



HEART & MIND TOWARDS EXCELLENCE

**RANCANG BANGUN CAMERA TRAP UNTUK PENGAMBILAN
GAMBAR SATWA SECARA OTOMATIS MENGGUNAKAN
RASPBERRY PI 3**



TUGAS AKHIR

Program Studi

S1 Sistem Komputer

INSTITUT BISNIS
DAN INFORMATIKA

stikom

Oleh:

SEPTIANTO RAKA PUTRA PRATAMA

14410200052

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

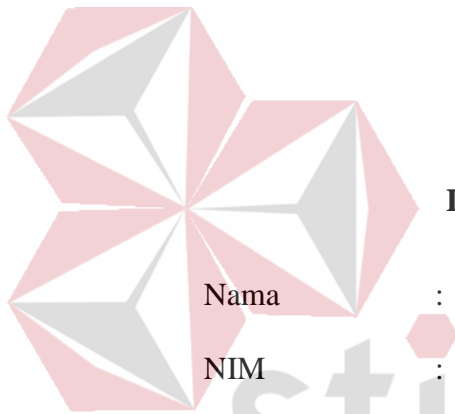
2018

**RANCANG BANGUN CAMERA TRAP UNTUK PENGAMBILAN
GAMBAR SATWA SECARA OTOMATIS MENGGUNAKAN
RASPBERRY PI 3**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Disusun Oleh:

Nama : Septianto Raka Putra Pratama
NIM : 14.41020.0052
Program : S1 (Strata Satu)
Jurusan : Sistem Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2018



“Tidak mengapa jika tidak menjadi orang hebat. Asalkan dapat menjaga hati orang tua. Bagiku keridhoan mereka jauh lebih penting”

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA



Ku persembahkan untuk

Papa, Mama dan Adik

Nenek, Kakek, dan Saudara

Kekasih dan teman-temanku

Yang selalu mendukung dan mendoakanku

Terima kasih atas semuanya

Semoga Tuhan selalu menyayangimu, Amin

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

TUGAS AKHIR
RANCANG BANGUN CAMERA TRAP UNTUK PENGAMBILAN
GAMBAR SATWA SECARA OTOMATIS MENGGUNAKAN
RASPBERRY PI 3

Dipersiapkan dan disusun oleh:

Septianto Raka Putra Pratama

NIM : 14.41020.0052

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada:

Susunan Dewan Penguji

Pembimbing

I. Harianto, S.Kom., M.Eng.

NIDN. 0722087701


II. Ira Guspasari, S.Si., M.T.

NIDN. 0710078601

Pembahas

I. Pauladie Susanto, S.Kom., M.T.

NIDN. 0729047501


16/8/18


16/8/18


16/8/2018

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



FAKULTAS TEKNIK
INFORMATIKA

stikom

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

27/8/18

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Septianto Raka Putra Pratama
NIM : 14410200052
Program Studi : S1 Sistem Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **RANCANG BANGUN *CAMERA TRAP* UNTUK
PENGAMBILAN GAMBAR SATWA SECARA
OTOMATIS MENGGUNAKAN RASPBERRY PI 3**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya, Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 16 Agustus 2018
Yang Menyatakan



6000
STIKOM SURABAYA

Septianto Raka Putra Pratama
Nim : 14.41020.0052

ABSTRAK

Di era yang serba membutuhkan informasi cepat banyak teknologi canggih yang bisa diterapkan di kehidupan. Salah satunya adalah *Camera Trap*. Di pasaran *Camera Trap* memang sudah ada, namun hanya memiliki satu buah sensor PIR dan posisi *camera* tidak dapat berputar. Untuk melengkapi kebutuhan informasi, dibuatlah *Camera Trap* dengan delapan buah sensor PIR dan *camera* dapat berputar 360°.

Pada penelitian ini menggunakan Raspberry Pi 3 sebagai pengendali utama, Sensor PIR HC-SR501 sebagai piranti pengindera pancaran infra merah dari objek, Raspberry Pi Camera Module Supports Night Vision sebagai *camera* yang dapat mengambil gambar pada malam hari secara jelas, dan 2 buah servo sebagai penggerak *camera* yang dapat berputar 360°.

Hasil penelitian ini menunjukkan bahwa alat ini dapat mengambil gambar secara otomatis dan *realtime* dengan jarak jangkauan sensor PIR terjauh yaitu 5 meter di sudut 0°. *Camera* dapat berputar 360° mengikut keberadaan satwa yang ditangkap oleh sensor PIR. *Life time* alat rata-rata selama 4,2 jam. Dapat menampung 49.489 gambar dengan rata-rata ukuran gambar 646.6 kB, pada memori 32 GB hingga penuh dapat terpakai 137,469 jam atau 5,728 hari. Pada kondisi kurang cahaya, alat ini mampu mengambil gambar objek dengan jelas pada jarak maksimal 3 meter. Kendali Motor Servo ini memiliki tingkat presisi 97,7708 %.

Kata kunci: *Camera Trap*, sensor PIR HC-SR501, raspberry Pi 3, servo

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur ke haddirat Allah SWT yang telah memberikan kekuatan, kesehatan lahir dan batin sehingga penulis dapat menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul “*Rancang Bangun Camera Trap untuk Pengambilan Gambar Satwa secara Otomatis Menggunakan Raspberry Pi 3*” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Institut Bisnis dan Informatika Stikom Surabaya

Pada kesempatan kali ini penulis juga ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu dan Kakak tercinta yang telah memberikan dukungan dan doa selama mengerjakan Tugas Akhir ini.
2. Pimpinan Institut Bisnis dan Informatika Stikom Surabaya yang telah banyak memberikan motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi Sistem Komputer Stikom Surabaya dan penguji yang senantiasa memberikan dukungan kepada penulis sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
4. Bapak Harianto, S.Kom., M.Eng. dan Ibu Ira Puspasari, S.Si., M.T., selaku dosen pembimbing satu dan dua yang telah membantu serta mendukung setiap kegiatan sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.

5. Seluruh dosen Pengajar Program Studi S1 Sistem Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Institut Bisnis dan Informatika Stikom Surabaya.
6. Teman-teman dari komunitas Cyber Robotic Stikom Surabaya yang mendukung dan membantu penulis selama masa dan penyusunan buku Tugas Akhir ini.
7. Seluruh pihak yang tidak dapat penulis tuliskan satu persatu yang telah membantu penulis secara langsung maupun tidak langsung.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang besar jika terdapat kata-kata yang salah serta menyinggung perasaan pembaca. Akhir kata penulis ucapkan banyak terima kasih yang besar kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, Agustus 2018

Penulis

DAFTAR ISI

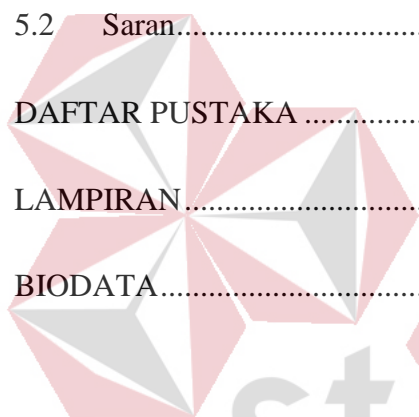
	Hal.
SURAT PERNYATAAN.....	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xviii
DAFTAR LAMPIRAN.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	6
2.1 <i>Camera Trap</i>	6
2.2 <i>Raspberry Pi Camera Module Supports Night Vision</i>	7
2.3 Sensor PIR HC-SR501	9
2.4 Raspberry Pi 3	12

2.4.1	Spesifikasi Raspberry Pi 3 Model B	15
2.4.2	Konektor Raspberry Pi 3 Model B.....	15
2.4.3	Raspberry Pi 3 Model B GPIO.....	17
2.5	Micro SD Card	18
2.6	Motor Servo	19
2.6.1	Jenis-jenis Motor Servo	22
2.6.2	Kegunaan Motor Servo	22
2.6.3	Pemberian Pulsa Pada Motor Servo.....	23
2.6.4	Pengendali Motor Servo.....	25
2.7	Power Bank	27
BAB III HASIL DAN PEMBAHASAN.....		29
3.1	Rancangan Penelitian.....	29
3.2	Rancangan Mekanik.....	29
3.2.1	Struktur Material	31
3.2.2	Ukuran Dimensi Rancang Bangun <i>Camera Trap</i>	32
3.2.3	Pembuatan Mekanik.....	32
3.3	Rancangan Perangkat Keras.....	36
3.3.1	Struktur Material	37
3.4	Rancangan Perangkat Lunak.....	39
3.4.1	Pembacaan Sensor PIR	41
3.4.2	Perencanaan Sensor PIR Menangkap Objek Lebih dari Satu	42

BAB IV METODE PENELITIAN	43
4.1 Pengujian <i>Camera</i>	43
4.1.1 Tujuan	43
4.1.2 Alat Yang digunakan.....	43
4.1.3 Prosedur Pengujian	43
4.1.4 Hasil Pengujian	44
4.2 Pengujian Motor Servo	46
4.2.1 Tujuan	46
4.2.2 Alat Yang digunakan.....	46
4.2.3 Prosedur Pengujian	46
4.2.4 Hasil Pengujian	48
4.3 Pengujian Sensor PIR.....	50
4.3.1 Tujuan	50
4.3.2 Alat Yang digunakan.....	50
4.3.3 Prosedur Pengujian	50
4.3.4 Hasil Pengujian	51
4.4 Pengujian Alat terhadap Gerakan Satwa.....	52
4.4.1 Tujuan	52
4.4.2 Alat yang Digunakan.....	53
4.4.3 Prosedur Pengujian	53
4.4.4 Hasil Pengujian	54

4.5	Pengujian Jarak	57
4.5.1	Tujuan	57
4.5.2	Alat yang Digunakan.....	57
4.5.3	Prosedur Pengujian	58
4.5.4	Hasil Pengujian	58
4.5.5	Hasil Foto	59
4.6	Pengujian Ketepatan Sudut	61
4.6.1	Tujuan	61
4.6.2	Alat yang Digunakan.....	61
4.6.3	Prosedur Pengujian	62
4.6.4	Hasil Pengujian	62
4.7	Pengujian Waktu Hidup <i>Camera Trap</i>	68
4.7.1	Tujuan	68
4.7.2	Alat yang Digunakan.....	68
4.7.3	Prosedur Pengujian	69
4.7.4	Hasil Pengujian	69
4.8	Pengujian Pengambilan Gambar pada Kondisi minim Cahaya	71
4.8.1	Tujuan	71
4.8.2	Alat yang Digunakan.....	71
4.8.3	Prosedur Pengujian	71
4.8.4	Hasil Pengujian	72

4.9	Pengujian Kebutuhan Ukuran <i>File</i> Gambar	73
4.9.1	Tujuan	73
4.9.2	Alat yang Digunakan.....	74
4.9.3	Prosedur Pengujian	74
4.9.4	Hasil Pengujian	75
BAB V PENUTUP.....		78
5.1	Kesimpulan	78
5.2	Saran.....	79
DAFTAR PUSTAKA		80
LAMPIRAN.....		81
BIODATA.....		96



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

DAFTAR GAMBAR

Gambar 2.1. <i>Camera Trap</i>	6
Gambar 2.2. <i>Raspberry Pi Camera Module Supports Night Vision</i>	7
Gambar 2.3. Antarmuka <i>Raspberry Pi Camera Module</i> dengan <i>Raspberry Pi 3 Board</i>	8
Gambar 2.4. <i>Port CSI</i> (Widipratama, 2017)	9
Gambar 2.5. <i>Sensor PIR HC-SR501</i>	10
Gambar 2.6. <i>Bagian-Bagian Sensor PIR (Sensor Gerak PIR (Passive Infra Red), 2012)</i>	10
Gambar 2.7. <i>Cara Kerja Sensor PIR (Sensor Gerak PIR (Passive Infra Red), 2012)</i>	12
Gambar 2.8. <i>Raspberry Pi 3 Board</i>	13
Gambar 2.9. <i>Blok Diagram Raspberry Pi 3</i>	14
Gambar 2.10. <i>I/O Raspberry Pi 3</i> (Widipratama, 2017)	16
Gambar 2.11. <i>Raspberry Pi 3 GPIO</i> (Pinout!, n.d.)	18
Gambar 2.12. <i>Micro SD Card</i>	19
Gambar 2.13. <i>Motor Servo</i>	20
Gambar 2.14. <i>Sistem Mekanik Motor Servo</i> (Widipratama, 2017).....	21
Gambar 2.15. <i>Pulsa Motor Servo</i> (Widipratama, 2017)	23
Gambar 2.16. <i>Contoh Posisi dan Waktu Pemberian Pulsa</i> (Widipratama, 2017). 24	
Gambar 2.17. <i>Pin-Pin Pada Motor Servo</i> (Widipratama, 2017)	25
Gambar 2.18. <i>Pergerakan Motor Servo Terhadap Perubahan Lebar Pulsa</i> (Widipratama, 2017)	26
Gambar 2.19. <i>Power Bank</i>	27

Gambar 3.1. Rancangan Penelitian	29
Gambar 3.2. Penempatan 8 buah sensor PIR	30
Gambar 3.3. Desain Rancangan <i>Camera Trap</i>	33
Gambar 3.4. Desain Rancangan <i>Camera Trap</i>	33
Gambar 3.5. <i>Camera Trap</i> tampak depan	34
Gambar 3.6. Bagian <i>Camera Module</i>	34
Gambar 3.7. Bagian Sensor PIR	35
Gambar 3.8. Bagian roda bebas	35
Gambar 3.9. Bagian Servo	36
Gambar 3.10. Bagian Raspberry Pi 3	36
Gambar 3.11. Diagram Perancangan Perangkat Keras	38
Gambar 3.12. <i>Flow Chart</i> Program.....	39
Gambar 3.13. <i>Flow Chart</i> Pembacaan Sensor PIR.....	41
Gambar 4.1. Hasil Gambar 1	45
Gambar 4.2. Hasil Gambar 2	45
Gambar 4.3. Hasil Gambar 3	45
Gambar 4.4. Pengujian Motor Servo 180°	49
Gambar 4.5. Pengujian Motor Servo 90°	49
Gambar 4.6. Hasil jika Sensor 1 mendeteksi objek	52
Gambar 4.7. Hasil jika Sensor 2 mendeteksi objek	52
Gambar 4.8. Penempatan <i>Camera Trap</i> di Taman Nasional Baluran Situbondo .	55
Gambar 4.9. Sensor PIR menangkap objek rusa.....	55
Gambar 4.10. Sensor PIR menangkap objek rusa 2.....	56

Gambar 4.11. Sensor PIR menangkap objek rusa 3.....	56
Gambar 4.12. Sensor PIR menangkap objek rusa 4.....	56
Gambar 4.13. Sensor PIR menangkap objek rusa 5.....	57
Gambar 4.14. Terdeteksi 5m.....	59
Gambar 4.15. Terdeteksi 4m.....	60
Gambar 4.16. Terdeteksi 3m.....	60
Gambar 4.17. Terdeteksi 2m.....	60
Gambar 4.18. Terdeteksi 1m.....	61



DAFTAR TABEL

Tabel 2.1. Spesifikasi Raspberry Pi 3	15
Tabel 2.2. Konektor Raspberry Pi 3	16
Tabel 3.1. Hasil Sudut dari kombinasi 8 Sensor PIR	31
Tabel 4.1. Percobaan Sudut Servo 1 dan Servo 2	48
Tabel 4.2. Pengujian Alat terhadap Gerakan Objek	54
Tabel 4.3. Pengujian Jarak	59
Tabel 4.4. Pengujian Ketepatan Sudut 360 derajat – 90 derajat	63
Tabel 4.5. Pengujian Ketepatan Sudut 112,5 derajat – 202,5 derajat	64
Tabel 4.6. Pengujian Ketepatan Sudut 225 derajat – 337,5 derajat	65
Tabel 4.7. <i>Error</i> Sudut 360° atau 0° sampai 157,5°	66
Tabel 4.8. <i>Error</i> Sudut 180° sampai 337,5°	67
Tabel 4.9. Pengujian Waktu Hidup <i>Camera Trap</i>	70
Tabel 4.10. Pengujian Pengambilan Gambar pada Kondisi minim Cahaya	72
Tabel 4.11. Pengujian Kebutuhan Ukuran File Gambar	75

DAFTAR LAMPIRAN

Lampiran 1. Program Keseluruhan	81
---------------------------------------	----



BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era yang serba modern dan membutuhkan informasi yang cepat, teknologi dibutuhkan karena dapat membantu pekerjaan manusia dalam di segala bidang kehidupan. *Camera Trap* atau kamera jebakan adalah kamera jarak jauh diaktifkan yang dilengkapi dengan sensor gerak atau sensor inframerah, atau menggunakan sinar sebagai pemicu. Terdapat kekurangan *Camera Trap* yang sudah ada, diantaranya hanya memiliki 1 buah sensor PIR dan posisi *camera* tidak dapat berputar sehingga sudut *camera* mengarah satu arah.

Di Indonesia contohnya *Camera Trap* dimanfaatkan dalam bidang perlindungan satwa-satwa langka dan endemik yang dilindungi oleh pemerintah. Usaha melindungi satwa tersebut juga dengan mengamati karakteristiknya seperti ukuran tubuh, morfologi, habitat untuk tempat tinggal, perilaku dan pola aktifitas serta perkembangbiakannya. Karena obyek diamati adalah hewan, maka tingkat kesulitannya tinggi. Apabila hewan atau satwa menemui manusia di habitatnya, maka akan sangat mungkin dirinya merasa terancam atau justru pola perilakunya berubah.

Dengan adanya bantuan suatu teknologi yang mampu mengintai setiap aktivitasnya, maka pengamat akan mendapatkan informasi mengenai satwa-satwa tersebut. Untuk itu digunakanlah *Camera Trap*. *Camera Trap* sudah terdapat di pasaran. Perbedaan dengan alat ini yaitu *camera* yang dapat berputar 360° dengan

bantuan sensor PIR yang akan menangkap objek di sisi-sisi *camera* dan memiliki tiang sehingga tidak ditempel di pohon.

Seperti penelitian yang telah dilakukan oleh Mohamad Zulfi Widipratama pada tahun 2017 yang berjudul Sistem Pemantau Keamanan Menggunakan Kamera dengan Metode *Background Subtraction*, dimana sistem didukung dengan Raspberry Pi 3 digunakan sebagai media kontrol. Raspberry Pi *Camera Board* digunakan sebagai mendeteksi gerakan manusia secara *real time* dan Motor Servo sebagai media menggerakkan kamera tersebut. Hasil yang didapat adalah sistem tersebut mampu mendeteksi gerakan manusia menggunakan kamera secara *real time* kemudian dapat menginformasikan kepada pemilik rumah (Widipratama, 2017).

