IMPLEMENTASI *RASPBERRY PI* UNTUK RANCANG BANGUN SISTEM KEAMANAN PINTU RUANG *SERVER* DENGAN PENGENALAN WAJAH MENGGUNAKAN METODE *TRIANGLE FACE*
(STUDI KASUS RUANG *SERVER* PPSDM MIGAS CEPU)

SKRIPSI

Oleh:

MULA AGUNG BARATA NIM. 1441183002

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

JULI 2017

IMPLEMENTASI *RASPBERRY PI* UNTUK RANCANG BANGUN SISTEM KEAMANAN PINTU RUANG *SERVER* DENGAN PENGENALAN WAJAH MENGGUNAKAN METODE *TRIANGLE FACE*
(STUDI KASUS RUANG *SERVER* PPSDM MIGAS CEPU)

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

MULA AGUNG BARATA NIM. 1441183002

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

JULI 2017

HALAMAN PERSETUJUAN

IMPLEMENTASI *RASPBERRY PI* UNTUK RANCANG BANGUN SISTEM KEAMANAN PINTU RUANG *SERVER* DENGAN PENGENALAN WAJAH MENGGUNAKAN METODE *TRIANGLE FACE* (STUDI KASUS RUANG *SERVER* PPSDM MIGAS CEPU)

Disusun oleh:

MULA AGUNG BARATA NIM. 1441183002

Laporan Akhir ini telah diuji pada tanggal 17 Juli 2017

Disetujui oleh:

1.	Penguji I	:	Dr. Eng. Faisal Rahutomo, ST., M. Kom. NIP. 19771116 200501 1 008	.....
2.	Penguji II	:	Putra Prima Arhandi, ST., M. Kom. NIP. 19861103 201404 1 001	.....
3.	Pembimbing I	:	Indra Dharma Wijaya, ST., M.MT. NIP. 19730510 200801 1 010	.....
4.	Pembimbing II	:	Usman Nurhasan, S. Kom., MT. NIP. 19860923 201504 1 001	.....

Mengetahui,

Ketua Jurusan Teknologi Informasi	Ketua Program Studi Manajemen Informatika
Rudy Ariyanto, S.T., M.Cs. NIP. 19711110 199903 1 002	Ir. Deddy Kusbianto P., M.MKom. NIP. 19621128 198811 1 001

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, Juli 2017

Mula Agung Barata

ABSTRAK

Barata, Mula Agung. "Implementasi *Raspberry Pi* Untuk Rancang Bangun Sistem Keamanan Pintu Ruang *Server* Dengan Pengenalan Wajah Menggunakan Metode *Triangle Face* (Studi Kasus Ruang *Server* PPSDM Migas Cepu)". **Pembimbing: (1) Indra Dharma Wijaya, ST., M.MT., (2) Usman Nurhasan, S. Kom., MT**

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Teknologi yang semakin berkembang telah mengubah pola pikir manusia dengan berbagai ide dan gagasan baru untuk memunculkan alat baru yang dapat membantu manusia dalam berbagai hal di kehidupan sehari-hari. Salah satu alat yang dibutuhkan manusia yaitu sistem keamanan ruang *server* yang sederhana dan berteknologi serta mudah untuk diaplikasikan dengan memperhatikan biaya pembuatan dan perawatannya.

Minicomputer raspberry pi merupakan sebuah alat yang praktis dalam segi dimensi dan memiliki fungsi yang kompleks untuk berbagai kebutuhan fungsi yang akan digunakan oleh manusia sebagai *micro controller, server* sampai dengan pengolahan citra *digital*. Penelitian dilakukan bertujuan untuk membantu memenuhi kebutuhan sistem keamanan ruang *server* yang mudah untuk diaplikasikan dan terjangkau dalam segi biaya pembuatan dan perawatan serta berteknologi, mengingat pentingnya keamanan data dan informasi yang tersimpan dalam *server* sehingga perlu pengamanan dalam mengakses ruang *server* pada suatu perusahaan.

Dengan memanfaatkan *minicomputer raspberry pi* sebagai pemroses dan *USB webcam* sebagai alat pendeteksi wajah yang kemudian akan diproses oleh *raspberry pi* dengan menggunakan *OpenCV* untuk menentukan wajah manusia, lalu wajah tersebut akan masuk pada proses pengenalan wajah dengan metode *triangle face* yang memanfaatkan perhitungan jarak antar fitur wajah seperti mata, hidung, dan mulut. Setelah wajah dikenali maka *raspberry pi* akan melakukan perintah pada *servo motor* untuk membuka pintu ruang agar dapat diakses oleh admin *server* pada suatu perusahaan. Berdasarkan pengujian sistem yang telah dilakukan, ternyata sistem pengenalan wajah menggunakan metode *Triangle face* ini memiliki tingkat keakuratan 75%, kesalahan positif 25% dan kesalahan negatif 0% sehingga dapat disimpulkan bahwa sistem ini cukup aman untuk diaplikasikan dalam sistem keamanan pintu ruang *server*.

Kata kunci: *Raspberry Pi*, Pengenalan Wajah *OpenCV*, *Triangle Face*, Pengolahan Citra

ABSTRACT

Barata, Mula Agung. *"Raspberry Pi Implementation for the Design of Server Room Door Security System with Facial Recognition Using Triangle Face Method (PPSDM Migas Cepu Server Room Case Study)".* ***Advisors: (1) Indra Dharma Wijaya, ST., M.MT., (2) Usman Nurhasan, S. Kom., MT.***

Thesis, Informatics Engineering Study Programme, Department of Information Technology, State Polytechnic of Malang, 2017.

The development of technology has changed the mindset of people providing them with new ideas to build tools that can assist them in everyday life. One of the tools that people need is a simple and technologically advanced server room door security system that is easy to apply by considering the manufacturing and maintenance costs.

The raspberry pi minicomputer is a practical tool in terms of dimensions and has a complex function for various functional needs that will be used by humans from microcontroller, server, to digital image processing. The research is aimed to meet the need of server room door security system that is easy to apply and affordable in terms of manufacturing and maintenance costs as well as technologically advanced, given the importance of data security and information stored in the server therefore security in accessing server room in a company is urgently required.

By utilizing raspberry pi minicomputer as a processor and USB webcam as a face detection tool which then will be processed by raspberry pi by using OpenCV to determine human face, then the recognized face will enter the facial recognition process with triangle face method which utilize calculation of distance between face features like eyes, nose, and mouth. Once the face is recognized then raspberry pi will perform the command on the servo motor to open the door room to be accessed by the server admin on a company. Based on the system testing that has been done, it shows that facial recognition system using Triangle face method has a level of accuracy of 75%, 25% positive error and 0% negative error and therefore it can be concluded that the system is safe enough to be applied in the server room door security system.

Keywords: Raspberry Pi, OpenCV Facial Recognition, Triangle Face, Image Processing.

KATA PENGANTAR

Puji Syukur selalu dihaturkan kehadirat Allah AWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan Skripsi dengan judul "Implementasi *Raspberry Pi* Untuk Rancang Bangun Sistem Keamanan Pintu Ruang *Server* Dengan Pengenalan Wajah Menggunakan Metode *Triangle Face* (Studi Kasus Ruang *Server* PPSDM Migas Cepu)".

Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku Ketua Jurusan Teknologi Informasi
2. Bapak Ir. Deddy Kusbianto P., M.MKom. selaku Ketua Program Studi Teknik Informatika.
3. Bapak Indra Dharma Wijaya, ST., M.MT. dan Usman Nurhasan, S. Kom., MT. selaku dosen pembimbing skripsi yang selalu membimbing dan memotivasi selama penyusunan skripsi hingga selesai.
4. Orang tua dan keluarga tercinta yang selalu mendoakan dan mendukung dari segi finansial maupun motivasi.
5. Sahabat terbaik yang selalu membantu dan mendoakan dalam segala kesulitan selama penyusunan skripsi dan pembuatan projeknya.
6. Teman-teman seangkatan yang selalu membantu dan memotivasi penulis dalam menyusun skripsi.
7. Dan seluruh pihak yang telah membantu dan mendukung lancarnya penyusunan Skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan Skripsi ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis

mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan Skripsi ini. Semoga laporan Skripsi ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, Juli 2017

Penulis

DAFTAR ISI

SAMPUL DEPAN i
HALAMAN JUDUL ii
HALAMAN PERSETUJUAN iii
HALAMAN PERNYATAAN iv
ABSTRAK v
ABSTRACT vi
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xii
DAFTAR TABEL xv
DAFTAR LAMPIRAN xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Batasan Masalah 3
1.5 Sistematika Penulisan 3
BAB II LANDASAN TEORI 5
2.1 Pengolahan Citra 5
2.2 OpenCV 5
2.3 Bahasa Pemograman Python 6
2.4 Minicomputer Raspberry Pi 7
2.5 Sistem Operasi Raspbian 8
2.6 Webcam 8
2.7 FSWebcam 9
BAB III METODOLOGI PENELITIAN 10
3.1 Studi Literatur 11
3.1.1 Observasi 11
3.1.2 Studi Pustaka 11
3.2 Analisa Kebutuhan 12
3.3 Desain Sistem dan Perancangan 13
3.4 Implementasi 13
3.5 Pengujian 14
3.5.1 Pendeteksian Wajah 14
3.5.2 Pemrosesan Wajah 16
3.6 Metode Triangle Face 16
3.6.1 Tahapan Segmentasi Warna Kulit 16
3.6.2 Tahapan Lokalisasi Wajah 17
3.6.3 Tahapan Pencarian Fitur-Fitur Wajah 18
3.6.4 Pengukuran Jarak Antar Fitur Wajah 20

BAB IV ANALISA DAN PERANCANGAN 22
4.1 Analisa Sistem 22
4.2 Gambaran Umum 22
4.3 Blok Diagram Sistem 23
4.4 Use Case Sistem 24
4.5 Skema Konfigurasi Sistem 24
4.6 Perancangan Database 25
4.7 Flowchart 27
4.8 Analisa Kebutuhan Perangkat Keras 30
4.8.1 Minicomputer raspberry pi 30
4.8.2 Acrylic Case Raspberry Pi 3 Model B + Pi Fan 31
4.8.3 USB Adaptor 32
4.8.4 VGA to HDMI Converter 33
4.8.5 Heatsink Raspberry Pi 33
4.8.6 MicroSD Card + Adapter 33
4.8.7 USB Webcam 34
4.8.8 Kabel UTP 35
4.8.9 Servo Motor 36
4.8.10 Rainbow Cable 36
4.8.11 GPIO Adapter Expansion Board 40 PIN 37
4.8.12 GPIO Bread Board 38
4.8.13 Kabel Jumper 38
4.8.14 Kotak Simulasi 39
4.8.15 Mur dan Baut 39
4.8.16 Engsel Pintu Acrylic 40
4.8.17 Kabel Ties 40
4.9 Analisa Kebutuhan Perangkat Lunak 41
4.9.1 Sistem Operasi Raspbian Jessie 41
4.9.2 SD Formatter 41
4.9.3 Win32 Disk Imager 42
4.9.4 PuTTY 42
4.9.5 VNCServer 42
4.9.6 Remote Desktop Connection 42
4.9.7 FSWebcam 43
4.9.8 OpenCV 43
4.9.9 Web Server 43
4.10 Perancangan Kotak Simulasi 43
4.11 Perancangan dan Perakitan Penyangga Dagu 46
4.12 Perancangan dan Perakitan Tripod Kamera 47
BAB V IMPLEMENTASI 48
5.1 Instalasi Perangkat Lunak 48
5.1.1 Instalasi Raspbian Jessie 48

5.1.2 Konfigurasi Raspberry pi 51
5.1.3 Instalasi OpenCV 57
5.1.4 Instalasi FSWebcam 62
5.1.5 Instalasi Web Server 64
5.2 Instalasi Perangkat Keras 65
5.2.1 Instalasi Raspberry pi pada Case 65
5.2.2 Instalasi USB Webcam 66
5.2.3 Instalasi Rainbow Cable 66
5.2.4 Instalasi GPIO Adapter Expansion Board 67
5.2.5 Instalasi Seluruh Perngkat ke Kotak Simulasi 68
5.2.6 Instalasi Semua Perangkat ke Dalam Kotak Simulasi 69
5.2.7 Instalasi Servo Motor 69
5.3 Perakitan Kotak Simulasi Ruang Server 70
5.4 Implementasi Perangkat Keras 71
5.5 Implementasi Perangkat Lunak 71
5.6 Pencocokan Hasil Pengukuran Nilai Citra Wajah dengan Database 72
BAB VI PENGUJIAN DAN PEMBAHASAN 74
6.1 Pengujian Fungsional 74
6.2 Pengujian Pendeteksian Fitur Wajah 74
6.3 Pengujian Pengukuran Jarak Antar Fitur Wajah 76
6.4 Pengujian Aplikasi Pengenalan Wajah 79
6.5 Variasi Pengujian Aplikasi Pengenalan Wajah 81
6.5.1 Pengujian dengan Variasi Jarak Pengambilan Citra Wajah 81
6.5.2 Pengujian Aplikasi dengan Variasi Nilai Thresholding 85
BAB VII PENUTUP 88
7.1 Kesimpulan 88
7.2 Saran 89
DAFTAR PUSTAKA 90
LAMPIRAN 91
LAMPIRAN 91

DAFTAR GAMBAR

Gambar 2.1 Minicomputer raspberry pi 7
Gambar 2.2 Web camera 8
Gambar 3.1 Tahapan penelitian 10
Gambar 3.2 Algoritma pendeteksian wajah 15
Gambar 3.3 Tahapan pada metode triangle face 16
Gambar 4.1 Blok diagram sistem 23
Gambar 4.2 Use case 24
Gambar 4.3 Skema konfigurasi sistem 25
Gambar 4.4 Flowchart register user 27
Gambar 4.5 Flowchart enter room 28
Gambar 4.6 Flowchart aplikasi 29
Gambar 4.7 Raspberry pi 3 model B 31
Gambar 4.8 Acrylic case raspberry pi 3 model B 32
Gambar 4.9 USB adaptor 32
Gambar 4.10 VGA to HDMI converter 33
Gambar 4.11 Heatsink pada raspberry pi 33
Gambar 4.12 Sandisk microSD card 34
Gambar 4.13 Logitech C170 USB webcam 35
Gambar 4.14 Kabel UTP 35
Gambar 4.15 Tower Pro SG90 servo motor 36
Gambar 4.16 Rainbow cable 37
Gambar 4.17 GPIO adapter expansion board 40 PIN 37
Gambar 4.18 GPIO bread board 38
Gambar 4.19 Kabel jumper 38
Gambar 4.20 Kotak simulasi 39
Gambar 4.21 Mur dan baut 39
Gambar 4.22 Engsel acrylic 40
Gambar 4.23 Kabel ties 40
Gambar 4.24 Desain sisi alas kotak simulasi 44
Gambar 4.25 Desain sisi kiri dan kanan kotak simulasi 44
Gambar 4.26 Desain sisa depan kotak simulasi 45
Gambar 4.27 Desain belakang kotak simulasi 45
Gambar 4.28 Perancangan penyangga dagu 46
Gambar 4.29 Hasil perakitan penyangga dagu 46
Gambar 4.30 Rancangan tripod kamera 47
Gambar 4.31 Hasil perakitan tripod kamera 47
Gambar 5.1 Formatting microSD card 49
Gambar 5.2 Write iso raspbian OS ke dalam microSD card 49
Gambar 5.3 Instalasi microSD card ke slot pada raspberry pi 50

Gambar 5.4 Pengecekan hasil konfigurasi perangkat keras 50
Gambar 5.5 Tampilan konfigurasi raspberry pi di terminal linux 51
Gambar 5.6 Menu interface options 52
Gambar 5.7 Raspberry pi configuration 52
Gambar 5.8 Pengaturan IP static pada laptop 53
Gambar 5.9 Pengaturan IP static pada raspberry pi 53
Gambar 5.10 Konfigurasi PuTTY 54
Gambar 5.11 Instalasi tightvncserver 54
Gambar 5.12 Instalasi xrdp 54
Gambar 5.13 Require password VNC 55
Gambar 5.14 Setting PuTTY untuk aplikasi VNCserver 55
Gambar 5.15 Login pada VNC Server 56
Gambar 5.16 Login pada Remote Desktop Connection 56
Gambar 5.17 Desktop sistem operasi raspbian 57
Gambar 5.18 Update Sistem 57
Gambar 5.19 Upgrade sistem 57
Gambar 5.20 Instalasi tools pengembang 58
Gambar 5.21 Instalasi image I/O packages 58
Gambar 5.22 Instalasi video I/O packages 58
Gambar 5.23 Install GTK development library 58
Gambar 5.24 Install extra dependensi 59
Gambar 5.25 Install python 2.7 dan python 3 file header 59
Gambar 5.26 Mengunduh arsip opencv-3.1.0 dan opencv_contrib 59
Gambar 5.27 Instalasi pip 59
Gambar 5.28 Instalasi virtualenv dan virtualenvwrapper 60
Gambar 5.29 Mengedit profiles 60
Gambar 5.30 Log on profiles 60
Gambar 5.31 Membuat virtual python environment 60
Gambar 5.32 Instalasi numpy 61
Gambar 5.33 Memulai instalasi OpenCV 61
Gambar 5.34 Build menggunakan CMake 61
Gambar 5.35 Kompilasi OpenCV 61
Gambar 5.36 Langkah akhir instalasi OpenCV 61
Gambar 5.37 Mengimpor python + OpenCV binding 62
Gambar 5.38 Update sistem 62
Gambar 5.39 Upgrade sistem 63
Gambar 5.40 Mendeteksi kamera terhubung sistem 63
Gambar 5.41 Instalasi fswebcam 63
Gambar 5.42 Update Sistem 64
Gambar 5.43 Instalasi apache2 64
Gambar 5.44 Konfigurasi apache2 64
Gambar 5.45 Lihat daftar aplikasi firewall pada apache 64

