gambar 2.7 Cara Kerja Sensor HC-SR04	13
Gambar 2.8 <i>Syntax</i> SMTP objek	14
Gambar 2.9 Pin GPIO Raspberry Pi.....	15
Gambar 2.10 Report Aktifitas Raspberry Pi.....	18
Gambar 3.1 Arsitektur Umum Sistem	20
Gambar 3.2 Perangkat keras sistem pengawas.....	22
Gambar 3.3 Rancangan isi notifikasi.....	23
Gambar 3.4 Alur diagram sistem	24
Gambar 3.5 Diagram alir data level 0	25
Gambar 4.1 Rangkaian Perangkat Keras.....	28
Gambar 4.2 Bagan-Bagan Perangkat Keras	29
Gambar 4.3 Implementasi Sensor PIR	31
Gambar 4.4 Implementasi Range Finder.....	32
Gambar 4.5 Implementasi Get Coordinate.....	33
Gambar 4.6 Implementasi <i>Capture</i>	33
Gambar 4.7 Implementasi HOG Descriptor.....	34
Gambar 4.8 Kode HTML untuk Notifikasi	34
Gambar 4.9 Implementasi Fungsi Kirim.....	35
Gambar 4.10 Hasil Implementasi Fungsi Kirim.....	36
Gambar 4.11 Hasil <i>Maps</i> dari Notifikasi.....	36
Gambar 5.1 Hasil dari Uji Coba 1	41
Gambar 5.2 Hasil dari Uji Coba 1	42
Gambar 5.3 Hasil dari Uji Coba 2.....	43
Gambar 5.4 Hasil dari Uji Coba 2	44
Gambar 5.5 Hasil dari Uji Coba 3	45
Gambar 5.6 Hasil dari Uji Coba 3	46
Gambar 5.7 Penggunaan CPU pada saat siaga.....	47

Gambar 5.8 Penggunaan CPU pada saat sistem dijalankan47
Gambar 5.9 Penggunaan memori Raspberry Pi pada mode siaga
.....48
Gambar 5.10 Penggunaan memori Raspberry Pi pada mode
menjalankan proses utama.....48

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Sesuai dengan perkembangan zaman pendidikan agama dinilai sebagai salah satu standar ilmu yang akhir akhir ini sering diterapkan sejajar dengan pendidikan formal. Pondok pesantren sebagai lembaga yang menerapkan pendidikan agama terpadu, sedangkan ilmu bidang lain dipandang sebagai alternatif yang menjanjikan untuk bentuk sekolah formal yang lain ada. Pondok pesantren juga membantu menerapkan pendidikan moral dan kedisiplinan, pesantren memberikan pendidikan moral yang bersinergi dengan pendidikan agama.

Namun, terkadang pendidikan kedisiplinan sulit diterapkan mengingat banyaknya santri dan sedikitnya sumber daya manusia. Untuk itu diperlukan sebuah alat bantu yang dapat membantu penerapan peraturan kedisiplinan. Salah satu peraturan pondok pesantren yang sering dilanggar oleh santri adalah keluar dari pondok pesantren pada saat jam belajar pondok, oleh karena itu diperlukan bantuan teknologi modern untuk membantu para ustad untuk mengawasi pondok.

Besar harapan penulis ini agar dapat membuat sistem untuk membantu para ustad mengawasi pondok pesantren dalam waktu 24 jam penuh dengan cara memanfaatkan teknologi Raspberry Pi, sensor PIR (*Passive Infra Red*) yang digunakan untuk mendeteksi adanya suatu gerakan, modul kamera yang digunakan untuk mengambil gambar dari suatu objek, dan sensor penghitung jarak agar dapat mengetahui titik terjadinya peristiwa tersebut. Tugas Akhir diharapkan dapat menerapkan teknologi modern di lingkungan pondok pesantren yang dipandang oleh masyarakat

luas bahwa pondok pesantren masih menggunakan fasilitas atau sistem yang masih kuno atau tradisional.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana membuat perangkat sistem yang dapat membantu ustad dalam hal pengawasan?
2. Bagaimana cara agar perangkat sistem dapat mendeteksi santri yang keluar melalui pagar pondok pesantren?
3. Bagaimana cara agar mendapatkan notifikasi dari perangkat sistem yang dibuat?

1.3 Batasan Masalah

Beberapa batasan masalah yang menjadi batas pada Tugas Akhir ini adalah:

1. Perangkat sistem ini hanya mendeteksi objek manusia yang bergerak.
2. Objek manusia pada gambar harus utuh.
3. Notifikasi, gambar, dan koordinat *map* dikirim melalui *email*.
4. Koordinat *map* perangkat Raspberry Pi bernilai statis.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah:

1. Dapat menerapkan kemajuan teknologi Raspberry Pi di lingkungan pondok pesantren.
2. Dapat membantu para ustad dalam hal menerapkan kedisiplinan pondok pesantren.

1.5 Manfaat

Tugas Akhir ini diharapkan mampu membantu meningkatkan keamanan dan kedisiplinan di lingkungan pondok pesantren dengan cara memanfaatkan teknologi Raspberry Pi dan

sensor yang sudah banyak dikembangkan di luar lingkungan pondok pesantren.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap pertama untuk memulai pengerjaan Tugas Akhir yaitu penyusunan proposal. Pada proposal ini, penulis mengajukan gagasan mengenai pemanfaatan teknologi Raspberry Pi, modul kamera Raspberry Pi, sensor PIR, dan sensor deteksi jarak.

2. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan Tugas Akhir sekaligus mempelajarinya. Mulai dari pengumpulan literatur, diskusi, serta pemahaman topik Tugas Akhir di antaranya tentang:

1. Perancangan perangkat sensor dari sistem pengawas perangkat keras Raspberry Pi.
2. Merancang sistem pengawas yang dapat bekerja dengan baik di lingkungan pondok pesantren.

3. Perancangan sistem

Pada tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep sistem yang dibuat. Dengan bekal teori, metode dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses tahap ini.

4. Implementasi perangkat lunak

Dalam pembuatan perangkat, digunakan beberapa teknologi untuk dapat menerapkan rancangan yang sudah ada, di antaranya:

- a. Raspbian
Merupakan sistem operasi Linux Debian yang dirancang untuk Raspberry Pi [1].
- b. Bahasa Pemrograman
Sistem ini dibangun dengan menggunakan bahasa pemrograman Python. Penggunaan bahasa pemrograman diharapkan dapat membantu menangani kebutuhan sistem pada perangkat sensor untuk mendeteksi adanya objek manusia yang bergerak dan kebutuhan perangkat Raspberry Pi saat mengirimkan notifikasi, gambar, dan koordinat *map* ke *email*.

5. Pengujian dan evaluasi

Proses pengujian dilakukan di lingkungan pondok pesantren dengan cara memasang sensor pada tempat tertentu, sehingga apabila terjadinya suatu pelanggaran yang dilakukan oleh santri yaitu melewati pagar pondok pesantren maka sistem akan segera mendeteksi adanya pelanggaran.

6. Penyusunan buku Tugas Akhir.

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari uji coba sistem yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

BAB III. PERANCANGAN PERANGKAT LUNAK DAN KERAS

Bab ini berisi tentang perancangan sistem, *flowchart*, dan perancangan sistem yang akan dibuat. Perancangan yang dibahas meliputi perancangan sistem, pengambilan gambar dan penghitungan jarak setelah terdeteksi oleh sensor infra merah, mengirimkan *email* yang berisi notifikasi, gambar, dan *map*.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

BAB V. PENGUJIAN DAN EVALUASI

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian fungsionalitas dan pengujian performa dalam beberapa skenario. Pengujian fungsionalitas merupakan pengujian jalannya aplikasi sesuai dengan perancangan sistem pada beberapa skenario yang telah ditentukan.

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan implementasi sistem pengawas batas teritorial menggunakan Raspberry Pi, Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat. Tinjauan pustaka ini juga digunakan sebagai dasar yang menunjang pengembangan sistem pengawas ruangan menggunakan Raspberry Pi, sensor, modul kamera.

2.1 Raspberry Pi

Raspberry Pi adalah sebuah *Single Board Computer* seukuran kartu kredit yang dikembangkan oleh Raspberry Foundation dari UK dengan tujuan mempromosikan pengajaran ilmu komputer di sekolah dasar. Raspberry Pi menggunakan *system on a chip* (SoC) dari Broadcom BCM2835, dan juga termasuk prosesor ARM1176JZF-S 700 MHz, GPU VideoCore IV dan penyimpanan data jangka panjang. Meskipun mempunyai hampir semua kemampuan yang dimiliki komputer biasa, namun kemampuan komputasi Raspberry Pi tidak sekuat komputer pada umumnya. Aplikasi-aplikasi *open source* pun bisa dipasang ke dalam komputer mini tersebut seperti LibreOffice, *web browser* ataupun *programming* [2].

Beberapa sistem operasi yang dapat digunakan pada Raspberry Pi:

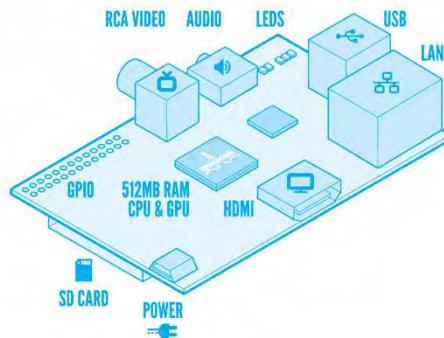
- Raspbian OS
- Arch Linux ARM
- Raspbmc
- OpenELEC

Gambar 2.1 menunjukkan bentuk fisik dari Raspberry Pi. Produksi Raspberry Pi sekarang diproduksi oleh China dan UK.



Gambar 2.1 Raspberry Pi

RASPBERRY PI MODEL B



Gambar 2.2 Fitur Raspberry Pi

Pada Gambar 2.2 ditunjukkan beberapa fitur yang disediakan dari Raspberry Pi yang terdiri dari *port* Lan, HDMI *output* untuk monitor, *power socket* sebagai sumber daya Raspberry Pi sebesar 5 volt 1 ampere, slot kartu micro SD yang mendukung sampai ukuran memori 32GB, *socket* GPIO yang digunakan untuk menghubungkan perangkat tambahan mikrokontroler, *socket* RCA video untuk tampilan ke TV *universal*, *socket* audio untuk luaran suara, lampu indikator *led* yang terdiri

dari lampu indikator *on* dan *off*, indikator jaringan yang sedang berjalan, kemudian terdapat 2 buah *socket* USB yang dapat digunakan untuk berbagai macam kebutuhan seperti *keyboard*, *WIFI*, *mouse*, *flasdisk*, ataupun sebuah modul.

Fungsi lain dari perangkat Raspberry Pi sangat varian diantaranya yaitu:

- *Home Automation* yaitu suatu *control* pusat otomatis yang dapat dikendalikan untuk berbagai fungsi seperti halnya pintu otomatis, pengendali AC ataupun membuat CCTV.
- Dapat dikombinasikan dengan berbagai sensor seerti sensor cahaya, sensor gerak, sensor suhu dan lain lainnya.
- Dapat dijadikan sebagai *server Network Attached Storage*.
- Dapat dikombinasikan dengan perangkat Arduino.