Camera pengintai yang dilengkapi dengan sensor PIR (*Passive InfraRed*) yang berfungsi sebagai penangkap informasi baik secara visual maupun audio visual mengenai satwa dengan mengambil gambar dari obyek secara otomatis. Terdapat 8 buah sensor PIR berfungsi untuk mengetahui posisi objek berada di kiri, kanan, depan atau belakang *camera*. Dari 8 sensor PIR menghasilkan 16 posisi/sudut *camera*. *Camera Trap* bekerja dengan menangkap panas yang dipancarkan oleh objek. Ketika panas ditangkap maka sensor PIR aktif. Kemudian mengarahkan *camera* pada objek lalu akan mengambil gambar objek tersebut dan diolah dengan mikrokomputer yaitu Raspberry Pi 3. Raspberry Pi 3 itulah yang menyimpannya ke dalam *memory*. *Camera Trap* tidak dapat mengambil objek berupa video karena keterbatasan media penyimpanan.

1.2 Rumusan Masalah

Permasalahan yang akan dihadapi dalam pengerjaan tugas akhir ini adalah bagaimana merancang sistem *Camera Trap* dengan pengambilan secara otomatis untuk mengamati karakteristik satwa. Adapun sub permasalahan yang akan dibahas:

1. Bagaimana menerapkan 8 buah sensor PIR sebagai pembaca gerak dari objek.
2. Bagaimana mengatur sudut dan mengarahkan *camera* pada objek untuk menangkap gambar secara maksimal.

1.3 Batasan Masalah

Dalam perancangan dan pembuatan terdapat beberapa batasan masalah, antara lain:

1. Perangkat pengindra sistem ini menggunakan sensor PIR.
2. Maksimal 3 sensor PIR berdekatan yang dapat membaca secara bersamaan.
3. Tidak membahas *video sender*.
4. *Camera Trap* kurang sesuai untuk jenis satwa yang terlalu besar / tinggi atau terlalu kecil, dan untuk jenis hewan air.
5. Tidak mampu menangkap gambar ≥ 5 meter.
6. Tidak membahas analisis katakarakteristik hasil foto.

1.4 Tujuan

Berdasarkan rumusan masalah yang diuraikan di atas, maka tujuan penelitian ini:

1. Menerapkan 8 buah sensor PIR sebagai pembaca gerak dari objek.

2. Mengatur sudut dan mengarahkan *camera* pada objek untuk menangkap gambar secara maksimal.

1.5 Manfaat

Manfaat yang akan diperoleh dari rancang bangun *Camera Trap* yaitu:

1. Dapat memberikan informasi tentang keberadaan satwa kepada konservator secara kontinu.
2. Dapat mengetahui kondisi satwa dan habitatnya.

1.6 Sistematika Penulisan

Berikut sistematika penulisan tugas akhir “Rancang Bangun *Camera Trap* untuk Pengambilan Gambar Satwa secara Otomatis Menggunakan Raspberry Pi 3”.

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan yang berisi penjelasan singkat pada masing-masing bab.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang teori yang mendukung tugas akhir ini. Teori tersebut terdiri dari *Camera Trap*, Pi Raspberry Pi *Camera Module Supports Night Vision*, Sensor PIR HC-SR501, Pi Raspberry Pi 3, *Micro SD Card*, Motor Servo, dan Power Bank.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang uraian Rancangan Penelitian, Rancangan Mekanik, Rancangan Perangkat Keras dan Rancangan Perangkat Lunak.

BAB IV IMPLEMENTASI DAN EVALUASI

Bab ini menjelaskan tentang implementasi dari sistem yang dibuat secara keseluruhan untuk mengetahui setiap fungsi bekerja sesuai dengan perancangan atau tidak. Jika ada yang tidak sesuai, maka perlu dilakukan evaluasi dan perbaikan sistem.

BAB V PENUTUP

Bab ini berisi kumpulan yang menjawab pertanyaan dalam perumusan masalah dan beberapa saran yang bermanfaat dalam pengembangan program di waktu yang akan datang.



BAB II

LANDASAN TEORI

2.1 *Camera Trap*

Kamera jebakan atau perangkap kamera atau *Camera Trap* adalah kamera jarak jauh diaktifkan yang dilengkapi dengan sensor gerak atau sensor inframerah, atau menggunakan sinar sebagai pemicu.

Camera Trapping adalah metode untuk menangkap hewan liar di film ketika para peneliti tidak hadir, dan telah digunakan dalam penelitian ekologi selama beberapa dekade. Selain untuk berburu dan melihat satwa liar, juga diaplikasikan dalam studi sarang ekologi, deteksi spesies langka, estimasi ukuran populasi dan kekayaan spesies, serta penelitian tentang penggunaan habitat dan pendudukan struktur bangunan manusia. Salah satu contoh *Camera Trap* diperlihatkan pada Gambar 2.1 (Kamera jebakan, n.d.).



Gambar 2.1. *Camera Trap*

2.2 Raspberry Pi Camera Module Supports Night Vision

Modul *camera* dengan *Adjustable-Focus* yang cocok untuk semua tipe Raspberry Pi. Modul *camera* ini sudah dilengkapi dengan *Infrared LED Board* yang mendukung fitur *Night Vision* sehingga memungkinkan untuk melihat dalam kondisi cahaya rendah. Contoh kamera seperti diperlihatkan pada Gambar 2.2.

Spesifikasi:

- a. Dapat digunakan untuk semua tipe Raspberry Pi.
- b. Menggunakan sensor *camera* OV5647 dengan resolusi 5MP.
- c. Tersedia 4 lubang sekrup yang digunakan untuk *Infrared LED board* sekaligus memberikan catu daya 3,3 V.
- d. Antarmuka: 15-pin CSI (*Camera Serial Interface*). Penerapan antarmuka dapat dilihat pada Gambar 3.
- e. Dimensi: 25mm x 24mm.
- f. Paket penjualan: RPi *Camera* (F) x1, *Infrared LED Board* (B) x2, 15-pin FFC (*opposite sides contact*) x1.



Gambar 2.2. Raspberry Pi Camera Module Supports Night Vision

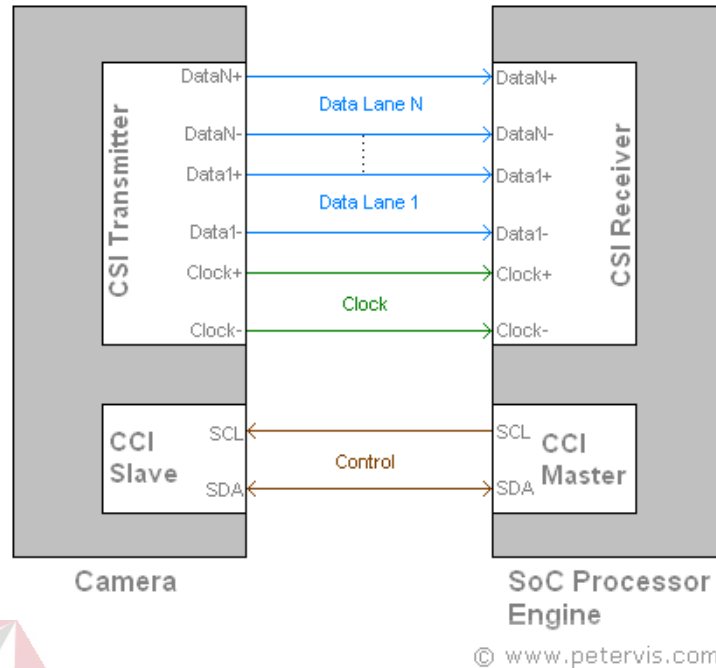
Spesifikasi *camera*:

- a. Ukuran CCD: *1/4inch*
- b. *Aperture* (F): 1.8
- c. *Focal Length*: 3.6mm (*adjustable*)
- d. Diagonal: 75.7°
- e. *Resolusi sensor*: 1080p



Gambar 2.3. Antarmuka Raspberry Pi *Camera Module* dengan Raspberry Pi 3
Board

Koneksi *camera* dengan Raspberry Pi 3 menggunakan kabel CSI (*Camera Serial Interface*) sebagai penghubung, port yang digunakan adalah port CSI, ditunjukkan pada Gambar 2.4.



Gambar 2.4. Port CSI (Widipratama, 2017)

Konektor CSI terdiri dari dua antarmuka yang kecil. Antarmuka pertama adalah untuk transfer data dan sinyal *clock* dari kamera ke prosesor dalam satu arah. Antarmuka kedua terdiri dari SCL / SDA, yang merupakan control dua arah.

2.3 Sensor PIR HC-SR501

Sensor gerak PIR (*Passive Infra Red*) adalah sensor yang berfungsi untuk pendeteksi gerakan yang bekerja dengan cara mendeteksi adanya perbedaan/perubahan suhu sekarang dan sebelumnya. Sensor gerak menggunakan modul pir sangat simpel dan mudah diaplikasikan karena Modul PIR hanya membutuhkan tegangan input DC 5V cukup efektif untuk mendeteksi gerakan hingga jarak 5 meter dan dengan kecepatan 0,2 detik. Ketika tidak mendeteksi gerakan, keluaran modul adalah *LOW*. Dan ketika mendeteksi adanya gerakan, maka keluaran akan berubah menjadi *HIGH*. Adapun lebar pulsa *HIGH* adalah $\pm 0,5$ detik. Sensitifitas Modul PIR yang mampu mendeteksi adanya gerakan pada jarak 5 meter memungkinkan kita membuat suatu alat pendeteksi gerak

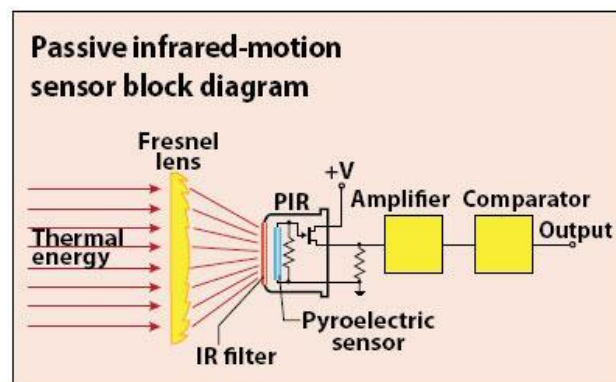
dengan keberhasilan lebih besar. Sensor PIR HC-SR501 yang diperlihatkan pada Gambar 2.5 (Sensor Gerak PIR (Passive Infra Red), 2012).



Gambar 2.5. Sensor PIR HC-SR501

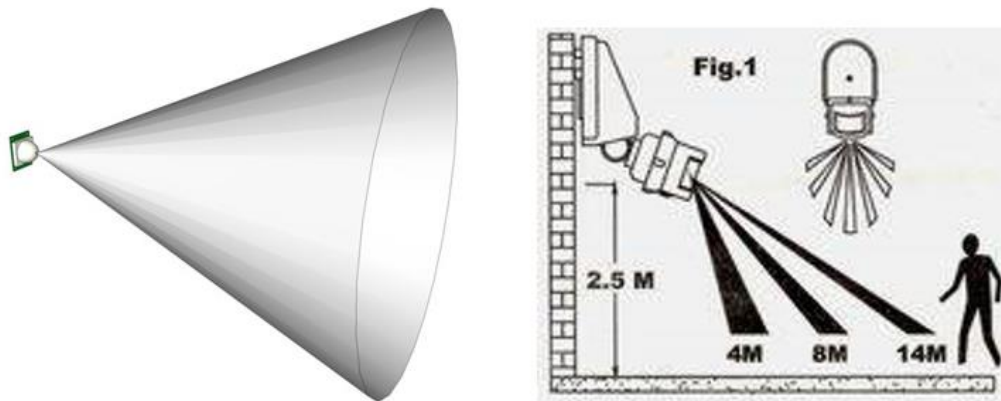
Sensor PIR terdiri dari beberapa bagian pada Gambar 2.6, yaitu:

- a. Lensa Fresnel
- b. Penyaring Infra Merah (Sensor)
- c. Sensor Pyroelektrik (Sensor)
- d. Penguat Amplifier
- e. Komparator



Gambar 2.6. Bagian-Bagian Sensor PIR (Sensor Gerak PIR (Passive Infra Red), 2012)

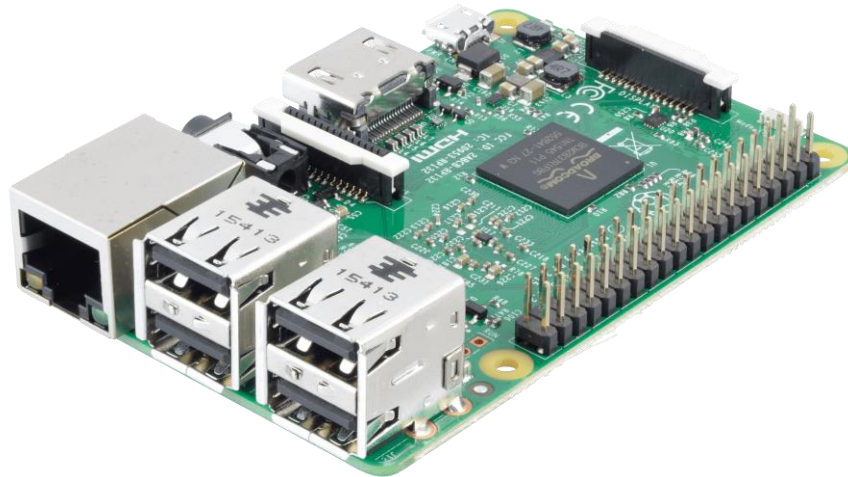
Pancaran infra merah masuk melalui lensa *Fresnel* dan mengenai sensor pyroelektrik, karena sinar infra merah mengandung energi panas maka sensor pyroelektrik akan menghasilkan arus listrik. Sensor pyroelektrik terbuat dari bahan *galium nitrida* (GaN), *cesium nitrat* (CsNo₃) dan *litium tantalate* (LiTaO₃). Arus listrik inilah yang akan menimbulkan tegangan dan dibaca secara analog oleh sensor. Kemudian sinyal ini akan dikuatkan oleh penguat dan dibandingkan oleh komparator dengan tegangan referensi tertentu (keluaran berupa sinyal 1-bit). Jadi sensor PIR hanya akan mengeluarkan logika 0 dan 1, logika 0 saat sensor tidak mendeteksi adanya pancaran infra merah dan logika 1 saat sensor mendeteksi infra merah. Sensor PIR didesain dan dirancang hanya mendeteksi pancaran infra merah dengan panjang gelombang 8-14 mikrometer. Diluar panjang gelombang tersebut sensor tidak akan mendeteksinya. Untuk manusia sendiri memiliki suhu badan yang dapat menghasilkan pancaran infra merah dengan panjang gelombang antara 9-10 mikrometer (nilai standar 9,4 mikrometer), panjang gelombang tersebut dapat terdeteksi oleh sensor PIR. (Secara umum sensor PIR memang dirancang untuk mendeteksi manusia). Sensor PIR memiliki jangkauan jarak yang bervariasi, tergantung karakteristik sensor. Proses penginderaan sensor PIR diperlihatkan pada Gambar 2.7. Umumnya, sensor PIR memiliki jangkauan pembacaan efektif hingga 5 meter. Sensor ini sangat efektif digunakan sebagai human detector.



Gambar 2.7. Cara Kerja Sensor PIR (Sensor Gerak PIR (Passive Infra Red), 2012)

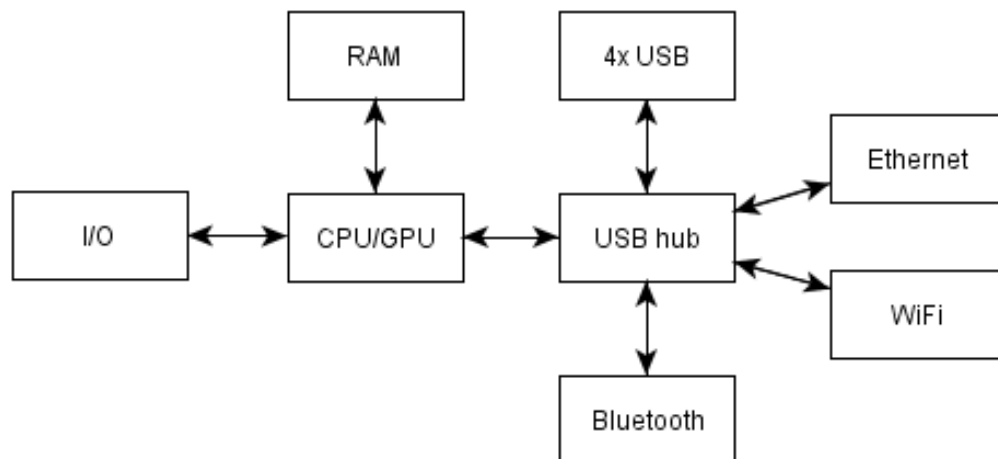
2.4 Raspberry Pi 3

Raspberry Pi 3 adalah komputer *Single Board Circuit* (SBC) yang memiliki ukuran kecil setara dengan ukuran ATM, KTP atau Kartu Kredit. Perangkat ini dapat digunakan untuk menjalankan program perkantoran, permainan, pemutar video beresolusi tinggi dan lain sebagainya. Raspberry Pi 3 ini sangat cocok untuk pada developer karena ukurannya yang kecil tetapi memiliki kinerja yang baik. Selain itu kelebihan Raspberry Pi 3 dibanding dengan mikrokontroler lain yaitu mempunyai Port I/O, display HDMI, koneksi USB dan banyak kelebihan lainnya. Raspberry Pi 3 dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang anggotanya merukan ahli komputer dari Universitas Cambridge, Inggris. Raspberry Pi 3 Board ditunjukkan pada Gambar 2.8.



Gambar 2.8. Raspberry Pi 3 Board

Terdapat 2 versi dari Raspberry yaitu model A dan model B. Perbedaan dari model A dan model B adalah kapasitas penyimpanannya, model B menggunakan penyimpanan yang lebih besar dari pada model A. Selain itu model B juga sudah menggunakan *Ethernet*, *Bluetooth* dan *WI-FI* sebagai sarana komunikasi. Penyimpanan data tidak didesain untuk menggunakan cakram keras atau *solid-state drive*, melainkan menggunakan mikro SD untuk menjalankan sistem dan media penyimpanan dalam jangka panjang. Pengguna dianjurkan menggunakan *Flashdisk* atau penyimpanan lain untuk menyimpan file yang berukuran besar. Perangkat Raspberry Pi 3 ini adalah hasil perkembangan dari perangkat yang dahulu yaitu Raspberry Pi 2. Banyak sekali perubahan yang terjadi mulai dari *RAM*, *processor* dan lain sebagainya.