Gambar 5.46 Lihat apache full 64
Gambar 5.47 Instalasi mysql-server 64
Gambar 5.48 Instalasi PHP 64
Gambar 5.49 Instalasi raspberry pi pada acrylic case..... 65
Gambar 5.50 Instalasi USB camera..... 66
Gambar 5.51 Instalasi Rainbow Cable..... 67
Gambar 5.52 Instalasi GPIO Adapter Expansion Board 67
Gambar 5.53 Instalasi Rainbow Cable pada Adapter Expansion..... 68
Gambar 5.54 Instalasi Semua Perangkat ke Papan Komponen..... 68
Gambar 5.55 Instalasi Seluruh Perangkat ke Kotak Simulasi..... 69
Gambar 5.56 Instalasi Servo Motor..... 70
Gambar 5.57 Hasil perakitan kotak simulasi 70
Gambar 5.58 Implementasi perangkat keras 71
Gambar 5.59 Pendeteksian dan pengukuran wajah 72
Gambar 5.60 Pencocokan Kemiripan Data Training dan Testing..... 73
Gambar 6.1 Pendeteksian Fitur Wajah 75
Gambar 6.2 Hasil Pengukuran Jarak Antar Fitur Wajah 76

DAFTAR TABEL

Tabel 4.1 Tabel admin 26
Tabel 4.2 Tabel login_face 26
Tabel 6.1 Pengujian Fungsional 74
Tabel 6.2 Hasil Pengukuran Wajah Yusron Hamidi 77
Tabel 6.3 Hasil Pengukuran Wajah Mula Agung Barata 78
Tabel 6.4 Hasil Pengukuran Wajah Ryan Indra 78
Tabel 6.5 Hasil Pengukuran Afnan Afif 79
Tabel 6.6 Hasil Pengujian Aplikasi 80
Tabel 6.7 Hasil pengujian dengan jarak pengambilan citra wajah 30 cm 82
Tabel 6.8 Hasil pengujian dengan jarak pengambilan citra wajah 50 cm 83
Tabel 6.9 Hasil pengujian dengan jarak pengambilan citra wajah 60 cm 84
Tabel 6.10 Hasil pengujian aplikasi dengan nilai thresholding 190 pixel 85
Tabel 6.11 Hasil pengujian aplikasi dengan nilai thresholding 220 pixel 86
Tabel 6.12 Hasil pengujian aplikasi dengan nilai thresholding 240 pixel 87

DAFTAR LAMPIRAN

Lampiran 1. Biodata Penulis 92
Lampiran 2. Surat Permohonan Ijin Obeservasi..... 93
Lampiran 3. Source code penyimpanan hasil pengukuran citra wajah..... 94
Lampiran 4. Source code pencocokan data training dan data testing. 99
Lampiran 5. Konfigurasi Perangkat Keras..... 100
Lampiran 6. Activity Control Bimbingan 101
Lampiran 7. Lembar Persetujuan Mengikuti Ujian Skripsi 104
Lampiran 8. Form Revisi Skripsi 105

BAB I PENDAHULUAN

1.1 Latar Belakang

Ruang *server* merupakan salah satu tempat penyimpanan asset bagi sebuah perusahaan atau kantor besar pada umumnya berupa data-data yang bernilai tinggi bagi perusahaan, pada penelitin ini penulis mengambil studi kasus pada ruang *server* PPSDM Migas Cepu, karena di dalam ruangan ini terdapat aplikasi dan data kantor yang semakin hari akan semakin bernilai bagi instansi yang bersifat vital dan rahasia, oleh karena itu ruangan ini harus selalu dalam kondisi yang baik. Sebuah ruang *server* harus memiliki standar mulai suhu udara, kelembaban, kebakaran dan akses masuk dari orang-orang yang tidak berkepentingan. Jadi, tidak semua karyawan memiliki hak akses penuh apalagi dapat dengan mudah keluar masuk ruangan ini.

Biometrik merupakan suatu teknologi yang memanfaatkan ciri khas dari tubuh manusia sebagai pembeda antara 1 orang dengan yang lainnya, misal sidik jari sebagai teknologi *finger print*, sidik mata, suara sebagai teknologi *voice detection* dan wajah sebagai teknologi *face detection* [1]. Semua teknologi tersebut di atas sudah banyak dikembangkan dalam berbagai aplikasi seperti absensi dan sistem keamanan. Dalam penelitian ini penulis menggunakan wajah sebagai teknologi *face detection* yang akan diimplementasikan pada sistem keamanan ruang *server* PPSDM Migas Cepu. Alasan yang melatar belakangi peneliti dalam menggunakan wajah sebagai objek yang akan diidentifikasi sistem karena wajah merupakan bagian tubuh yang dimiliki setiap manusia yang sulit untuk dimanipulasi atau diduplikasi. Namun, dalam penelitian ini memerlukan sistem yang dapat mengenali dan mengidentifikasi wajah setiap orang yang sebelumnya sudah dilakukan perekaman sampel gambar dari masing-masing orang yang memiliki hak akses penuh terhadap ruang *server* yang sudah lebih dulu tersimpan dalam *database*. Sistem mengharuskan seseorang untuk tidak bergerak selama wajahnya dalam proses identifikasi agar akurat dalam pembacaannya.

Dalam penelitian ini, peneliti menggunakan metode *triangle face* dalam mengidentifkasi atau melakukan klasifikasi wajah dengan *webcam* lalu menyimpannya ke dalam *database* dan menggunakan *raspberry pi* yang merupakan salah satu *single board computer* sebagai pemrosesnya. Metode *triangle face* mengidentifikasi wajah seseorang dengan mendeteksi jarak-jarak antara mata kanan dengan mata kiri, mata kanan dengan mulut, mata kiri dengan mulut, mata kanan dengan hidung, mata kanan dengan hidung [1].

Berdasarkan uraian diatas, penulis mengajukan penelitian dengan judul "IMPLEMENTASI *RASPBERRY PI* UNTUK RANCANG BANGUN SISTEM KEAMANAN PINTU RUANG *SERVER* DENGAN PENGENALAN WAJAH MENGGUNAKAN METODE *TRIANGLE FACE"*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem yang dapat mengenali wajah yang disimulasikan sebagai sistem keamanan pintu ruang *server* PPSDM Migas Cepu?
2. Bagaimana mengimplementasikan sistem pengenalan wajah menggunakan metode *triangle face* pada teknologi *face recognition*?
3. Bagaimana merancang perangkat keamanan pintu *server* pengenalan wajah yang diintegrasikan dengan *webcam* sebagai pendeteksi wajah dan *raspberry pi* sebagai pemrosesnya?

1.3 Tujuan

Tujuan yang ingin dicapai oleh penulis dari peneitian ini adalah merancang perangkat *raspberry pi* untuk keamanan ruang *server* dengan pengenalan wajah menggunakan metode *triangle face* yang disimulasikan dengan beberapa perangkat keras pendukung.

1.4 Batasan Masalah

Perancangan dan simulasi dari *raspberry pi* untuk rancang bangun sistem keamanan pintu ruang *server* dengan pengenalan wajah menggunakan metode *triangle face* memiliki batasan masalah sebagai berikut:

1. Tidak membahas variasi latar belakang gambar yang diambil kamera, variasi pengambilan citra wajah, aksesoris wajah, pola wajah dan manipulasi wajah.
2. Tidak membahas secara rinci mengenai perangkat keras yang digunakan dalam penelitian terutama perangkat keras pendukung simulasi pintu otomatis yang dibuat.
4. Memproses *image* wajah yang memiliki ukuran 320 x 240 dengan format *JPEG*.
5. Sistem hanya mendeteksi satu wajah pada 1 *frame* dalam proses pendeteksian pada jarak 40 cm.

1.5 Sistematika Penulisan

Secara garis besar materi skripsi ini terbagi dalam beberapa bab yang tersusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisikan uraian yang memuat tentang segala hal yang melatar belakangi dilakukannya penelitian ini yang terdiri dari latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan naskah skripsi.

BAB II LANDASAN TEORI

Bab ini menguraikan tentang teori yang menjadi referensi dalam melaksanakan penelitian mengenai implementasi *raspberry pi* untuk rancang bangun sistem keamanan pintu ruang *server*.

BAB III METODOLOGI PENELITIAN

Pada bab ini berisi mengenai metode atau tahapan penelitian yang ditempuh oleh penulis yang meliputi metode pengambilan data dan pengembangan sistem.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisa dan perancangan sistem keamanan ruang *server* yang meliputi *flowchart, block digram, use case* dan skema konfigurasi sistem.

BAB V IMPLEMENTASI

Pada bab ini menjelaskan bentuk implementasi *raspberry pi* untuk rancang bangun sistem keamanan pintu ruang *server* yang meliputi spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk implementasi sistem yang dikembangkan dan bagaimana langkah dalam menjalankan sistem.

BAB VI PENGUJIAN DAN PEMBAHASAN

Bab ini berisikan mengenai analisa hasil uji coba dari implemantasi sistem yang dikembangkan dengan metode *triangle face.*

BAB VII PENUTUP

Bab ini dibagi menjadi dua sub bab, kesimpulan yang menjawab permasalahan yang dihadapi dan saran yang berisikan solusi alternatif untuk permasalahan yang terjadi pada penelitian ini.

DAFTAR PUSTAKA

Berisikan catatan semua sumber yang digunakan sebagai rujukan dalam penulisan laporan skripsi ini.

BAB II LANDASAN TEORI

2.1 Pengolahan Citra

Citra (*image*) merupakan salah satu komponen multimedia yang memegang peranan penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi, maksudnya sebuah gambar dapat memberikan informasi yang lebih banyak dari pada informasi tersebut disajikan dalam bentuk kata-kata. Citra adalah suat representasi, kemiripan, atau imitasi dari suatu objek. Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik [5]. Dengan istilah lain bahwa pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh *bit-bit* tertentu.

Citra ada 2 macam yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal *analog*, misalnya mata manusia dan kamera *analog*. Citra diskrit diasilkan melalui proses digitalisasi terhadap citra kontinu.

2.2 *OpenCV*

OpenCV adalah suatu *library* gratis yang dikembangkan oleh *developer-developer Intel Corporation*. Library ini terdiri dari fungsi-fungsi *computer vision* dan *API (Aplication Programming Interface)* untuk *image processing* yang *high level* maupun *low level* dan sebagai optimasi aplikasi *realtime. OpenCV* sangat disarankan untuk programmer yang akan berkutat pada bidang *computer vision*, karena librarynya mampu membuat aplikasi yang handal di bidang *digital vision* dan mempunyai fitur yang mirip dengan cara pengolahan visual pada manusia [2].

Berikut ini adalah beberapa fitur pada *library OpenCV*:

6. Manipulasi data gambar (alokasi memori, melepaskan memori, *setting* serta konversi gambar).
7. *Image*/*Video I/O* (Bisa menggunakan kamera yang sudah didukung oleh *library* ini).
8. Manipulasi matriks dan vektor serta terdapat juga *routines linier algebra* (*products, solvers, eigenvalues, SVD*)
9. *Image* processing dasar (*filtering, edge detection*, pendeteksian tepi, sampling dan interpolasi, konversi warna, operasi morfologi, histograms, *image pyramids*)
10. Analisis struktural
11. Kalibrasi kamera
12. Pendeteksian gerak
13. Pengenalan objek (hewan, manusia, tumbuhan dll)
14. *Basic GUI* (*Display* gambar/*video, mouse/keyboard* kontrol, *scrollbar*)
15. *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* [1].

2.3 Bahasa Pemograman *Python*

Python adalah bahasa pemrograman model skrip (*scripting language)* yang berorientasi obyek. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi. *Python* merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan *source code*nya, *debugge*r dan *profiler*,

antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, *GUI* (antarmuka pengguna grafis), dan basis datanya [3].

Beberapa fitur yang dimiliki *python* adalah:

1. Memiliki kepustakaan yang luas dan telah disediakan modul-modul.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan *layout* kode sumber yang memudahkan pengecekan,
4. Pembacaan kembali dan penulisan ulang kode sumber.
5. Berorientasi obyek.
6. Dapat dibangun dengan bahasa *python* maupun *C/C++*.

2.4 *Minicomputer Raspberry Pi*

Raspberry pi adalah sebuah *SBC (single-board computer)* seukuran kartu kredit. *raspberry pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan *SOC* (*System-on-a-Chip) ARM* yang dikemas dan diintegrasikan di atas *PCB* (papan sirkuit). *Raspberry Pi* ini mampu bekerja layaknya komputer pada umumnya dengan kemampuan untuk menjalankan sistem operasi Linux dan aplikasinya seperti *LibreOffice*, multimedia *(audio* dan *video*), peramban web, ataupun *programming*. [2]

Gambar 2.1 *Minicomputer raspberry pi*

2.5 Sistem Operasi *Raspbian*

Merupakan sebuah sistem operasi berbasis Linux *distro Debian* yang dapat dioptimalkan untuk penggunaan komputer mini *raspberry pi*. Sistem operasi ini memiliki beberapa program standard dan beberapa program pembantu untuk dapat menjalankan perangkat keras dari komputer mini *raspberry pi* ini [2].

Dalam sistem operasi ini sudah lebih lengkap daripada sistem yang murni digunakan di komputer pada umumnya karena memiliki lebih dari 350.00 paket dan *library precompiled* yang tersaji dalam bentuk format yang mudah untuk diinstalisasi pada *raspberry pi*.

2.6 *Webcam*

Webcam (singkatan dari *web camera*) adalah sebutan bagi kamera *realtime* (bermakna keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui *World Wide Web*, program instant messaging, atau aplikasi *video call.* Sebuah web camera yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar, *casing (cover),* termasuk casing depan dan casing samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di casing depan yang berguna untuk memasukkan gambar, kabel *support*, yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector,* kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang *web camera.*

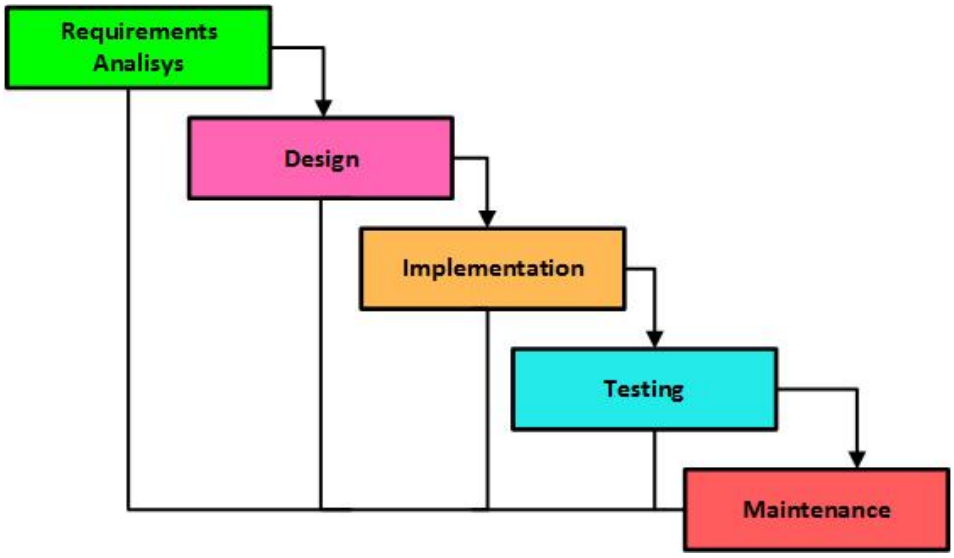
Gambar 2.2 *Web camera*

2.7 *FSWebcam*

FSWebcam adalah suatu aplikasi webcam atau modul kamera yang dapat melakukan ambil gambar melalui media tertentu seperti kamera atau *webcam.* Aplikasi ini dapat digunakan untuk mereduksi *noise* yang ditimbulkan oleh kamera saat melakukan perintah ambil gambar dan selain itu aplikasi ini dapat melakukan kompresi ukuran gambar yang dihasilkan berdasarkan format yang diinginkan seperti *PNG* atau *JPEG.* Hasil dari proses ambil gambar ini dapat diolah lagi sesuai dengan kebutuhan dan keinginan pengguna, seperti pengaturan resolusi gambar yang diinginkan, info gambar yang ingin ditampilkan pada hasil akhir tangkapan gambar pada kamera, selain itu *fswebcam* dapat melakukan perintah ambil gambar secara berurutan dan dapat disesuaikan waktu pengambilannya.

BAB III METODOLOGI PENELITIAN

Pada penelitian ini merupakan penelitian eksperimental yaitu penelitian dengan cara mencatat langsung hasil pengujian sebagai media pengumpulan data termasuk hasil dari analisa kebutuhan sistem. Selanjutnya membuat desain sistem yang akan dirancang, desain yang dirancang meliputi desain perancangan dan skema konfigurasi perangkat lunak maupun perangkat keras yang digunakan dalam perancangan sistem. Lalu mengimplementasikan sistem yang sudah dirancang. Dalam tahap implementasi ini, perancangan sistem yang sudah dibuat kemudian diterapkan sesuai desain yang sudah dibuat. Masing-masing perangkat baik perangkat lunak maupun perangkat keras dihubungkan agar dapat berintegrasi dan membentuk sistem yang sudah dirancang. Setelah tahap implementasi maka dilanjutkan dengan tahap pengujian sistem, bagaimana sistem berjalan dan memperbaiki jika terjadi kekurang akuratan dalam perancangan. Tahapan yang terakhir adalah *maintenance*, tahapan ini meliputi tahapan pemeliharaan sistem serta pengembangan sistem dalam penambahan fitur sesuai dengan kebutuhan *user*. Pada gambar 5.1 di bawah ini menunjukkan tahapan dalam penelitian.



Gambar 3.1 Tahapan penelitian

3.1 Studi Literatur

Pada tahap ini, peneliti mempelajari berbagai literatur sebagai sumber untuk penelitian melalui pengumpulan data, referensi dari berbagai buku dan sumber dari *internet* atau sumber lain yang diperlukan dalam merancang dan mengimplementasikan sistem yang dikembangkan dan berkaitan untuk penulisan skripsi ini.

3.1.1 Observasi

Observasi adalah metode pengumpulan data melalui pengamatan langsung atau peninjauan secara cermat dan langsung di lapangan atau lokasi penelitian. Dalam hal ini, peneliti dengan berpedoman kepada desain penelitiannya perlu mengunjungi lokasi penelitian untuk mengamati langsung berbagai hal atau kondisi yang ada di lapangan. Penemuan ilmu pengetahuan selalu dimulai dengan observasi dan kembali kepada observasi untuk membuktikan kebenaran ilmu pengetahuan tersebut.[6]

Dalam penelitian ini observasi yang dilakukan oleh peneliti adalah melihat fakta di lapangan untuk mengetahui permasalahan yang terjadi sehingga dapat menarik kesimpulan yang mengarah pada solusi yang bisa menyelesaikan permaslahan yang ada. Observasi ini juga merupakan langkah dalam pengumpulan data yang dapat diperoleh dari berbagai sumber baik secara langsung maupun tidak langsung.