2.2 HC-SR501 Passive Infrared Sensor (PIR)

PIR adalah sebuah perangkat sensor dimana perangkat tersebut berguna untuk mendeteksi adanya gerakan suatu objek di sekitarnya. Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi hanya menerima radiasi sinar infra merah dari luar, ketika sebuah gerakan terdeteksi dengan adanya suatu sumber radiasi dengan suhu tertentu yaitu manusia melewati sumber infra merah disekitarnya dengan suhu berbeda, maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor. Tipe PIR yang khusus digunakan untuk mendeteksi manusia adalah tipe HC-SR501 *Body Sensor Module Pyroelectric Infrared* [3].

HC-SR501 *Body Sensor Module Pyroelectric Infrared* didasarkan pada teknologi inframerah dengan sensitivitas yang tinggi dan modus operasi *ultra-low-voltage*. Sensor ini memiliki *voltage* 5V-20V, *Power Comsumption* 65mA, TTL output mencapai 3.3V, 0V. Suhu yang dideteksi memiliki temperature 20-

70 derajat. jarak sensor bisa mencapai 5-7 meter dengan sudut pancaran mencapai kurang lebih 120 derajat [3].

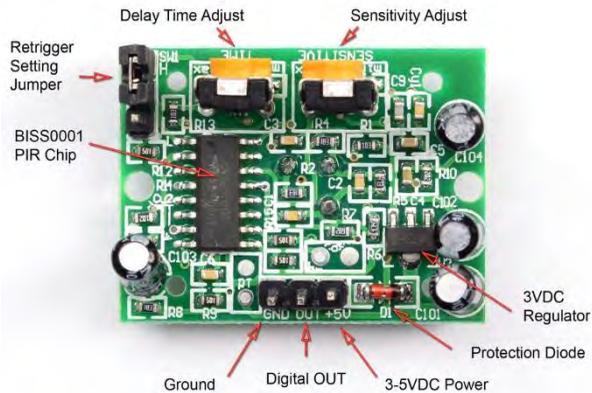


Gambar 2.3 Sensor HC-SR501 PIR

Sensor PIR terdiri dari beberapa bagian seperti ditunjukkan pada gambar 2.4 yaitu:

- BISS0001 PIR Chip (chip decoder yang digunakan)
- Retrigger Setting Jumper
- Delay Time Adjust
- 3VDC Regulator
- Protection Diode
- 3-5VDC Power (VCC)
- Digital OUT
- Ground

Pembacaan sensor hanya akan mengeluarkan logika 0 dan 1, 0 pada saat sensor tidak mendeteksi adanya pancara infra merah dan 1 pada saat sensor mendeteksi infra merah. Sensor PIR dirancang hanya mendeteksi pancaran infra merah dengan panjang gelombang 8-14 mikrometer. Diluar panjang gelombang tersebut maka sensor tidak akan mendeteksi gerakan. Untuk manusia sendiri memiliki suhu badan yang dapat menghasilkan pancaran infra merah dengan panjang gelombang antara 9-10 mikrometer, Secara umum sensor PIR dirancang untuk mendeteksi manusia.



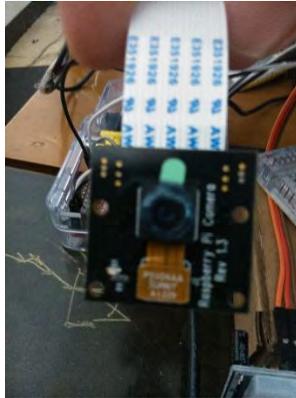
Gambar 2.4 Bagian bagian PIR [3]

2.3 Modul Kamera Raspberry Pi NoIR

Modul kamera Raspberry Pi NoIR adalah modul kamera yang dirancang khusus *add-on* untuk Raspberry Pi yang tidak memiliki *IR cut filter* yang dipasang seperti pada kamera biasa, dengan hanya memasangkan ke salah satu soket kecil di permukaan atas Raspberry Pi. *Interface* soket ini menggunakan *CSI interface* yang khusus, yang mana dirancang khusus untuk *interfacing* pada kamera.

Gambar 2.5 merupakan bentuk fisik dari modul kamera Raspberry Pi NoIR dimana modul tersebut memiliki ukuran sangat kecil, sekitar 25mm x 20mm x 9mm, dan juga memiliki berat hanya 3g. Modul kamera memiliki resolusi 5 megapixel, dan memiliki lensa fokus yang tetap pada *onboard*, kamera ini mampu mengambil gambar paling tinggi 2592 x 1944 pixel gambar statis, dan juga mendukung 1080p30, 720p60 dan 640 x 480p 60/90 video [4].

Versi NoIR adalah hal yang terbaik karena dapat digunakan pada malam hari, karena modul kamera Raspberry Pi NoIR dapat memancarkan cahaya inframerah.



Gambar 2.5 Modul Kamera Raspberry Pi NoIR

2.4 Ranging Detector Mod Distance Sensor HC-SR04

Sensor HC-SR04 adalah sensor yang berfungsi sebagai deteksi jarak antara sensor dengan sebuah objek.



Gambar 2.6 Sensor HC-SR04

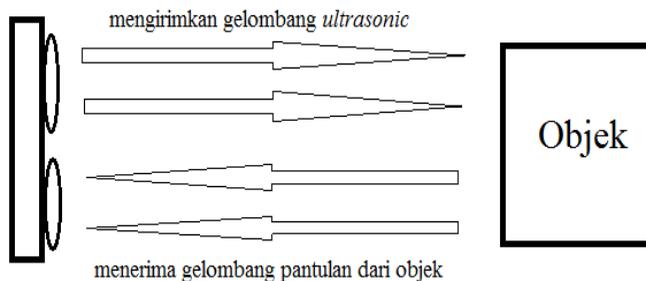
Gambar 2.6 menunjukkan bentuk fisik dari sensor. Jarak maksimum yang dapat terdeteksi oleh sensor ini sekitar 5 meter dengan akurasi hingga 3mm. Seperangkat sensor ini terdiri dari

ultrasonic transmitters, receiver dan *control circuit* [5]. Prinsip dasar kerja dari sensor ini adalah:

1. Menggunakan IO trigger untuk paling sedikit dengan sinyal sebesar 10us.
2. Secara otomatis mengirimkan gelombang *ultrasonic* sebesar 40 kHz dan mendeteksi apakah ada sinyal balik.
3. Jika terdapat sinyal yang kembali, melalui sinyal tinggi maka waktu output durasi IO adalah waktu pengiriman *ultrasonic* untuk kembali.

$$\text{Test distance} = (\text{high level time} \times \text{velocity of sound}) / 2$$

(34000cm/s) / 2



Gambar 2.7 Cara Kerja Sensor HC-SR04

Secara singkat dijelaskan bahwa cara kerja sensor HC-SR04 untuk mendeteksi jarak ialah dengan cara memancarkan gelombang ultrasonic dan apabila terdapat objek didepannya maka gelombang ultrasonic akan memantul kembali ke sensor HC-SR04.

2.5 Raspbian OS

Raspbian adalah sistem operasi Linux *distro* Debian yang dioptimalkan untuk penggunaan perangkat komputer mini Raspberry Pi. Sistem operasi ini memiliki beberapa *program* standard dan beberapa alat bantu untuk dapat menjalankan perangkat keras dari komputer min Raspberry Pi. Di dalam sistem

operasi Raspbian ini sudah lengkap daripada sistem operasi yang murni pada komputer umum, karena Raspbian berisi lebih dari 35.000 paket dan *pre-compiled* perangkat lunak yang tersaji dalam bentuk format yang mudah untuk diinstalasi dalam komputer mini Raspberry Pi [1].

Dengan 35.000 paket Raspbian, Raspbian sudah memiliki performa yang bagus yang dibangun dan telah selesai pada juni tahun 2012. Bagaimanapun, Raspbian pi masih dibawah developer aktif dan masih membutuhkan perkembangan yang jauh lebih baik, untuk paket paket lain memungkinkan masih bisa dibangun lagi.

2.6 SMTP (*Simple Mail Transfer Protocol*)

Simple Mail Transfer Protocol atau SMTP adalah sebuah protocol yang dimana berfungsi untuk membantu dalam hal pengiriman *email* dan penjaluran pengiriman *email*. Dalam bahasa pemrograman Python disediakan `smtplib` modul, yang dimana dapat menerjemahkan sebuah objek *client session* SMTP yang digunakan untuk melakukan pengiriman *email* ke semua perangkat internet yang mendukung pembacaan SMTP Daemon [6].

```
Import smtplib
smtpObj = smtplib.SMTP([host [, port [, local_hostname]])
```

Gambar 2.8 Syntax SMTP objek

Pada Gambar 2.8 adalah sebuah *syntax* sederhana untuk membuat sebuah SMTP objek, yang dimana dapat digunakan untuk mengirim *email*.

Berikut ini adalah detail parameter Gambar 2.8

- Host: tempat dimana untuk menjalankan SMTP *server*, dapat juga digunakan sebagai penggunaan alamat IP dari sebuah *domain*.
- Port: jalur data yang digunakan untuk pertukaran data SMTP *server*.

- *Local_hostname*: berisi *webmail service*, seperti contoh menggunakan google (smtp.google.com) atau *webmail its* (smtp.its.ac.id).

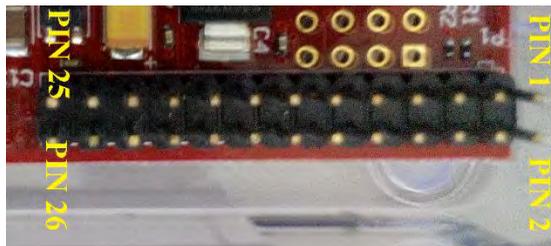
SMTP objek memiliki sebuah metode yang disebut “*sendmail*” yang digunakan untuk mengirimkan pesan. Terdapat 3 buah parameter yaitu:

- *Sender*: parameter yang berisi alamat dari pengirim.
- *Receiver*: parameter yang berisi alamat untuk penerima.
- *Message*: parameter yang berisi pesan dengan format tertentu.

2.7 Socket GPIO Raspberry Pi

General Purpose Input/Output atau yang biasa disebut *socket GPIO* adalah pin yang dimana berfungsi sebagai *socket input/output* dari perangkat keras dan dapat dikendalikan melalui perangkat lunak. *Socket GPIO* dapat digunakan untuk menghubungkan beberapa perangkat tambahan seperti sensor infra merah, layar LCD, lampu LED ataupun untuk menambahkan perangkat Arduino. Kemampuan dari *socket GPIO* adalah antara lain sebagai berikut:

- Pin GPIO dapat diaktifkan dan di non aktifkan.
- Pin GPIO dapat dikonfigurasi untuk menerima *input* dan mengirim *output*.
- Nilai *input* dari GPIO adalah bernilai 1 (*high*) dan 0 (*low*).