Gambar 2.9. Blok Diagram Raspberry Pi 3

Blok Diagram Raspberry Pi 3 Model B ditunjukkan pada Gambar 2.9. mempunyai *input* dan *output* antara lain:

- a. HDMI, sebagai penghubung ke LCD Monitor yang mempunyai *port* HDMI atau dengan converter HDMI ke VGA untuk bisa digunakan pada LCD Monitor yang memiliki *port* VGA.
- b. Audio *output* menggunakan jack 3,5mm, sebagai *output* untuk *speaker* atau sejenisnya yang menggunakan jack 3,5mm.
- c. 4 buah *port* USB
- d. 40 *pin* I/O
- e. CSI (*Camera Serial Interface*), digunakan khusus untuk kamera yang menggunakan CSI sebagai antarmuka.
- f. DSI (*Display Serial Interface*)
- g. LAN *port* (*Network*)
- h. WiFi
- i. *Bluetooth*

- j. *SD Card slot* untuk *SD Card memory* penyimpanan sistem operasi juga sebagai penyimpanan file.

2.4.1 Spesifikasi Raspberry Pi 3 Model B

Beberapa spesifikasi Raspberry Pi 3 Model B dapat dilihat pada Tabel 2.1.

Tabel 2.1. Spesifikasi Raspberry Pi 3

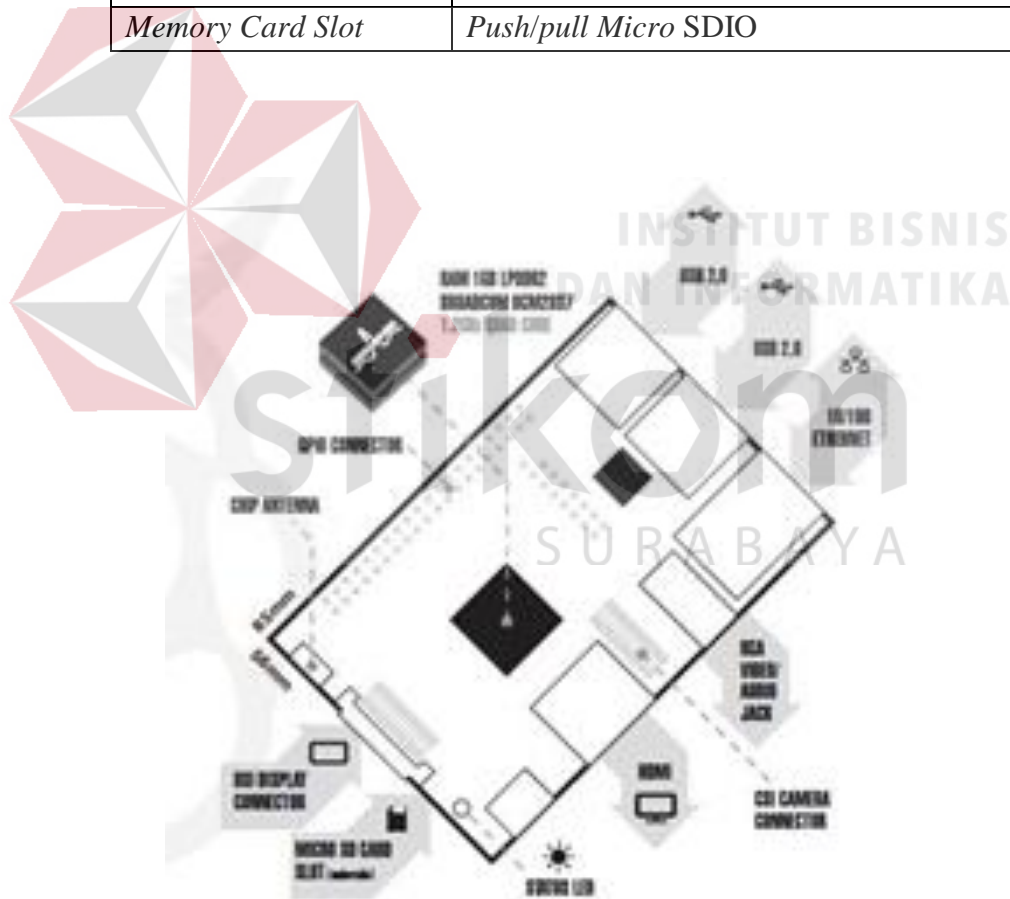
<i>Processor</i>	Broadcom BCM2387 chipset 1.2Ghz <i>Quad-Core</i> ARM Cortex-A53 802.11 b/g/n <i>Wireless LAN</i> and <i>Bluetooth</i> 4.1 (<i>Bluetooth Classic</i> and LE)
GPU	<i>Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware- accelerated OpenVG, and 1080p30 H.264 high- profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure</i>
<i>Memory</i>	1GB LPDDR2
<i>Operating System</i>	Boots from Micro SD card, <ul style="list-style-type: none"> • Raspbian • Ubuntu MATE • Snappy Ubuntu Core • Windows 10 IoT Core • RISC OS • Debian • Arch Linux ARM • Android
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5 Volt 2.5A

2.4.2 Konektor Raspberry Pi 3 Model B

Dalam Raspberry Pi 3 Model B, terdapat beberapa konektor sebagai I/O, yang dapat dilihat pada Tabel 2.2, dan gambar I/O ditunjukkan pada Gambar 2.10.

Tabel 2.2. Konektor Raspberry Pi 3

<i>Ethernet</i>	10/100 BaseT Ethernet socket
<i>Video Output</i>	HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)
<i>Audio Output</i>	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
<i>GPIO Connector</i>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<i>Camera Connector</i>	15-pin MIPI Camera Serial Interface (CSI-2)
<i>Display Connector</i>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock
<i>Memory Card Slot</i>	Push/pull Micro SDIO



Gambar 2.10. I/O Raspberry Pi 3 (Widipratama, 2017)

Key Benefits:

- a. *Low cost*

- b. *10x faster processing*
- c. *Consistent board format*
- d. *Added connectivity*

Key Applications:

- a. *Low cost PC/tablet/laptop*
- b. *Media center*
- c. *Industrial/Home automation*
- d. *Print server*
- e. *Web camera*
- f. *Wireless access point*
- g. *IoT applications*
- h. *Robotics*
- i. *Server/cloud server*
- j. *Security monitoring*
- k. *Gaming*
- l. *Environmental sensing/monitoring (e.g. weather station)*

2.4.3 Raspberry Pi 3 Model B GPIO

GPIO pada Raspberry Pi 3 ini terdiri dari 40 pin, yang ditampilkan pada Gambar 2.11, dengan berbagai fungsi masing – masing, juga terdapat keluaran *power* GND, 5V dan 3,3V. Selain sebagai I/O pada beberapa *pin* GPIO juga terdapat juga fungsi untuk komunikasi serial diantaranya I²C, SPI dan Serial Komunikasi UART.

		Pin No.			
3.3V	1	2	5V		
GPIO2	3	4	5V		
GPIO3	5	6	GND		
GPIO4	7	8	GPIO14		
GND	9	10	GPIO15		
GPIO17	11	12	GPIO18		
GPIO27	13	14	GND		
GPIO22	15	16	GPIO23		
3.3V	17	18	GPIO24		
GPIO10	19	20	GND		
GPIO9	21	22	GPIO25		
GPIO11	23	24	GPIO8		
GND	25	26	GPIO7		
DNC	27	28	DNC		
GPIO5	29	30	GND		
GPIO6	31	32	GPIO12		
GPIO13	33	34	GND		
GPIO19	35	36	GPIO16		
GPIO26	37	38	GPIO20		
GND	39	40	GPIO21		

Key	
Power +	UART
GND	SPI
I ² C	GPIO

Gambar 2.11. Raspberry Pi 3 GPIO (Pinout!, n.d.)

2.5 Micro SD Card

Kartu *memory* adalah salah satu komponen penting bagi pemilik smartphone atau pun tablet. Kartu ini dapat menanggulangi keterbatasan *memory* internal pada sebuah *smartphone*. Saat ini kartu *memory* yang paling banyak digunakan adalah kartu *memory* dari jenis SD, yang terdiri dari *memory SD card*, *MiniSD*, dan *MicroSD*. *Micro SD Card* diperlihatkan pada Gambar 2.12.



Gambar 2.12. *Micro SD Card*

SD adalah singkatan dari *Secure Digital*, yang merupakan *memory Card flash ultra* kecil yang dirancang untuk menyediakan *memory* berkapasitas tinggi dalam ukuran yang kecil. Kartu *memory* SD sendiri bisa dikategorikan berdasarkan dua kriteria, yaitu ukurannya dan kapasitasnya (Inilah Perbedaan Memori SD Card, Mini SD, dan Micro SD, n.d.).

Micro SD Card ini memiliki kapasitas 16 GB. Sebesar 3 GB dari kapasitas tersebut digunakan untuk program dari Raspberry Pi 3. Sehingga yang bisa digunakan untuk menyimpan file gambar sebesar 13 GB atau 13.000.000 kb. Sedangkan untuk mengetahui banyaknya gambar yang tersimpan dalam *memory* tersebut menggunakan rumus:

$$\text{Banyak gambar} = \frac{\text{Kapasitas } \textit{Micro SD Card}}{\text{Rata - rata ukuran file gambar}} \quad (1)$$

2.6 Motor Servo

Motor Servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotor-nya akan diinformasikan kembali ke rangkaian kontrol yang

ada di dalam Motor Servo. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu Motor Servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel Motor Servo (Motor Servo, 2014).

Motor Servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, Motor Servo dapat dimodifikasi agar bergerak kontinyu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagianbagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

Motor Servo adalah motor yang mampu bekerja dua arah CW (*Clock Wise*) dan CCW (*Counter Clock Wise*) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Motor Servo tampak pada Gambar 2.13.



Gambar 2.13. Motor Servo

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian kontrol elektronik dan gear dalam untuk mengendalikan pergerakan dan sudut angularnya. Sistem mekanik Motor Servo tampak pada Gambar 2.14.



Gambar 2.14. Sistem Mekanik Motor Servo (Widipratama, 2017)

Motor Servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena gear dalamnya. Kecepatan servo bergerak adalah 0,83 rad/detik.

Lebih dalam dapat digambarkan bahwa sebuah Motor Servo memiliki:

- a. Jalur kabel: *power*, *ground*, dan *control*
- b. Sinyal control mengendalikan posisi
- c. Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.
- d. Konstruksi didalamnya meliputi *internal gear*, *potensiometer*, dan *feedback control*.

2.6.1 Jenis-jenis Motor Servo

a. Motor Servo Standar 180°

Motor Servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan deteksi masing-masing sudut mencapai 90° sehingga total deteksi sudut dari kanan – tengah – kiri adalah 180°.

b. Motor Servo *Continuous*

Motor Servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan deteksi sudut putar (dapat berputar secara kontinyu). Untuk menggunakan servo jenis *Continuous* maka diperlukan perhitungan tersendiri sebagai penentu sudut ketika dibutuhkan.

Keterangan:

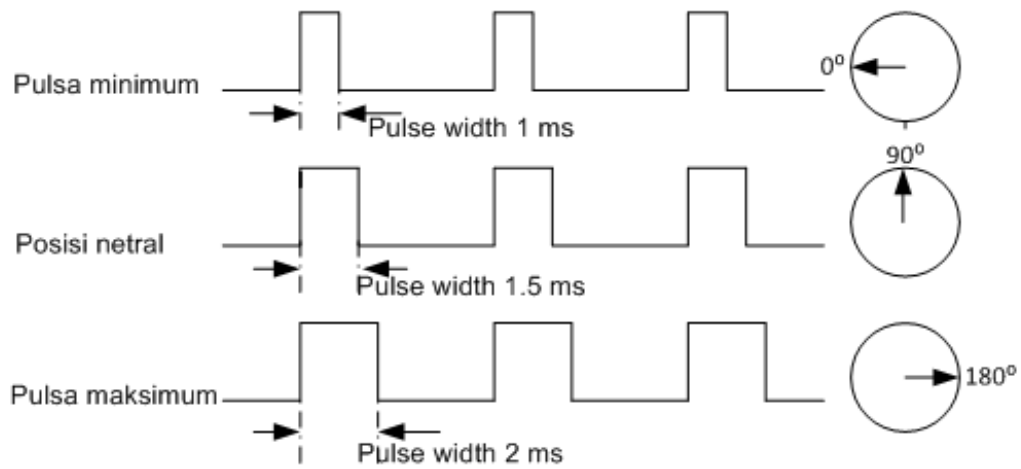
- CW (*Counter Wise*) yaitu putaran motor searah dengan arah putaran jarum jam.
- CCW (*Counter Clock Wise*) yaitu putaran motor berlawanan dengan arah putaran jarum jam.

2.6.2 Kegunaan Motor Servo

Kebanyakan Motor Servo digunakan sebagai:

- a. *Manipulators.*
- b. *Moving Camera.*
- c. *Robot Arms.*

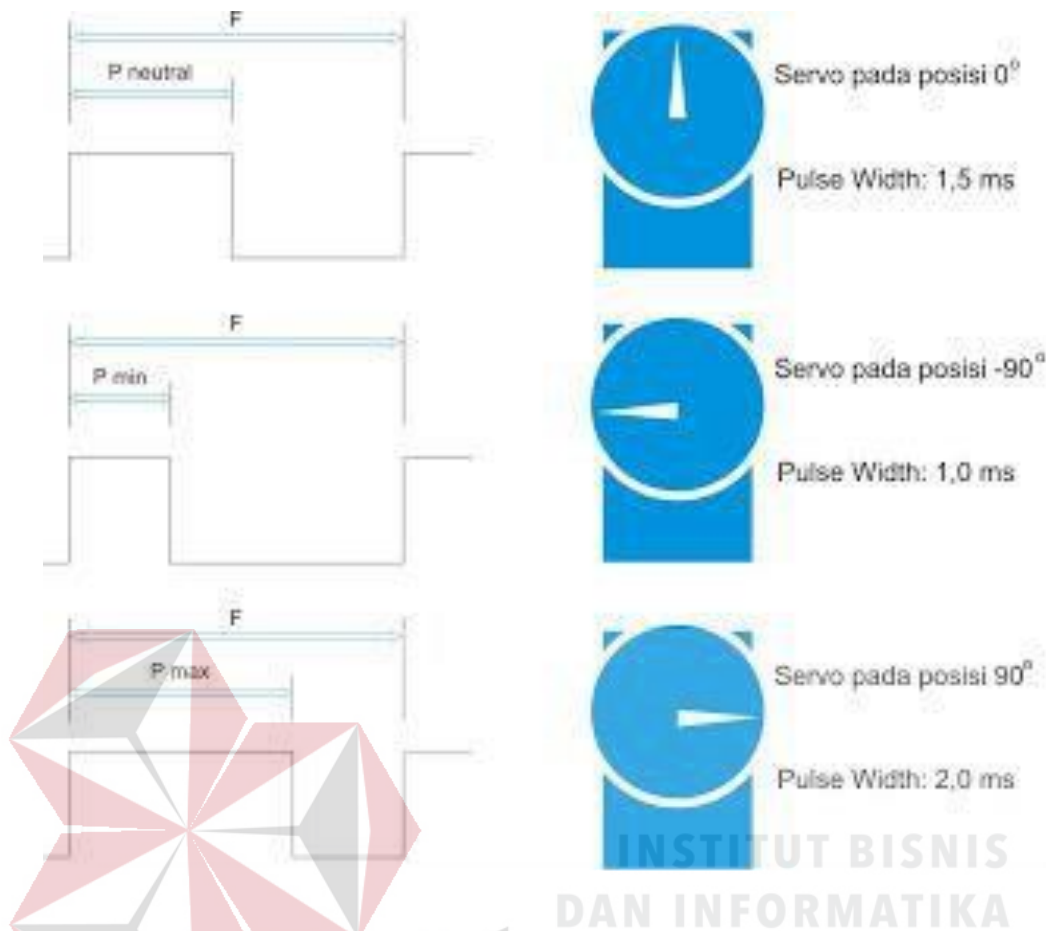
2.6.3 Pemberian Pulsa Pada Motor Servo



Gambar 2.15. Pulsa Motor Servo (Widipratama, 2017)

Contoh dimana bila diberikan pulsa dengan besar 1.5ms mencapai gerakan 90 derajat, maka bila diberikan data kurang dari 1.5 ms maka posisi mendekati 0 derajat dan bila diberikan data lebih dari 1.5 ms maka posisi mendekati 180 derajat. Contoh Posisi dan Waktu Pemberian Pulsa tampak pada Gambar 2.15 dan Gambar 2.16.

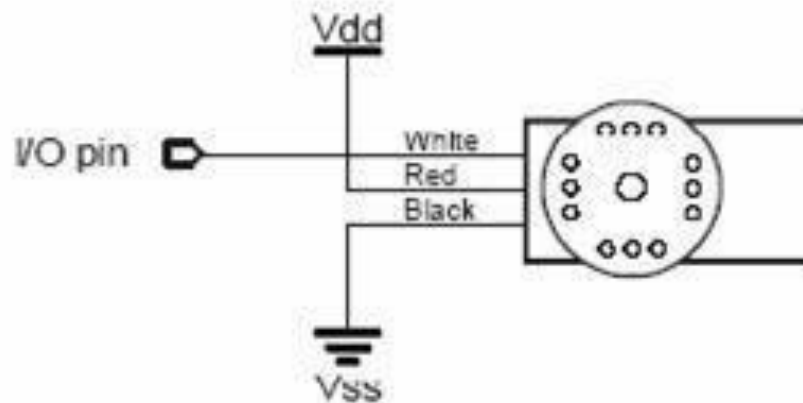
SURABAYA



Gambar 2.16. Contoh Posisi dan Waktu Pemberian Pulsa (Widipratama, 2017)

- Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50Hz.
- Dimana pada saat sinyal dengan frekuensi 50Hz tersebut dicapai pada kondisi Ton duty cycle 1.5ms, maka rotor dari motor akan berhenti tepat di tengah- tengah (sudut 0° / netral).
- Pada saat Ton *duty cycle* dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya Ton *duty cycle*, dan akan bertahan diposisi tersebut.
- Dan sebaliknya, jika Ton *duty cycle* dari sinyal yang diberikan lebih dari 1.5ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut

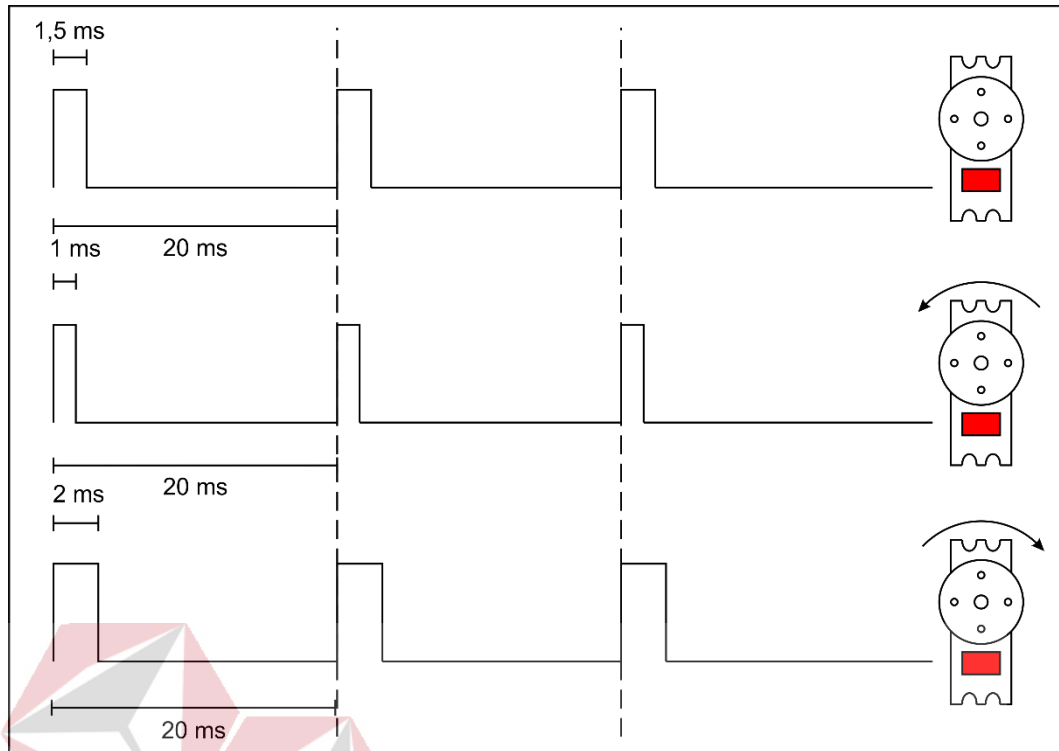
yang linier pula terhadap besarnya Ton *duty cycle*, dan bertahan diposisi tersebut.