3.1.2 Studi Pustaka

Studi pustaka adalah suatu pembahasan yang berdasarkan pada buku-buku referensi yang bertujuan untuk memperkuat materi pembahasan maupun sebagai dasar untuk menggunakan rumus-rumus tertentu dalam menganalisa dan mendesain suatu struktur. Studi pustaka digunakan untuk memecahkan masalah yang ada, baik untuk menganalisa faktor-faktor dan data pendukung maupun untuk merencanakan konstruksi, maka pada bagian ini kami menguraikan secara global pemakaian rumus-rumus dan persamaan yang akan digunakan untuk memecahkan masalah yang ada.

3.2 Analisa Kebutuhan

Pada tahap analisa kebutuhan ini merupakan tahapan perencanaan sebelum melakukan penelitian. Peneliti menganalisa segala kebutuhan yang akan digunakan selama proses penelitian meliputi data yang akan diolah, perangkat lunak dan perangkat keras yang akan digunakan.

Data yang digunakan untuk implementasi sistem pengenalan wajah untuk sistem keamanan pintu ruang *server* ini adalah citra digital *user*. Citra digital *user* yang digunakan yaitu citra standar yang didapatkan dari peng-*capture*-an menggunakan *usb webcam*. Pengujian citra digital tersebut menggunakan citra digital dengan format **JPG atau JPEG* dan resolusi sesuai dengan *webcam* yang digunkaan, setelah citra digital didapatkan selanjutnya akan diimplementasikan menggunakan metode *triangle face* untuk proses klasifikasinya.

Selain kebutuhan data, pada analisa kebutuhan ini juga mendefinisikan mengenai kebutuhan perangkat keras dan perangkat lunak yang digunakan selama penelitian antara lain:

1. Adapun perangkat keras utama yang dibutuhkan adalah sebagai berikut:
   - *Minicomputer raspberry pi*
   - *USB Adaptor*
   - *MicroSD card +adapter*
   - *USB Webcam*
   - Kabel *UTP*
   - *Servo motor*
   - Kabel *jumper*
   - Kotak simulasi
2. Adapun perangkat lunak yang dibutuhkan adalah sebagai berikut:
   - Sistem operasi *raspbian jessie*
   - *SD Formatter*
   - *Win32 Disk Imager*
   - *PuTTY*
   - *VNCServer*
   - *Remote Desktop Connection*

- *FSWebcam*
- *OpenCV*

3.3 Desain Sistem dan Perancangan

Tahap desain sistem dan perancangan merupakan tahapan perancangan kerangka kerja yang dibangun dengan mendefiniskan dan menggambarkan sejumlah kerangka akifitas kerja yang diaplikasikan ke semua proyek pengembangan sistem. Model proses yang digambarkan untuk pengembangan sistem dipilih berdasarkan jenis proyek, metode, alat bantu yang digunakan dan perangkat lunak pendukung yang dibutuhkan.

Adapun perancangan yang dibuat adalah sebagai berikut:

1. *Block* Diagram Sistem
2. *Use Case* Sistem
3. Skema Konfigurasi Sistem

3.4 Implementasi

Tahap implementasi ini merupakan tahapan yang menjelaskan mengenai langkah dalam perakitan alat atau perangkat keras yang digunakan, proses instalasi perangkat lunak yang digunakan, proses konfigurasi dan integrasi antara perangkat keras dan perangkat lunak sampai dengan penggunaan alat yang sudah dirancang.

Proses yang dilakukan dalam tahapan implementasi antara lain:

1. Menyiapkan perangkat keras dan perangkat lunak sesuai analisa kebutuhan
2. Instalasi sistem operasi *Raspbian Jessie* pada *raspberry pi*
3. Menghubungkan *raspberry pi* dengan *notebook* untuk melakukan *remote desktop* menggunakan *UTP cable* model *crossover*
4. Melakukan konfigurasi *raspberry pi* yang diperlukan
5. Instalasi *OpenCV* pada sistem operasi *raspbian Jessie*
6. Instalasi *USB webcam* ke *raspberry pi*
7. Menghubungkan *GPIO breadboard* ke *raspberry p*i menggunakan kabel *rainbow* beserta *adapter*nya

8. Menghubungkan *servo motor* dengan *raspberry pi* untuk bisa dikontrol melalui *GPIO breadboard*

3.5 Pengujian

Pada penelitian ini, pengujian sistem pengenalan wajah yang diimplementasikan pada sistem keamanan pintu ruang *server* menggunakan metode *triangle face* sebagai metode pendeteksiannya.

Metode *triangle face* adalah salah satu metode yang digunakan untuk mengenali wajah dalam suatu citra digital. Cara kerja metode ini yaitu dengan cara mendeteksi fitur-fitur wajah seseorang yang telah diinputkan, fitur-fitur wajah ini di jadikan parameter untuk dihitung jarak antar fiturnya yang membentuk segitiga yang sering disebut *triangle face*. Fitur-fitur wajah tersebut yaitu mata kana, mata kiri, hidung, mulut, tinggi wajah dan lebar wajah. Jarak yang dicari menggunakan fitur-fitur wajah adalah [1]:

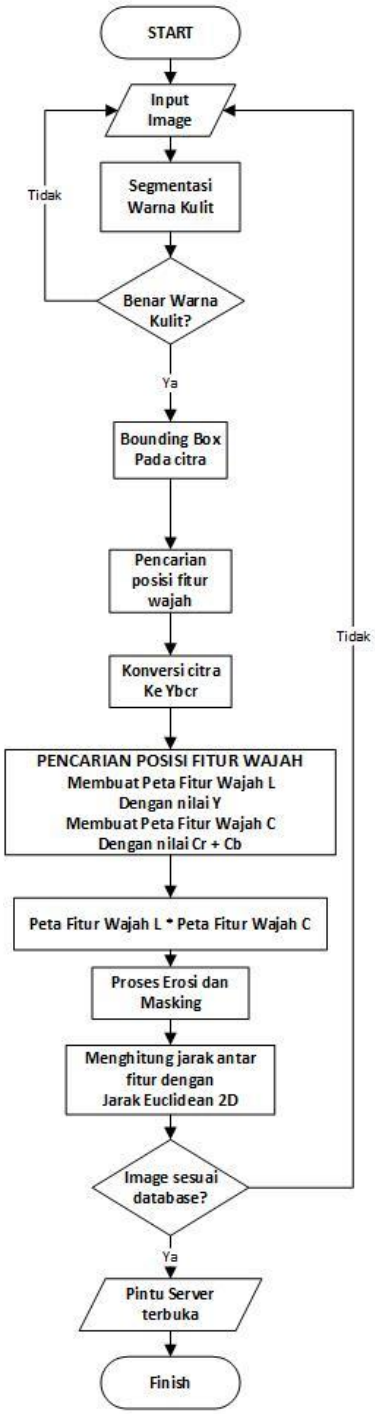
1. Jarak mata kanan-mulut
2. Jarak mata kiri-mulut
3. Jarak mata kanan-hidung
4. Jarak mata kiri-hidung
5. Jarak antar mata

Langkah dari proses implementas metode *triangle face* pada sistem keamana ruang *server* adalah sebagai berikut:

3.5.1 Pendeteksian Wajah

Tahap awal dalam pengenalan wajah adalah mendeteksi wajah yang diposisikan di depan kamera untuk mengambil citra wajah yang kemudian dideteksi apakah wajah yang terdeteksi merupakan wajah manusia atau bukan, lalu dicocokkan sebagai klasifikasi wajah yang terdeteksi dengan data wajah yang telah tersimpan dalam *database* citra yang telah disiapkan untuk menentukan mana wajah orang sebagai citra yang akan dikenali oleh sistem.

Adapun algoritma atau alur dalam pendeteksian wajah sebagai citra yang akan diolah dalam sistem yang dirancang sesuai dengan metode *triangle face* adalah sebagai berikut:



Gambar 3.2 Algoritma pendeteksian wajah

3.5.2 Pemrosesan Wajah

Setelah tahap pendeteksian wajah *(face detection)*, tahapan selanjutnya yaitu pemrosesan gambar yang telah dideteksi oleh kamera. Pemrosesan di sini menggunakan *minicomputer raspberry pi* dengan *library OpenCV* sebagai *tool*nya lalu menggunakan metode *triangle face.*

3.6 Metode *Triangle Face*

Dalam penelitian ini, citra masukan akan dikenali akan melalui beberapa proses dalam metode *triangle face* yang dilakukan dalam empat tahap, antara lain:

1. Tahap segmentasi warna kulit
2. Tahap lokalisasi wajah
3. Tahap pencarian fitur-fitur wajah
4. Tahap perhitungan pengukuran jarak antar fitur wajah

Gambar 3.3 Tahapan pada metode *triangle face*

3.6.1 Tahapan Segmentasi Warna Kulit

Berdasarkan percobaan yang dilakukan ternyata didapatkan bahwa segmentasi warna kulit memiliki warna pokok yang dominan, yaitu warna merah *(Red)*. Berdasarkan pada hal tersebut, maka dapat dipisahkan mana saja kandidat-kandidat warna kulit yang terdapat dalam gambar yan diambil dari *webcam.* Dalam segmentasi ini digunakan persamaan untuk membedakan antara warna kulit dan yang buka warna kulit. Persamaan yang digunakan adalah sebagai berikut (Eckert, 2002) :

$$\begin{pmatrix} Er \\ Eg \\ Eb \end{pmatrix} \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}701 & -0{,}587 & -0{,}114 \\ 0{,}701 & -0{,}587 & 0{,}886 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Contoh perhitungan suatu image untuk warna kulit dengan nilai R lebih besar dari pada G dan B, sehingga didapatkan hasil nilai Eg > 0. Sedangkan untuk

yang bukan warna kulit memiliki nilai R sama dengan G dan B, atau R lebih kecil dari G atau B, maka Eg < 0. Jadi secara matematis hubungan Eg dengan komponen RGB adalah sebagai berikut:

1. Eg = 0,701 * R – 0,587 * G – 0,114 * B, faktor pengali untuk R (0,701) merupakan jumlah faktor pengali dari G dan B. jika nilai R > G dan R > B, maka nilai Eg sudah pasti bernilai > 0.
2. Sebagai contoh perhitungan warna kulit yang penulis berikan adalah bernilai R=253, G=249, dan B=243, maka nilai Eg: Eg = 0,0701*253 – 0,587*249 – 0,114*243 Eg = 3,488
3. Sebagai contoh perhitungan yang bukan warna kulit yang penulis berikan adalah memiliki komponen warna R=94, G=96, dan B=85, maka nilai Eg: Eg = 0,701*94 – 0,587*96 – 0,114*85 Eg = -0,148

Dari hasil perhitungan pada tahap diatas, menggunakan rumus persamaan Eg = 0,701*R – 0,587*G – 0,114*B, didapatkan bahwa untuk warna kulit wajah nilai Eg < 179 dan Eg ≥ 0 dengan komponen G dan B yang bernilai nol. Untuk selain warna kulit wajah nilai Eg < 0, yang bisa diartikan bahwa nilai warna putih (R = 255, G = 255, B = 255) yang ditambahkan Eg + 1. Berdasarkan percobaan, selain warna kulit memiliki nilai G dan B maksimum yaitu masing-masing 255.

3.6.2 Tahapan Lokalisasi Wajah

Tahap lokalisasi wajah merupakan tahap lanjutan dalam proses pendeteksian wajah. Tahap ini melanjutkan hasil yang didapat dari tahap sebelumnya yaitu tahap segmentasi warna kulit. Tahap ini bertujuan untuk memisahkan mana yang merupakan wajah atau bukan bagian wajah.

Adapun cara yang digunakan dalam menseleksi kandidat mana yang wajah dan mana yang bukan wajah yaitu dengan analisis matematis sebagai berikut:

$$Eg = 0{,}701*R - 0{,}587*G - 0{,}114*B$$

Hasil yang didapatkan dari formula di atas masih belum seperti yang diinginkan dan masih belum bisa dibuat acuan dalam pengambilan keputusan sebagai citra wajah. Untuk itu diperlukannya tambahan rumus dan perhitungan

baru yang diharapkan dapat menjadikan hasil menjadi lebih baik. Adapun rumusan yang ditambahkan adalah:

$$\text{Wajah} = C_r^2 . (C_r^2 .- \eta . C_r / C_b)^2 \qquad (1)$$

$$\text{di mana } \eta = 0{,}95 \frac{\frac{1}{n}\sum_{(x,y)\in FG} C_r(x,y)}{\frac{1}{n}\sum_{(x,y)\in FG} C_r(x,y) / C_b(x,y)} \qquad (2)$$

Semua tahapan ini dilakukan pada seluruh region yang memiliki nilai warna 1 pada citra biner. Setelah didapatkan citra dari hasil perhitungan formula di atas tidak langsung semata-mata diambil kesimpulan bahwa citra yang memiliki nilai 1 adalah wajah. Masih harus diberikan batasan-batasan seberapa besar citra wajah yang akan diambil. Nilai-nilai yang terdapat dalam koordinat tersebut digunakan untuk membentuk segi empat yang mengelilingi wajah *(bounding box)* yang terdapat dalam citra seseorang. Setelah didapatkan *bounding box* maka selanjutnya akan dilakukan pemotongan/ cropping citra berdasarkan Bounding box pada citra diam tunggal awal.

3.6.3 Tahapan Pencarian Fitur-Fitur Wajah

3.6.3.1 Pemrosesan Pencarian Posisi Mata

Secara logika dalam area wajah akan terdapat tepi-tepi yang jelas yang membedakan antara bagian-bagian wajah seperti mata, hidung, dan mulut. Untuk mendeteksi tepi-tepi tersebut digunakan metode deteksi tepi. Dalam hal ini area wajah membentang dari sisi atas wajah sampai sisi bawah wajah.

Dalam proses pencarian posisi mata pertama yang dilakukan adalah dengan membangun dua peta mata yang terpisah, satu dari komponen chrominance dan yang lainnya dari komponen pencahayaan. Kedua peta ini kemudian digabungkan ke dalam peta mata tunggal. Peta mata dari kroma didasarkan pada pengamatan terhadap tinggi rendahnya nilai Cb dan Cr yang ditemukan di sekitar mata. Kedua peta mata tersebut adalah PetaMataC dan PetaMataL. Pada pembentukan PetaMataC komposisi yang paling membentuk karakternya adalah nilai Cb dan Cr yang berada pada citra yang telah di konversi kedalam citra YCbCr. Sedangkan

pada pembentukan PetaMataL yang banyak dihitung adalah nilai Y pada Citra YCbCr itu sendiri. Adapun perhitungan yang dilakukan menggunakan rumus:

$$\text{Peta MataC} = \frac{1}{3}\{(C_b^2) + (C_r)^2 + (C_b/C_r)\}$$

$$\text{Peta MataL} = \frac{Y\,(x,y) \oplus g\sigma(x,y)}{Y\,(x,y) \ominus g\sigma(x,y) + 1}$$

Setelah didapatkan PetaMataC dana PetaMataL maka untuk mendapatkan PetaMata digunakan:

PetaMata = (PetaMataC) **AND** (PetaMataL)

Secara garis besar proses pendeteksian posisi mata dilakukan melalui tahapan sebagai berikut:

1. Menyiapkan hasil pemotongan/*cropping* wajah yang telah dilakukan pada proses sebelumnya.
2. Mengubahnya kedalam citra Ycbcr.
3. Memisahkan citra menjadi tiga jenis citra berdasarkan nilai Y, Cb, Cr pada citra itu sendiri.
4. Membuat PetaMataL pada persamaan 2 dengan menggunakan nilai Y yang telah diketahui.
5. Membuat PetaMataC pada persamaan 1 dengan menggunakan nilai Cb dan Cr pada yang telah didapatkan. 6. Setelah didapatkan PetaMataC dan PetaMataL maka dilakukan penggabungan untuk mendapatkan PetaMata (PetaMataC * PetaMataL).
6. Setelah didapatkan proses penggabungan mata akan didapatkan PetaMata yang telah dapat diketahui letak mata. Namun untuk mencari mendapatkan hasil yang lenbih jernih dan bersih maka akan dilakukan *erosi* dan *masking.*

Setelah didapatkan proses penggabungan mata akan didapatkan PetaMata yang telah dapat diketahui letak mata. Namun untuk mencari mendapatkan hasil

yang lebih jernih dan bersih maka akan dilakukan *erosi* dan *masking* pada citra PetaMata agar pembacaan posisi mata dapat lebih mudah dilakukan.

3.6.3.2 Tahap Pencarian Posisi Hidung

Pada pencarian posisi hidung mempunyai cara yang tidak jauh berbeda dengan cara pencarian posisi mata, pada pencarian posisi hidung ini sebenarnya juga telah didapatkan pada pencarian posisi mata namun menggunakan koordinat yang berbeda untuk pendeteksiannya. Pada pendeteksian posisi hidung menggunakan koordinat acuan bernilai 1 pada 1/3 tinggi wajah dan 1/3 lebar wajah bagian tengah.

3.6.3.3 Pemrosesan Pencarian Posisi Mulut

Dalam pencarian posisi mulut, dilakukan dengan mempersempit area pencarian posisi mulut dalam area wajah. Posisi mulut manusia terletak pada batas atas dan batas bawah tertentu dari sisi panjang dan lebar wajah. Berdasarkan hasil percobaan, batas atas yang dapat dijadikan acuan dan prediksi keberadaan mulut adalah 0,25 x tinggi wajah yang diukur dari sisi bawah wajah. Sedangkan batas bawah adalah 0,96 x tinggi wajah yang diukur dari sisi atas wajah.