Gambar 2.9 Pin GPIO Raspberry Pi

Pada Gambar 2.9 menunjukkan beberapa pin dari pin GPIO yang terdapat di Raspberry Pi, setiap pin memiliki kegunaannya masing-masing dan digunakan untuk berbagai perangkat tambahan [7]. Berikut ini adalah fungsi-fungsi dari pin GPIO:

- Pin 1 berfungsi sebagai pin *power* dengan daya 3.3V
- Pin 2 berfungsi sebagai pin *power* dengan daya 5V
- Pin 3 berfungsi sebagai pin GPIO2
- Pin 4 berfungsi sebagai pin *power* dengan daya 5V
- Pin 5 berfungsi sebagai GPIO3
- Pin 6 berfungsi sebagai pin *ground*
- Pin 7 berfungsi sebagai pin GPIO4
- Pin 8 berfungsi sebagai pin GPIO14 dan dapat berfungsi sebagai pin untuk melakukan pengiriman data yang sama seperti pada kabel serial
- Pin 9 berfungsi sebagai *ground*
- Pin 10 berfungsi sebagai pin GPIO15 dan dapat berfungsi sebagai pin untuk melakukan penerimaan data yang sama seperti pada kabel serial
- Pin 11 berfungsi sebagai pin GPIO17
- Pin 12 berfungsi sebagai pin GPIO18
- Pin 13 berfungsi sebagai pin GPIO27
- Pin 14 berfungsi sebagai pin *ground*
- Pin 15 berfungsi sebagai pin GPIO22
- Pin 16 berfungsi sebagai pin GPIO23
- Pin 17 berfungsi sebagai pin *power* dengan daya 3.3V
- Pin 18 berfungsi sebagai pin GPIO24
- Pin 19 berfungsi sebagai pin GPIO20
- Pin 20 berfungsi sebagai pin *ground*
- Pin 21 berfungsi sebagai pin GPIO9
- Pin 22 berfungsi sebagai pin GPIO25
- Pin 23 berfungsi sebagai pin GPIO11
- Pin 24 berfungsi sebagai pin GPIO8
- Pin 25 berfungsi sebagai *ground*
- Pin 26 berfungsi sebagai pin GPIO7

2.8 Open CV

OpenCV adalah suatu *library* gratis yang dikembangkan oleh *developer-developer* Intel Corporation. *Library* ini terdiri dari fungsi-fungsi *computer vision* dan API (*Application Programming Interface*) untuk *image processing high level* maupun *low level* dan sebagai optimasi aplikasi *realtime*. OpenCV sangat disarankan untuk programmer yang akan berkecukupan pada bidang *computer vision*, karena *library*nya mampu membuat aplikasi yang handal dibidang *digital vision* dan mempunyai fitur yang mirip dengan cara pengolahan visual pada manusia [8].

Berikut ini adalah beberapa fitur pada *library* OpenCV:

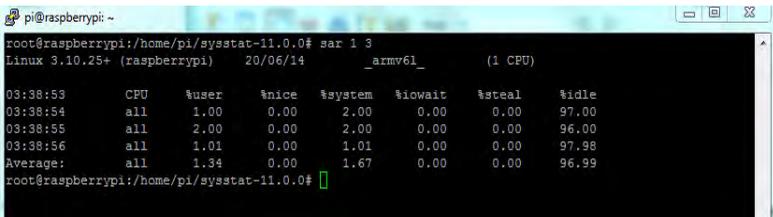
- Manipulasi data gambar (alokasi memori, melepaskan memori, setting serta konversi gambar)
- Image/Video I/O (Bisa menggunakan kamera yang sudah didukung oleh *library* ini)
- Manipulasi matriks dan vektor serta terdapat juga *routines* linier algebra (*products, solvers, eigenvalues, SVD*)
- *Image processing* dasar (filtering, edge detection, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi, *histograms, image pyramids*)
- Analisis struktural
- Kalibrasi kamera
- Pendeteksian gerak
- Pengenalan objek (Hewan, Manusia, Tumbuhan dll)
- Basic GUI (*Display* gambar/video, mouse/keyboard kontrol, *scrollbar*)
- *Image Labelling* (*line, conic, polygon, text drawing*)

Dengan memanfaatkan *library* pada openCV yaitu fungsi HOG (*Histogram Oriented of Gradient*) *descriptor*, maka suatu image dapat dideteksi objek tersebut atau manusia atau bukan.

HOG *descriptor* sendiri merupakan suatu fitur dari gambar yang digunakan untuk menghitung vektor gradien pada area tertentu sehingga dihasilkan output berupa vektor yang nantinya diklasifikasi oleh *support vector machine* [9].

2.9 SAR (System Activity Report)

SAR (*System Activity Report*) adalah aplikasi yang berbasis Solaris dan digunakan untuk memantau kinerja dari suatu proses perangkat CPU seperti contoh digunakan untuk memantau kinerja aktivitas CPU, memantau penggunaan memori, memantau preangkat yang sedang bekerja, maupun dapat digunakan untuk memantau kinerja dari suatu jaringan yang sedang terhubung dengan CPU. SAR didistribusikan melalui Linux dan melalui Linux dan melalui paket aplikasi sysstat [10].



```

pi@raspberrypi: ~
root@raspberrypi:/home/pi/sysstat-11.0.0# sar 1 3
Linux 3.10.25+ (raspberrypi) 20/06/14 _armv6l_ (1 CPU)
03:38:53 CPU %user %nice %system %iowait %steal %idle
03:38:54 all 1.00 0.00 2.00 0.00 0.00 97.00
03:38:55 all 2.00 0.00 2.00 0.00 0.00 96.00
03:38:56 all 1.01 0.00 1.01 0.00 0.00 97.98
Average: all 1.34 0.00 1.67 0.00 0.00 96.99
root@raspberrypi:/home/pi/sysstat-11.0.0#

```

Gambar 2.10 Report Aktifitas Raspberry Pi

Pada Gambar 2.10 ditunjukkan bagaimana contoh pemakaian aplikasi SAR pada Raspberry Pi yang sedang dalam posisi siaga, terlihat total presentase dari pemakaian CPU yang sedang terpakai pada saat Raspberry Pi sedang dalam kondisi siaga.

BAB III

PERANCANGAN PERANGKAT LUNAK DAN KERAS

Perancangan merupakan bagian penting dari pembuatan perangkat lunak yang berupa perencanaan-perencanaan secara teknis mengenai sistem yang akan dibuat. Bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Penjelasan mencakup deskripsi umum aplikasi hingga perancangan proses, alur sistem.

3.1 Deskripsi Sistem Secara Umum

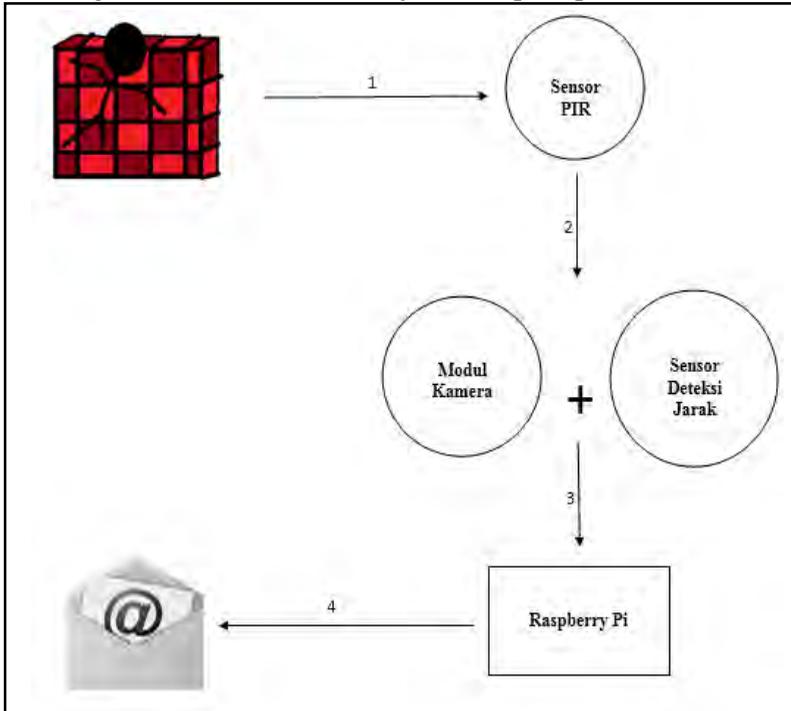
Pada Tugas Akhir ini akan dibangun suatu sistem pengawas batas teritorial pondok pesantren berbasis Raspberry Pi. Sistem ini dirancang dengan studi kasus untuk mengawasi batas teritorial pondok pesantren dari pelanggaran santri yang keluar pondok tanpa izin dan lemahnya pengawasan dari para ustad.

Sistem ini, pertama kali dilakukan ialah mengambil data dari input sensor infra merah yang dimana berfungsi sebagai penangkap suhu panas tubuh seseorang yang sedang keluar dari pondok pesantren yang sedang diawasi oleh rangkaian alat ini, sensor infra merah akan bernilai logika 0 dan 1, 0 saat sensor tidak mendeteksi adanya pancaran infra merah dan 1 saat sensor mendeteksi. Kemudian setelah sistem dapat mendeteksi adanya suatu gerakan maka modul kamera akan mendeteksi objek yang bergerak tersebut apakah manusia atau bukan setelah gambar objek bergerak tersebut diambil gambarnya. Setelah terdeteksi objek tersebut adalah manusia, maka terdapat sensor deteksi jarak yang akan mendeteksi jarak antara rangkaian alat ini dengan objek manusia tersebut, setelah mendapatkan jarak antara alat dengan objek manusia maka jarak tersebut akan ditambahkan dengan letak koordinat alat agar dapat menemukan koordinat *map* objek manusia tersebut.

Untuk Raspberry Pi itu sendiri berfungsi sebagai pusat dari semua perangkat yang dihubungkan baik itu sensor infra merah, modul kamera, dan sensor deteksi jarak.

3.2 Arsitektur Umum Sistem

Rancangan arsitektur sistem ditunjukkan seperti pada Gambar 3.1.



Gambar 3.1 Arsitektur Umum Sistem

Pada Gambar 3.1 dijelaskan alur kerja dari arsitektur umum sistem adalah sebagai berikut:

1. Dengan cara mendeteksi suhu panas dari tubuh seseorang yang akan keluar dari pondok pesantren tersebut dengan

- menggunakan sensor infra merah yang sudah diintegrasikan melalui *socket* GPIO dengan Raspberry Pi.
2. Setelah sensor infra merah menangkap suhu panas tubuh manusia yang tidak sengaja terkena pancaran infra merah tersebut, maka kamera akan mengambil gambar objek bergerak tersebut, lalu gambar tersebut dideteksi apakah objek tersebut benar-benar manusia atau bukan. Setelah terdeteksi objek tersebut adalah manusia maka gambar akan disimpan di *file storage* Raspberry Pi. Kemudian sensor deteksi jarak akan melakukan tugasnya yaitu menghitung jarak antara rangkaian alat ini dengan objek manusia.
 3. Semua input data yang telah didapat dari sensor infra merah, modul kamera dan sensor deteksi jarak akan disimpan di memori Raspberry Pi.
 4. Data input yang telah disimpan dalam *storage* Raspberry Pi. Kemudian akan dikirim ke *email* agar para ustad dapat secara mudah dapat mengetahui informasi dari rangkaian alat ini.