Gambar 2.17. Pin-Pin Pada Motor Servo (Widipratama, 2017)

2.6.4 Pengendali Motor Servo

Pin-pin pada Motor Servo dapat dilihat pada Gambar 2.17. Terdapat tiga utas kabel dengan warna merah, hitam, dan kuning. Kabel merah dan hitam harus dihubungkan dengan sumber tegangan 4-6volt DC agar Motor Servo dapat bekerja normal. Sedangkan kabel berwarna kuning adalah kabel data yang dipakai untuk mengatur arah gerak dan posisi servo. Pergerakan Motor Servo terhadap perubahan lebar pulsa tampak pada Gambar 2.18.



Gambar 2.18. Pergerakan Motor Servo Terhadap Perubahan Lebar Pulsa

(Widipratama, 2017)

Posisi servo dikendalikan oleh *pulsewidth* dari sinyal 50 Hz PWM. Oleh karena itu, perlu mengubah urutan PWM pada 50 Hz. Periode sinyal adalah $1/50 = 0,02$ detik, atau 20 milidetik. *DutyCycles* spesifik yang mengarah ke posisi penuh kiri dan kanan penuh. Kiri penuh di *DutyCycle* = 2, dan kanan penuh di *DutyCycle* = 12. Dengan nilai-nilai ini, dapat membuat persamaan linear yang akan memberi sudut yang diinginkan antara 0 dan 180. Untuk sudut yang diinginkan 0, maka harus menerapkan *DutyCycle* of 2. Ini akan menjadi titik (0,2). Sekarang untuk sudut yang diinginkan 180, maka harus menerapkan *DutyCycle* of 12 dan akan menjadi titik (180,12) (RASPBerry PI LESSON 28: CONTROLLING A SERVO ON RASPBerry PI WITH PYTHON, 2015).

Kemiringan garis adalah:

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)} = \frac{(12 - 2)}{(180 - 0)} = \frac{10}{180} = \frac{1}{18}$$

(2)

Mendapatkan persamaan garis menggunakan rumus titik kemiringan.

$$y - y_1 = m(x - x_1)$$

$$y - 2 = \frac{1}{18} \times (x - 0)$$

$$y = \frac{1}{18} \times x + 2 \quad (3)$$

Dimasukkan ke dalam variabel aktual, dapatkan:

$$\text{DutyCycle} = \frac{1}{18} \times (\text{DesiredAngle}) + 2 \quad (4)$$

2.7 Power Bank



Gambar 2.19. Power Bank

Power Bank adalah sebagai pengisi daya *gadget* saat sedang berada diluar dan jauh dari sumber listrik. Fungsi Power Bank dapat disebut juga sebagai

penyimpan daya atau dapat dianalogikan sebagai baterai cadangan, namun untuk penggunaannya tidak perlu mencopot batrei *handphone*, cukup menancapkan kabel seperti saat kita mengisi daya menggunakan *charger* biasa. Bentuk Power Bank dapat dilihat pada Gambar 2.19.

Power Bank dapat digunakan untuk mengisi daya baterai *Camera Trap* agar *Camera Trap* tetap hidup. Untuk menghitung berapa lama *camera* dapat bertahan digunakan rumus berikut:

$$\text{Waktu hidup camera trap} = \frac{\text{Kapasitas Power Bank}}{\text{Konsumsi energi camera trap}} \quad (5)$$

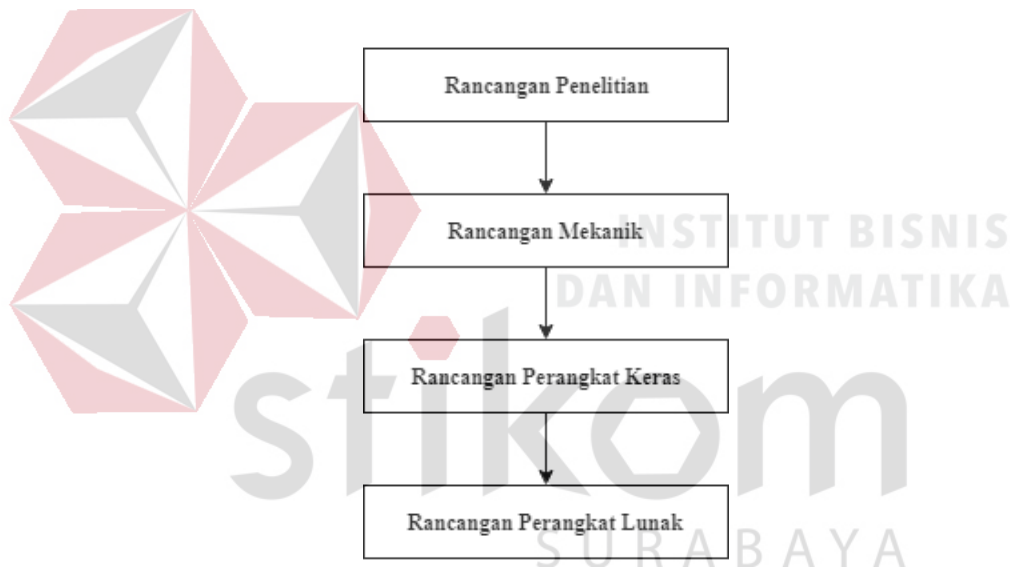


BAB III

METODE PENELITIAN

3.1 Rancangan Penelitian

Pada bab ini akan dibahas mengenai beberapa tahap mulai dari Rancangan Penelitian, Rancangan mekanik, Rancangan Perangkat Keras, Rancangan Perangkat Lunak seperti pada Gambar 3.1. Perancangan ini dibuat supaya pengerjaan tahapan selanjutnya dapat dilakukan dengan lancar dan berurutan.



Gambar 3.1. Rancangan Penelitian

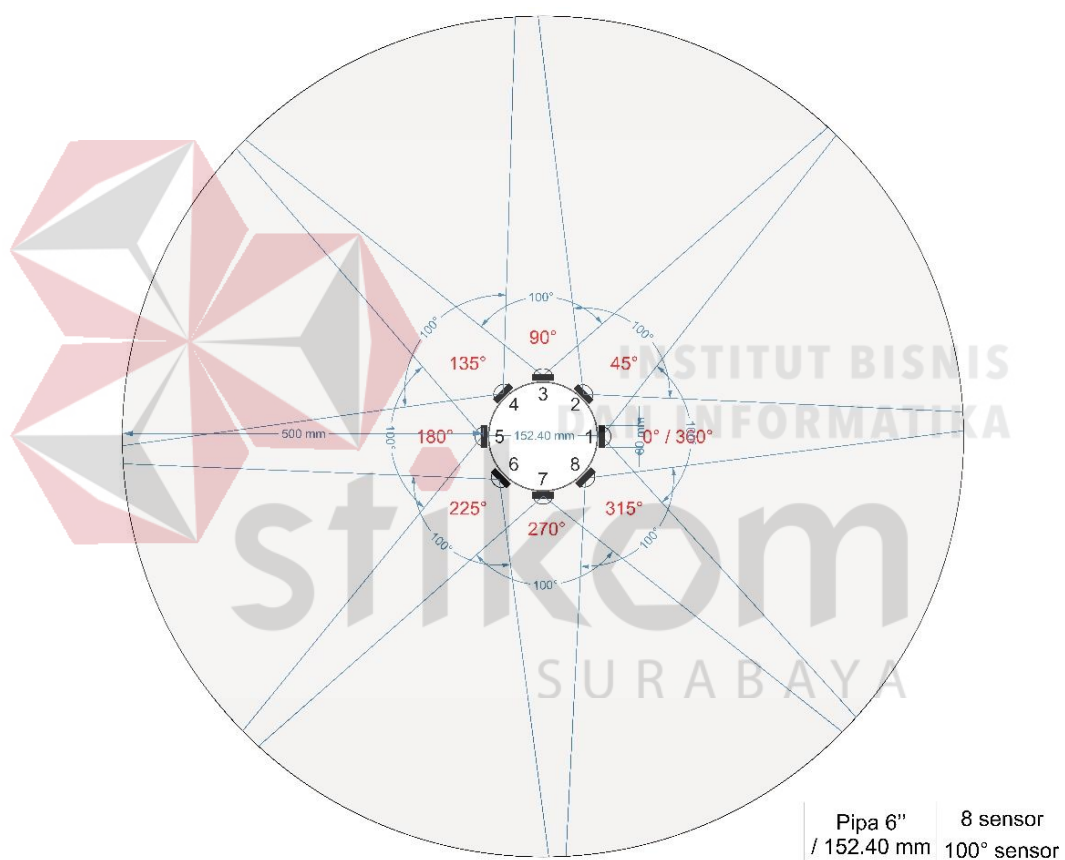
3.2 Rancangan Mekanik

Pada tahap ini, penulis menuliskan tahapan-tahapan yang dibutuhkan dalam perencanaan dalam pembuatan mekanik dari *Camera Trap*. Tahapan ini dibuat agar alur kerja dari mesin dapat diketahui dengan mudah.

Pada tahap pertama, penulis akan membuat pipa. Dimana pipa ini berfungsi sebagai tiang *Camera Trap*. Pada tahap ini pipa di bagi menjadi 2 bagian yaitu

bagian bawah dan atas. Bagian bawah sebagai tempat 8 sensor PIR dan Motor Servo bawah sedangkan bagian atas sebagai tempat Raspberry Pi 3, Power Bank, *accu*, *camera* dan Motor Servo.

Alat ini terdapat 2 buah Motor Servo, setiap Motor Servo berkerja masing masing sesuai sudut yang telah ditentukan. Terdapat 8 sensor PIR yang dapat menghasilkan area tertentu. Penempatan sensor dapat dilihat pada Gambar 3.2.



Gambar 3.2. Penempatan 8 buah sensor PIR

Dari perhitungan manual didapatkan hasil 16 sudut yang nantinya akan mengarahkan *camera* sesuai sensor yang bernilai 1. Dari 8 sensor tersebut dapat dikombinasikan maksimal dengan 2 sensor PIR. Hasil sudut dapat dilihat pada Tabel 3.1.

Tabel 3.1. Hasil Sudut dari kombinasi 8 Sensor PIR

Sensor 8	Sensor 7	Sensor 6	Sensor 5	Sensor 4	Sensor 3	Sensor 2	Sensor 1	Hasil Sudut
0	0	0	0	0	0	0	1	0°
0	0	0	0	0	0	1	0	45°
0	0	0	0	0	0	1	1	22.5°
0	0	0	0	0	1	0	0	90°
0	0	0	0	0	1	1	0	67.5°
0	0	0	0	0	1	1	1	45°
0	0	0	0	1	0	0	0	135°
0	0	0	0	1	1	0	0	112.5°
0	0	0	0	1	1	1	0	90°
0	0	0	1	0	0	0	0	180°
0	0	0	1	1	0	0	0	157.5°
0	0	0	1	1	1	0	0	135°
0	0	1	0	0	0	0	0	225°
0	0	1	1	0	0	0	0	202.5°
0	0	1	1	1	0	0	0	180°
0	1	0	0	0	0	0	0	270°
0	1	1	0	0	0	0	0	247.5°
0	1	1	1	0	0	0	0	225°
1	0	0	0	0	0	0	0	315°
1	0	0	0	0	0	0	1	337.5°
1	0	0	0	0	0	1	1	0°
1	1	0	0	0	0	0	0	292.5°
1	1	0	0	0	0	0	1	315°
1	1	1	0	0	0	0	0	270°
				1 Sensor Terdekteksi				
				2 Sensor Terdekteksi				
				3 Sensor Terdekteksi				

3.2.1 Struktur Material

Dalam penelitian ini bahan material mekanik yang digunakan penulis dalam pembuatan rancang bangun *Camera Trap* adalah sebagai berikut:

1. Pipa pvc 6 inch.

2. Dop pipa pvc 6 inch.
3. Siku Aluminium.
4. Mur dan baut.
5. Akrilik.
6. Roda bebas.
7. Plat Aluminium.
8. Cat spray.
9. Besi

3.2.2 Ukuran Dimensi Rancang Bangun *Camera Trap*

Berikut ini adalah ukuran dimensi dari rancang bangun *Camera Trap* setelah pemasangan komponen:

1. Tinggi rancang bangun : 100 cm
2. Diameter rancang bangun : 17 cm
3. Volume rancang bangun : 22.686,5 cm³

3.2.3 Pembuatan Mekanik

Berikut ini adalah desain rancangan mekanik *Camera Trap* pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3. Desain Rancangan *Camera Trap*

Gambar 3.4. Desain Rancangan *Camera Trap*

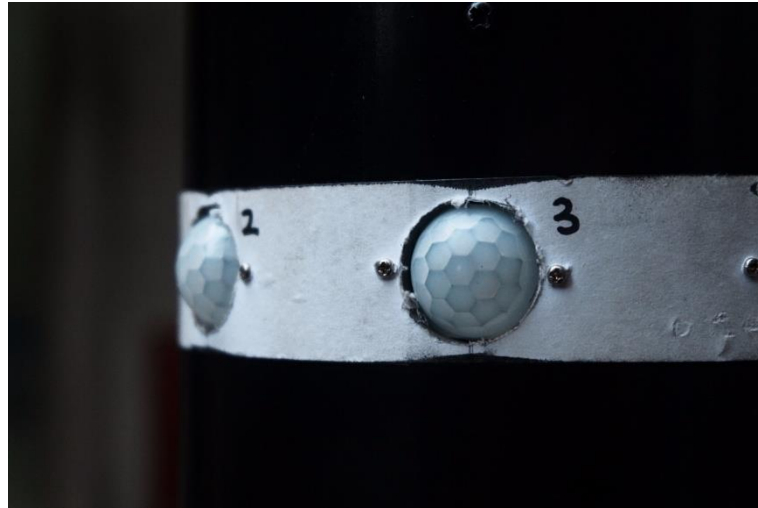
Berikut ini adalah hasil rancangan mekanik *Camera Trap* pada Gambar 3.5 sampai Gambar 3.10:



Gambar 3.5. *Camera Trap* tampak depan



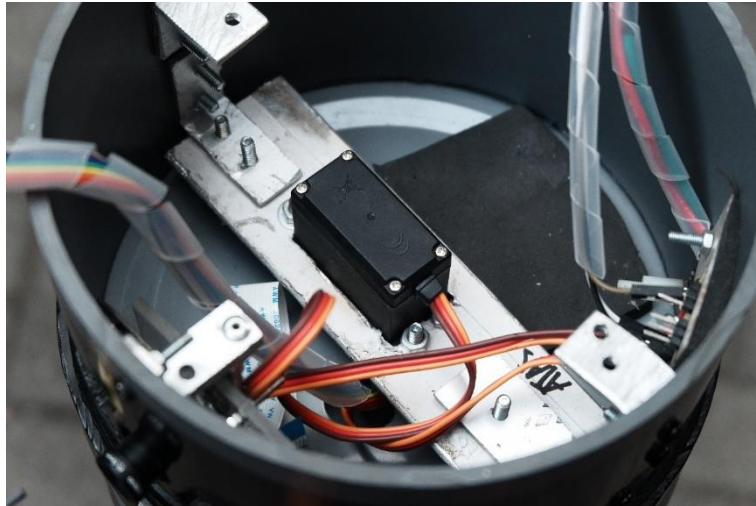
Gambar 3.6. Bagian *Camera Module*



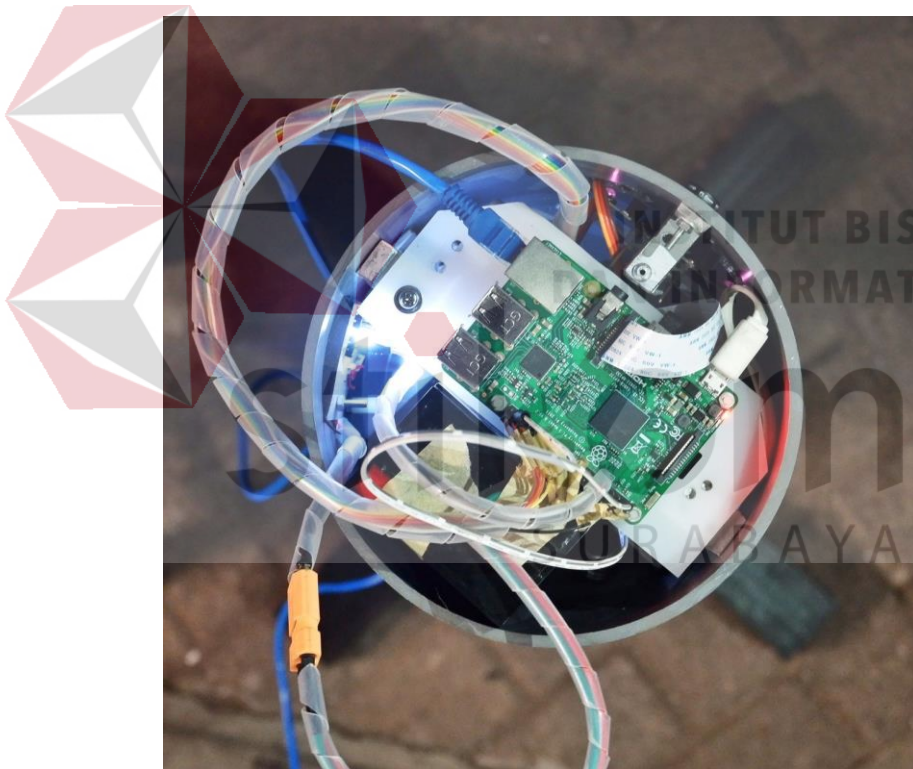
Gambar 3.7. Bagian Sensor PIR



Gambar 3.8. Bagian roda bebas



Gambar 3.9. Bagian Servo



Gambar 3.10. Bagian Raspberry Pi 3

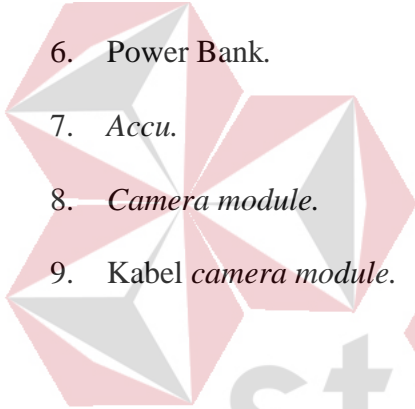
3.3 Rancangan Perangkat Keras

Pada tahap ini, penulis menuliskan perangkat keras yang dibutuhkan dalam perencanaan pembuatan *Camera Trap*.