3.6.4 Pengukuran Jarak Antar Fitur Wajah

Setelah semua fitur wajah teridentifikasi dan diketahui posisi-posisinya maka akan dilakukan penghitungan jarak antar fitur wajah yaitu:

1. Jarak mata kanan-mulut
2. Jarak mata kiri-mulut
3. Jarak mata kanan-hidung
4. Jarak mata kiri-hidung
5. Jarak antar mata

Jarak antar fitur ini dihitung dengan menggunakan persamaan Jarak Euclidean 2D:

$$\sqrt{dx^2 + dy^2}\ d_{12} = \qquad (1)$$

di mana,

$$dx = x_2 - x_1 \qquad (2)$$

$$dy = y_2 - y_1 \qquad (3)$$

Berdasarkan literatur yang diambil dari jurnal sebagai bahan referensi dalam penelitian dan penyusunan skripsi ini didapatkan bahwa setiap pengujian sistem

menggunakan 30 sampel dari 9 orang maka akan dibuat persentase keakuratan dari sistem tersebut, untuk mengetahui persentase sistem ini maka hasil pengujian di atas dapat dimasukkan dalam persamaan berikut ini [1]:

$$\text{Keakuratan} = \frac{\textit{Hasil Benar}}{\textit{Banyaknya Pengujian}} \times 100\%$$

$$\text{Keakuratan} = \frac{28}{30} \times 100\%$$

$$\text{Keakuratan} = 93{,}3\%$$

$$\text{Kesalahan} = 100\% - \text{Keakuratan}$$

$$\text{Kesalahan} = 100\% - 93{,}3\%$$

$$\text{Kesalahan} = 6{,}7\%$$

Dari perhitungan di atas menunjukkan hasil dari kerja sistem pengenalan wajah, didapatkan 93,3% keakuratan dan 6,7% kesalahan. Namun kesalahan yang didapatkan merupakan kesalahan positif. Kesalahan positif adalah kesalahan sistem yang dipakai namun tidak berakibat fatal pada sistem keamanan yang dikembangkan.

BAB IV ANALISA DAN PERANCANGAN

4.1 Analisa Sistem

Analisa sistem merupakan suatu penjabaran mengenai komponen-komponen penyusun sistem pada penelitian ini baik perangkat lunak maupun perangkat keras yang digunakan. Selain itu, analisa sistem juga menjabarkan gambaran umum mengenai sistem sistem yang dirancang.

Sistem yang akan dibangun adalah sistem keamanan pintu ruang *server* yang mengimplementasi *minicomputer raspberry pi sebagai tool*nya. Perancangan sistem keamanan pintu ruang *server* tersebut dibangun sesuai dengan kebutuhan-kebutuhan sebagai berikut:

1. Membangun sistem keamanan berteknologi *face recognition* yang dapat menjamin keamanan ruang *server.*
2. Mengimplementasikan metode *triangle face* sebagai metode pendeteksian wajah sesuai dengan tahapan yang ada di dalamnya.
3. Mengimplementasikan keoptimalan *minicomputer raspberry pi* yang memiliki dimensi kecil, biaya terjangkau dan kebutuhan daya listrik yang kecil serta menggunakan *USB Webcam* untuk mendeteksi wajah.
4. Melakukan pengujian sistem untuk diukur performa kinerja dalam pengamanan pintu *ruang server.*

4.2 Gambaran Umum

Sistem keamanan pintu ruang *server* yang dibangun bertujuan untuk memberikan pengamanan pada ruang *server* yang di dalamnya menyimpan data-data perusahaan yang penting dan sangat berharga bagi perusahaan. Pengamanan ruang tersebut tidak menggunakan kunci layaknya pintu pada kamar atau rumah, melainkan menggunakan citra wajah untuk membuka pintu dengan deteksi kamera yang telah dipasang di dekat pintu.

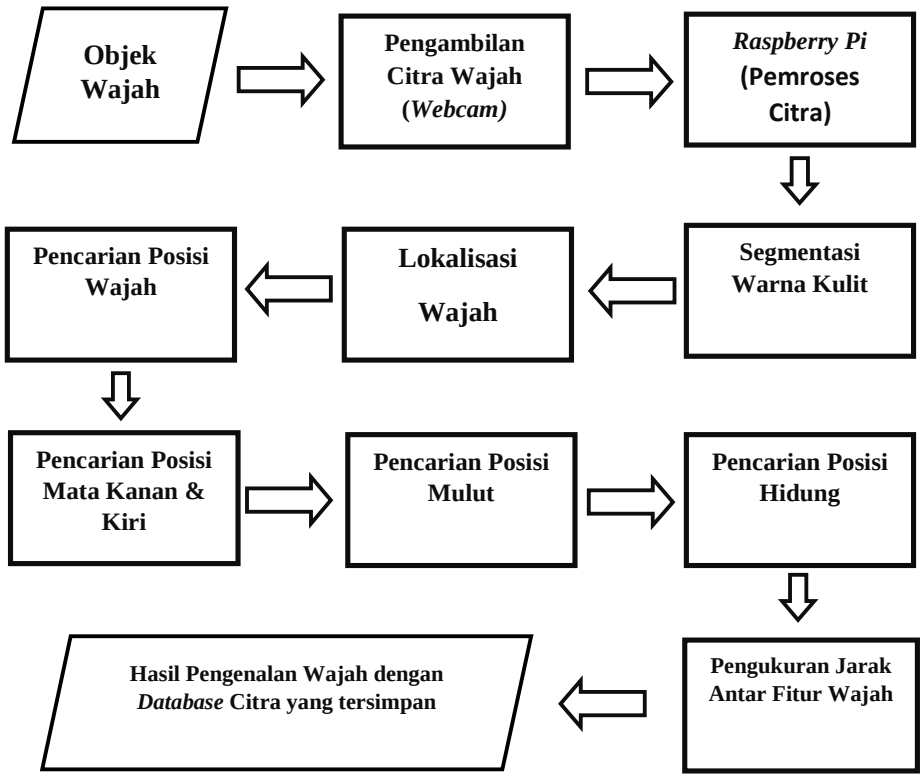
Proses awal yang dilakukan adalah admin yang diberikan hak akses pada ruang *server* memposisikan wajah di depan kamera, kemudian kamera akan mendeteksi wajah. Citra wajah yang dideteksi kemudian akan dikenali oleh sistem dan dicocokkan dengan citra yang sebelumnya telah tersimpan dalam *database.*

Jika citra wajah yang dideteksi kamera tidak dikenali sistem, maka sistem memberikan peringatan bahwa citra wajah tidak sesuai. Jika citra wajah sesuai dengan citra yang sebelumnya sudah tersimpan di *database* maka pintu akan terbuka secara otomatis.

4.3 Blok Diagram Sistem

Unit penangkap citra wajah pengguna terdiri atas *webcam* yang terhubung ke *raspberry pi* melalui jalur *port USB*. Unit sentral berupa *raspberry pi* yang bertugas mengendalikan sistem dan mengendalikan perangkat simulasi pintu *server* yang akan membuka dan menutup jika hasil deteksi citra cocok dengan citra yang ada di *database* sistem.

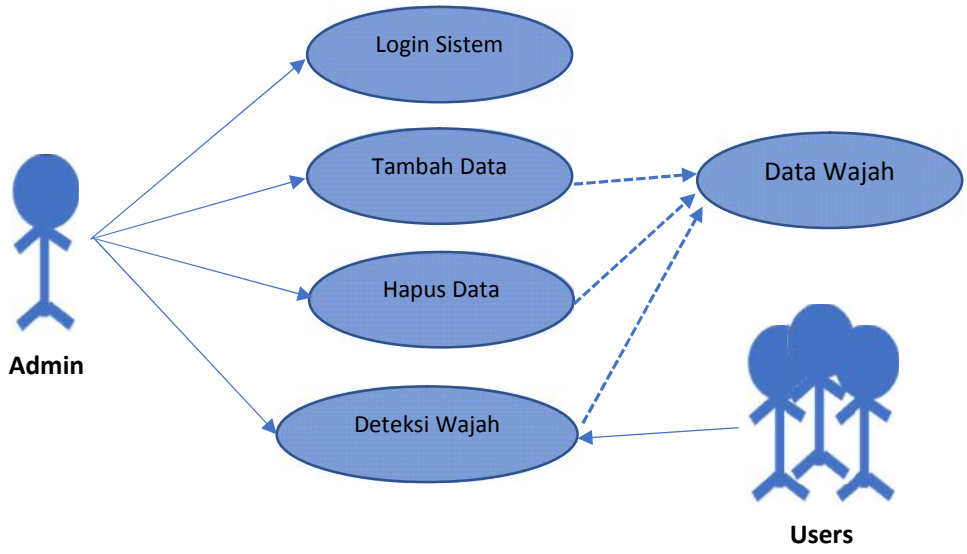
Pada blok diagram sistem ini menunjukkan alur dari pendeteksian citra wajah dan pemrosesan dengan menggunakan metode *triangle face*. Mulai dari objek wajah yang akan dideteksi oleh kamera dan akan diolah oleh sistem pengolahan citra.



Gambar 4.1 Blok diagram sistem

4.4 *Use Case* Sistem

Pada gambar 4.2 di bawah ini merupakan *use case* dari sistem yang dirancang. *Use case* yang dibuat menggambarkan masing-masing aktor yang memiliki hak akses terhadap sistem sesuai dengan *level* aktor. Pada sistem yang dirancang hanya memiliki 2 aktor, yaitu *admin* dan *users*. *Admin* memiliki hak akses secara menyeluruh terhadap sistem mulai *login* sistem, tambah data, hapus data dan pendeteksian wajah untuk akses ruang. *Users* hanya memiliki akses untuk masuk pada ruang *server* dengan pendeteksian wajah.

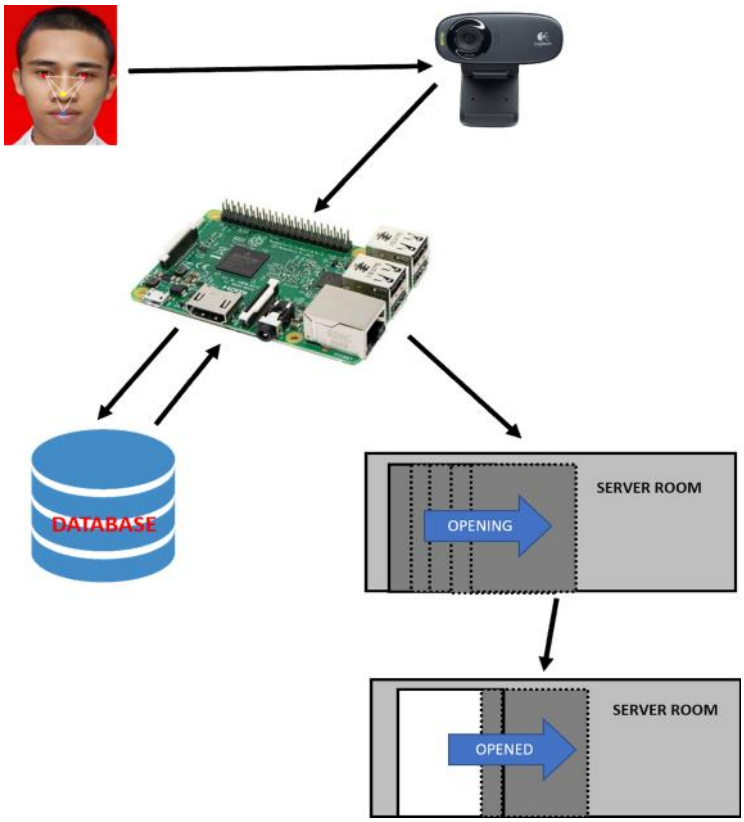


Gambar 4.2 *Use case*

4.5 Skema Konfigurasi Sistem

Dalam perancangan pemodelan konfigurasi sistem keamanan pintu *server* ini terdiri dari beberapa perangkat keras yang terinstalasi sesuai dengan konfigurasi yang telah didesain sebelumnya. Pemodelan dari konfigurasi sistem yang dirancang digambarkan pada skema konfigurasi sistem keamaan ruang *server* yang terdiri dari beberapa perangkat keras penyusun. Perangkat *webcam* sebagai perangkat pendeteksi citra wajah yang dihubungkan dengan *raspberry pi* melalui *USB port* dan *raspberry pi* sebagai perangkat pemroses, *database storage* dengan memanfaatkan micro*SD card* yang terinstal pada *raspberry pi* sekaligus sebagai tempat sistem operasi *raspbian* terinstal dan kontroller yang akan

menggerakkan *servo motor* yang akan membuat pintu terbuka secara otomatis setelah citra wajah yang dideteksi *webcam* dan cocok dengan citra yang tersimpan dalam *database*. Pada Gambar 4.3 merupakan skema konfigurasi sistem dalam perancangan sistem keamanan pintu ruang *server*.



Gambar 4.3 Skema konfigurasi sistem

4.6 Perancangan *Database*

Dalam penelitian ini membutuhkan *database* yang akan digunakan sebagai wadah untuk menampung data hasil dari pengukuran wajah sebagai tahapan dalam mendaftarkan *user* yang diberikan hak akses untuk memasuki ruang *server*. Pada saat pengambilan citra wajah maka sistem akan secara langsung mengolah citra dan mengukur jarak antar fitur wajah kemudian hasil pengukuran disimpan ke dalam *database*. *Database* juga digunakan untuk pemanggilan data pada waktu sistem mengenali citra wajah yang ditangkap dengan mengukur jarak antar fitur wajah kemudian dicocokkan dengan data hasil pengukuran jarak antar fitur wajah.

Dalam perancangan *database* ini terdiri dari 2 tabel yang terdiri dari tabel admin dan login_face. Tabel admin menyimpan data akses yang digunakan untuk melakukan *login* pada sistem dan tabel login_face yang digunakan untuk menyimpan data hasil pengukuran jarak antar fitur wajah oleh sistem. Adapun rancangan *database* yang akan digunakan dalam penelitian ini dapat dilihat pada tabel 4.1 dan tabel 4.2 di bawah ini. Pada tabel 4.1 menunjukkan tabel admin yang berisikan *username* dan *password* yang dimiliki admin untuk melakukan *login* pada aplikasi.

Tabel 4.1 Tabel admin

Tabel admin
username varchar
password varchar

Pada tabel 4.2 menunjukkan tabel login_face yang berisikan data hasil pengukuran jarak antar wajah yang diambil oleh sistem menggunakan kamera. Kedua tabel yang dibuat tidak memiliki relasi, melainkan masing-masing berdiri sendiri karena masing-masing tabel memiliki fungsi yang berbeda. Data yang dimasukkan pada tabel login_face ini akan digunakan untuk proses klasifikasi atau pencocokan data hasil pengukuran jarak antar wajah yang diambil oleh sistem, jika terjadi kecocokan antara hasil pengukuran wajah yang dilakukan dengan data ukuran wajah yang telah tersimpan dalam database maka pintu ruang *server* akan terbuka.

Tabel 4.2 Tabel login_face

Tabel login_face	
id_face	int
NIP	int
nama	varchar
luas_wajah	int
makan_mulut	int
makin_mulut	int
makan_hidung	int
makin_hidung	int
jarak_mata	int

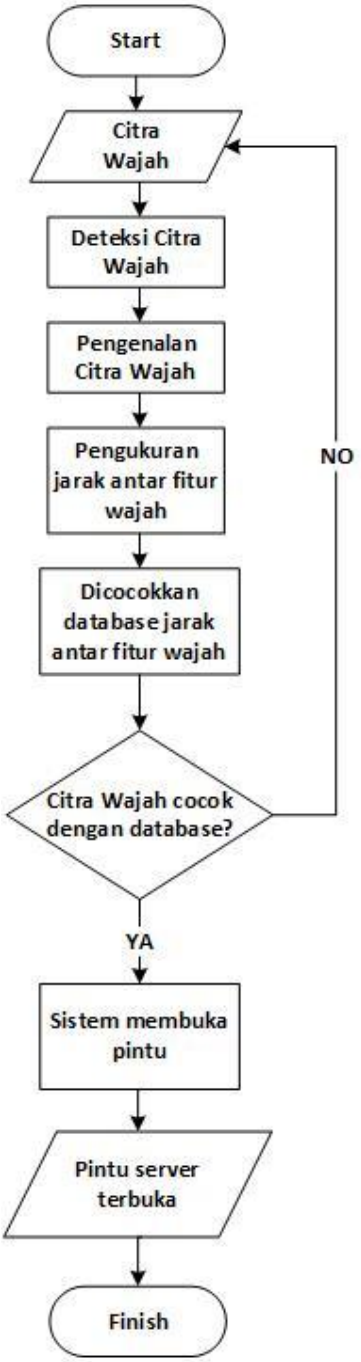
4.7 *Flowchart*

Sistem keamanan pintu ruang *server* yang dikembangkan memiliki alur sistem yang menggambarkan bagaimana proses sistem dapat bekerja sesuai dengan tujuan pengembangan sistem. Alur sistem digambarkan menggunakan *flowchart.* Untuk memudahkan dalam membaca alur sistem, pada bagian ini penulis membagi alur sistem dengan beberapa *flowchart*, antara lain *flowchart register user*, *flowchart enter room* dan *flowchart* dari aplikasi yang dikembangkan.

Adapun *flowchart* sistem dari *register user* dapat dilihat pada gambar 4.4 di bawah ini yang menggambarkan bagaimana proses sistem dalam mendeteksi wajah dan mengukur jarak antar fitur wajah kemudian menyimpannya dalam *database.*

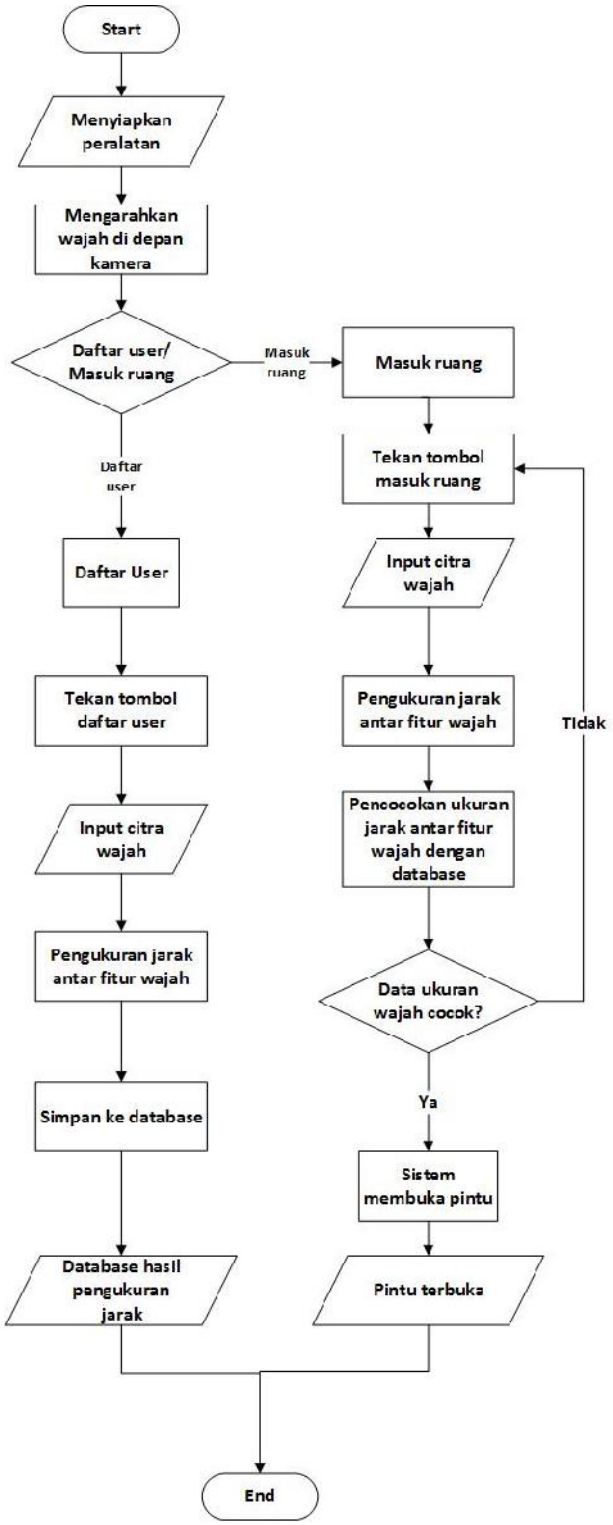
Gambar 4.4 *Flowchart register user*

Sistem menyimpan hasil pengukuran wajah ke dalam *database* kemudian data yang disimpan akan digunakan untuk klasifikasi dengan data wajah yang akan diambil dalam proses membuka pintu ruang *server*. Adapun alur sistem dalam membuka ruang digambarkan pada *flowchart* yang dapat dilihat pada gambar 4.5 di bawah ini.