3.3 Perancangan Perangkat Keras

Perancangan perangkat keras pada penelitian ini menjelaskan tentang perangkat keras apa saja yang digunakan untuk mengimplementasikan Tugas Akhir ini. Tugas akhir ini membutuhkan perangkat keras sebagai berikut:

- 1 buah mini komputer Raspberry Pi model B versi China.
- 1 buah sensor PIR tipe HC-SR501
- 1 buah sensor deteksi jarak HC-SR04
- 1 buah modul kamera Raspberry Pi NoIR
- 1 buah *breadboard* 400 *tie point*
- 11 buah kabel pelangi *male to female*
- 1 buah kabel LAN
- 1 buah *powerbank* oren 2600mah
- 1 buah SDHC *card* Sandisk Ultra 8GB

- 1 buah kabel konverter VGA ke HDMI



Gambar 3.2 Perangkat keras sistem pengawas

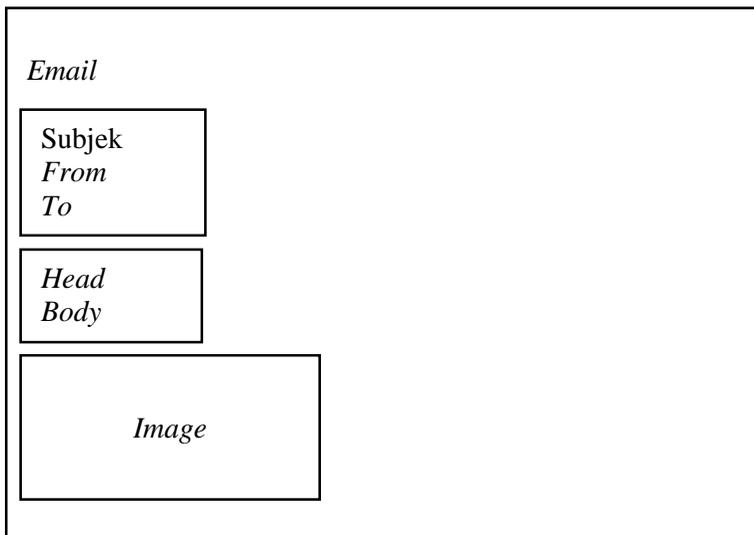
Pada Gambar 3.2 menunjukkan beberapa kabel pelangi yang akan digunakan untuk menghubungkan *socket* GPIO ke *breadboard*. Perangkat keras diatas juga menunjukkan Raspberry Pi sebagai mainboard atau sebagai pusat dari semua perangkat yang akan dihubungkan baik itu sensor PIR, sensor deteksi jarak, *wireless* adapter maupun dengan modul kamera Raspberry Pi. Sensor PIR bertugas sebagai mendeteksi gerakan dari seseorang yang telah melewati rangkaian alat tersebut. Kemudian sensor deteksi jarak bertugas untuk menghitung jarak antara Raspberry Pi dengan objek yang tertangkap oleh kamera. Untuk *storage*, Raspberry Pi menggunakan SDHC *card* sebagai tempat penyimpanan berkas.

Seluruh perangkat sistem ini memakai daya suplai listrik yang berasal dari *powerbank* berukuran 2600mah. Untuk perantara ke jaringan internet, perangkat sistem ini menggunakan *wireless*

USB *adapter* agar dapat mengirimkan notifikasi berupa gambar dan teks melalui *email*.

3.4 Perancangan Antarmuka

Perancangan antarmuka mempunyai tujuan untuk memudahkan dalam memanfaatkan sistem ini dengan mudah. Rancangan antarmuka pada sistem ini hanya meliputi antarmuka hasil notifikasi yang akan diterima pengguna berupa isi notifikasi pada *email*. Rancangan antarmuka notifikasi ditunjukkan pada Gambar 3.3 yang meliputi adanya subjek, pengirim, penerima, *head*, *body*, dan *file attachment* berupa gambar. Untuk isi notifikasi disusun menggunakan bahasa HTML agar terlihat rapi dan mudah dipahami oleh penerima notifikasi.

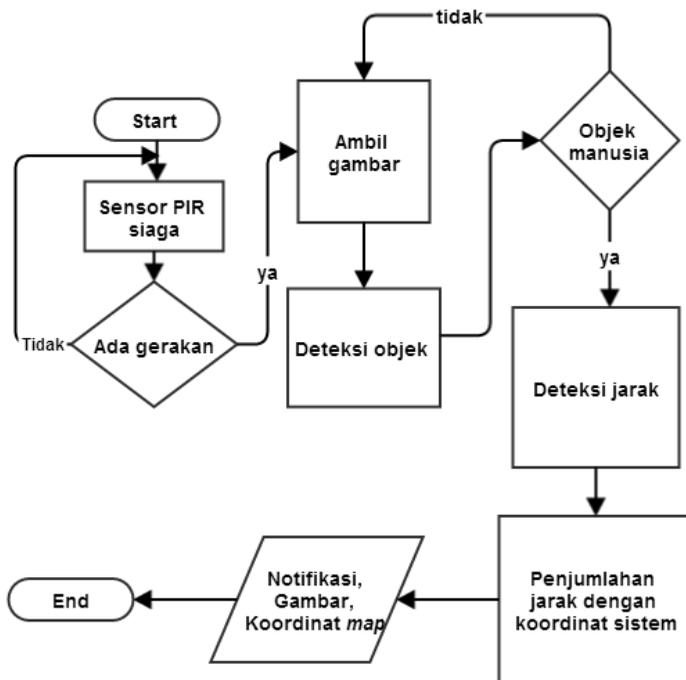


Gambar 3.3 Rancangan isi notifikasi

3.5 Diagram Alir Aplikasi

Pada Gambar 3.4 menunjukkan diagram alir dari proses kerja dari sistem pengawas menggunakan Raspberry Pi. Pertama

proses inialisasi *start* kemudian mengaktifkan sensor PIR dan modul kamera dari Raspberry Pi agar sensor dan kamera pada posisi siaga, lalu jika terjadi gerakan maka sensor mendeteksi gerakan tersebut dan langsung memberikan sinyal ke Raspberry Pi agar modul kamera melakukan perintah ambil gambar, setelah itu gambar akan dideteksi menggunakan HOG *descriptor* agar dapat mengetahui apakah objek tersebut manusia atau bukan. Kemudian sensor deteksi jarak menghitung jarak antara objek manusia tersebut dengan sensor. Setelah mendapatkan gambar yang telah dicek dan jarak, selanjutnya hasil jarak dijumlahkan dengan koordinat Raspberry Pi untuk menentukan titik objek manusia. Kemudian Raspberry Pi juga dapat mengirim notifikasi, gambar, dan koordinat *map* ke *email* tujuan.

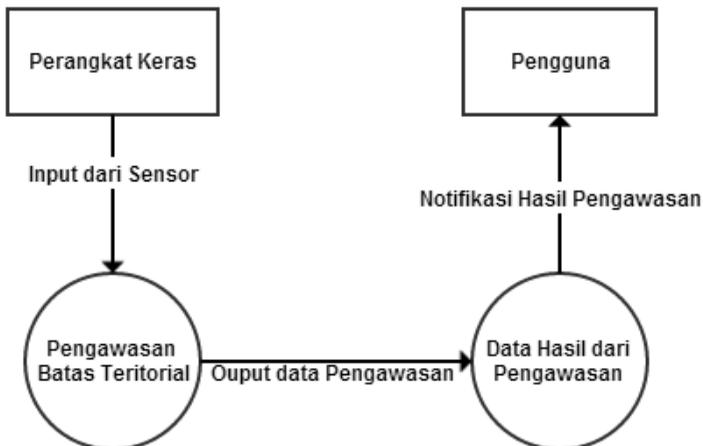


Gambar 3.4 Alur diagram sistem

3.6 Perancangan Diagram Alir Data *Level 0*

Diagram alir data *level 0* menggambarkan fungsionalitas sistem beserta aktor yang terlibat. Seperti yang terlihat pada Gambar 3.5 ditunjukkan bahwa terdapat satu pengguna yang menggunakan sistem ini. Pengguna akan menerima hasil dari pengawasan batas teritorial berupa notifikasi, gambar dan koordinat *map* melalui *email* yang telah didaftarkan pada sistem ini.

Perangkat sistem ini diawali dengan *input* dari hasil pendeteksian gerak melalui sensor infra merah. Kemudian nilai dari *input* sensor infra merah akan *trigger* kamera dan sensor deteksi jarak yang terhubung dengan Raspberry Pi untuk melakukan perintah ambil gambar dan mendeteksi objek gambar tersebut, kemudian sensor deteksi jarak akan menghitung jarak antara perangkat sistem dengan objek manusia. Setelah mendapatkan data informasi pengawasan batas teritorial, maka data tersebut dikirim ke pengguna melalui *email*.



Gambar 3.5 Diagram alir data level 0

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung pengembangan sebagai berikut.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah komputer dan perangkat Raspberry Pi. Spesifikasi dari perangkat tersebut adalah sebagai berikut:

- Komputer AMD FX – 4300 Quad-Core 3.8GHz dan 4 GB memori DDR3.
- Laptop ASUS K401 Core 2 Duo
- Sensor PIR, sensor deteksi jarak, modul kamera
- Raspberry Pi model B versi China.

4.1.2 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Microsoft Windows 7 Ultimate 64 bit sebagai sistem operasi.
- Raspbian Wheezy 3.10.
- Bahasa pemrograman Python mengimplementasikan aplikasi dan mengintegrasikan perangkat keras yang terhubung.
- PuTTY sebagai aplikasi untuk melakukan *remote console* terhadap Raspberry Pi.

4.2 Implementasi Perangkat Keras

Pada subbab ini akan dijelaskan bagaimana implementasi perangkat lunak pada Raspberry Pi agar semua perangkat keras dapat bekerja dengan baik dan sesuai dengan susunan yang ditentukan agar sistem pengawasan batas teritorial ini dapat berjalan dengan sempurna.



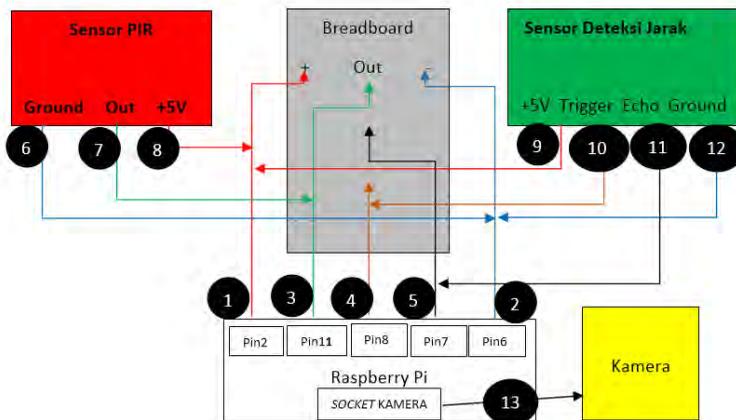
Gambar 4.1 Rangkaian Perangkat Keras

Pada Gambar 4.1 menunjukkan hasil pembuatan perangkat keras yang akan digunakan pada sistem pengawasan batas teritorial pondok pesantren menggunakan Raspberry Pi, perangkat keras ini diawali dengan sebuah *prototype* untuk menguji apakah perangkat keras dapat berfungsi sebagai sistem pengawas batas teritorial pondok pesantren. berikut penjabaran dari perangkat keras yang digunakan:

- 1 buah mini komputer Raspberry Pi model B versi China.
- 1 buah sensor PIR tipe HC-SR501
- 1 buah sensor deteksi jarak HC-SR04
- 1 buah modul kamera Raspberry Pi NoIR
- 1 buah *breadboard* 400 *tie point*
- 11 buah kabel pelangi *male to female*

- 1 buah *wireless USB adapter* EW-7811Un
- 1 buah *powerbank* oren 2600mah
- 1 buah *SDHC card* Sandisk Ultra 8GB
- 1 buah *casing* mika persegi

Pengujian fungsi perangkat sensor melibatkan semua komponen perangkat keras. *Prototype* perangkat keras pada tugas akhir ini menggunakan sebuah komputer mini Raspberry Pi model B versi China yang dilengkapi dengan kartu SD berukuran 8GB dan sudah memiliki sistem operasi Raspbian Wheezy, sensor PIR, sensor deteksi jarak, modul kamera Raspberry Pi NoIR, *breadboard* 400 *tie point*, dan komputer mini Raspberry Pi yang berguna sebagai pusat dari perangkat perangkat keras lainnya.