3.3.1 Struktur Material

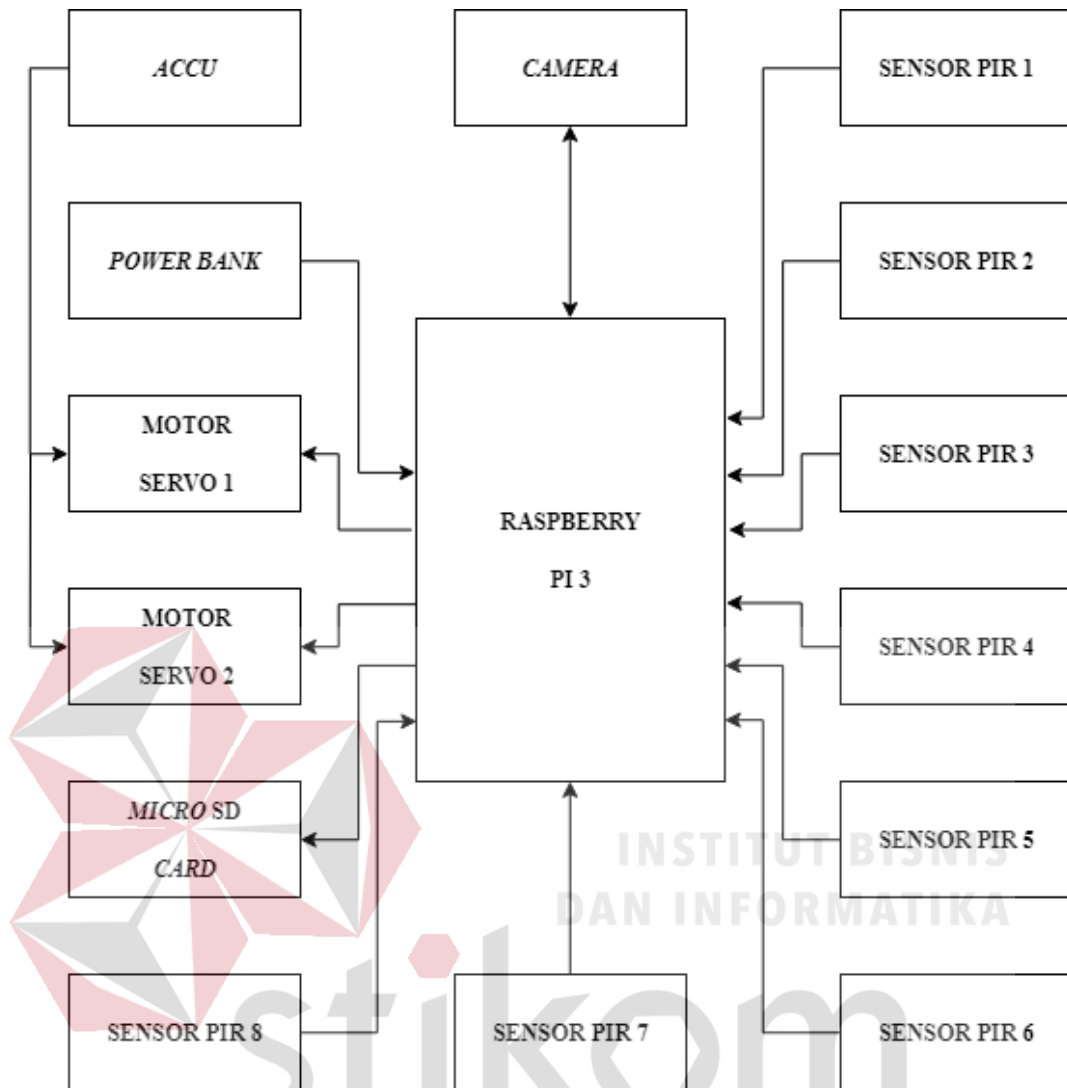
Dalam penelitian ini bahan material perangkat keras yang digunakan penulis dalam pembuatan rancang bangun *Camera Trap* adalah sebagai berikut:

1. Raspberry Pi 3.
2. Kabel *jumper*.
3. Kabel USB.
4. 8 buah sensor PIR.
5. 2 Motor Servo.
6. Power Bank.
7. *Accu*.
8. *Camera module*.
9. Kabel *camera module*.



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA



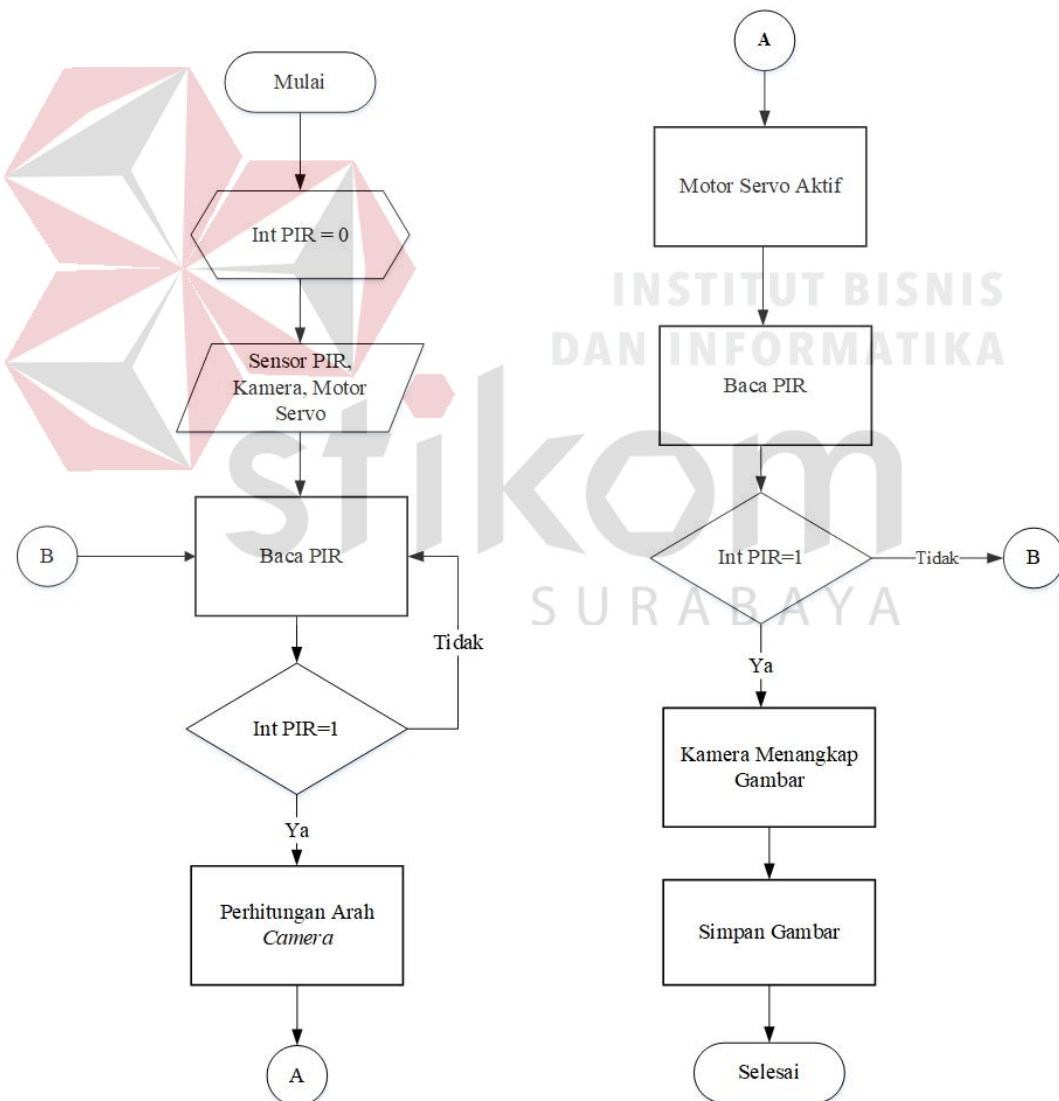
Gambar 3.11. Diagram Perancangan Perangkat Keras

Dari Gambar 3.11. dapat dilihat bahwa sistem secara keseluruhan terbagi menjadi beberapa bagian. Bagian tersebut diantaranya 8 *input* sensor PIR untuk menangkap keberadaan objek, Power Bank untuk power *supply* Raspberry Pi 3 dan Accu untuk power *supply* Motor Servo. Sedangkan untuk *output* yaitu Motor Servo untuk mengarahkan *camera* kepada objek, *camera* untuk mengambil gambar objek, *micro* SD untuk penyimpanan hasil gambar, serta menggunakan raspberry pi 3

sebagai otak dari *input* dan *output* yang berfungsi untuk mengatur jalannya sistem *input* dan *output*.

3.4 Rancangan Perangkat Lunak

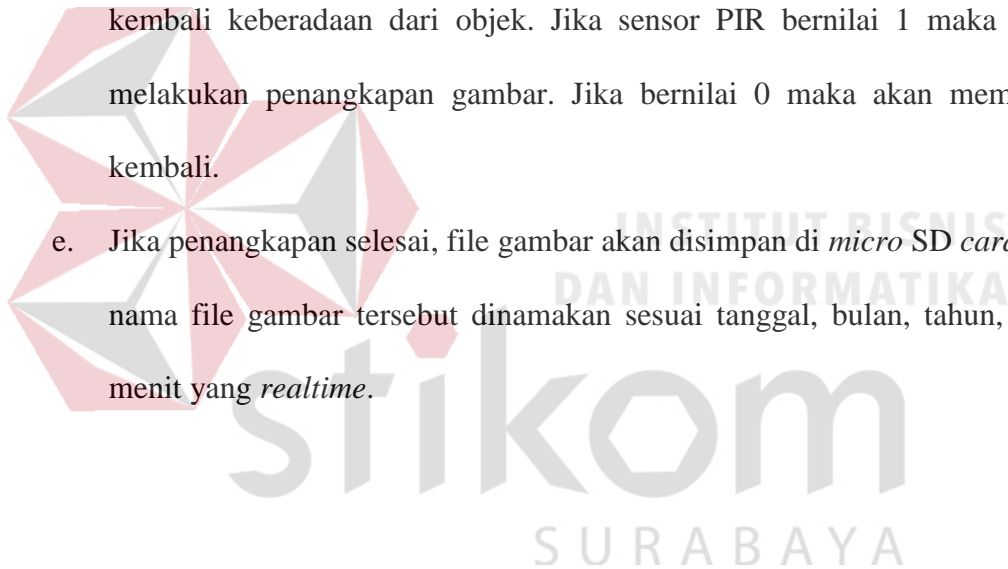
Perangkat lunak yang digunakan pada *Camera Trap* ini yaitu python sebagai *tools* pemograman Raspberry Pi 3. Apabila Raspberry Pi 3 telah diprogram maka dapat mengoperasikan komponen-komponen pendukung, yaitu *camera*, dan SD *card* serta di program untuk mengoprasikan 8 buah sensor PIR dan Motor Servo.



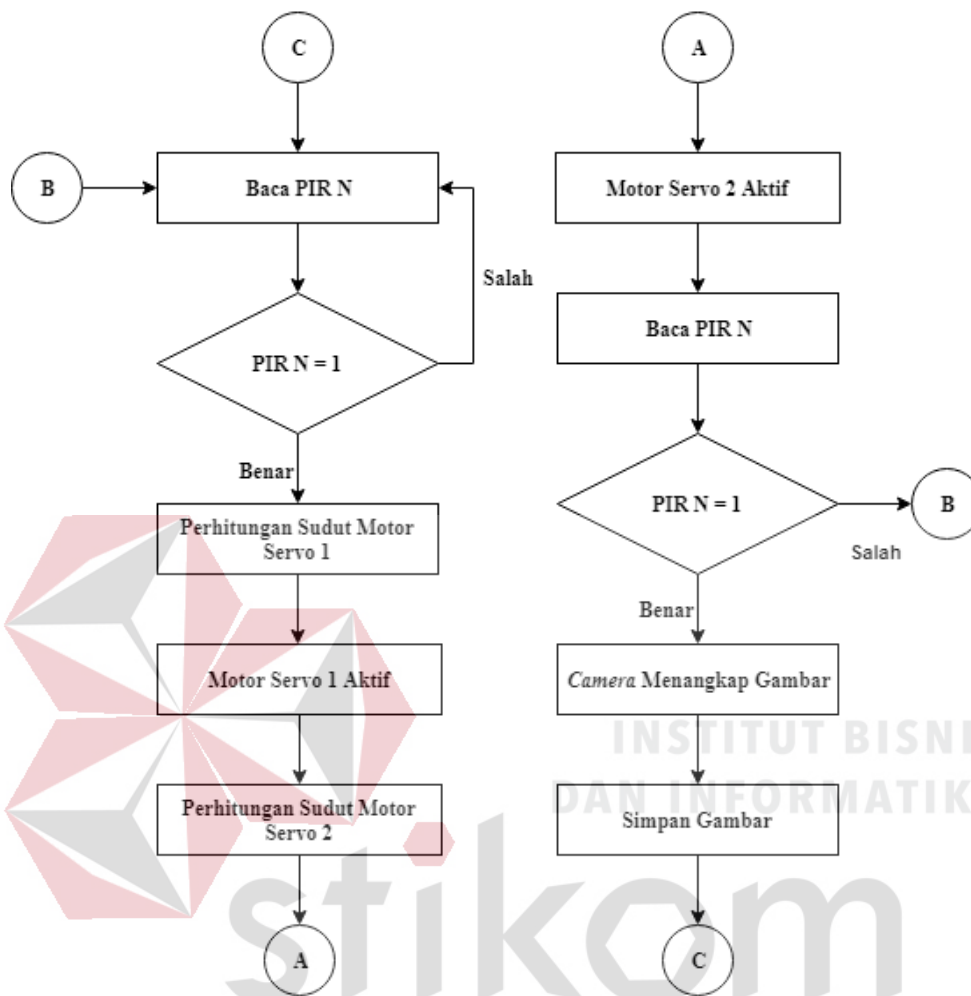
Gambar 3.12. Flow Chart Program

Proses dari *Flow Chart* Gambar 3.12 adalah sebagai berikut:

- a. Pertama sensor PIR dalam kondisi 0. Artinya dalam keadaan awal sensor PIR.
- b. Sensor PIR siap, artinya menunggu objek yang melintas di depannya.
- c. Jika sensor PIR bernilai 1 maka terdapat objek di depannya. Lalu mengaktifkan Motor Servo untuk mengarahkan posisi *camera* menghadap ke objek. Jika tidak, maka sensor PIR akan membaca ulang.
- d. Ketika posisi *camera* mengarah ke objek maka akan dilakukan pengecekan kembali keberadaan dari objek. Jika sensor PIR bernilai 1 maka akan melakukan penangkapan gambar. Jika bernilai 0 maka akan membaca kembali.
- e. Jika penangkapan selesai, file gambar akan disimpan di *micro SD card* dan nama file gambar tersebut dinamakan sesuai tanggal, bulan, tahun, jam, menit yang *realtime*.



3.4.1 Pembacaan Sensor PIR



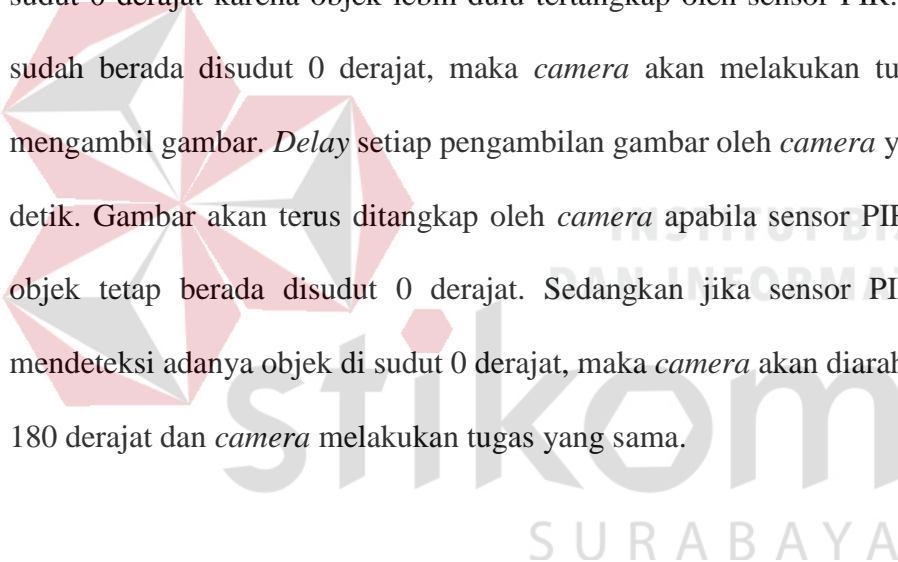
Gambar 3.13. *Flow Chart* Pembacaan Sensor PIR

Semua sensor PIR mulanya dalam kondisi 0 yaitu sensor tidak menangkap adanya objek. Jika keberadaan objek tertangkap oleh sensor PIR N maka kondisinya menjadi 1 dan melanjutkan ke proses selanjutnya yaitu menghitung sudut Motor Servo 1 lalu Motor Servo 1 aktif atau berputar ke sudut tertentu (0 derajat sampai 180 derajat). Jika Motor Servo 1 mencapai sudutnya maka Motor Servo 2 aktif atau berputar ke sudut tertentu (180 derajat sampai 360 derajat). Dalam hal ini arah *camera* tepat pada objek. Sensor PIR N akan membaca lagi apakah objek masih

pada posisi awal atau tidak. Jika masih terdapat objek dengan posisi yang sama maka *camera* akan menangkap gambar dan menyimpan ke *memory SD*. *Flow Chart* Pembacaan Sensor PIR 1 dapat dilihat pada Gambar 3.13.