Gambar 4.5 *Flowchart enter room*

Flowchart yang menggambarkan alur kerja dari aplikasi dapat dilihat pada gambar 4.6 di bawah ini.



Gambar 4.6 *Flowchart* aplikasi

4.8 Analisa Kebutuhan Perangkat Keras

Pada penelitian ini membutuhkan perangkat keras untuk mengimplementasikan sistem keamanan pintu ruang *server* sesuai dengan hasil analisa kebutuhan. Adapun perangkat keras meliputi alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Minicomputer raspberry pi*
2. *Acrylic case raspberry pi* model *B*
3. *USB Adaptor*
4. *VGA to HDMI converter*
5. *Heatsink raspberry pi*
6. *MicroSD card +adapter*
7. *USB Webcam*
8. Kabel *UTP*
9. *Servo motor*
10. *Rainbow Cable 40 cable*
11. *GPIO adapter expansion board 40 PIN*
12. *GPIO bread board*
13. Kabel *jumper*
14. Kotak simulasi
15. Mur baut
16. Engsel pintu *acrylic*
17. Kabel *ties*

4.8.1 *Minicomputer raspberry pi*

Perangkat keras utama yang digunakan dalam penelitian adalah *minicomputer raspberry pi*. *Raspberry pi* yang digunakan menggunakan generasi terbaru yaitu *Raspberry pi 3 model B* yang sebenarnya memiliki spesifikasi yang sama dengan model sebelumnya yaitu *raspberry pi 2 model B.* Keunggulan yang dimiliki *raspberry* generasi terbaru ini yaitu sudah dilengkapi dengan *wireless LAN adapter* dan *bluetooth*. Semakin lengkap fitur yang dimiliki, sehingga semakin menunjang sebuah penelitian yang memanfaatkan *minicomputer* ini. Adapun spesifikasi dari *Raspberry pi 3 model B* adalah sebagai berikut:

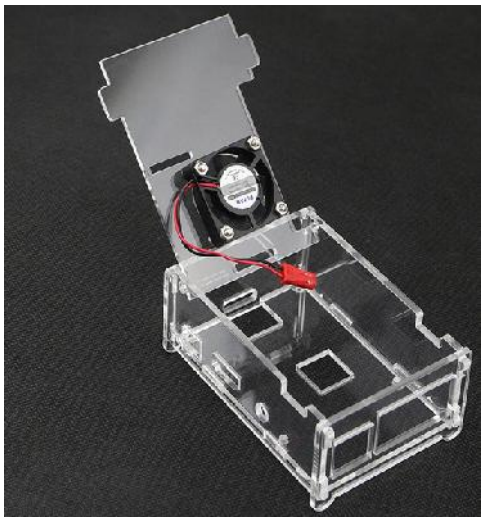
- *Processor* : *1.2GHz 64-bit quad-core CPU ARMv8*
- *Wireless* : *802.11n*
- *Bluetooth* : *Bluetooth 4.1 Bluetooth Low Energy (BLE)*
- *Memory* : *1GB RAM*
- *Port USB* : *4 Port USB*
- *PIN Out* : *40 PIN GPIO*
- *HDMI* : *1 Port HDMI*
- *Ethernet* : *1 Ethernet Port*
- *Port Camera* : *Kamera Antarmuka (Csi)*
- *Port Display* : Tampilan Antarmuka (Dsi)
- *Storage* : *Slot microSD card*
- *VGA* : *Inti Grafis Videocore Iv 3d (Vga on Board)*

Gambar 4.7 *Raspberry pi* 3 model B

4.8.2 *Acrylic Case Raspberry Pi 3 Model B + Pi Fan*

Keamanan *raspberry pi* selama melakukan penelitian perlu diperhatikan dari bahaya korsleting, terjaga dari debu dan kotoran, untuk itu memerlukan sebuah

case untuk *raspberry pi* yang presisi dengan bahan *acrylic* dan dilengkapi dengan *pi fan* untuk menjaga kestabilan suhu *raspberry pi.*



Gambar 4.8 *Acrylic case raspberry pi* 3 model B

4.8.3 *USB Adaptor*

Raspberry pi yang digunakan memerlukan sumber daya dari sebuah *adaptor*. Sumber daya akan menyuplai listrik untuk *raspberry pi* sesuai dengan kebutuhan daya listrik. Minimum *adaptor adaptor* yang direkomendasikan untuk sebagai *power supply* untuk *raspberry* yaitu memiliki tegangan 5*volt* dan minimal arus listrik 2 *amphere.* Pada penelitian ini menggunakan *adaptor* merk DVE dengan tegangan 5*volt* dan arus 2 *amphere.*

Gambar 4.9 *USB adaptor*

4.8.4 *VGA to HDMI Converter*

Dalam melakukan konfigurasi awal, *raspberry pi* memerlukan *VGA to HDMI converter* yang berfungsi untuk menampilkan *output* tampilan hingga proses konfigurasi awal selesai dan dapat di*remote* menggunakan *PC/laptop.*

Gambar 4.10 VGA to HDMI converter

4.8.5 *Heatsink Raspberry Pi*

Heatsink berbahan aluminium berukuran 1 x 1cm yang terpasang pada *processor dan RAM raspberry pi* bertujuan untuk mengurangi panas selama *raspberry pi* bekerja. Seperti konsep dasar fisika yaitu panas bisa dirambatkan melalui benda konduktor seperti besi, tembaga, alumunium, emas dan lain-lain.

Gambar 4.11 *Heatsink pada raspberry pi*

4.8.6 *MicroSD Card + Adapter*

Storage yang digunakan untuk meletakkan sistem operasi dan menyimpan data pada *raspberry pi* menggunakan *microSD card.* Pada penelitian ini

menggunakan *microSD card* dengan merk Sandisk dengan ukuran 16 GB sesuai dengan ukuran yang direkomendasikan. Setelah melakukan instalasi *raspbian OS* dan aplikasi yang memerlukan proses lamau dalam instalasinya disaranan untuk melakukan *backup microSD card* agar tidak kehilangan data dan tidak mengulangi dari awal lagi dalam instalasi *raspbian OS* dan aplikasi yang digunakan. Ketika jika diperlukan untuk instalasi ulang cukup melakukan *write* pada *microSD* dengan aplikasi *Win32Disk Imager. Adapter microSD* digunakan ketika menyalin *file ISO* sistem operasi *raspbian* menggunakan *PC/laptop.*

Gambar 4.12 *Sandisk microSD card*

4.8.7 *USB Webcam*

Piranti yang digunakan untuk melakukan deteksi dan pengenalan wajah dalam penelitian ini menggunaka *USB Web Camera* dengan merk *Logitech* seri *Logitech C170* dengan spesifikasi sebagai berikut:

- Panggilan *video* (640 x 480 *pixel*) dengan sistem yang direkomendasikan
- Merekam *video* hingga 1024 x 768 *pixel*
- Teknologi *Logitech Fluid Crystal*™*
- Foto: Hingga 5 *megapixel* (ditingkatkan menggunakan perangkat lunak)
- Mikrofon bawaan dengan pengurangan *noise*
- Bersertifikat *Hi-Speed USB 2.0* (direkomendasikan)
- Klip *universal* cocok dengan berbagai *laptop*, monitor *LCD* atau *CRT*

Dengan spesifikasi kamera seperti yang tersebut di atas, pada penelitian ini dirasa sudah memenuhi spesifikasi *minimum* yang dibutuhkan dalam penelitian ini untuk mendeteksi dan mengenali wajah manusia.

Gambar 4.13 Logitech C170 *USB webcam*

4.8.8 Kabel *UTP*

Dalam melakukan *remote desktop raspberry pi* melalui komputer/*laptop* membutuhkan penghubung yaitu menggunakan kabel *UTP* dengan model konektor yang dirakit dengan model *crossover* karena mengingat bahwa untuk menghubungkan *2 devices* atau perankat yang sejenis yaitu *minicomputer* dengan komputer/*laptop* sama dengan menghubungkan komputer dengan komputer. Kabel *UTP* model *crossover* memiliki 2 ujung konektor yang berbeda dalam urutan perakitan kabel. Dalam menguhubungkan kedua piranti menggunakan kabel lebih baik dalam komunikasinya dibandingkan melalui media *wireless*.

Gambar 4.14 Kabel *UTP*

4.8.9 *Servo Motor*

Sistem mekanik pada pembuka dan penutup pintu pada kotak simulasi menggunakan *servo motor* dengan merk *Tower Pro* seri *Micro Servo SG90*. Adapun spesifikasi dari *servo motor* yang digunakan adalah sebagai berikut:

- Dimensi (pajang x lebar x tinggi) = 22 x 11,5 x 27 mm
- Berat bersih = 9 gram (10,6 gram bila kabel dan konektor ikut ditimbang)
- Torsi maksimum/*stall torque* = 1,2 kg
- Rentang sudut putaran 180°
- Catu daya operasional = 4 ~ 7,2 *Volt DC*
- *Operating Voltage* = 4.0 to 7.2 *volts*
- Kecepatan pada 4,8 *VDC* tanpa beban = 0,12 detik per 60° (57,6 *rpm*)
- *Dead band*= 10µs
- Kecepatan operasi pada 4.8V (kondisi tanpa beban) = 0.12 sec/ 60 *degrees*
- Panjang kabel = 248 mm
- *Connector Wire Length* = 9.75 *inches* (248 mm)
- Tipe konektor: universal "S" (*Futaba/JR/Berg*/dll)
- Bahan gir = plastik

Gambar 4.15 Tower Pro SG90 *servo motor*

4.8.10 *Rainbow Cable*

Rainbow cable atau biasanya disebut kabel pelangi ini difungsikan untuk menggantikan kabel *jumper* yang digunakan untuk menghubungkan antara 40 *PIN GPIO raspberry pi* ke *GPIO bread board.* Sehingga instalasi perangkat keras

lebih tertata rapi dan mudah dalam instalasi kabel *jumper* yang akan dihubungkan dengan *raspberry pi.*

Gambar 4.16 *Rainbow cable*

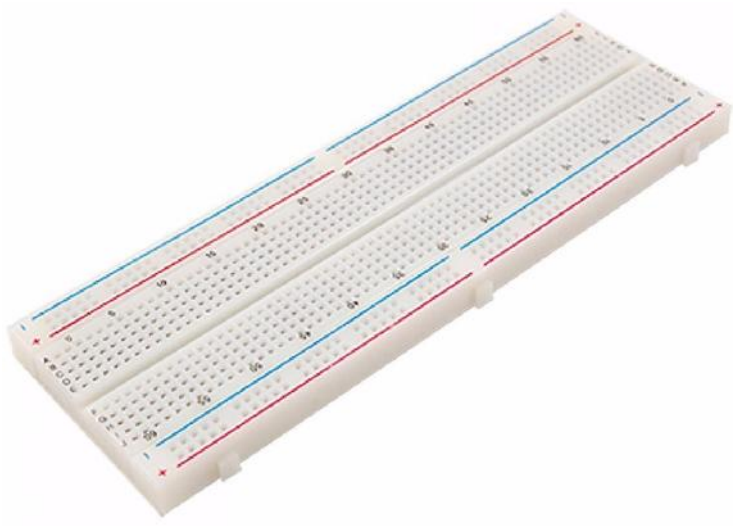
4.8.11 *GPIO Adapter Expansion Board 40 PIN*

PIN GPIO raspberry yang dihubungkan pada *GPIO bread board* menggunakan *rainbow cable* ini tidak langsung bisa dihubungkan, melainkan masih membutuhkan *adapter* agar rainbow cable bisa terhubung pada *GPIO bread board.*

Gambar 4.17 *GPIO adapter expansion board 40 PIN*

4.8.12 *GPIO Bread Board*

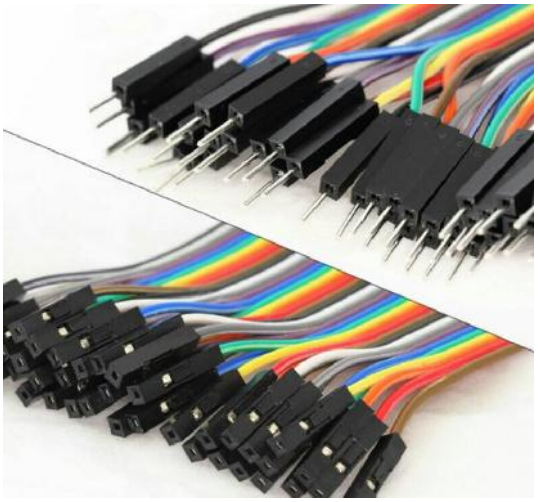
GPIO bread board atau yang biasa disebut dengan *project board* ini digunakan untuk mencolokkan semua alat yang digunakan dalam penelitian yang terhubung dengan 40 *PIN GPIO* pada *raspberry pi*.



Gambar 4.18 *GPIO bread board*

4.8.13 Kabel *Jumper*

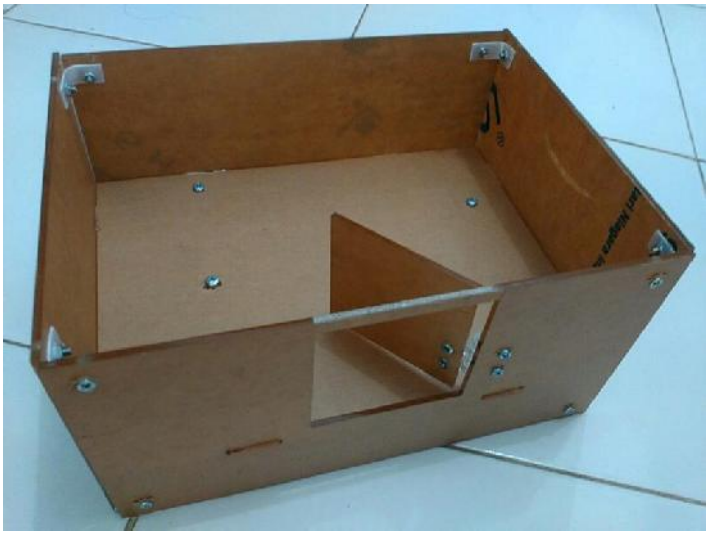
Kabel *jumper* memiliki fungsi yang sama dengan *rainbow cable*, yang mana menghubungkan antara 40 *PIN GPIO* pada *raspberry pi* dengan alat yang dikontrol seperti lampu *led*, *servo motor*, *solenoid* ataupun sensor-sensor yang digunakan dalam sebuah penelitian.



Gambar 4.19 Kabel *jumper*

4.8.14 Kotak Simulasi

Dalam penelitian ini membutuhkan kotak simulasi yang digunakan sebagai simulasi ruang *server* dan dilengkapi dengan pintu yang terbuka jika sebuah wajah yang dideteksi sistem dapat dikenali kemudian pintu terbuka secara otomatis Karena digerakkan oleh *servo motor* atas kontrol *raspberry pi*. Bahan yang digunakan untuk pembuatan kotak simulasi adalah *acrylic* yang didesain menyerupai sebuah ruangan dilengkapi sebuah pintu.



Gambar 4.20 Kotak simulasi

4.8.15 Mur dan Baut

Kotak simulasi yang digunakan dalam penelitian ini dalam perakitannya membutuhkan mur dan baut yang digunakan untuk menyatukan berbagai sisi dari kotak simulasi. Kotak simulasi yang dibuat terdiri dari beberapa bagian, yaitu bagian alas, sisi kanan dan kiri, sisi depan dan belakang yang semuanya disatukan dengan mur dan buat begitu juga pintu pada ruang simulasi dipasang beserta engsel pintu dengan mur dan baut.

Gambar 4.21 Mur dan baut

4.8.16 Engsel Pintu *Acrylic*

Pintu pada kotak simulasi terpasang dan dapat bergerak terbuka dan tutup karena terpasang dengan menggunakan engsel. Engsel yang digunakan berbahan *acrylic* sesuai dengan bahan dasar dari kotak simulasi, kemudian membuat lubang pada engsel untuk memasang mur dan baut yang digunakan untuk tempat mur dan baut.

Gambar 4.22 Engsel *acrylic*

4.8.17 Kabel *Ties*

Selain mur dan baut yang digunakan untuk menyatukan beberapa bagian kotak simulasi yang terdiri dari berbagai sisi, kabel *ties* juga dibutuhkan untuk memasang beberapa peralatan agar dapat menempel pada kotak simulasi. *Servo motor* dan *GPIO bread board* dipasang dengan kabel *ties* sehingga lebih presisi dalam pemasangannya.