Gambar 4.2 Bagan-Bagan Perangkat Keras

Pada Gambar 4.2 menunjukkan bagan *socket* GPIO untuk menghubungkan komputer mini Raspberry Pi dengan sensor PIR dan sensor deteksi jarak. Dibutuhkan beberapa kabel GPIO pelangi untuk menghubungkan *socket* dari kedua sensor ke Raspberry Pi melalui beberapa *socket* GPIO. Dengan menggunakan *breadboard* pin dari kedua sensor tersebut dapat dijalankan bersamaan dengan cara: kabel

socket GPIO nomor 2 dan 6 pada bagian *vcc* dan *ground* ditancapkan pada *breadboard* dengan tanda positif untuk *vcc* dan negatif untuk *ground* seperti pada nomor 1 dan 2. Sedangkan untuk *socket* GPIO11, GPIO7, GPIO8 ditancapkan pada *breadboard* bagian *output* seperti pada nomor 3, 4, dan 5.

Pada sensor PIR terdapat tiga pin utama agar sensor dapat berjalan dengan baik. Ketiga pin tersebut ialah *vcc*, *ground*, dan *digital pin out*. Pin pertama pada sensor PIR yaitu *vcc* yang digunakan sebagai pin *power* yang harus diberi daya 5 volt maka ditancapkan pada lubang *breadboard* yang sejajar dengan kabel nomor 8 dan sejajar dengan kabel nomor 1 yang telah tertancap pada tanda positif, lalu pin *ground* pada sensor PIR ditancapkan pada lubang *breadboard* dengan kabel nomor 6 dan sejajar dengan dengan kabel nomor 2 dan pin yang terakhir yaitu pin *digital out* pada sensor PIR akan dihubungkan ke lubang *breadboard* dengan kabel nomor 7 dan sejajar dengan kabel *socket* GPIO11 yang telah tertancap pada *breadboard* dengan kabel nomor 3.

Untuk sensor deteksi jarak memiliki empat pin, kedua pin sama dengan dari dua pin sensor PIR yaitu *vcc* dan *ground*, untuk itu pemasangan kabel ditancapkan sejajar dengan tanda positif dan negative dengan kabel nomor 9 dan 12. Lalu pin yang ketiga yaitu pin *trigger* akan ditancapkan ke *breadboard* dengan kabel nomor 10 dan ditancapkan sejajar dengan tempat pemasangan pin GPIO8 dari *socket* Raspberry Pi dengan kabel nomor 4, kemudian untuk pin yang terakhir yaitu pin *echo* ditancapkan pada *breadboard* dengan kabel nomor 11 sejajar dengan pin *socket* GPIO7 yang telah ditancapkan di *breadboard*. Untuk pemasangan modul kamera Raspberry Pi NoIR dipasang tepat pada *socket* kamera Raspberry Pi pada nomor 13.

Rangkaian perangkat sistem ini diletakkan sekitar 1-2 meter dari kemungkinan tempat pelanggaran santri dan diletakkan diatas tembok, agar keseluruhan perangkat dapat bekerja dengan baik maka membutuhkan suplai daya listrik dari sebuah *powerbank* yang telah terisi dayanya agar tidak terjadi kekurangan daya pada saat uji coba. Dan menggunakan *wifi adapter* sebagai koneksi internet.

4.3 Implementasi Perangkat Lunak

Pada subbab ini akan dijelaskan bagaimana implementasi perangkat lunak pada Raspberry Pi agar semua perangkat keras dapat bekerja dengan baik dan sesuai dengan susunan yang ditentukan agar sistem pengawasan batas teritorial ini dapat berjalan dengan sempurna.

4.3.1 Implementasi sensor

- Implementasi sensor PIR

Fungsi monitor pada Gambar 4.3 menunjukkan kondisi apabila *GPIO.input(self._gpio_input)* bernilai satu maka terdeteksi adanya suatu gerakan dan melakukan perintah selanjutnya, setelah itu sensor kembali ke mode siaga kembali.

```
def monitor(self):
    while self.active:
        self._current_state=
        GPIO.input(self._gpio_input)
        if self._current_state == 1:
            self._notify()
            self._current_state = 0
            time.sleep(self._delay)
        time.sleep(0.01)
```

Gambar 4.3 Implementasi Sensor PIR

- Implementasi sensor deteksi jarak

Untuk melakukan proses deteksi jarak pada sistem ini telah ditunjukkan pada Gambar 4.4, apabila *GPIO.output(self._gpio_trigger, False)* bernilai false maka sensor memulai untuk mengirimkan gelombang *ultrasonic* ke objek manusia dan selama *GPIO.input(self._gpio_echo)* bernilai 0 artinya pin *echo* belum mendapatkan pantulan gelombang *ultrasonic* maka gelombang *ultrasonic* masih dalam status dikirim ke objek manusia tersebut. Kemudian

apabila `GPIO.input(self._gpio_echo)` bernilai 1 maka pin *echo* telah menerima pantulan gelombang *ultrasonic* dari objek.

```
GPIO.output(self._gpio_trigger, True)
time.sleep(0.00001)
GPIO.output(self._gpio_trigger, False)
start = time.time()
while GPIO.input(self._gpio_echo) == 0:
    start = time.time()
while GPIO.input(self._gpio_echo) == 1:
    stop = time.time()
elapsed = stop-start
distance = elapsed * 34000
distance2 = float(distance) /float( 2)
```

Gambar 4.4 Implementasi Range Finder

Kemudian mencari *time elapsed* yang didapatkan dari selisih waktu mulai gelombang terkirim dengan waktu gelombang kembali. Kemudian *time elapsed* dikonversi ke centimeter, setelah didapatkan jarak dengan ukuran *centimeter* jarak akan dibagi dua karena jarak total dari *time elapsed* adalah jarak gelombang dikirim dan kembali.

- Implementasi *get coordinate*

Untuk menemukan titik koordinat objek pada sistem ini ditunjukkan pada Gambar 4.5, cukup dengan menggunakan *library* pada Python yaitu *geopy* [11]. Untuk koordinat sistem diinisialisasi secara statis, dan untuk *bearing* bernilai 90 derajat. Untuk parameter dari *VincentyDistance* [11] adalah jarak dalam hitungan kilometer, untuk *destination* parameternya adalah *lat1*, *lat2* dan *bearing*.

```

lat1 = -7.279725
lon1 = 112.797120
b = 90.0
d = distance
origin = geopy.Point(lat1, lon1)
destination =
VincentyDistance(kilometers=d).destination(origin
, b)
lat2, lng2 = destination.latitude,
destination.longitude

```

Gambar 4.5 Implementasi Get Coordinate

4.3.2 Implementasi kamera

- Implementasi *Capture Image*

Proses mengambil gambar dari modul kamera Raspberry Pi NoIR dapat dijalankan melalui potongan *code* seperti pada Gambar 4.6, dengan menggunakan *library* picamera [12] dapat secara mudah melakukan perintah *capture image*. *My_file* adalah variabel berisi hasil *capture image*.

```

self._camera = picamera.PiCamera()
my_file = join(self._base_directory,
self._current_subdirectory,
str(self._current_index) + ".jpg")
self._camera.capture(my_file, 'jpeg', quality=50)

```

Gambar 4.6 Implementasi Capture

- Implementasi *HOG descriptor*

Dengan menggunakan *library* openCV dan memanfaatkan fiturnya yaitu *HOG descriptor* [13]. Perangkat sistem ini dapat mendeteksi keberadaan objek manusia pada suatu gambar. Gambar 4.7 adalah potongan *code* yang digunakan untuk mendeteksi suatu gambar dengan ukuran *multi-scale windows*.

```

storage = CreateMemStorage(0)
img = LoadImage(my_file)
found = list(HOGDetectMultiScale(img,
storage, win_stride=(8,8),padding=(32,32),
scale=1.05, group_threshold=2))

```

Gambar 4.7 Implementasi HOG Descriptor

4.3.3 Implementasi SMTP

- Implementasi notifikasi *email*

Gambar 4.8 menunjukkan isi dari notifikasi yang nantinya akan dikirim ke *email*.

```

text_html =
'''<html>
<head>
<br>Assalamulaikum<br/>
</head>
<body>
Ada terjadinya pelanggaran santri<br/>
ini titik koordinatnya<br/>
<a
href='http://maps.google.com/?q=%s,%s'>silahkan
klik
disini</a><br/>
dengan jarak '%s' <br/>
</body></html>''' % (lat, lng, range2)

```

Gambar 4.8 Kode HTML untuk Notifikasi

- Implementasi pengiriman notifikasi melalui *email*

Fungsi kirim pada Gambar 4.9 digunakan untuk mengirim hasil hasil dari perangkat yang akan dikirim ke *email* sebagai notifikasi. Dengan menggunakan *library email* pada Python [14], parameter dari fungsi kirim ialah hasil dari informasi perangkat sensor dan modul yaitu *latitude*, *longitude*, jarak, dan gambar. Pada Gambar 4.8 adalah isi dari variabel `text_html`.