3.4.2 Perencanaan Sensor PIR Menangkap Objek Lebih dari Satu

Apabila sensor PIR menangkap objek lebih dari satu, misalkan objek A berada di sudut 0 derajat dan datang lebih dulu dari pada objek B yang berada di sudut 180 derajat, maka sistem akan melakukan perhitungan arah *camera* ke arah sudut 0 derajat karena objek lebih dulu tertangkap oleh sensor PIR. Jika *camera* sudah berada disudut 0 derajat, maka *camera* akan melakukan tugasnya yaitu mengambil gambar. *Delay* setiap pengambilan gambar oleh *camera* yaitu selama 2 detik. Gambar akan terus ditangkap oleh *camera* apabila sensor PIR mendeteksi objek tetap berada disudut 0 derajat. Sedangkan jika sensor PIR tidak lagi mendeteksi adanya objek di sudut 0 derajat, maka *camera* akan diarahkan ke sudut 180 derajat dan *camera* melakukan tugas yang sama.



BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan membahas pengujian yang telah dilakukan penulis terhadap perangkat keras serta perangkat lunak dan sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui kerja dari sistem berjalan dengan baik atau tidak.

4.1 Pengujian *Camera*

4.1.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan Alat dalam menangkap gambar.

4.1.2 Alat Yang digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut:

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Memory*.
5. Laptop.

4.1.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
3. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.
4. Menjalankan program menggunakan perintah *sudo python cobacamera.py* pada LX-Terminal.

```
import time
from picamera import PiCamera
from datetime import datetime, timedelta

camera = PiCamera(resolution=(1280,720))
camera.rotation=0

def foto():
    next_hour = datetime.now()
    camera.capture('/home/pi/camera_trap/HASIL_FOTO/%s.jpg'% next_hour)
    print(next_hour)
    time.sleep(2)
```

5. Amati hasil dari program tersebut.

4.1.4 Hasil Pengujian

Dari prosedur pengujian di atas, telah mendapatkan hasil gambar yang dapat di lihat pada gambar 4.1 sampai gambar 4.3.



Gambar 4.1. Hasil Gambar 1



Gambar 4.2. Hasil Gambar 2



Gambar 4.3. Hasil Gambar 3

4.2 Pengujian Motor Servo

4.2.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan Alat dalam menggerakkan Motor Servo sesuai sudut yang ditentukan.

4.2.2 Alat Yang digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut:

1. Raspberry Pi 3.
2. Power Bank.
3. Motor Servo.
4. *Memory*.
5. Laptop.

4.2.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *Accu*.
2. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
3. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.

4. Menjalankan program menggunakan perintah *sudo python cobaservo.py* pada LX-Terminal.

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.OUT)

#servo
pwm1 = GPIO.PWM(4,50)
pwm1.start(10)
pwm2 = GPIO.PWM(17,50)
pwm2.start(10)

try:
    while 1:
        print("Sudut 180 derajat")
        DC=1./18.*(180)+2
        pwm1.ChangeDutyCycle(DC)
        DC=1./18.*(175)+2
        pwm2.ChangeDutyCycle(DC)
        print("Sudut 90 derajat")
        DC=1./18.*(180)+2
        pwm1.ChangeDutyCycle(DC)
        DC=1./18.*(85)+2
        pwm2.ChangeDutyCycle(DC)
```

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

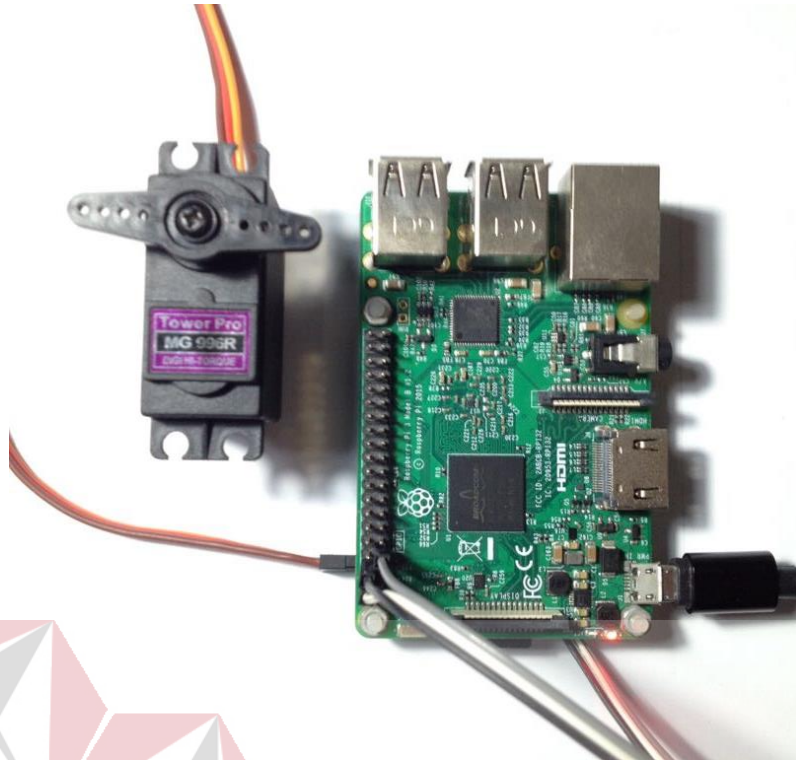
5. Amati hasil dari program tersebut.

4.2.4 Hasil Pengujian

Dari prosedur pengujian di atas, telah mendapatkan hasil gambar yang dapat di lihat pada gambar 4.4 dan 4.5. dan Tabel 4.1. terdapat sudut yang di inginkan dan hasil sudut servo percobaan. Hasil sudut servo 1 percobaan dan hasil sudut servo 2 percobaan di masukan ke dalam *code* Python.

Tabel 4.1. Percobaan Sudut Servo 1 dan Servo 2

No.	Sudut (°)	Sudut Servo 1 (°)	Sudut Servo 1 Hasil Percobaan (°)	Sudut Servo 2 (°)	Sudut Servo 2 Hasil Percobaan (°)
1.	360/0	180	180	0	6
2.	22.5	180	180	22.5	20
3.	45	180	180	45	43
4.	67.5	180	180	67.5	63
5.	90	180	180	90	85
6.	112.5	180	180	112.5	110
7.	135	180	180	135	130
8.	157.5	180	180	157.5	155
9.	180	180	180	180	175
10.	202.5	22.5	22.5	0	20
11.	225	45	45	0	20
12.	247.5	67.5	67.5	0	20
13.	270	90	90	0	20
14.	292.5	112.5	112.5	0	20
15.	315	135	135	0	20
16.	337.5	157.5	157.5	0	20



Gambar 4.4. Pengujian Motor Servo 180°



Gambar 4.5. Pengujian Motor Servo 90°

4.3 Pengujian Sensor PIR

4.3.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan Alat dalam menangkap suatu objek.

4.3.2 Alat Yang digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut:

1. Raspberry Pi 3.
2. Power Bank.
3. Sensor PIR
4. *Memory*.
5. Laptop.

4.3.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
2. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
3. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.

4. Menjalankan program menggunakan perintah *sudo python cobasensor.py* pada LX-Terminal.

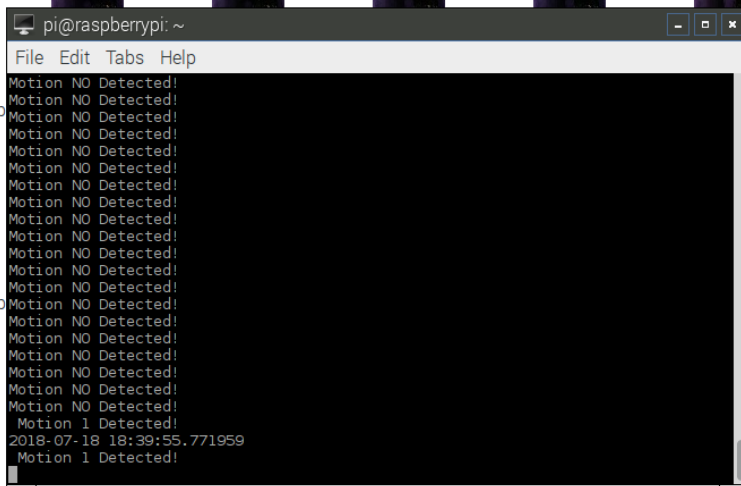
```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(21, GPIO.IN)
GPIO.setup(20, GPIO.IN)
try:
    while 1:
        if (GPIO.input(21) == 1):
            print("Motion 1 Detected!")
        elif (GPIO.input(20) == 1):
            print("Motion 2 Detected!")
        else:
            print("Motion NO Detected!")
```

5. Amati hasil dari program tersebut.

4.3.4 Hasil Pengujian

Dari prosedur pengujian di atas, telah mendapatkan hasil gambar yang dapat di lihat pada gambar 4.6 dan 4.7.

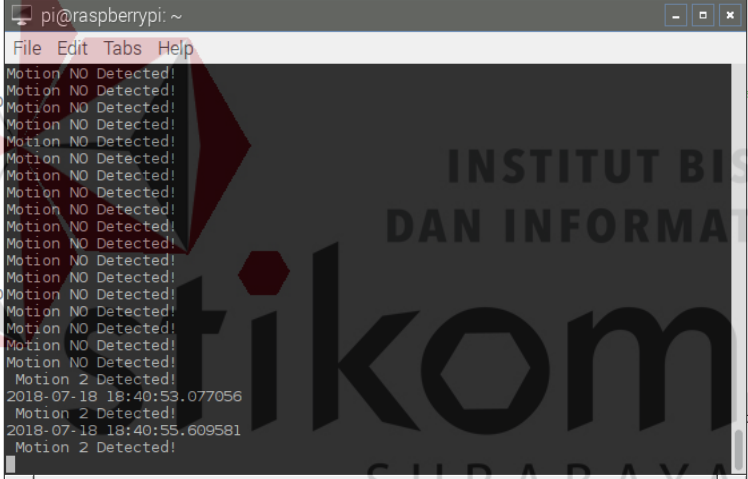


```

pi@raspberrypi: ~
File Edit Tabs Help
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion 1 Detected!
2018-07-18 18:39:55.771959
Motion 1 Detected!

```

Gambar 4.6. Hasil jika Sensor 1 mendeteksi objek