Gambar 4.23 Kabel *ties*

4.9 Analisa Kebutuhan Perangkat Lunak

Pada penelitian ini membutuhkan perangkat lunak untuk mengimplementasikan sistem keamanan pintu ruang *server* sesuai dengan hasil analisa kebutuhan. Adapun perangkat lunak yang dalam penelitian ini adalah sebagai berikut:

1. Sistem operasi *raspbian jessie*
2. *SD Formatter*
3. *Win32 Disk Imager*
4. *PuTTY*
5. *VNCServer*
6. *Remote Desktop Connection*
7. *FSWebcam*
8. *OpenCV*
9. *Web Server*

4.9.1 Sistem Operasi *Raspbian Jessie*

Raspbian adalah sistem operasi gratis yang berdasarkan pada *debian* dan dioptimisasi untuk perangkat keras *raspberry pi.* Dalam penelitian ini, *raspberry pi* menggunakan sistem operasi *raspbian jessie with pixel* yang terinstall pada *microSD card* dan dipasang pada *raspberry pi. Raspbian jessie with pixel* ini merupakan. Sistem operasi yang digunakan dalam penelitian ini dapat diunduh paada alamat https://www.raspberrypi.org/downloads/raspbian/.

4.9.2 *SD Formatter*

Perangkat lunak ini digunakan untuk menformat semua kartu memori *SD* dan tidak berbayar dalam pengunduhannya, kartu memori *SDHC* dan kartu memori *SDXC. SD Card Formatter* menyediakan akses cepat dan mudah ke kemampuan penuh kartu memori *SD, SDHC* dan *SDXC. Formater SD* Card dibuat khusus untuk kartu memori dengan menggunakan standar *SD/SDHC/SDXC.* Dalam penelitian ini menggunakan *SD Formatter V4.0* yang dapat diunduh pada alamat https://www.sdcard.org/downloads/formatter_4/.

Setelah diunduh kemudian diinstall pada *PC/laptop* lalu memasukkan kartu memori *SD* ke *card reader* kemudian melakukan proses *formatting SD card*.

4.9.3 *Win32 Disk Imager*

Win32 Disk Imager adalah aplikasi *open source* yang bisa dipakai *untuk* menulis berkas *image CD* atau *DVD* ke *USB* atau kartu *SD* untuk membuat *drive* berisi *disk virtual*. Hanya perlu melakukan *unzip* (program tidak perlu instalasi). Pilih berkas *image* yang Anda inginkan dari *HD* dan perangkat yang ingin Anda tulis. Dalam beberapa detik, *pendrive* atau kartu memori akan menjadi *CD virtual*. *Win32 Disk Imager* membuat berkas *image* dengan menyalin data di perangkat penyimpan. Aplikasi yang digunakan pada penelitian ini adalah *Win32 Disk Imager* versi 0.7 yang dapat diunduh pada alamat https://win32-disk-imager.id.uptodown.com/windows/download secara gratis.

4.9.4 *PuTTY*

PuTTY adalah sebuah aplikasi *open-source yang* memanfaatkan *protokol* jaringan seperti *SSH* dan *Telnet*. *PuTTY* memanfaatkan *protokol* tersebut untuk mengaktifkan sesi *remote* pada komputer. Aplikasi *PuTTY* dapat diunduh pada alamat http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

4.9.5 *VNCServer*

VNC Server merupakan salah satu aplikasi yang digunakan untuk melakukan *remote desktop* pada sebuah *device* seperti computer, *laptop* atau perangkat *android*. Aplikasi *VNC Server* yang digunakan pada penelitian ini adalah *VNCServer* versi 6.0.3 yang dapat diunduh pada alamat https://www.realvnc.com/download/vnc/windows/ yang diperuntukkan untuk sistem operasi *windows*.

4.9.6 *Remote Desktop Connection*

Selain menggunakan *VNCServer*, untuk melakukan *remote desktop* pada *raspberry pi* melalui sebuah *laptop* dapat juga dilakukan dengan aplikasi yang sudah disediakan di dalam sistem operasi *windows*. Sehingga tidak perlu melakukan pengunduhan aplikasi cukup *log on* dengan *IP address* dan *password*.

4.9.7 *FSWebcam*

FSWebcam adalah aplikasi yang digunakan untuk melakukan *capture* gambar pada *raspberry pi*. Aplikasi ini dapat diunduh melalu *terminal linux* pada *raspberry pi* yang berintegrasi dengan *usbwebcam.* Sehingga kamera dapat digunakan untuk melakukan deteksi wajah dalam penelitian ini.

4.9.8 *OpenCV*

Pada penelitian ini *OpenCV* digunakan untuk pengolahan citra. Perangkat lunak ini dapat diunduh dan diinstall pada sistem operasi *raspbian* dengan melalui *terminal linux.* Langkah dalam mengunduh dan memasang aplikasi ini dijelaskan pada bagian implementasi.

4.9.9 *Web Server*

Web Server adalah aplikasi yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari *HTTP* atau *HTTPS* pada klien. Pada penelitian ini, peneliti menggunakan *Apache2* sebagai *web server* yang bertanggung jawab pada request-response HTTP dan *logging* informasi secara detail serta menggunakan *MYSQL* sebagai *web server* dalam manajemen basis data.

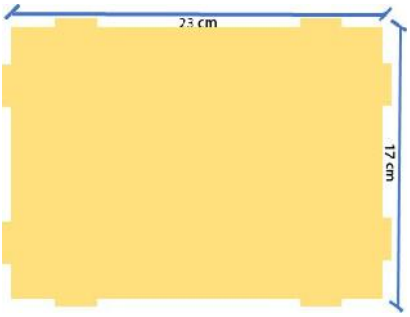
4.10 Perancangan Kotak Simulasi

Kotak simulasi pada peneitian ini digunakan sebagai simulasi ruang *server* yang nantinya pintu pada kotak dapat terbuka dan tertutup secara otomatis atas petintah dari sistem, sekaligus sebagai tempat untuk meletakkan seluruh perangkat keras yang digunakan pada penelitian ini. Bahan dari kotak simulasi yang dibuat adalah *acrylic* yang dirasa lebih mudah dalam menemukan bahan, mudah dalam pembuatan, biaya yang murah dan hasil yang lebih rapi daripada menggunakan bahan lain seperti triplek atau kardus yang dirasa kurang presisi dan kurang solid untuk dijadikan sebagai kotak simulasi. Kotak simulasi yang dibuat memiliki dimensi yang dapat dikatakan kecil yaitu berukuran 23 x 17 x 12 cm.

Kotak simulasi yang dibuat terdiri dari 5 sisi, yaitu sisi alas sebagai dasar dimana papan komponen ditempelkan, 2 sisi kanan dan kiri serta 2 sisi depan dan

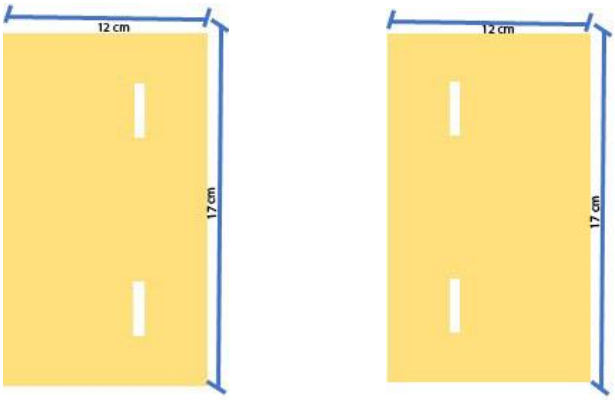
belakang. Dalam perakiran, kotak dirancang memiliki lubang pengait antar bagian dan dibaut di masing-masing sisi pojok atas dan bawah kotak. Sehingga kotak terlihat rapi dan rapat dalam rakitannya. Pada gambar 4.24 di bawah ini merupakan kerangka dari perakitan kotak simulasi yang akan digunakan dalam penelitian. Kotak simulasi akan ditunjukkan dari beberapa sisi, sisi alas, sisi kanan dan kiri, sisi depan dan belakang.

Pada gambar 4.24 merupakan desain sisi alas dari kotak simulasi yang memliki ukuran 23 x 17 cm dan memiliki 6 lidah yang digunakan untuk mengaitkan sisi kanan kiri dan depan belakang pada sisi alas lalu diberikan mur baut pada masing-masing pojok kotak simulasi.



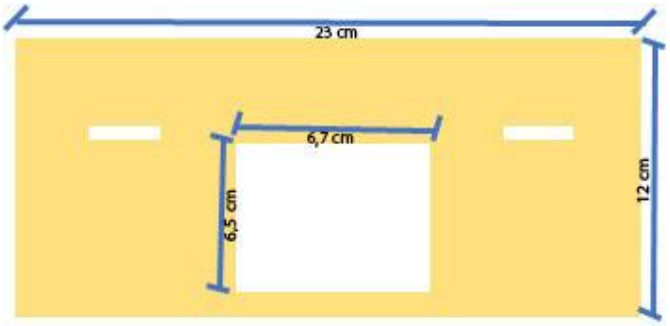
Gambar 4.24 Desain sisi alas kotak simulasi

Pada gambar 4.25 menunjukkan desain sisi kanan dan kiri kotak simulasi yang masing-masing sisi memiliki 2 lubang untuk dikaitkan dengan lidah pada sisi alas kotak simulasi. Sehingga sisi kanan dan kiri kotak dapat dengan mudah dipasang sebelum dipasang mur dan baut. Kedua sisi kanan dan kiri memiliki ukuran yang sama yaitu 17 x x12 cm.



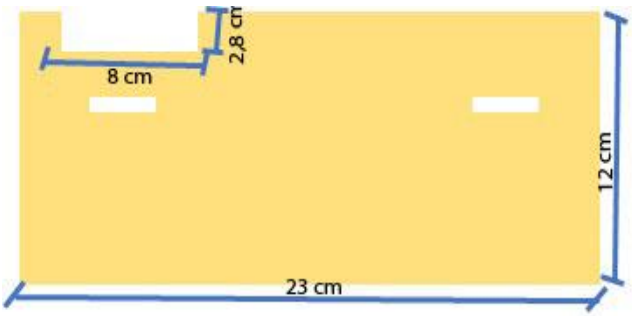
Gambar 4.25 Desain sisi kiri dan kanan kotak simulasi

Desain sisi depan kotak berbeda dengan desain sisi kanan dan kiri maupun sisi belakang kotak simulasi. Perbedaan ini terletak pada ukurannya. Desain sisi depan kotak simulasi memiliki ukuran 23 x 12 cm. Pada sisi depan kotak terdapat lubang berbentuk kotak berukuran 6,7 x 6.5 cm yang didesain sebagai simulasi pintu ruang *server*. Sehingga ukuran pintu pada simulasi ruang *server* sama dengan lubang berbentuk kotak pada desain sisi depan kotak simulasi. Selain itu juga terdapat 2 lubang pengait yang digunakan untuk mengaitkan dengan sisi alas kotak simulasi.



Gambar 4.26 Desain sisa depan kotak simulasi

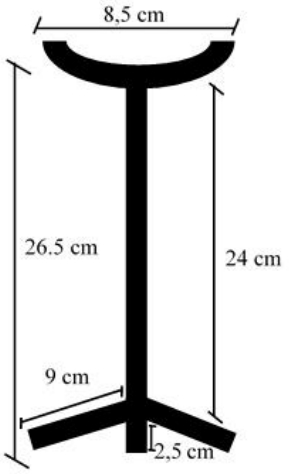
Bagian terakhir yaitu sisi belakang kotak simulasi yang memiliki ukuran sama dengan sisi depan kotak yaitu berukuran 23 x 12 cm tetapi terdapat lubang kotak di bagian bawah dengan ukuran 8 x 2,8 cm yang berfungsi untuk tempat perangkat *raspberry pi* jika sudah terpasang pada kotak masih mudah untuk melakukan konfigurasi. Sehingga masih dapat dengan mudah memanfaatkan 1 *port USB* dan 4 *port LAN adapter* pada perangkat *raspberry pi.* Pada desain sisi belakang kotak juga terdapat 2 lubang pengait untuk mengaitkan antara sisi belakang dengan sisi alat kotak simulasi.



Gambar 4.27 Desain belakang kotak simulasi

4.11 Perancangan dan Perakitan Penyangga Dagu

Dalam pengambilan citra wajah memerlukan keseragaman citra yang akan diolah sehingga dapat dimanfaatkan dalam penelitian ini. Untuk mendapatkan citra yang seragam memerlukan tempat kamera dan tempat objek yang bersifat *fix* atau tetap dengan memperhatikan jarak antara kamera dan objek yang akan diambil. Untuk itu diperlukan perancangan sebuah *tripod* sebagai tempat kamera dan sebuah tiang yang digunakan untuk menyangga dagu. Dalam perancangan *tripod* dan penyangga dagu sangat memperhatikan ukurannya. Adapun perancangan dari penyangga dagu dapat dilihat pada gambar 4.28 di bawah ini.



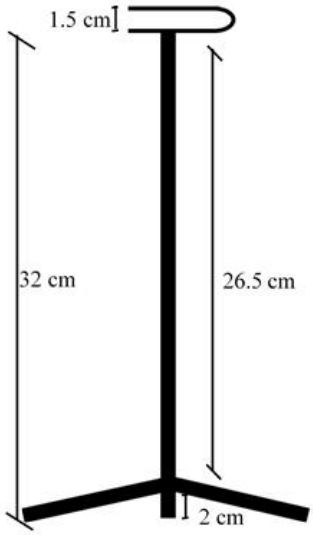
Gambar 4.28 Perancangan penyangga dagu

Hasil dari perakitan penyangga dagu yang telah dirancang sebelumnya dapat dilihat pada gambar 4.29 di bawah ini.

Gambar 4.29 Hasil perakitan penyangga dagu

4.12 Perancangan dan Perakitan *Tripod* Kamera

Adapun perancangan dari *tripod* kamera yang dirancang sesuai dengan ukuran yang telah didesain dapat dilihat pada gambar 4.30 di bawah ini.



Gambar 4.30 Rancangan *tripod* kamera

Setelah perancangan tripod kamera selesai lalu dilanjutkan perakitan *tripod* yang berbahan dasar besi kemudian di bentuk dan rakit dengan cara pengelasan. Berikut hasil dari perakitan *tripod* kamera dapat dilihat pada gambar 4.231

Gambar 4.31 Hasil perakitan *tripod* kamera

BAB V IMPLEMENTASI

5.1 Instalasi Perangkat Lunak

Pada penelitian ini memerlukan perangkat keras yang sudah terpasang perangkat lunak yang digunakan selama penelitian. Instalasi perangkat lunak meliputi instalasi sistem operasi pada *raspberry pi* hingga instalasi aplikasi yang diperlukan selama penelitian yang memerlukan konfigurasi masing-masing.

Instalasi perangkat lunak pada penelitian ini dilakukan secara bertahap, adapun tahapan dalam instalasi perangkat lunak adalah sebagai berikut:

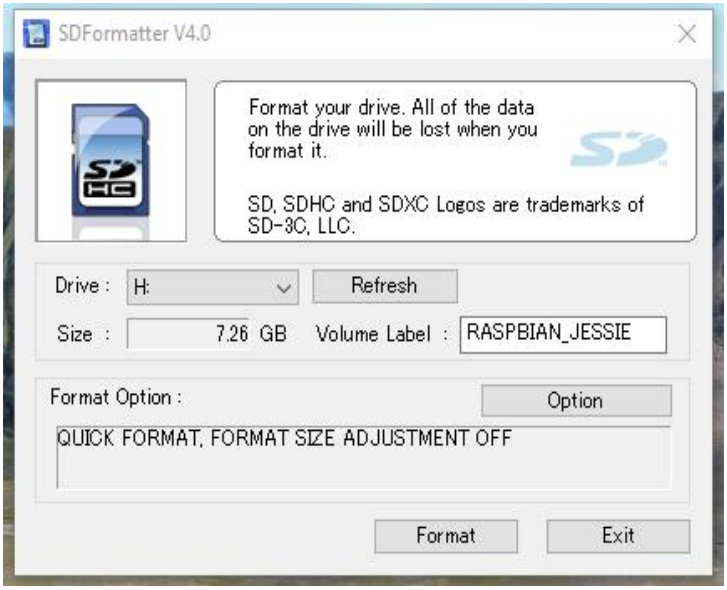
1. Instalasi *Raspbian Jessie*
2. Konfigurasi *Raspberry pi*
3. Instalasi *OpenCV*
4. Instalasi *FSWebcam*
5. Instalasi *Web Server*

5.1.1 Instalasi Raspbian Jessie

Tahapan instalasi *raspbian jessie* merupakan langkah awal dalam instalasi perangkat lunak pada penelitian ini, karena *raspbian jessie* merupakan sistem operasi yang digunakan pada perangkat *raspberry pi* yang berbasis linux. Sistem operasi *raspbian jessie* terdari dari 2 versi, yaitu *raspbian jessie with pixel* untuk versi *based GUI* atau berbasis *GUI* dan *raspbian jessie lite* dengan versi yang minimalis atau tidak berbasis *GUI* melainkan semua konfigurasi dilakukan pada terminal linux. Sistem operasi *raspbian jessie with pixel* dapat diunduh pada *link https://www.raspberrypi.org/downloads/raspbian/* yang berupa file *ISO.* Instalasi *raspbian jessie with pixel* ini terdiri dari beberapa langkah. Langkah-langkah dalam instalasi *raspbian jessie with pixel* antara lain:

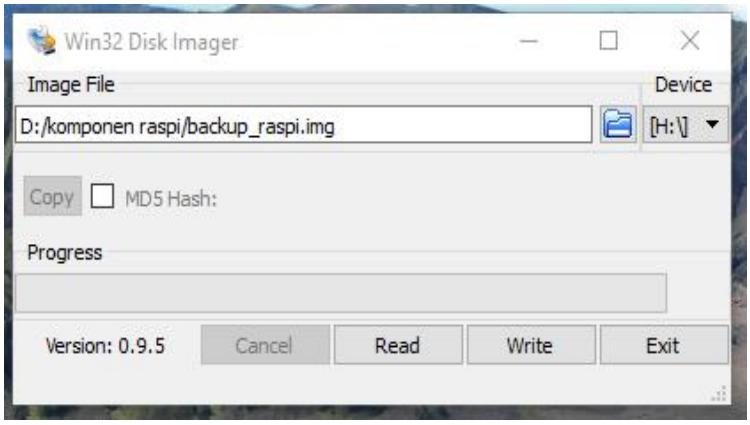
1. Mengunduh file *ISO* sistem operasi *raspbian jessie with pixel.*
2. Mengunduh aplikasi *SD Formatter* untuk melakukan *formatting* pada *microSD card* baru agar terformat dengan *NOOBS.*
3. Mengunduh aplikasi *Win32DiskImager* untuk melakukan *write* file *ISO raspbian jessie* ke dalam *microSD card.*

4. Setelah selesai melakukan pengunduhan beberapa file aplikasi untuk mempersiapkan proses instalasi sistem operasi *raspbian*, kemudian menyiapkan *microSD.*
5. Melakukan format *microSD card* sebelum melakukan *write* file *ISO* pada *microSD card* dengan aplikasi *SD Formatter* dapat dilihat pada gambar 5.1 di bawah ini.