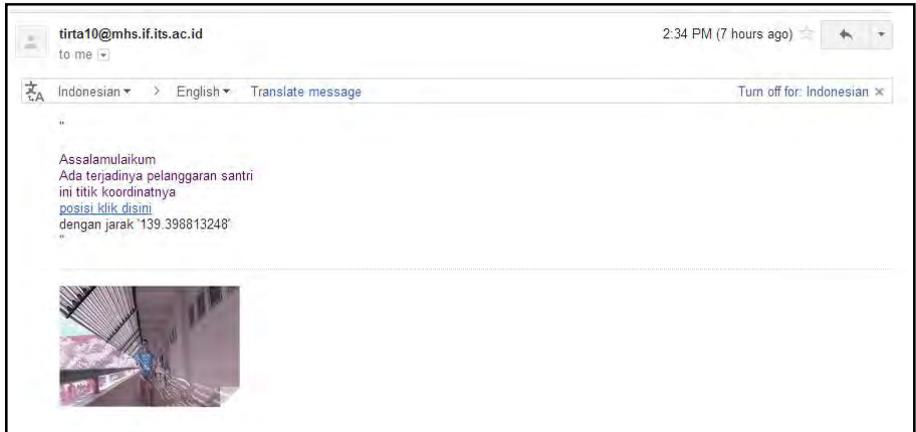
```

        def kirim(self, lat, lng, range2,
filename=''):
        msgRoot= MIMEMultipart('related')
        msgRoot['Subject']= 'NOTIFIKASI'
        msgRoot['From'] = self.strFrom
        msgRoot['To']= self.strTo
        msgRoot.preamble = 'This is a multi-part
message in MIME format'
        text_html = text_html
        msgText = MIMEText (text_html, 'html')
        msgRoot.attach(msgText)
        fp = open(filename, 'rb')
        img = MIMEImage(fp.read())
        fp.close()
        img.add_header('Content-ID', '<image>')
        msgRoot.attach(img)
587) smtp = smtplib.SMTP('smtp.its.ac.id',
        smtp.starttls()
        smtp.login('tirta10@mhs.if.its.ac.id',
'adhanyusi')
        smtp.sendmail(self.strFrom, self.strTo,
msgRoot.as_string())
        smtp.quit()

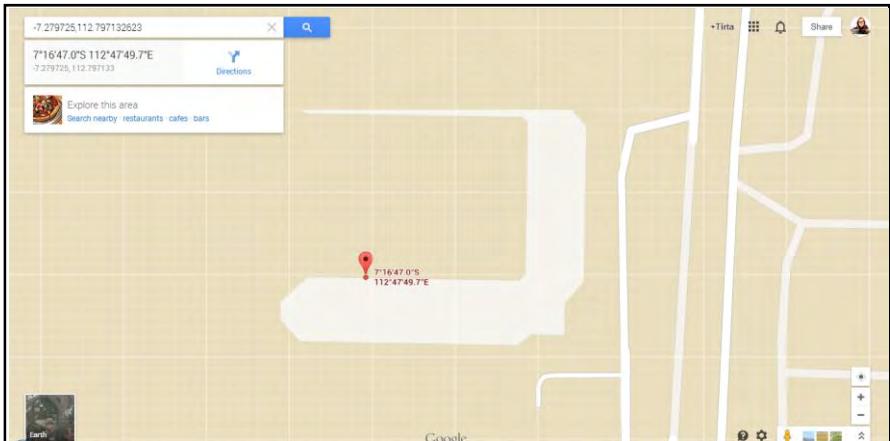
```

Gambar 4.9 Implementasi Fungsi Kirim

Pada Gambar 4.10 adalah hasil dari implementasi fungsi kirim berupa pesan *email*, yang terdiri dari informasi koordinat, informasi jarak, dan yang terakhir informasi gambar. Untuk informasi *map* bisa didapatkan dengan menekan teks yang bersifat *link*. Setelah menekan *link* tersebut maka akan diarahkan langsung ke *google maps* seperti ditunjukkan pada Gambar 4.11.



Gambar 4.10 Hasil Implementasi Fungsi Kirim



Gambar 4.11 Hasil Maps dari Notifikasi

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan uji dari segi fungsionalitas dan performa dari sistem yang telah dirancang. Uji coba fungsionalitas dan performa akan dibagi ke dalam beberapa skenario uji coba.

5.1 Lingkungan Uji Coba

Pada subbab ini, dijelaskan mengenai gambaran dari lingkungan yang digunakan sebagai uji coba sistem. Uji coba akan dilakukan pada area suatu tembok atau pagar dan perangkat sistem diletakkan searah vertical dengan tembok atau pagar, dimaksudkan agar dapat mendeteksi objek manusia yang akan melompat dari tembok. Pengiriman notifikasi akan secara langsung dikirim melalui *email* menggunakan koneksi internet dari wifi terdekat dengan memasang *wifi* adaptor yang telah terhubung dengan Raspberry Pi. Lingkungan uji coba memiliki spesifikasi sebagai berikut:

- Area dekat tembok, pagar dan batas suatu area.
- Perangkat sistem pengawas batas teritorial pondok pesantren (Raspberry Pi, sensor PIR, modul kamera NoIR, sensor deteksi jarak)
- *Powerbank* 2600mah Oren
- 1 buah *wireless* USB *adapter* EW-7811Un
- Komputer AMD FX – 4300 Quad-Core 3.8GHz dan 4 GB memori DDR3
- Laptop ASUS K401 Core 2 Duo
- 1 buah Monitor Advance
- 1 buah converter HDMI ke VGA
- PuTTY

5.2 Uji Coba Fungsionalitas Sistem

Uji coba fungsionalitas merupakan pengujian terhadap jalannya fungsi-fungsi utama yang ada pada perangkat sistem. Pada perangkat sistem ini terdiri dari Raspberry Pi, sensor PIR, sensor deteksi jarak, dan modul kamera NoIR. Uji coba fungsionalitas ini dilakukan pada keseluruhan perangkat sistem. Program ini dijalankan pada Raspberry Pi dengan *meremote* menggunakan PuTTY. Ada beberapa macam fungsionalitas utama yang merupakan bagian dari sistem ini yaitu:

5.2.1 Uji Coba Fungsionalitas Pendeteksian Objek Bergerak dengan Sensor PIR

Pendeteksian objek bergerak menggunakan sensor PIR merupakan bagian inti dari sistem karena sensor PIR disini berperan penting sebagai penerima sinyal dari panas tubuh manusia yang akan melewati suatu tembok yang akan dipantau oleh perangkat sistem ini. Sensor PIR akan menangkap gelombang yang dipancarkan oleh panas tubuh seseorang. Pada uji coba ini sensor PIR di uji berdasarkan jarak, hasil uji coba ditunjukkan pada Tabel 5.1.

Tabel 5.1 Uji Coba Fungsionalitas Sensor PIR Berdasarkan Jarak

No.	Jarak	Keberhasilan
1	1 meter	Sukses
2	2 meter	Sukses
3	3 meter	Sukses
4	4 meter	Sukses
5	5 meter	Sukses

6	6 meter	Sukses
7	7 meter	Sukses
8	8 meter	Gagal
9	9 meter	Gagal
10	10 meter	Gagal

5.2.2 Uji Coba Fungsionalitas Pendeteksian Letak Objek dengan Sensor Deteksi Jarak

Pendeteksian letak jarak objek dari sensor merupakan bagian fitur dari perangkat sistem ini yang digunakan untuk menemukan jarak dari sensor dengan objek bergerak tersebut. Hasil uji coba ditunjukkan pada Tabel 5.2, dari data yang telah didapat sensor deteksi jarak tiper HC-SR04 hanya mampu mendeteksi objek sampai dengan 5 meter.

Tabel 5.2 Uji Coba Fungsionalitas Sensor Deteksi Jarak

No.	Jarak	Jarak yang didapat dari sensor deteksi jarak
1	1 meter	105.64 cm
2	2 meter	212.60 cm
3	3 meter	315.11 cm
4	4 meter	401.90 cm
5	5 meter	499.20 cm
6	6 meter	499.27 cm
7	7 meter	499.98 cm
8	8 meter	499.98 cm

5.2.3 Pengujian Perangkat Sistem Utama

Pengujian perangkat sistem utama ialah uji coba yang dilakukan oleh semua bagian sistem yang sudah terhubung dan sudah berurutan jalannya perangkat sistem ini. Dimulai dengan sensor PIR yang mendeteksi objek bergerak tersebut, kemudian dilanjutkan ke modul kamera NoIR yang bertugas mengambil gambar dan mendeteksi objek pada gambar tersebut. Lalu menghitung jarak objek tersebut dan menghitung koordinatnya, dan yang terakhir informasi objek tersebut dikirim melalui *email* sebagai notifikasi.

5.2.4 Skenario Uji Coba 1

Skenario uji coba 1 adalah pengujian perangkat sistem yang diletakkan di lantai 3 gedung selatan Teknik Informatika. Waktu skenario uji coba ini berdurasi 1 jam lebih 45 menit, mulai pukul 13.00-14.45 wib, pada waktu siang hari.

Tabel 5.3 Hasil Uji Coba 1

Jarak	135.917186737 cm
Koordinat	-7.279725,112.797132308
Posisi Objek Manusia	Menghadap ke samping
Sensor PIR	Bekerja
Waktu	Siang
Pukul	1:44 PM
Tempat	Lantai 3 gedung selatan Teknik Informatika

Dari hasil uji coba pada Tabel 5.3 dapat diketahui beberapa informasi dari perangkat sistem ini yaitu jarak, koordinat map,

waktu dan gambar yang ditunjukkan pada Gambar 5.1. Sedangkan untuk hasil *map* ditunjukkan pada Gambar 5.2. Gambar dari hasil uji coba didapatkan dari hasil *capture* oleh modul kamera Raspberry Pi setelah sensor PIR bekerja dan mendeteksi adanya suatu gerakan. Gambar menunjukkan adanya gerakan manusia yang akan mendekati tembok atau pagar. Kemudian *map* didapatkan dari perhitungan jarak oleh sensor deteksi jarak sehingga diketahui koordinat dari objek bergerak tersebut.



Gambar 5.1 Hasil dari Uji Coba 1

Untuk koordinat *map* pada notifikasi bersifat *link* agar dapat langsung menuju ke *google maps* jika link tersebut diclick. Pada gambar *map* menunjukkan titik kordinat dari terjadinya suatu pelanggaran oleh santri.



Gambar 5.2 Hasil dari Uji Coba 1

5.2.5 Skenario Uji Coba 2

Skenario uji coba 2 adalah pengujian perangkat sistem yang diletakkan di lantai 1 plasa lama mengarah tempat parkir gedung Teknik Informatika. Waktu skenario uji coba ini berdurasi 2 jam, mulai pukul 09.00-11.00 wib, pada waktu pagi hari.

Tabel 5.4 Hasil Uji Coba 2

Jarak	439.398813248 cm
Koordinat	-7.27929265926,112.797464663
Posisi Objek Manusia	Menghadap ke samping
Sensor PIR	Bekerja
Waktu	Pagi
Pukul	10:33 AM

Tempat	Arah tempat parkir gedung Teknik Informatika
--------	--

Dari hasil uji coba pada Tabel 5.4 dapat diketahui beberapa informasi dari perangkat sistem ini yaitu jarak, koordinat map, waktu dan gambar yang ditunjukkan pada Gambar 5.3. Sedangkan untuk *map* ditunjukkan pada Gambar 5.4. Gambar dari hasil uji coba didapatkan dari hasil *capture* oleh modul kamera Raspberry Pi setelah sensor PIR bekerja dan mendeteksi adanya suatu gerakan.



Gambar 5.3 Hasil dari Uji Coba 2

Kemudian *map* didapatkan dari perhitungan jarak oleh sensor deteksi jarak sehingga diketahui koordinat dari objek bergerak tersebut. Untuk koordinat map pada notifikasi bersifat *link* agar dapat langsung menuju ke *google maps* jika link tersebut *diclick*.



Gambar 5.4 Hasil dari Uji Coba 2

5.2.6 Skenario Uji Coba 3

Skenario uji coba 3 adalah pengujian perangkat sistem yang diletakkan di lantai 3 selatan gedung Teknik Informatika. Waktu skenario uji coba ini berdurasi 2 jam, mulai pukul 18.00-20.00 wib, pada waktu senja.

Tabel 5.5 Hasil Uji Coba 3

Jarak	160.106182098 cm
Koordinat	-7.27973947718,112.79712
Posisi Objek Manusia	Menghadap ke depan
Sensor PIR	Bekerja
Waktu	Senja
Pukul	18.45 PM

Tempat	Lantai 3 selatan gedung Teknik Informatika
--------	--

Dari hasil uji coba pada Tabel 5.4 dapat diketahui beberapa informasi dari perangkat sistem ini yaitu jarak, koordinat map, waktu dan gambar yang ditunjukkan pada Gambar 5.5. Sedangkan untuk *map* ditunjukkan pada Gambar 5.6. Gambar dari hasil uji coba didapatkan dari hasil *capture* oleh modul kamera Raspberry Pi setelah sensor PIR bekerja dan mendeteksi adanya suatu gerakan.