```

pi@raspberrypi: ~
File Edit Tabs Help
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion NO Detected!
Motion 2 Detected!
2018-07-18 18:40:53.077056
Motion 2 Detected!
2018-07-18 18:40:55.609581
Motion 2 Detected!

```

Gambar 4.7. Hasil jika Sensor 2 mendeteksi objek

4.4 Pengujian Alat terhadap Gerakan Satwa

4.4.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan Alat dalam mendeteksi gerakan satwa.

4.4.2 Alat yang Digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut.

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Accu*.
5. 2 buah Motor Servo.
6. Satwa.
7. 8 buah Sensor PIR.
8. *Memory*.
9. Laptop.

4.4.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* Accu.
3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Desktop* pada windows 10.

6. Menjalankan program menggunakan perintah *sudo python programTA.py* pada LX-Terminal.
7. Amati hasil dari program tersebut.

4.4.4 Hasil Pengujian

Dari prosedur pengujian di atas, meskipun objek tersebut berupa satwa, manusia atau benda dengan skenario yang berbeda-beda namun sensor PIR mendeteksi adanya suhu tubuh, maka hasil dinyatakan terdeteksi. Sebaliknya, jika sensor PIR tidak mendeteksi adanya suhu tubuh, maka hasil dinyatakan tidak terdeteksi. Apabila LX-Terminal mendapatkan hasil *Motion Detection* maka dinyatakan sebagai gerakan satwa. Hasil dari pengujian alat terhadap gerakan objek dapat dilihat pada Tabel 4.2. dan penempatan *Camera Trap* di Taman Nasional Baluran Situbondo pada Gambar 4.8. sampai Gambar 4.13.

Tabel 4.2. Pengujian Alat terhadap Gerakan Objek

No.	Skenario	Keterangan	Keberhasilan
1.	Rusa	Hewan	Berhasil
2.	Rusa	Hewan	Berhasil
3.	Rusa	Hewan	Berhasil
4.	Rusa	Hewan	Berhasil
5.	Rusa	Hewan	Berhasil



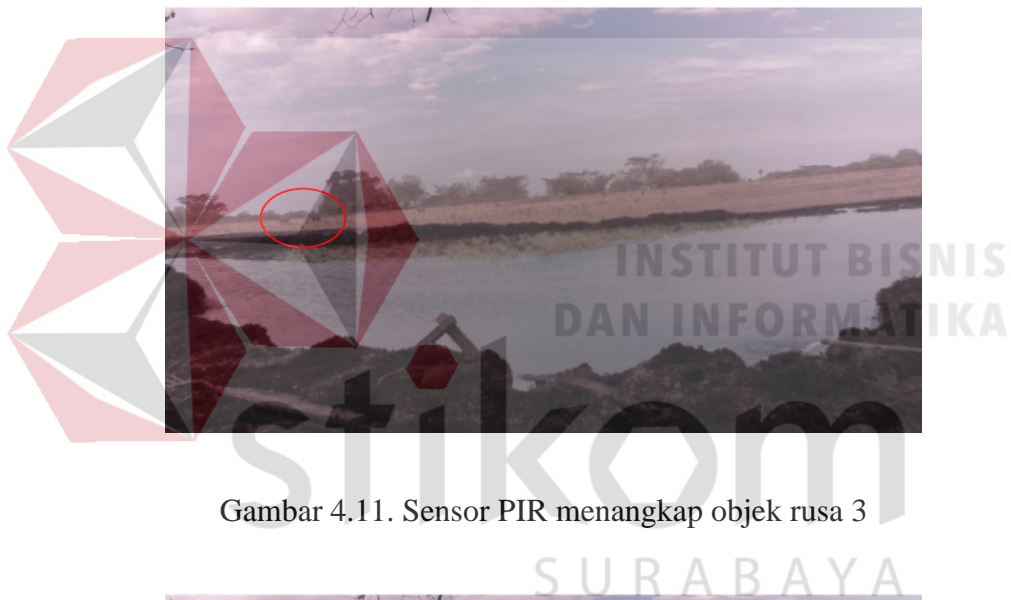
Gambar 4.8. Penempatan *Camera Trap* di Taman Nasional Baluran Situbondo



Gambar 4.9. Sensor PIR menangkap objek rusa



Gambar 4.10. Sensor PIR menangkap objek rusa 2



Gambar 4.11. Sensor PIR menangkap objek rusa 3



Gambar 4.12. Sensor PIR menangkap objek rusa 4



Gambar 4.13. Sensor PIR menangkap objek rusa 5

4.5 Pengujian Jarak

4.5.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan sensor PIR dan *camera* terhadap jarak yang ditangkap.

4.5.2 Alat yang Digunakan

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Accu*.
5. 2 buah Motor Servo.
6. Satwa atau manusia.
7. 8 buah Sensor PIR.
8. *Memory*.
9. Laptop.

4.5.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* Accu.
3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.
6. Menentukan objek pada jarak 1m sampai 8m.
7. Menjalankan program menggunakan perintah *sudo python programTA.py* pada LX-Terminal.
8. Amati hasil dari program tersebut.

4.5.4 Hasil Pengujian

Dari prosedur pengujian di atas, apabila keberadaan objek kurang dari 5 meter, maka hasil dinyatakan terdeteksi. Jika lebih dari 5 meter, maka hasil dinyatakan tidak terdeteksi. Pengujian jarak dapat dilihat pada Tabel 4.3.

Tabel 4.3. Pengujian Jarak

No.	Jarak (m)	Keberhasilan
1.	1	Terdeteksi
2.	2	Terdeteksi
3.	3	Terdeteksi
4.	4	Terdeteksi
5.	5	Terdeteksi
6.	6	Tidak Terdeteksi
7.	7	Tidak terdeteksi
8.	8	Tidak terdeteksi

4.5.5 Hasil Foto

Hasil gambar dapat dilihat pada Gambar 4.14. sampai Gambar 4.8.



Gambar 4.14. Terdeteksi 5m



Gambar 4.15. Terdeteksi 4m



Gambar 4.16. Terdeteksi 3m



Gambar 4.17. Terdeteksi 2m



Gambar 4.18. Terdeteksi 1m

4.6 Pengujian Ketepatan Sudut

4.6.1 Tujuan

Pengujian dari proses ini untuk mengetahui sudut Motor Servo sesuai dengan sudut yang ditangkap oleh sensor PIR.

4.6.2 Alat yang Digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut.

1. Raspberry Pi 3.
2. *Camera Module.*
3. Power Bank.
4. *Accu.*
5. 2 buah Motor Servo.
6. Satwa.
7. 8 buah Sensor PIR.
8. *Memory.*
9. Laptop.

4.6.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* Accu.
3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Desktop* pada windows 10.
6. Menentukan objek pada sudut yang diinginkan.
7. Menjalankan program menggunakan perintah *sudo python programTA.py* pada LX-Terminal.
8. Amati hasil dari sudut *camera*.

4.6.4 Hasil Pengujian

Dari prosedur di atas, *error* sudut yang dihasilkan dijadikan acuan sebagai pengujian terhadap kepresisian kontrol. Terdapat 30 kali percobaan yang mendapatkan hasil yang berbeda. Pada Tabel 4.4. menjelaskan bahwa pengujian ketepatan sudut 360 derajat sampai 90 derajat yang menghasilkan rata-rata dan *error* di setiap sudut.

Tabel 4.4. Pengujian Ketepatan Sudut 360 derajat – 90 derajat

Pengujian ke-	Sudut (°)				
	360	22.5	45	67.5	90
	Hasil Sudut yang dihasilkan (°)				
1.	358	20.5	42	62.5	85
2.	360	17.5	45	65.5	88
3.	357	18.5	40	67.5	90
4.	358	19.5	45	67.5	90
5.	360	22.5	42	63	85.5
6.	360	22.5	45	67.5	90
7.	357	22.5	45	64.5	87
8.	360	19	41	67.5	90
9.	356.5	20.5	42	63	85.5
10.	360	22.5	40.5	60.5	83
11.	358	31	39	67.5	90
12.	357	16.5	45	63	85.5
13.	360	22.5	38	61.5	84
14.	358	19	45	70.5	93
15.	364	15.5	36	60	82.5
16.	357	22.5	41	70.5	93
17.	357	24.5	45	62.5	85
18.	360	18	34	67.5	90
19.	357	22.5	39	63.5	86
20.	363	22.5	36.5	67.5	90
21.	360	18	39	64	86.5
22.	358	16.5	40	65.5	88
23.	360	22.5	45	67.5	90
24.	354	15.5	39	67.5	90
25.	356	22.5	41	63.5	86
26.	360	13.5	45	67.5	90
27.	358	18.5	43	64.5	87
28.	360	25.5	39	67.5	90
29.	357	24.5	42	62	84.5
30.	360	22.5	39	65.5	88
Rata-rata	358.7	20.7	41.3	65.3	87.8

Pada Tabel 4.5. menjelaskan bahwa pengujian ketepatan sudut 112,5 derajat sampai 202,5 derajat yang menghasilkan rata-rata dan *error* di setiap sudut.

Tabel 4.5. Pengujian Ketepatan Sudut 112,5 derajat – 202,5 derajat

Pengujian ke-	Sudut (°)				
	112.5	135	157.5	180	202.5
	Hasil Sudut yang dihasilkan (°)				
1	108	134	155	177	200.5
2.	106.5	134	157	180	202.5
3.	112.5	130.5	154	175	199.5
4.	105.5	134	157	180	202.5
5.	104.5	132	157	177	202.5
6.	103	134	157	180	202.5
7.	108.5	134	156	180	201.5
8.	112.5	134	157	176	202.5
9.	112.5	134	157	177	202.5
10.	110.5	134	153	175.5	198.5
11.	112.5	130	157	174	202.5
12.	112.5	131	157	180	202.5
13.	108.5	130.5	152.5	173	202.5
14.	112.5	132	151	180	200.5
15.	109.5	134	157	171	202.5
16.	112.5	134	150	176	202.5
17.	112.5	130	149	180	202.5
18.	112.5	134	148	169	202.5
19.	109	139	157	174	202.5
20.	112.5	134	154	171.5	199.5
21.	110.5	134	157	174	202.5
22.	112.5	134	157	175	202.5
23.	108.5	134	162	180	202.5
24.	109.5	138	157	174	202.5
25.	112.5	134	157	176	202.5
26.	109.5	134	157	180	202.5
27.	115.5	128	153	178	198.5
28.	112.5	134	157	174	202.5
29.	112.5	132	157	177	202.5
30.	108.5	131	152	174	197.5
Rata-rata	110.4	133.2	155.3	176.3	201.7

Pada Tabel 4.6. menjelaskan bahwa pengujian ketepatan sudut 225 derajat sampai 337,5 derajat yang menghasilkan rata-rata dan *error* di setiap sudut.

Tabel 4.6. Pengujian Ketepatan Sudut 225 derajat – 337,5 derajat

Pengujian ke-	Sudut (°)					
	225	247.5	270	292.5	315	337.5
	Hasil Sudut yang dihasilkan (°)					
1.	222	247.5	270	292.5	315	337.5
2.	222	243.5	270	292.5	314	336.5
3.	218.5	247.5	268	289	315	337.5
4.	222	244.5	270	292.5	315	337.5
5.	220	247.5	270	290.5	311	333.5
6.	222	247.5	270	292.5	315	333.5
7.	222	247.5	270	292.5	315	334.5
8.	222	244	270	292.5	315	333
9.	222	244.5	267	292.5	313	332
10.	222	245.5	270	292.5	315	337.5
11.	218	247.5	270	288.5	315	330.5
12.	219	243.5	270	289.5	315	329.5
13.	218.5	244.5	274	289	315	328.5
14.	220	247.5	270	290.5	315	337.5
15.	222	244.5	270	292.5	312	334.5
16.	222	250.5	266	292.5	315	337.5
17.	218	245.5	270	288.5	311.5	337.5
18.	222	244.5	270	292.5	313	334
19.	227	247.5	270	297.5	315	337.5
20.	222	243.5	270	292.5	315	337.5
21.	222	247.5	268	292.5	311	337.5
22.	222	244.5	270	292.5	315	337.5
23.	222	247.5	270	292.5	312	337.5
24.	226	247.5	273	296.5	315	334.5
25.	222	247.5	270	292.5	315	337.5
26.	222	244	270	292.5	315	337.5
27.	216	250.5	267	286.5	315	337.5
28.	222	245.5	272	292.5	312	337.5
29.	220	247.5	270	290.5	315	337.5
30.	219	243.5	270	289.5	315	333
Rata-rata	221.2	246.1	269.8	291.7	314.2	335.5

Tabel 4.7. *Error Sudut 360° atau 0° sampai 157,5°*

No.	Sudut (°)							
	360	22.5	45	67.5	90	112.5	135	157.5
	Hasil <i>Error</i>							
1.	2	2	3	5	5	4.5	1	2.5
2.	0	5	0	2	2	6	1	0.5
3.	3	4	5	0	0	0	4.5	3.5
4.	2	3	0	0	0	7	1	0.5
5.	0	0	3	4.5	4.5	8.5	3	0.5
6.	0	0	0	0	0	9.5	1	0.5
7.	3	0	0	3	3	4	1	1.5
8.	0	3.5	4	0	0	0	1	0.5
9.	3.5	2	3	4.5	4.5	0	1	0.5
10.	0	0	4.5	7	7	2	1	4.5
11.	2	8.5	6	0	0	0	5	0.5
12.	3	6	0	4.5	4.5	0	4	0.5
13.	0	0	7	6	6	4	4.5	5
14.	2	3.5	0	3	3	0	3	6.5
15.	4	7	9	7.5	7.5	3	1	0.5
16.	3	0	4	3	3	0	1	7.5
17.	3	2	0	5	5	0	5	8.5
18.	0	4.5	11	0	0	0	1	9.5
19.	3	0	6	4	4	3.5	4	0.5
20.	3	0	8.5	0	0	0	1	3.5
21.	0	4.5	6	3.5	3.5	2	1	0.5
22.	2	6	5	2	2	0	1	0.5
23.	0	0	0	0	0	4	1	4.5
24.	6	7	6	0	0	3	3	0.5
25.	4	0	4	4	4	0	1	0.5
26.	0	9	0	0	0	3	1	0.5
27.	2	4	2	3	3	3	7	4.5
28.	0	3	6	0	0	0	1	0.5
29.	3	2	3	5.5	5.5	0	3	0.5
30.	0	0	6	2	2	4	4	5.5

Tabel 4.8. *Error* Sudut 180° sampai 337,5°

No.	Sudut (°)							
	180	202.5	225	247.5	270	292.5	315	337.5
	Hasil <i>Error</i>							
1.	3	2	3	0	0	0	0	0
2.	0	0	3	4	0	0	1	1
3.	5	3	6.5	0	2	3.5	0	0
4.	0	0	3	3	0	0	0	0
5.	3	0	5	0	0	2	4	4
6.	0	0	3	0	0	0	0	4
7.	0	1	3	0	0	0	0	3
8.	4	0	3	3.5	0	0	0	4.5
9.	3	0	3	3	3	0	2	5.5
10.	4.5	4	3	2	0	0	0	0
11.	6	0	7	0	0	4	0	7
12.	0	0	6	4	0	3	0	8
13.	7	0	6.5	3	4	3.5	0	9
14.	0	2	5	0	0	2	0	0
15.	9	0	3	3	0	0	3	3
16.	4	0	3	3	4	0	0	0
17.	0	0	7	2	0	4	3.5	0
18.	11	0	3	3	0	0	2	3.5
19.	6	0	2	0	0	5	0	0
20.	8.5	3	3	4	0	0	0	0
21.	6	0	3	0	2	0	4	0
22.	5	0	3	3	0	0	0	0
23.	0	0	3	0	0	0	3	0
24.	6	0	1	0	3	4	0	3
25.	4	0	3	0	0	0	0	0
26.	0	0	3	3.5	0	0	0	0
27.	2	4	9	3	3	6	0	0
28.	6	0	3	2	2	0	3	0
29.	3	0	5	0	0	2	0	0
30.	6	5	6	4	0	3	0	4.5

Dari perhitungan *error* pada Tabel 4.7. dan Tabel 4.8. maka jika di jumlah semua akan menghasilkan jumlah 1070 dan di bagi dengan banyaknya percobaan dikali 16 sudut maka menghasilkan jumlah 480.

$$\text{Prosentase Rata – rata Error} = \frac{1070}{480} = 2,2292$$

Dari percobaan sebanyak 30 kali pada Tabel 4.4 sampai Tabel 4.6 dapat disimpulkan bahwa *error* sudut pada Motor Servo memiliki nilai selisih yang kecil. Hal ini tersebut dibuktikan dari *error* yang terbesar yaitu 11 %. Persentase rata-rata *error* sebesar 2,2292 % yang berarti kendali Motor Servo ini memiliki tingkat presisi 97,7708 %.

4.7 Pengujian Waktu Hidup *Camera Trap*

4.7.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan *Camera Trap* dapat bertahan berapa lama.

4.7.2 Alat yang Digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut.

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Accu*.
5. 2 buah Motor Servo.
6. Satwa.
7. 8 buah Sensor PIR.
8. *Memory*.
9. Laptop.

4.7.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* Accu.
3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.
6. Menjalankan program menggunakan perintah *sudo python programTA.py* pada LX-Terminal.
7. Amati hasil dari Power Bank tersebut.

4.7.4 Hasil Pengujian

Camera Trap membutuhkan energi yang besar karena selalu dalam kondisi hidup, sedangkan kebutuhan energi LED IR tidak terlalu besar karena hanya hidup saat kondisi lingkungan gelap dan terdapat satwa yang melintas. Kapasitas energi yang tersimpan pada Power Bank sebesar $5,11 \text{ V} \times 10050 \text{ mAH} = 51,3555 \text{ VAH}$.

Pengukuran besarnya arus yang dibutuhkan oleh *Camera Trap* dengan sumber catu daya Power Bank adalah 2,4A. Hal ini berarti bahwa *Camera Trap* membutuhkan daya sebesar $5,11\text{V} \times 2,4\text{A} = 12,264 \text{ VA}$ untuk beroperasi. Jika

Power Bank memiliki energi sebesar 51,3555 maka menggunakan persamaan 1 waktu hidup kamera dapat ditentukan.

$$\text{Waktu hidup camera} = \frac{51,3555}{12,264} = 4,1875 \text{ jam}$$

Jadi secara teoritis *camera* dapat hidup secara terus menerus selama 4,1875 jam. Pengujian secara praktek dilakukan selama 1 hari yang berbeda dengan cara menyalakan alat ini kemudian sensor PIR diberi trigger berupa objek yang melewati alat tersebut. Data hasil pengujian dapat dilihat pada Tabel 4.9.

Tabel 4.9. Pengujian Waktu Hidup *Camera Trap*

Percobaan	Tanggal	Kondisi <i>Camera</i>		Waktu Hidup (jam)
		Hidup	Mati	
1	4.7.2017	6:00	9:48	3,48
2	4.7.2017	22:00	1:42	3,42
3	4.7.2017	22:00	1:52	3,52
4	14.7.2017	9:00	12:43	3,43
5	17.7.2017	22.00	1:58	3,58
Rata-rata ketahanan <i>Camera Trap</i>				3,486

Efektifitas *Camera Trap* ditentukan dengan membandingkan rata-rata ketahanan kamera yang diperoleh dalam pengujian dengan nilai waktu hidup *camera* secara teoritis. Persentasi efektifitas daya tahan *Camera Trap* adalah sebesar:

$$\text{Efektifitas camera} = \frac{3,486}{4,1875} \times 100\% = 83,24776119\%$$

4.8 Pengujian Pengambilan Gambar pada Kondisi minim Cahaya

4.8.1 Tujuan

Pengujian dari proses ini untuk mengetahui kemampuan *Camera Trap* dapat bekerja secara optimal pada kondisi minim cahaya.

4.8.2 Alat yang Digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut.

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Accu*.
5. 2 buah Motor Servo.
6. Satwa.
7. 8 buah Sensor PIR.
8. *Memory*.
9. Laptop.

4.8.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.

1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* *Accu*.

3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Dekstop* pada windows 10.
6. Menjalankan program menggunakan perintah *sudo python programTA.py* pada LX-Terminal.
7. Amati hasil dari gambar.

4.8.4 Hasil Pengujian

Dari prosedur di atas, pada cahaya minimal keberadaan objek yang tertangkap kamera ditentukan secara subyektif. Dimana kemampuan *Camera Trap* dalam menangkap objek ditentukan berdasarkan pendapat orang yang melihat gambar hasil tangkapan *camera*. Pengujian Pengambilan Gambar pada Kondisi minim Cahaya dapat dilihat pada Tabel 4.10.

Tabel 4.10. Pengujian Pengambilan Gambar pada Kondisi minim Cahaya

No.	Jarak (m)	Hasil Gambar	Keterangan
1.	5		Terlihat kurang jelas

No.	Jarak (m)	Hasil Gambar	Keterangan
2.	4		Terlihat kurang jelas
3.	3		Terlihat jelas
4.	2		Terlihat jelas
5.	1		Terlihat jelas

4.9 Pengujian Kebutuhan Ukuran *File* Gambar

4.9.1 Tujuan

Pengujian dari proses ini untuk mengetahui rata-rata ukuran *file* gambar tersimpan di *memory*.

4.9.2 Alat yang Digunakan

Peralatan yang dibutuhkan untuk pengujian alat ini adalah sebagai berikut.

1. Raspberry Pi 3.
2. *Camera Module*.
3. Power Bank.
4. *Accu*.
5. 2 buah Motor Servo.
6. Satwa.
7. 8 buah Sensor PIR.
8. *Memory*.
9. Laptop.

4.9.3 Prosedur Pengujian

Langkah-langkah yang dilakukan untuk melakukan pengujian sistem adalah seperti berikut.




1. Menyambungkan kabel CSI *camera* dengan Raspberry Pi 3.
2. Menyambungkan kabel data Motor Servo dengan Raspberry Pi 3 dan menyambungkan kabel *ground* dan *vcc* dengan *ground* dan *vcc* Accu.
3. Menyambungkan 8 buah sensor PIR dengan Raspberry Pi 3.
4. Menyambungkan Power Bank dan dihubungkan dengan *port power* pada Raspberry Pi 3, maka Raspberry Pi 3 dalam keadaan *on*.
5. Mengaktifkan laptop untuk mengakses OS Raspberry Pi 3 menggunakan aplikasi *Remote Desktop* pada windows 10.



6. Menjalankan program menggunakan perintah `sudo python programTA.py` pada LX-Terminal.
7. Amati ukuran *file* dari gambar.

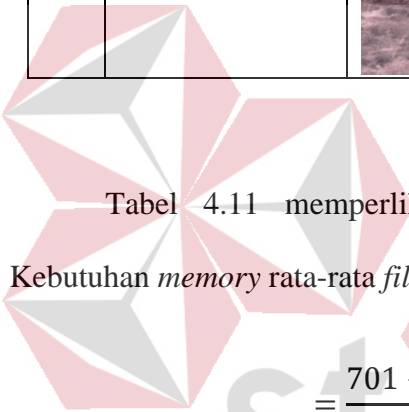
4.9.4 Hasil Pengujian

Dari prosedur di atas, setiap ukuran *file* gambar yang tersimpan akan dibagi banyaknya pengujian untuk didapat ukuran rata-rata setiap *file* gambar. Jika rata-rata tersebut sudah didapatkan maka akan diketahui jumlah gambar yang tersimpan di *memory*. Pengujian kebutuhan ukuran file gambar dapat dilihat pada Tabel 4.11.

Tabel 4.11. Pengujian Kebutuhan Ukuran File Gambar

No.	Waktu	Hasil Gambar	Ukuran File
1.	2018-07-14 11:00:16		701kB
2.	2018-07-14 11:32:15		741 kB
3.	2018-07-14 15:26:21		483 kB

No.	Waktu	Hasil Gambar	Ukuran File
4.	2018-07-14 10:59:43		631 kB
5.	2018-07-14 10:59:41		677 kB



Tabel 4.11 memperlihatkan kebutuhan masing-masing *file* gambar.

Kebutuhan *memory* rata-rata *file* gambar adalah:

$$\begin{aligned}
 &= \frac{701 + 741 + 483 + 631 + 677}{5} \\
 &= \frac{3233}{5} = 646,6 \text{ kB}
 \end{aligned}$$

Jika keberadaan satwa tersebar merata dalam satu hari maka kapasitas *memory* yang berukuran 32 GB dapat menyimpan gambar sebanyak,

$$\text{Jumlah Gambar} = \frac{32.000.000 \text{ kB}}{646,6 \text{ kB}}$$

$$\text{Jumlah Gambar} = 49.489 \text{ gambar}$$

Hasil pengambilan gambar disimpan pada *memory* dengan penamaan berkas sesuai dengan waktu pengambilan gambar tersebut (*realtime*). Hal ini bertujuan untuk memudahkan mengetahui kapan objek pengambil gambar.

$$x = 10 \times 49.489$$

$$x = 494.890$$

$$x = \frac{494.890}{1}$$

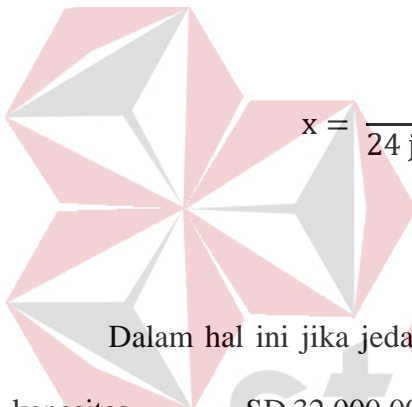
$$x = 494.890 \text{ detik}$$

Atau

$$x = \frac{494.890}{24 \text{ jam} \times 60 \text{ menit} \times 60 \text{ detik}}$$

$$x = 5,728 \text{ hari}$$

Dalam hal ini jika jeda atau *delay* pengambilan gambar selama 10 detik, kapasitas *memory* SD 32.000.000 kB dan rata-rata ukuran *file* gambar sebesar 646.6 kB maka *memory* akan penuh selama pemakaian 494.890 detik atau 5,728 hari.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan dapat disimpulkan bahwa:

1. Telah terealisasi rancang bangun *Camera Trap* berbasis Raspberry Pi 3 yang mampu menggerakkan *camera* sesuai dengan posisi satwa dengan menggunakan sensor PIR sebagai pancaran infra merah dari objek. Dengan perhitungan manual didapatkan hasil 16 sudut. Terdapat kombinasi tiga sensor PIR yang dapat bekerja bersamaan.
2. Setelah sudut sudah ditetapkan maka sudut *camera* mengikuti sesuai arah objek dengan menggunakan dua buah servo 180°. Motor Servo 1 sebagai servo bawah yang mengatur sudut 0° - 180°, sedangkan Motor Servo 2 sebagai servo atas yang mengatur sudut 180° - 360°. Pada pengujian, *Error* sudut pada Motor Servo memiliki nilai selisih yang kecil. Hal ini dibuktikan dari *error* yang terbesar yaitu 11 %. Persentase rata-rata *error* sebesar 2,2292 % yang berarti kendali Motor Servo ini memiliki tingkat presisi 97,7708 %.
3. *Camera trap* ini dapat mendeteksi hewan yaitu rusa. Jarak jangkauan sensor PIR terjauh yaitu 5 meter di sudut 0°. *Life time* alat ini rata-rata selama 4,2 jam. Dapat menampung 49.489 gambar dengan rata-rata ukuran gambar 646.6 kB, pada memori 32 GB hingga penuh dapat terpakai 137,469 jam

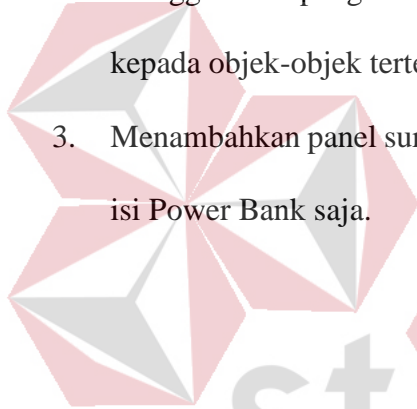
atau 5,728 hari. Pada kondisi kurang cahaya, alat ini mampu mengambil gambar objek dengan jelas pada jarak maksimal 3 meter.

5.2 Saran

Dalam pembuatan *Camera Trap* ini terdapat beberapa saran untuk perbaikan

penelitian yang akan datang:

1. Menggunakan Motor Servo yang tidak menimbulkan suara yang keras.
2. Menggunakan pengolahan citra agar pengambilan gambar dapat difokuskan kepada objek-objek tertentu saja.
3. Menambahkan panel surya agar waktu hidup *camera* tidak bergantung pada isi Power Bank saja.



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

DAFTAR PUSTAKA

- Inilah Perbedaan Memori SD Card, Mini SD, dan Micro SD.* (n.d.). Retrieved from Carisinyal: <https://carisinyal.com/memori-sd-card/>
- Kamera jebakan.* (n.d.). Retrieved from RIAU: http://riau.kabel.web.id/en1/1066-962/Kamera-Jebakan_135130_riau-kabel.html
- Motor Servo.* (2014, Desember 14). Retrieved from Zona Elektro: <http://zoniaelektro.net/motor-servo/>
- Pinout!* (n.d.). Retrieved from Pinout: <https://pinout.xyz/>
- RASPBERRY PI LESSON 28: CONTROLLING A SERVO ON RASPBERRY PI WITH PYTHON.* (2015, 6 2). Retrieved from www.toptechboy.com: <http://www.toptechboy.com/raspberry-pi/raspberry-pi-lesson-28-controlling-a-servo-on-raspberry-pi-with-python/>
- Sensor Gerak PIR (Passive Infra Red).* (2012, Juli 23). Retrieved from E-BELAJAR ELEKTRONIKA: <http://e-belajarelektronika.com/sensor-gerak-pir-passive-infra-red/>
- Widipratama, M. Z. (2017). *Sistem Pemantau Keamanan Menggunakan Kamera dengan Metode Background Subtraction*. Surabaya: Tugas Akhir Institut Bisnis dan Informatika Stikom Surabaya.

LAMPIRAN

Lampiran 1. Program Keseluruhan