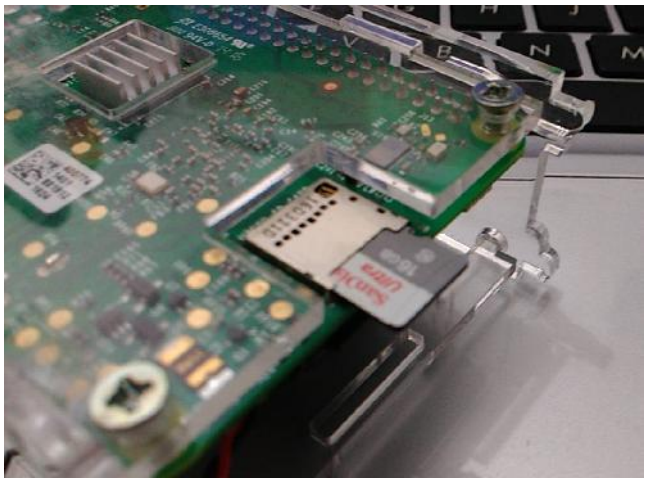
Gambar 5.1 *Formatting microSD card*

6. Memasukkan sistem operasi *raspbian jessie* ke dalam *microSD* dengan aplikasi *Win32DiskImager* dapat dilihat pada gambar 5.2 di bawah ini.



Gambar 5.2 *Write iso raspbian OS* ke dalam *microSD card*

7. Setelah file *ISO raspbian jessie* dimasukkan pada *microSD* kemudian dimasukkan ke slot *microSD* pada *raspberry pi* seperti pada gambar 5.3 di bawah ini.



Gambar 5.3 Instalasi *microSD card* ke *slot* pada *raspberry pi*

8. *Raspberry pi* siap digunakan dalam penelitian, setelah terpasang *microSD card* yang sudah terisi file *raspbian jessie.* Untuk pertama kali *raspberry pi* dijalankan menggunakan *LED monitor* sebagai *output* tampilan dengan menggunakan *converter VGA to HDMI* dan seperangkat *keyboard* serta *mouse* sebagai *input device* dapat dilihat pada gambar 5.4 di bawah ini.

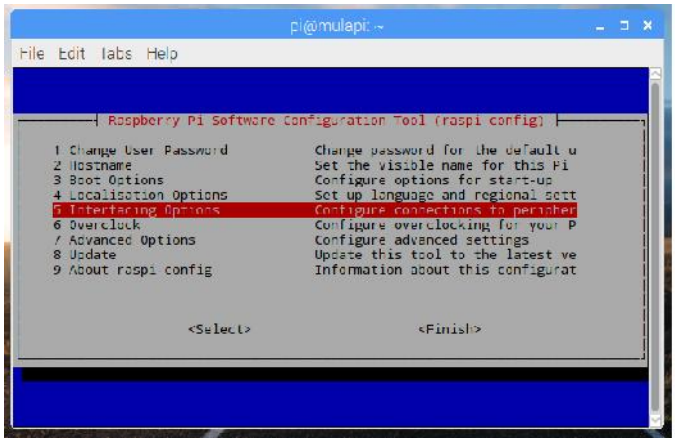
Gambar 5.4 Pengecekan hasil konfigurasi perangkat keras

9. *Raspberry pi* dengan sistem operasi *raspbian jessie with pixel* siap digunakan dalam penelitian. Tujuan utama *raspberry pi* dijalankan dengan tampilan pada *LED Monitor* dengan *input device* berupa *keyboard* dan *mouse* agar memudahkan dalam proses konfigurasi awal.

5.1.2 Konfigurasi *Raspberry pi*

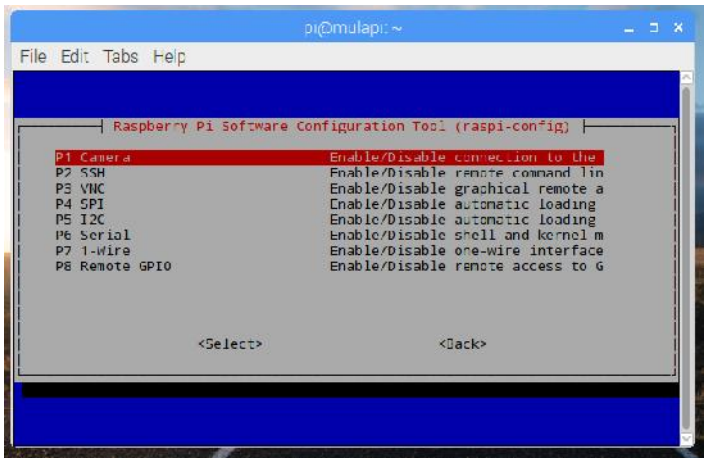
Setelah melakukan instalasi sistem operasi pada *raspberry pi* sebagai langkah awal dalam proses instalasi perangkat keras diperlukan proses konfigurasi pada *raspberry pi.* Konfigurasi yang dilakukan antara lain konfigurasi dasar yang ada pada *raspberry pi.*

Konfigurasi awal yang dilakukan karena dibutuhkan dalam penelitian pada *interface* antara lain mengaktifkan *interface* untuk *camera, SSH* dan *remote GPIO.* Konfigurasi dapat dilakukan dengan *user interface* ataupun dengan terminal linux. Perintah yang digunakan untuk melakukan konfigurasi pada terminal linux adalah *"$ sudo raspi-config"* lalu menekan *enter*, maka akan muncul halaman untuk melakukan konfigurasi. Ada beberapa pilihan pengaturan pada halaman tetapi pada penelitian ini cukup melakukan beberapa konfigurasi saja. Konfigurasi yang dilakukan antara lain *expands file system* yang bertujuan untuk mencakup semua ruang yang tersedia pada *microSD card.* Memastikan bahwa semua kartu penyimpanan *SD* tersedia untuk sistem operasi. Tampilan dari perintah *$ sudo raspi-config* dapat dilihat pada gambar 5.5. Untuk melakukan konfigurasi *raspberry pi* maka masuk pada menu *Interfacing Options,* untuk melakukan *expands file system* bisa masuk pada menu *Advanced Options.*



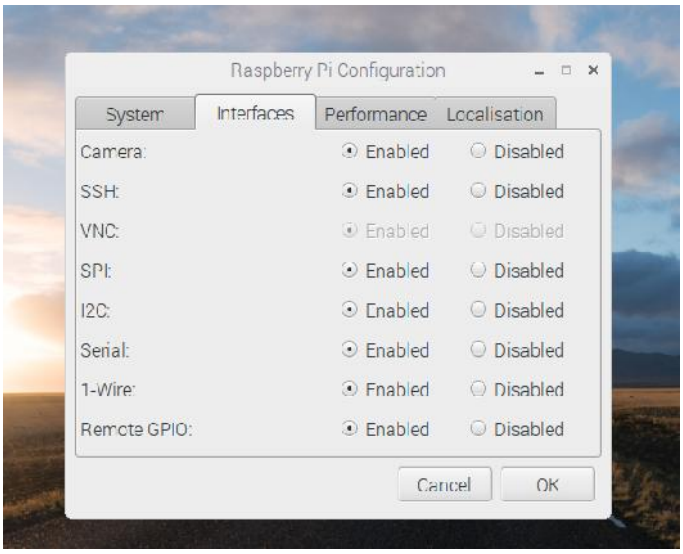
Gambar 5.5 Tampilan konfigurasi *raspberry pi* di *terminal linux*

Beberapa pilihan pada menu *Interfacing Options* yang digunakan untuk melakukan konfigurasi dasar dapat dilihat pada gambar 5.6 di bawah ini.



Gambar 5.6 *Menu interface options*

Adapun tampilan *user interface* dari konfigurasi *raspberry pi* dapat dilihat pada gambar 5.7.

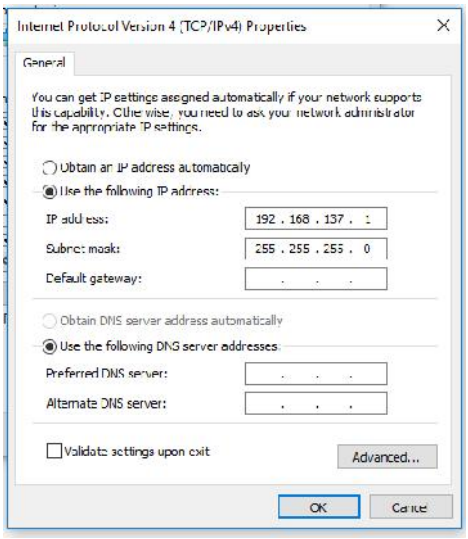


Gambar 5.7 *Raspberry pi configuration*

Setelah *raspberry pi raspberry pi* terkonfigurasi terutama konfigurasi *SSH* dan *VNC* sudah *enabled* maka *raspberry pi* siap digunakan dengan melakukan *remote desktop* melalui *laptop*. Sehingga *raspberry pi* dapat digunakan tanpa menggunakan piranti seperti *mouse*, *keyboard* dan *monitor*, cukup menggunakan fitur *remote desktop connection* atau *VNC server*. Pada penelitian ini, *raspberry*

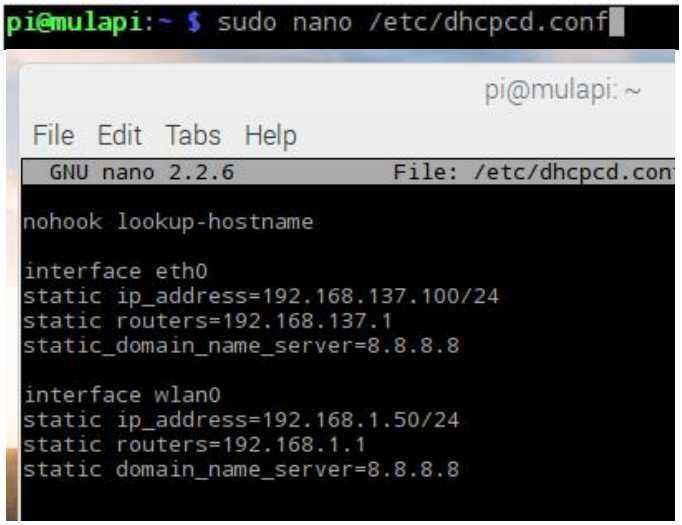
diakses dengan menggunakan fitur *remote desktop connection* pada *laptop* yang merupakan fitur yang sudah ada pada sistem operasi *windows.* Langkah untuk melakukan *remote desktop* pada *raspberry pi* adalah sebagai berikut:

1. Mengunduh dan menginstal aplikasi *PuTTY* untuk mengaktifkan sesi remote pada komputer pada halaman https://putty.id.softonic.com/
2. Mengatur IP Address untuk Ethernet0 pada *laptop* menjadi *static I* dapat dilihat pada gambar 5.8 di bawah ini.



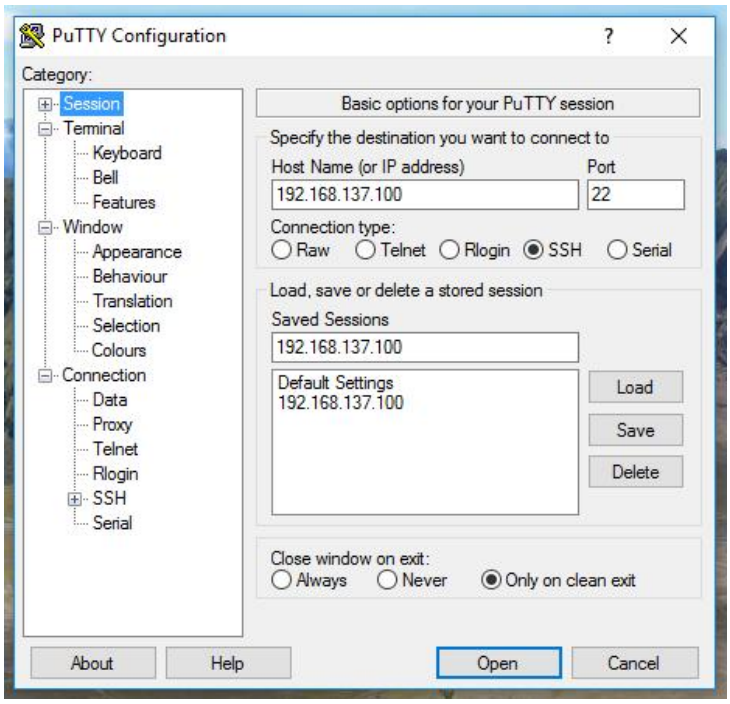
Gambar 5.8 Pengaturan *IP static* pada *laptop*

3. Mengatur *IP Address* untuk Ethernet0 pada *raspberry pi* menjadi *static* dapat dilihat pada gambar 5.9 di bawah ini.



Gambar 5.9 Pengaturan *IP static* pada *raspberry pi*

4. Menjalankan aplikasi *PuTTY* dan melakukan beberapa pengaturan dapat dilihat pada gambar 5.10 di bawah ini.



Gambar 5.10 Konfigurasi *PuTTY*

5. Membuka *terminal linux* dan menjalankan beberapa perintah untuk melakukan *remote desktop* dapat dilihat pada gambar 5.11 dan gambar 5.12 di bawah ini.

```
pi@mulapi:~ $ sudo apt-get install tightvncsecrver
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package tightvncsecrver
```

Gambar 5.11 Instalasi *tightvncserver*

```
pi@mulapi:~ $ sudo apt-get install xrdp
Reading package lists... Done
Building dependency tree
Reading state information... Done
xrdp is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

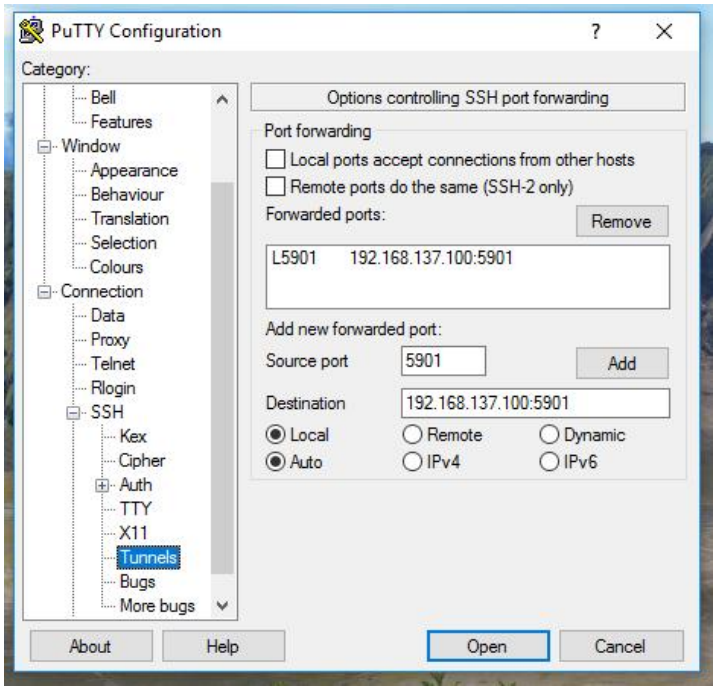
Gambar 5.12 Instalasi *xrdp*

6. *VNCServer* yang telah terinstall memerlukan password untuk mengaksesnya seperti pada gambar 5.13 di bawah ini.

```
pi@mulapi:~ $ vncserver :1

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:

New 'X' desktop is mulapi:1

Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/mulapi:1.log
```

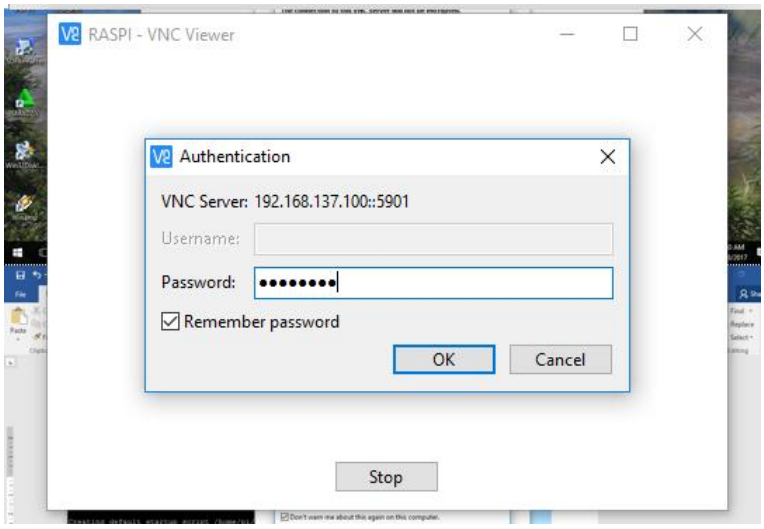
Gambar 5.13 *Require password VNC*

7. Melakukan pengaturan kembali pada aplikasi PuTTY pada menu submenu *SSH* pada menu *Connection* lalu memilih *Tunnels* kemudian mengisikan *IP Address raspberry pi* sebagai *hostname*, *port* yang digunakan dan menyimpan pengaturannya baru kemudian *login* pada aplikasi *VNC Server* dan *Remote Desktop* seperti pada gambar 5.14 di bawah ini.

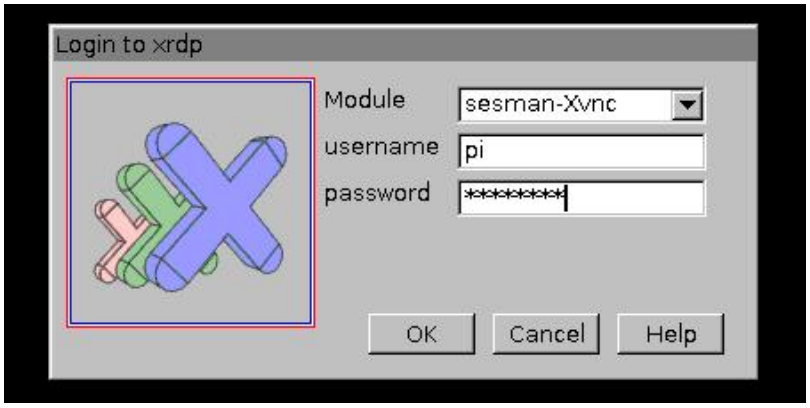


Gambar 5.14 *Setting PuTTY* untuk aplikasi *VNCserver*

8. Selesai melakukan instalasi *tightvncserver*, *xrdp* dan *verifikasi password vncserver* maka langkah selanjutnya yaitu *login vnc server* dan *remote desktop connection* dengan memasukkan *IP Address* dari *raspberry pi* dan password yang sudah diverifikasi seperti pada gambar 5.15 dan gambar 5.16 di bawah ini.



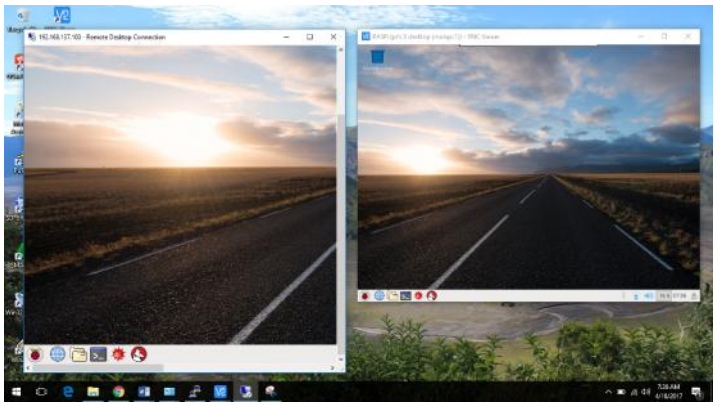
Gambar 5.15 *Login* pada *VNC Server*



Gambar 5.16 *Login* pada *Remote Desktop Connection*

9. Setelah berhasil *login* maka *raspberry pi* dapat diakses menggunakan *laptop* tanpa memerlukan *keyboard*, *mouse* dan *LED Monitor*. Jika *IP address*, *password* dan *port* sesuai dengan pengaturan, maka akan masuk pada *desktop* sistem operasi *raspbian* pada *raspberry pi* melalui *Remote Desktop Connection*.dan aplikasi *VNC Server*. *Raspberry pi* siap digunakan untuk penelitian tanpa

harus menggunakan *input* dan *output device* layaknya sebuah *personal computer* seeprti pada gambar 5.17 di bawah ini.



Gambar 5.17 Desktop sistem operasi raspbian

5.1.3 Instalasi *OpenCV*

Instalasi *OpenCV* pada penelitian ini memerlukan waktu yang lama dengan estimasi waktu selam 2 jam. Instalasi *OpenCV* memerlukan koneksi *internet* yang stabil untuk keperluan *update, upgrade* sistem dan mengunduh beberapa *package* yang dibutuhkan selama instalasi *OpenCV,* instalasi dilakukan melalui *terminal linux.* Adapun langkah-langkah dalam instalasi *OpenCV* dapat dilihat pada gambar 5.18 sampai dengan gambar 5.37 di bawah ini sebagai berikut:

1. Membuka *terminal linux* dan melakukan perintah *update* sistem