Gambar 5.5 Hasil dari Uji Coba 3

Kemudian *map* didapatkan dari perhitungan jarak oleh sensor deteksi jarak sehingga diketahui koordinat dari objek bergerak tersebut. Untuk koordinat map pada notifikasi bersifat *link* agar dapat langsung menuju ke *google maps* jika link tersebut *click*.



Gambar 5.6 Hasil dari Uji Coba 3

5.3 Uji Coba Performa

Uji coba performa dilakukan untuk melihat aktivitas yang dihasilkan oleh perangkat sistem pada tugas akhir ini. Uji coba ini bertujuan untuk mengetahui pemakaian CPU dan memori pada saat menjalankan perangkat sistem ini, kemudian untuk mengetahui kapasitas gambar yang dihasilkan dari modul kamera.

5.3.1 Uji Coba Penggunaan CPU Raspberry Pi

Uji coba ini dilakukan untuk mengukur seberapa besar pemakaian kapasitas CPU yang digunakan oleh *prosesor* Raspberry Pi. Pada uji coba ini digunakan aplikasi yang berjalan pada *terminal* Linux yang bernama SAR (*System Activiyy Report*). Skenario uji coba akan dilakukan pada 2 keadaan berbeda. Yang pertama kondisi dimana Raspberry Pi siaga yaitu

dimana Raspberry Pi dalam posisi *stand by* dan yang kedua pada saat Raspberry Pi sedang menjalankan proses utama, proses utama adalah fase dimana Raspberry Pi menjalankan seluruh perangkat keras yang digunakan sensor PIR, modul kamera, sensor deteksi jarak dan diteruskan dengan mengirim notifikasi melalui *email*.

```

root@raspberrypi:/home/pi# sar 1 5
Linux 3.10.25+ (raspberrypi) 29/06/14 _armv6l_ (1 CPU)
12:41:48 CPU %user %nice %system %iowait %steal %idle
12:41:49 all 0.00 0.00 1.03 0.00 0.00 98.97
12:41:50 all 1.02 0.00 1.02 0.00 0.00 97.96
12:41:51 all 0.00 0.00 3.96 0.00 0.00 96.04
12:41:52 all 1.01 0.00 1.01 0.00 0.00 97.98
12:41:53 all 1.02 0.00 1.02 0.00 0.00 97.96
Average: all 0.61 0.00 1.62 0.00 0.00 97.77

```

Gambar 5.7 Penggunaan CPU pada saat siaga

Gambar 5.7 menunjukkan penggunaan CPU pada Raspberry pi saat mode siaga, percobaannya dilakukan selama 5 kali dan tiap percobaannya diukur selama 1 detik, terlihat pada gambar bahwa pada saat Raspberry pi dalam mode siaga penggunaan CPU nya sangat kecil. Berbanding terbalik dengan Raspberry Pi pada saat menjalankan sistem pengawasan batas territorial pondok pesantren, ditunjukkan pada Gambar 5.8.

```

root@raspberrypi:/home/pi# sar 1 5
Linux 3.10.25+ (raspberrypi) 29/06/14 _armv6l_ (1 CPU)
11:49:20 CPU %user %nice %system %iowait %steal %idle
11:49:21 all 98.00 0.00 2.00 0.00 0.00 0.00
11:49:22 all 99.00 0.00 1.00 0.00 0.00 0.00
11:49:23 all 95.00 0.00 5.00 0.00 0.00 0.00
11:49:24 all 95.00 0.00 5.00 0.00 0.00 0.00
11:49:25 all 98.00 0.00 2.00 0.00 0.00 0.00
Average: all 97.00 0.00 3.00 0.00 0.00 0.00

```

Gambar 5.8 Penggunaan CPU pada saat sistem dijalankan

Tabel 5.6 menunjukkan perbandingan dari penggunaan CPU Raspberry Pi.

Tabel 5.6 Tabel Penggunaan CPU

No.	Penggunaan Presentase CPU (%)	
	Pada saat siaga	Pada saat proses utama
1	0.00	98.00

2	1.02	99.00
3	0.00	95.00
4	1.01	95.00
5	1.02	97.00

5.3.2 Uji Coba Penggunaan Memori Raspberry Pi.

Uji coba ini dilakukan untuk mengukur seberapa besar pemakaian kapasitas memori yang digunakan oleh Raspberry Pi. Pada uji coba ini digunakan aplikasi yang berjalan pada *terminal* Linux yang bernama SAR (*System Activiyy Report*). Skenario uji coba akan dilakukan pada 2 keadaan berbeda. Yang pertama kondisi dimana Raspberry Pi siaga yaitu dimana Raspberry Pi dalam keadaan *stand by* dan yang kedua pada saat Raspberry Pi menjalankan proses utama yaitu dimana Raspberry Pi menjalankan seluruh perangkat keras secara berurutan. Gambar 5.9 menunjukkan pemakaian memori pada saat Raspberry Pi dalam mode siaga.

```

root@raspberrypi:/home/pi/TA# free -t -m
              total        used        free      shared    buffers     cached
Mem:          374           60          313           0           10          26
-/+ buffers/cache:          23          350
Swap:          99           0           99
Total:        474           60          413
root@raspberrypi:/home/pi/TA#

```

Gambar 5.9 Penggunaan memori Raspberry Pi pada mode siaga

Berbanding terbalik dengan Raspberry pi pada saat menjalankan sistem pengawasan batas territorial pondok pesantren seperti ditunjukkan pada Gambar 5.10.

```

root@raspberrypi:/home/pi/TA# free -t -m
              total        used        free      shared    buffers     cached
Mem:          374          115          258           0           12          57
-/+ buffers/cache:          43          328
Swap:          99           0           99
Total:        474          115          358
root@raspberrypi:/home/pi/TA#

```

Gambar 5.10 Penggunaan memori Raspberry Pi pada mode menjalankan proses utama

Tabel 5.7 menunjukkan perbandingan pemakaian memori antara mode siaga dengan Raspberry Pi saat sistem menjalankan proses utama.

Tabel 5.7 Tabel Pemakaian Memori

No.	Pemakaian Memori (MB)	
	Pada saat siaga	Pada saat proses utama
1	60	121

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap perangkat sistem pengawasan pondok pesantren menggunakan teknologi Raspberry Pi dapat disimpulkan antara lain:

1. Perangkat sistem ini dapat mendeteksi objek manusia yang bergerak hingga berjarak 7 meter.
2. Dengan menggunakan perangkat sistem ini, ustad dapat mengetahui area yang paling sering terjadi pelanggaran santri.
3. Perangkat sistem ini menggunakan sumber daya CPU dan memori yang cukup intensif.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem pengawasan batas teritorial pondok pesantren:

1. Penambahan *GPS module* pada Raspberry Pi agar dapat mengetahui titik koordinatnya secara dinamis.
2. Penggunaan modem pada koneksi internet sistem ini, apabila wifi mati sistem tetap berjalan semestinya.
3. Penambahan fitur *server-client* pada sistem ini agar kinerja Raspberry Pi menjadi ringan.

DAFTAR PUSTAKA

- [1] "Raspbian OS," [Online]. Available: <http://www.raspbian.org/>. [Accessed mei 1 2014].
- [2] "Raspberry Pi: FAQs," [Online]. Available: <http://www.raspberrypi.org/faqs#introWhatIs>. [Accessed 11 Maret 2014].
- [3] "HC-SR501 Human Sensor Module Pyroelectric Infrared," [Online]. Available: http://www.icstation.com/product_info.php?products_id=1390#UxV_FvmSy3B. [Accessed 2 Maret 2014].
- [4] "5MP Raspberry Pi Camera Board NoIR," [Online]. Available: <https://www.modmypi.com/raspberry-pi-noir-camera-board>. [Accessed 1 Maret 2014].
- [5] "Ultrasonic Ranging Detector Mod HC-SR04 Distance Sensor," [Online]. Available: <http://www.sainsmart.com/ultrasonic-ranging-detector-mod-hc-sr04-distance-sensor.html>. [Accessed 1 Maret 2014].
- [6] "SMTP," [Online]. Available: http://www.tutorialspoint.com/python/python_sending_email.htm. [Accessed 10 mei 2014].
- [7] "Raspberry Pi GPIO," [Online]. Available: <http://raspberrypiwebservice.com/gpio/gpio.html>. [Accessed 12 mei 2014].
- [8] "OpenCV," [Online]. Available: <http://docs.opencv.org/modules/core/doc/intro.html>. [Accessed 13 mei 2014].
- [9] "Object Detection," [Online]. Available: http://docs.opencv.org/modules/gpu/doc/object_detection.html#gpu-hogdescriptor-detectmultiscale. [Accessed juni 2014].

- [10] "Sysstat," [Online]. Available: <http://sebastien.godard.pagesperso-orange.fr/documentation.html>. [Accessed juli 2014].
- [11] "Library geopy," [Online]. Available: <http://geopy.readthedocs.org/en/latest/>. [Accessed Mei 2014].
- [12] "picamera," [Online]. Available: <http://picamera.readthedocs.org/en/release-1.3/>. [Accessed Mei 2014].
- [13] "HOG," [Online]. Available: http://docs.opencv.org/modules/gpu/doc/object_detection.html#gpu-hogdescriptor-detectmultiscale. [Accessed Mei 2014].
- [14] "library MIME," [Online]. Available: <https://docs.python.org/2/library/email.html>. [Accessed Mei 2014].

BIODATA PENULIS



Tirta Taruna Ramadhan biasa dipanggil dengan nama Adhan, lahir di Gresik, pada tanggal 10 Maret 1994. Penulis menempuh pendidikan mulai dari SD Muhammadiyah 6 (2000-2006), MTS Akselerasi PP. Amanatul Ummah (2006-2008), MA Akselerasi PP. Amanatul Ummah (2008-2010) dan melanjutkan studi di S-1 Teknik Informatika ITS (2010-2014).

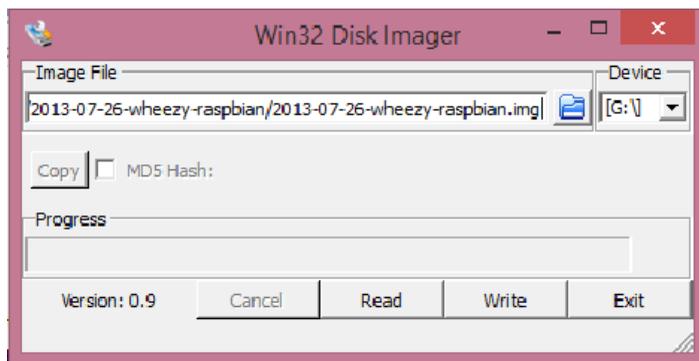
Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTTC). Kemudian aktif dalam

mengikuti organisasi CSS Mora ITS. Penulis juga menjadi administrator Grid Computing Laboratorium (GCL). Penulis juga menghabiskan masa perkuliahan dengan menyalurkan hobi penulis seperti futsal, bermain dota, dan berolahraga.