```
import time

import RPi.GPIO as GPIO

from picamera import PiCamera

from datetime import datetime, timedelta

camera = PiCamera(
    resolution=(1280,720))
camera.rotation=0
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(21, GPIO.IN)
GPIO.setup(20, GPIO.IN)
GPIO.setup(16, GPIO.IN)
GPIO.setup(12, GPIO.IN)
GPIO.setup(7, GPIO.IN)
GPIO.setup(8, GPIO.IN)
GPIO.setup(25, GPIO.IN)
GPIO.setup(24, GPIO.IN)

#servo

pwm1 = GPIO.PWM(4,50)
```

```

pwm1.start(10)

pwm2 = GPIO.PWM(17,50)

pwm2.start(10)

print("Motion Sensor Alarm (CTRL+C to exit)")

print("Ready")

#takecamera

def foto():

    next_hour = datetime.now()

    camera.capture('/home/pi/camera_trap/HASIL_FOTO/%s.jpg' % next_hour)

    print(next_hour)

    time.sleep(10)

try:

    while 1:

        if (GPIO.input(24) == 1) and (GPIO.input(21) == 1) and (GPIO.input(20) == 1):

            print("Motion 8 , 1 , 2 Detected!")

            print(" ")

            DC=1./18.*(180)+2

            pwm1.ChangeDutyCycle(DC)

            print("A . 180")

            DC=1./18.*(6)+2

            pwm2.ChangeDutyCycle(DC)

            print("B . 6")

            print(" ")

            if (GPIO.input(24) == 1) and (GPIO.input(21) == 1) and (GPIO.input(20) == 1):

```

```

    foto()
else:
    print("Motion NO Detected Again!")
elif(GPIO.input(21) == 1) and (GPIO.input(20) == 1) and (GPIO.input(16) == 1):
    print("Motion 1 , 2 , 3 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(43)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 43")
    print(" ")
    if (GPIO.input(21) == 1) and (GPIO.input(20) == 1) and (GPIO.input(16) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")
elif (GPIO.input(20) == 1) and (GPIO.input(16) == 1) and (GPIO.input(12) == 1):
    print("Motion 2 , 3 , 4 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(85)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 85")
    print(" ")

```

```

if (GPIO.input(20) == 1) and (GPIO.input(16) == 1) and (GPIO.input(12) == 1):
    foto()
else:
    print("Motion NO Detected Again!")
elif (GPIO.input(16) == 1) and (GPIO.input(12) == 1) and (GPIO.input(7) == 1):
    print("Motion 3 , 4 , 5 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(130)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 130")
    print(" ")
    if (GPIO.input(16) == 1) and (GPIO.input(12) == 1) and (GPIO.input(7) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")
elif (GPIO.input(12) == 1) and (GPIO.input(7) == 1) and (GPIO.input(8) == 1):
    print("Motion 4 , 5 , 6 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(175)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 175")

```

```

print(" ")
if (GPIO.input(12) == 1) and (GPIO.input(7) == 1) and (GPIO.input(8) == 1):
    foto()
else:
    print("Motion NO Detected Again!")
elif (GPIO.input(7) == 1) and (GPIO.input(8) == 1) and (GPIO.input(25) == 1):
    print("Motion 5 , 6 , 7 Detected!")
    print(" ")
    DC=1./18.*(45)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 45")
    DC=1./18.*(20)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 20")
    print(" ")
    if (GPIO.input(7) == 1) and (GPIO.input(8) == 1) and (GPIO.input(25) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")
elif (GPIO.input(8) == 1) and (GPIO.input(25) == 1) and (GPIO.input(24) == 1):
    print("Motion 6 , 7 , 8 Detected!")
    print(" ")
    DC=1./18.*(90)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 90")
    DC=1./18.*(20)+2
    pwm2.ChangeDutyCycle(DC)

```



```

print("B . 20")

print(" ")

if (GPIO.input(8) == 1) and (GPIO.input(25) == 1) and (GPIO.input(24) == 1):

    foto()

else:

    print("Motion NO Detected Again!")

elif (GPIO.input(25) == 1) and (GPIO.input(24) == 1) and (GPIO.input(21) == 1):

    print("Motion 7 , 8 , 1 Detected!")

    print(" ")

    DC=1./18.*(135)+2

    pwm1.ChangeDutyCycle(DC)

    print("A . 135")

    DC=1./18.*(20)+2

    pwm2.ChangeDutyCycle(DC)

    print("B . 20")

    print(" ")

if (GPIO.input(25) == 1) and (GPIO.input(24) == 1) and (GPIO.input(21) == 1):

    foto()

else:

    print("Motion NO Detected Again!")

elif (GPIO.input(21) == 1) and (GPIO.input(20) == 1):

    print("Motion 1 & 2 Detected!")

    print(" ")

    DC=1./18.*(180)+2

    pwm1.ChangeDutyCycle(DC)

    print("A . 180")

```

```

DC=1./18.*(20)+2
pwm1.ChangeDutyCycle(DC)
print("B . 20")
print(" ")
if ((GPIO.input(21) == 1) and (GPIO.input(20) == 1)):
    foto()
else:
    print("Motion NO Detected!")
elif (GPIO.input(20) == 1) and (GPIO.input(16) == 1):
    print("Motion 2 & 3 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(63)+2
    pwm1.ChangeDutyCycle(DC)
    print("B . 63")
    print(" ")
    if ((GPIO.input(20) == 1) and (GPIO.input(16) == 1)):
        foto()
    else:
        print("Motion NO Detected!")
elif (GPIO.input(16) == 1) and (GPIO.input(12) == 1):
    print("Motion 3 & 4 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)

```

```

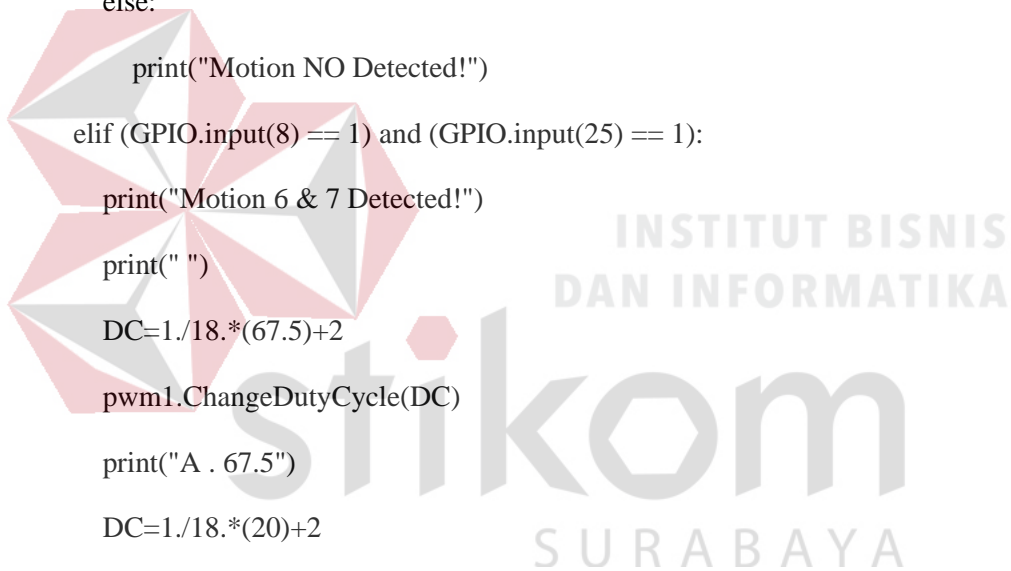
print("A . 180")
DC=1./18.*(110)+2
pwm1.ChangeDutyCycle(DC)
print("B . 110")
print(" ")
if ((GPIO.input(16) == 1) and (GPIO.input(12) == 1)):
    foto()
else:
    print("Motion NO Detected!")
elif (GPIO.input(12) == 1) and (GPIO.input(7) == 1):
    print("Motion 4 & 5 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(155)+2
    pwm1.ChangeDutyCycle(DC)
    print("B . 155")
    print(" ")
    if ((GPIO.input(12) == 1) and (GPIO.input(7) == 1)):
        foto()
    else:
        print("Motion NO Detected!")
elif (GPIO.input(7) == 1) and (GPIO.input(8) == 1):
    print("Motion 5 & 6 Detected!")
    print(" ")

```

```

DC=1./18.*(22.5)+2
pwm1.ChangeDutyCycle(DC)
print("A . 22.5")
DC=1./18.*(20)+2
pwm1.ChangeDutyCycle(DC)
print("B . 20")
print(" ")
if ((GPIO.input(7) == 1) and (GPIO.input(8) == 1)):
    foto()
else:
    print("Motion NO Detected!")
elif (GPIO.input(8) == 1) and (GPIO.input(25) == 1):
    print("Motion 6 & 7 Detected!")
    print(" ")
    DC=1./18.*(67.5)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 67.5")
    DC=1./18.*(20)+2
    pwm1.ChangeDutyCycle(DC)
    print("B . 20")
    print(" ")
if ((GPIO.input(8) == 1) and (GPIO.input(25) == 1)):
    foto()
else:
    print("Motion NO Detected!")
elif (GPIO.input(25) == 1) and (GPIO.input(24) == 1):
    print("Motion 7 & 8 Detected!")

```

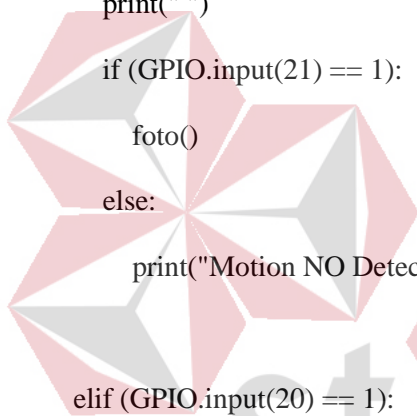


```

print(" ")
DC=1./18.*(112.5)+2
pwm1.ChangeDutyCycle(DC)
print("A . 112.5")
DC=1./18.*(20)+2
pwm1.ChangeDutyCycle(DC)
print("B . 20")
print(" ")
if ((GPIO.input(25) == 1) and (GPIO.input(24) == 1)):
    foto()
else:
    print("Motion NO Detected!")
elif (GPIO.input(24) == 1) and (GPIO.input(21) == 1):
    print("Motion 8 & 1 Detected!")
    print(" ")
    DC=1./18.*(157.5)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 157.5")
    DC=1./18.*(20)+2
    pwm1.ChangeDutyCycle(DC)
    print("B . 20")
    print(" ")
if ((GPIO.input(24) == 1) and (GPIO.input(21) == 1)):
    foto()
else:
    print("Motion NO Detected!")

```

```
elif (GPIO.input(21) == 1):  
    print("Motion 1 Detected!")  
    print(" ")  
    DC=1./18.*(180)+2  
    pwm1.ChangeDutyCycle(DC)  
    print("A . 180")  
    DC=1./18.*(6)+2  
    pwm2.ChangeDutyCycle(DC)  
    print("B . 6")  
    print(" ")  
    if (GPIO.input(21) == 1):  
        foto()  
    else:  
        print("Motion NO Detected Again!")  
elif (GPIO.input(20) == 1):  
    print("Motion 2 Detected!")  
    print(" ")  
    DC=1./18.*(180)+2  
    pwm1.ChangeDutyCycle(DC)  
    print("A . 180")  
    DC=1./18.*(43)+2  
    pwm2.ChangeDutyCycle(DC)  
    print("B . 43")  
    print(" ")  
    if (GPIO.input(20) == 1):  
        foto()
```



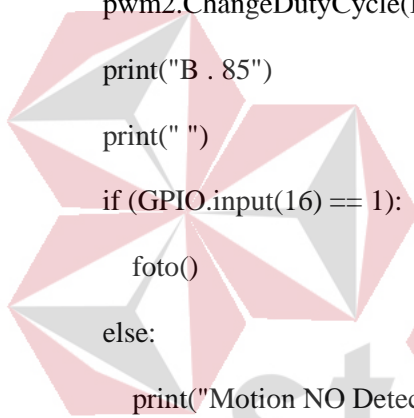
```

else:
    print("Motion NO Detected Again!")

elif (GPIO.input(16) == 1):
    print("Motion 3 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(85)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 85")
    print(" ")
    if (GPIO.input(16) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")

elif (GPIO.input(12) == 1):
    print("Motion 4 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(130)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 130")
    print(" ")
    if (GPIO.input(12) == 1):

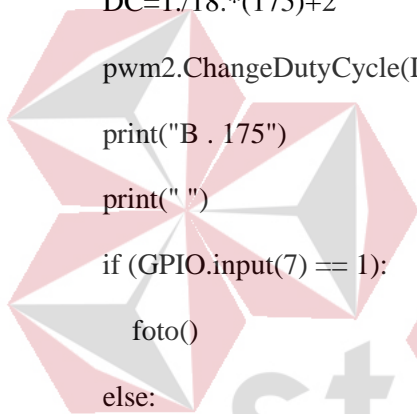
```



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

```
        foto()
    else:
        print("Motion NO Detected Again!")
elif (GPIO.input(7) == 1):
    print("Motion 5 Detected!")
    print(" ")
    DC=1./18.*(180)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 180")
    DC=1./18.*(175)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 175")
    print(" ")
    if (GPIO.input(7) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")
elif (GPIO.input(8) == 1):
    print("Motion 6 Detected!")
    print(" ")
    DC=1./18.*(45)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 45")
    DC=1./18.*(20)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 20")
    print(" ")
```



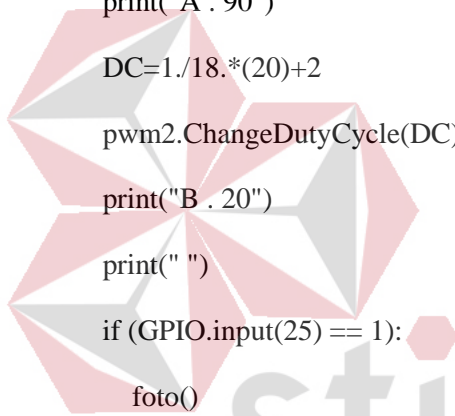
INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA


```
if (GPIO.input(8) == 1):
    foto()
else:
    print("Motion NO Detected Again!")

elif (GPIO.input(25) == 1):
    print("Motion 7 Detected!")
    print(" ")
    DC=1./18.*(90)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 90")
    DC=1./18.*(20)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 20")
    print(" ")
    if (GPIO.input(25) == 1):
        foto()
    else:
        print("Motion NO Detected Again!")

elif (GPIO.input(24) == 1):
    print("Motion 8 Detected!")
    print(" ")
    DC=1./18.*(135)+2
    pwm1.ChangeDutyCycle(DC)
    print("A . 135")
    DC=1./18.*(20)+2
    pwm2.ChangeDutyCycle(DC)
    print("B . 20")
```



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

```
print(" ")  
if (GPIO.input(24) == 1):  
    foto()  
else:  
    print("Motion NO Detected Again!")  
else:  
    print("Motion NO Detected!")  
    pwm1.ChangeDutyCycle(0)  
except KeyboardInterrupt:  
    print("Quit")  
    GPIO.cleanup()
```



BIODATA



Nama : Septianto Raka Putra Pratama
Tempat Lahir : Jember
Tanggal Lahir : 20 September 1994
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Puri Lidah Kulon No. 17
Surabaya

RIWAYAT PENDIDIKAN

2010 – 2013 : SMK Penerbangan Juada, Sidoarjo
2007 – 2010 : SMP Negeri 2 Jenggawah, Jember
2001 – 2007 : SD Muhammadiyah 1 Jember

PERGURUAN TINGGI

Program Studi : Sistem Komputer
Fakultas : Teknologi dan Informatika
Nama PT : Insitut Bisnis dan Informatika
Stikom Surabaya