Selama kuliah di teknik informatika ITS penulis mengambil bidang minat *Net Centric Computing* (NCC). Komunikasi dengan penulis melalui: adhanjr@gmail.com

LAMPIRAN

- Memasang Raspbian Wheezy 3.10
Proses instalasi sistem operasi pada Raspberry Pi adalah hal yang harus dilakukan dikarenakan Raspberry Pi seperti halnya suatu perangkat komputer perlu dipasang sistem operasi. Langkah- langkah untuk melakukan instalasi sistem operasi Raspbian Wheezy akan dijelaskan ke beberapa tahapan sebagai berikut:
 1. Siapkan sebuah kartu SD yang berukuran minimal 4GB.
 2. Unduh aplikasi *Win32 Disk Imager* untuk melakukan format kartu SD yang akan digunakan sebagai wadah *file system* operasi Raspbian Wheezy.

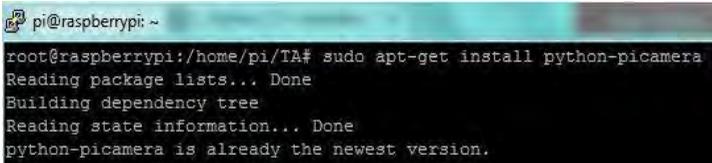


Gambar A.1 Aplikasi Win32

3. Pada Gambar A.1 ditunjukkan tampilan aplikasi *Win32 Disk Manager*, pilih *drive* kartu SD yang ingin di format dan sorot berkas *image* dari sistem operasi Raspbian Wheezy.
4. Klik *Write* dan tunggu sampai proses nya selesai dan kartu SD siap digunakan untuk menjalankan sistem operasi Raspberry Pi.

- Memasang *library* picamera

Proses instalasi *library* picamera digunakan agar modul kamera dari Raspberry Pi dapat melakukan perintah pengambilan gambar. Cara *install library* picamera sangat mudah dengan perintah pada Gambar A.2.



```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/TA# sudo apt-get install python-picamera
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-picamera is already the newest version.
```

Gambar A.2 Install Library Picamera

- Memasang *library* geopy 0.99

Hal yang pertama dilakukan dalam proses instalasi *library* geopy ialah mengunduh *package* geopy 0.99 di <https://pypi.python.org/pypi/geopy>.

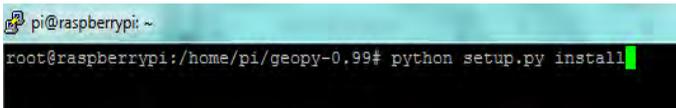


```
root@raspberrypi:/home/pi# tar -xf geopy-0.99.tar.gz
```

Gambar A.3 Extract File Geopy

Kemudian hasil unduhan di *extract* dengan perintah pada Gambar A.3 dan hasil *extract* ialah berupa folder geopy-0.99.

Masuk ke folder geopy-0.99 lalu lakukan perintah seperti pada Gambar A.4.



```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/geopy-0.99# python setup.py install
```

Gambar A.4 Install Geopy

- Memasang OpenCV

Langkah langkah *install library* OpenCV pada Raspberry Pi:

1. Memastikan sistem operasi Raspbian telah *update*:

```
sudo apt-get update
sudo apt-get upgrade
```

Gambar A.5 Perintah *Update* dan *Upgrade*

2. *Install* dependensi

Pada Gambar A.6 dan Gambar A.7 adalah perintah *install* dependensi.

```
sudo apt-get -y install build-essential
cmake      cmake-curses-gui      pkg-config
libpng12-0 libpng12-dev      libpng++-dev
libpng3     libpnglite-dev      zlib1g-dbg
zlib1g     zlib1g-dev      pngtools      libtiff4-dev
```

Gambar A.6 Perintah *Install* Dependensi

```
sudo apt-get -y install libjpeg8
libjpeg8-dev libjpeg8-dbg libjpeg-progs
ffmpeg      libavcodec-dev      libavcodec53
libavformat53          libavformat-dev
libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev
libxine1-ffmpeg libxine-dev libxine1-
bin libunicap2 libunicap2-dev swig
libv4l-0      libv4l-dev      python-numpy
libpython2.6 python-dev python2.6-dev
libqt42.0-dev
```

Gambar A.7 Perintah *Install* Dependensi

3. Kemudian *download library* OpenCV dari URL <http://opencv.org/downloads.html>, kemudian di *unzip file* *opencv-2.4.8.zip*. Lalu buat folder *release* atau *build* dengan perintah *mkdir release* atau *build* kemudian masuk ke folder tersebut.

4. Yang terakhir dalam proses instalasi ialah membuat perintah pada Gambar A.8:

```
make  
sudo make install
```

Gambar A.8 Perintah *Build* OpenCV

- Memasang Sysstat

Hal yang pertama dilakukan pertama kali ialah mengunduh paket sysstat dengan perintah seperti pada Gambar A.9:

```
Wget http://pagesperso-  
orange.fr/sebastien.godard/sysstat-
```

Gambar A.9 Unduh Paket Sysstat

Kemudian ekstrak hasil dari unduh dengan perintah *tar xfv sysstat-11.0.0.tar.gz*, kemudian masuk ke folder hasil ekstrak sysstat-11.0.0. Lalu gunakan perintah seperti pada Gambar A.10 untuk melakukan instalasi.

```
./configure --enable-install-cron
```

Gambar A.10 Perintah *Install* Sysstat

- Fungsi menjalankan sensor

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import time
import geopy
from geopy.distance import VincentyDistance

class RangeFinder:

    def __init__(self, gpio_trigger, gpio_echo):
        self._gpio_trigger = gpio_trigger
        self._gpio_echo = gpio_echo
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(self._gpio_trigger, GPIO.OUT)
        GPIO.setup(self._gpio_echo, GPIO.IN)
        GPIO.output(self._gpio_trigger, False)
        time.sleep(0.001)

    def __del__(self):
        GPIO.setup(self._gpio_trigger, GPIO.IN) #
        reset to input for safety.

    def get_coordinate(self):
        GPIO.output(self._gpio_trigger, True)
        time.sleep(0.00001)
        GPIO.output(self._gpio_trigger, False)
        start = time.time()

        while GPIO.input(self._gpio_echo) == 0:
            start = time.time()

        while GPIO.input(self._gpio_echo) == 1:
            stop = time.time()

        elapsed = stop-start
        distance = elapsed * 34000
        distance2 = float(distance) /float( 2)
```

```

distance = (float(distance)
/float( 2))/float(100000)
    lat1 = -7.279725
    lon1 = 112.797120
    b = 90.0
    d = distance
    origin = geopy.Point(lat1, lon1)
    destination =
VincentyDistance(kilometers=d).destination(origin
, b)
        lat2, lng2 = destination.latitude,
destination.longitude
    return lat2, lng2, distance2

class MotionDetector:

    active = True

    def __init__(self, cleanup, gpio_input,
delay):
        #self._callback = callback
        self._cleanup = cleanup
        self._gpio_input = gpio_input
        self._current_state = 0
        self._previous_state = 0
        self._delay = delay

        GPIO.setmode(GPIO.BCM)

        GPIO.setup(self._gpio_input, GPIO.IN)
        while GPIO.input(self._gpio_input) == 1:
            self._current_state = 0

def __del__(self):
    pass

def monitor(self):

    while self.active:

        self._current_state =
GPIO.input(self._gpio_input)

```

```

if self._current_state == 1:
    self._notify()
    self._current_state = 0
    time.sleep(self._delay)

    time.sleep(0.01)

def _notify(self):
    #self._callback()
    self._cleanup()

```

Gambar B.1 Implementasi pada Sensor

- Fungsi mengirim *email*

```

import smtplib
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart

class SendEmail:
    def __init__(self, strFrom, strTo):
        self.strFrom = strFrom
        self.strTo = strTo
    def kirim(self, lat, lng, range2,
filename=''):
        msgRoot= MIMEMultipart('related')
        msgRoot['Subject']= 'NOTIFIKASI'
        msgRoot['From'] = self.strFrom
        msgRoot['To']= self.strTo
        msgRoot.preamble ='This is a multi-
part message in MIME format'
text_html = '''<html>
                                <head>
                                </head>

                                <body><br>Assalamulaikum<br/>
                                Ada terjadinya pelanggaran santri<br/>
                                ini titik koordinatnya<br/>
                                <a href='http://maps.google.com/?q=%s,%s'>posisi
                                klik disini</a><br/> dengan jarak '%s' <br/>

```

```

</body></html>'''' % (lat, lng, range2)
    msgText = MIMEText (text_html, 'html')
    msgRoot.attach(msgText)
    fp = open(filename, 'rb')
    img = MIMEImage(fp.read())
    fp.close()
    img.add_header('Content-ID', '<image>')
    msgRoot.attach(img)
    smtp = smtplib.SMTP('smtp.its.ac.id', 587)
    smtp.starttls()
    smtp.login('tirta10@mhs.if.its.ac.id',
'adhanyusi')
    smtp.sendmail(self.strFrom, self.strTo,
msgRoot.as_string())
    smtp.quit()

```

Gambar B. 2 Implementasi Mengirim ke *Email*

- Fungsi mengambil gambar

```

#!/usr/bin/env python

import RPi.GPIO as GPIO
from PiPy.sensor import MotionDetector,
RangeFinder
from emails.email3 import SendEmail
import picamera
import threading
from datetime import datetime
from cv import *
import time
import os
from os.path import join
from os import mkdir
from datetime import datetime
class MotionEventHandler:
    def __init__(self, base_directory):
        self._base_directory = base_directory
        self._current_index = 1
        self._current_subdirectory = None

```

```

self._os = os
self._camera = picamera.PiCamera()
self.rf = RangeFinder(8, 7)
self.mm = SendEmail('tirta10@mhs.if.its.ac.id',
'adhanjr@gmail.com')
def capture(self):
    if not self.is_capturing():
        self.start_capturing()
        self._last_capture = datetime.now()
        my_file = join(self._base_directory,
self._current_subdirectory,
str(self._current_index) + ".jpg")

#self._camera.capture(join(self._base_directory,
self._current_subdirectory,
str(self._current_index) + ".jpg"), 'jpeg',
quality=25)
self._camera.capture(my_file, 'jpeg', quality=50)
        lat2, lng2, distance2 =
self.rf.get_coordinate()
        print "capture"
storage = CreateMemStorage(0)
img = LoadImage(my_file)
found = list(HOGDetectMultiScale(img, storage,
win_stride=(8,8),padding=(32,32), scale=1.05,
group_threshold=2))
print "Hog detect"
        if not found:
            print "not found"
            self._os.remove(my_file)
        else:
            print "found"
            self.mm.kirim(lat2, lng2, distance2,
my_file)
            self._current_index += 1
def is_capturing(self):
    if self._last_capture is None or
(datetime.now() - self._last_capture).seconds >
10:
        print "make new condition "
        return False
    return True

```

```
def start_capturing(self):
    self._current_index = 1
    self._current_subdirectory =
time.strftime("%Y_%m_%d_%H:%M:%S")
    mkdir(join(self._base_directory,
self._current_subdirectory))

if __name__ == '__main__':
    meh = MotionEventHandler("captures")
    md = MotionDetector(meh.capture, 11,
delay=5)
    t = threading.Thread(target=md.monitor,
args=())
    try:

        t.start()
        while t.isAlive():
            time.sleep(1)

    except KeyboardInterrupt:
        md.active = False

    finally:
        GPIO.cleanup()
```

Gambar B. 3 Implementasi Mengambil Gambar