```
pi@mulapi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org jessie InRelease [22.9 kB]
Get:2 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
Get:3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,981 kB]
Get:4 http://archive.raspberrypi.org jessie/main armhf Packages [145 kB]
```

Gambar 5.18 *Update* Sistem

2. Melakukan perintah *upgrade* sistem

```
pi@mulapi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  chromium-browser libgl1-mesa-dri pprompt rpi-chromium-mods sonic-pi
  wolfram-engine xserver-xorg-input-all
```

Gambar 5.19 *Upgrade* sistem

3. Menginstal beberapa alat pengembang, termasuk *CMake*, sebagai pendukung dalam mengkonfigurasi selama proses instalasi *OpenCV*.

```
pi@mulapi:~ $ sudo apt-get install build-essential git cmake pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version.
git is already the newest version.
pkg-config is already the newest version.
cmake is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

Gambar 5.20 Instalasi *tools* pengembang

4. Menginstal beberapa paket gambar *I/O* yang memungkinkan untuk memuat berbagai format file gambar dari *disk*. Contoh *format file* tersebut termasuk *JPEG*, *PNG*, *TIFF*, dll.

```
pi@mulapi:~ $ sudo apt-get install libavcodec-dev libswscale-dev libv4l-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libavcodec-dev is already the newest version.
libswscale-dev is already the newest version.
libv4l-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
pi@mulapi:~ $
```

Gambar 5.21 Instalasi *image I/O packages*

5. *Video I/O packages*, *library* ini memungkinkan kita untuk membaca berbagai *format file video* dari disk serta bekerja secara langsung dengan *video stream.*

```
pi@mulapi:~ $ sudo apt-get install libxvidcore-dev libx264-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libx264-dev is already the newest version.
libxvidcore-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

Gambar 5.22 Instalasi *video I/O packages*

6. *OpenCV library* dilengkapi dengan sub-modul bernama *highgui* yang digunakan untuk menampilkan gambar ke layar kami dan membangun *basic GUI.* Dalam rangka untuk mengkompilasi modul *highgui*, kita perlu menginstal *GTK development library*.

```
pi@mulapi:~ $ sudo apt-get install libgtk2.0-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libgtk2.0-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

Gambar 5.23 *Install GTK development library*

7. Banyak operasi dalam *OpenCV* misalnya operasi matriks dapat dioptimalkan lebih lanjut dengan menginstal beberapa dependensi ekstra.

```
pi@mulapi:~ $ sudo apt-get install libatlas-base-dev gfortran
Reading package lists... Done
Building dependency tree
Reading state information... Done
gfortran is already the newest version.
libatlas-base-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

Gambar 5.24 *Install extra* dependensi

8. *Library* optimasi ini sangat penting untuk perangkat *raspberry pi*. Lalu menginstal kedua *Python 2.7* dan *Python 3 file header* dengan tujuan agar bisa mengkompilasi *OpenCV* dengan *binding Python*.

```
pi@mulapi:~ $ sudo apt-get install python2.7-dev  python3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python2.7-dev is already the newest version.
python3-dev is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
```

Gambar 5.25 *Install python* 2.7 dan *python* 3 *file header*

9. Setelah dependensi selesai diinstal, langkah selanjutnya mengunduh arsip *OpenCV* 3.1.0 dari repositori *OpenCV* resmi. Lalu mengekstraknya dengan perinta *unzip.*

```
pi@mulapi:~ $ cd ~

pi@mulapi:~ $ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0
zip

pi@mulapi:~ $ unzip opencv.zip

pi@mulapi:~ $ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contr
ib/archive/3.1.0.zip

pi@mulapi:~ $ unzip opencv_contrib.zip
```

Gambar 5.26 Mengunduh arsip *opencv-3.1.0* dan *opencv_contrib*

10. Sebelum mengkompilasi *python* terlebih dahulu menginstal *pip.*

```
pi@mulapi:~ $ wget https://bootstrap.pypa.io/get-pip.py

pi@mulapi:~ $ sudo python get-pip.py
```

Gambar 5.27 Instalasi *pip*

11. Berikutnya instalasi *virtualenv* dan *virtualenvwrapper*

```
pi@mulapi:~ $ sudo pip install virtualenv virtualenvwrapper

pi@mulapi:~ $ sudo rm -rf ~/.cache/ pip
```

Gambar 5.28 Instalasi *virtualenv* dan *virtualenvwrapper*

12. Setelah *virtualenv* dan *virtualenvwrapper* telah dipasang, kita perlu memperbarui file ~ / *.profile* untuk memasukkan baris berikut di bagian bawah file

```
pi@mulapi:~ $ ls -a .
.               Desktop          opencv_contrib-3.1.0     .themes
..              Documents        opencv_contrib.zip       .thumbnails
1t.jpg          Downloads        opencv_face_features.py  ts1t.jpg
.asoundrc       .gconf           opencv.zip               tst.jpg
.bash_history   get-pip.py       .oracle_jre_usage        Videos
.bash_logout    .gstreamer-0.10  Pictures                 .virtualenvs
.bashrc         .idlerc          .pki                     .vnc
.cache          .local           .profile                 webca,
cobacam.py      mula.jpg         Public                   webcam
coba.jpg        Music            python_games             .Xauthority
.config         oldconffiles     .python_history          .xsession-errors
.dbus           opencv-3.1.0     Templates                .xsession-errors.old
pi@mulapi:~ $ nano .profile
# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

#virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

Gambar 5.29 Mengedit *profiles*

13. Setelah *profiles* diperbarui selanjutnya kemudian melakukan *logoff* untuk memastikan perubahan yang dilakukan. Lalu masuk kembali dengan *log on.*

```
pi@mulapi:~ $ source ~/.profile
```

Gambar 5.30 *Log on profiles*

14. Berikutnya, membuat *virtual python environment* yang akan digunakan untuk pengembangan *computer vision.*

```
pi@mulapi:~ $ mkvirtualenv cv
New python executable in /home/pi/.virtualenvs/cv/bin/python
Installing setuptools, pip, wheel...done.
(cv) pi@mulapi:~ $
```

Gambar 5.31 Membuat *virtual python environment*

15. Setelah selesai membuat *virtual python environment* melanjutkan instalasi *numpy*.

```
(cv) pi@mulapi:~ $ pip install numpy
```

Gambar 5.32 Instalasi *numpy*

16. Setealah proses instalasi dependensi dan unsur pendukung lain selesai maka melanjutkan instalasi *OpenCV*, dipastikan berada di *virtual cv environment* saat melakukan instalasinya dengan menjalankan *cv* terlebih dahulu.

```
(cv) pi@mulapi:~ $ workon cv
(cv) pi@mulapi:~ $ cd ~/opencv-3.1.0
(cv) pi@mulapi:~/opencv-3.1.0 $ mkdir build
```

Gambar 5.33 Memulai instalasi *OpenCV*

17. Kemudian melakukan *build* dengan menggunakan *CMake*.

```
(cv) pi@mulapi:~ $ workon cv
(cv) pi@mulapi:~ $ cd ~/opencv-3.1.0
(cv) pi@mulapi:~/opencv-3.1.0 $ mkdir build
(cv) pi@mulapi:~/opencv-3.1.0 $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTA
LL_PREFIX=/usr/local -D INSTALL_PYTHON_EXAMPLES=ON -D OPENCV_EXTRA_MODULES_PATH=
~/opencv_contrib-3.1.0/modules -D BUILD_EXAMPLES=ON ..
```

Gambar 5.34 *Build* menggunakan *CMake*

18. Berikutnya melakukan kompilasi *OpenCV* yang memakan waktu lama, kurang lebih 1-2 jam tergantung dengan koneksi *internet*

```
(cv) pi@mulapi:~/opencv-3.1.0 $ sudo make -j4
(cv) pi@mulapi:~/opencv-3.1.0 $ make clean
(cv) pi@mulapi:~/opencv-3.1.0 $ make
(cv) pi@mulapi:~/opencv-3.1.0 $ sudo make install
(cv) pi@mulapi:~/opencv-3.1.0 $ sudo make install ldconfig
```

Gambar 5.35 Kompilasi *OpenCV*

19. Langkah akhir dalam proses instalasi *OpenCV*

```
pi@mulapi:~/opencv-3.1.0 $ cd ~
pi@mulapi:~ $ ls /usr/local/lib/python3.4/site-packages
cv2.cpython-34m.so
pi@mulapi:~ $
```

Gambar 5.36 Langkah akhir instalasi *OpenCV*

20. Memverifikasi bahwa instalasi *OpenCV* Anda bekerja dengan benar. Membuka terminal baru lalu menjalankan dan workon perintah, dan kemudian mencoba untuk mengimpor *python + OpenCV binding*.

```
pi@mulapi:~ $
pi@mulapi:~ $ source ~/.profile
pi@mulapi:~ $ workon cv
(cv) pi@mulapi:~ $ python3
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2. __version__
'3.1.0'
>>>
```

Gambar 5.37 Mengimpor *python + OpenCV binding*

5.1.4 Instalasi *FSWebcam*

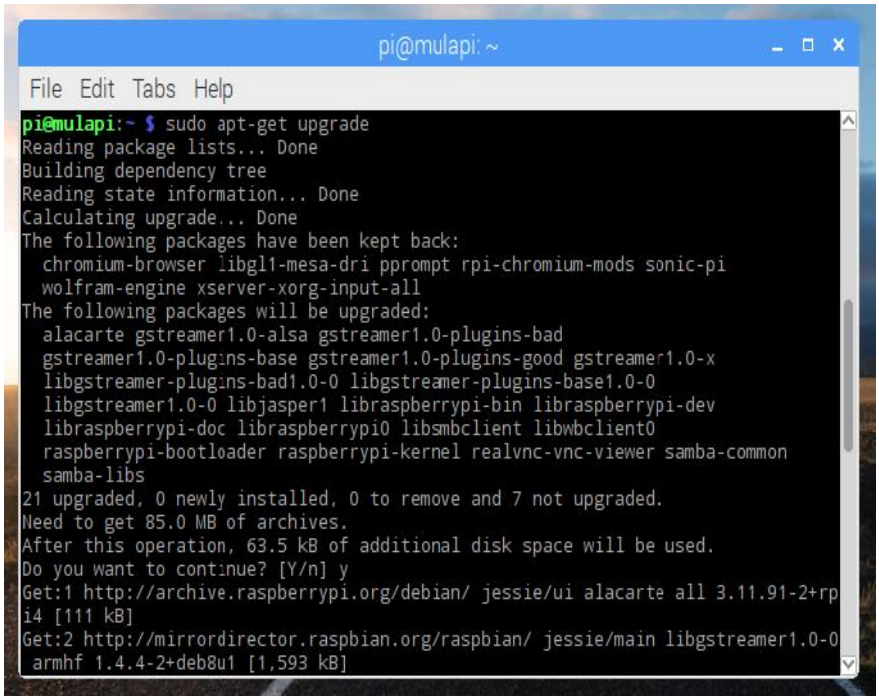
Pada proses instalasi aplikasi *fswebcam* atau instalasi aplikasi apapun di sistem operasi Linux direkomendasikan untuk melakukan *update* dan *upgrade* sistem agar terjadi pembaruan sistem yang dapat berintegrasi dengan aplikasi yang dipasang. Langkah-langkah dalam instalasi *fswebcam* dapat dilihat pada gambar 5.38 sampai dengan gambar 5.41 di bawah ini sebagai berikut:

1. Masuk pada terimal linux terlebih dahulu melakukan *update* sistem dengan mengetikkan perintah *$ sudo apt-get update*

```
pi@mulapi: ~
File  Edit  Tabs  Help
pi@mulapi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org jessie InRelease [22.9 kB]
Get:2 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
Get:3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,981 kB]
Get:4 http://archive.raspberrypi.org jessie/main armhf Packages [145 kB]
Get:5 http://archive.raspberrypi.org jessie/ui armhf Packages [57.9 kB]
Ign http://archive.raspberrypi.org jessie/main Translation-en_GB
Ign http://archive.raspberrypi.org jessie/main Translation-en
Ign http://archive.raspberrypi.org jessie/ui Translation-en_GB
Ign http://archive.raspberrypi.org jessie/ui Translation-en
Get:6 http://mirrordirector.raspbian.org jessie/contrib armhf Packages [37.5 kB]
Get:7 http://mirrordirector.raspbian.org jessie/non-free armhf Packages [70.3 kB
]
Get:8 http://mirrordirector.raspbian.org jessie/rpi armhf Packages [1,356 B]
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/contrib Translation-en
Ign http://mirrordirector.raspbian.org jessie/main Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/main Translation-en
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/non-free Translation-en
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org jessie/rpi Translation-en
Fetched 9,331 kB in 53s (174 kB/s)
Reading package lists... 98%
```

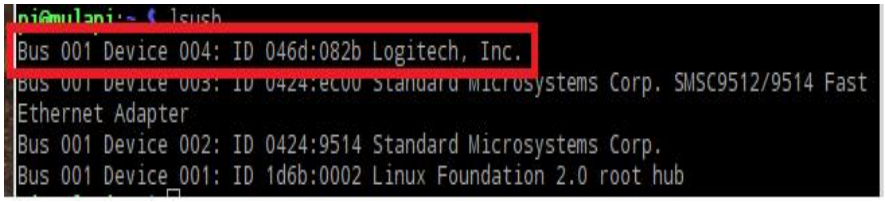
Gambar 5.38 *Update* sistem

2. Kemudian mengetikkan perintah *$ sudo apt-get upgrade*



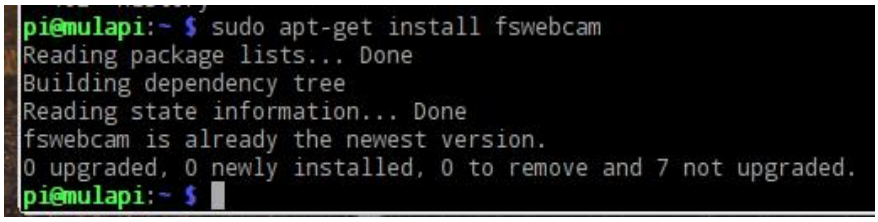
Gambar 5.39 *Upgrade* sistem

3. Setelah proses *update* dan *upgrade* selesai maka kita lihat apakah kamera yang akan dipake sudah terdeteksi pada sistem. Perintah yang diketikkan yaitu $ *lsusb*



Gambar 5.40 Mendeteksi kamera terhubung sistem

4. Jika kamera sudah terdeteksi maka melakukan instalasi aplikasi *fswebcam* dengan mengetikkan perintah *$ sudo apt-get install fswebcam*



Gambar 5.41 Instalasi *fswebcam*

5.1.5 Instalasi *Web Server*

Web server yang digunakan pada penelitian ini antara lain *apache2*, *MySQL* dan *PHP*. Adapun proses instalasinya dapat dilihat pada gambar 5.42 sampai dengan gambar 5.48 di bawah ini sebagai berikut:

1. Membuka *terminal linux* dan melakukan perintah *update* sistem

```
pi@mulapi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org jessie InRelease [22.9 kB]
Get:2 http://mirrordirector.raspbian.org jessie InRelease [14.9 kB]
Get:3 http://mirrordirector.raspbian.org jessie/main armhf Packages [8,981 kB]
Get:4 http://archive.raspberrypi.org jessie/main armhf Packages [145 kB]
```

Gambar 5.42 *Update* Sistem

2. Selanjutnya melakukan konfiguras sistem pada *apache2*

```
pi@mulapi:~ $ sudo apt-get install apache2
```

Gambar 5.43 Instalasi *apache2*

3. Setelah selesai melakukan konfigurasi *apache2* selanjutnya *restart apache2* agar terjadi perubahan setelah dilakukan konfigurasi

```
pi@mulapi:~ $ sudo systemctl restart apache2
```

Gambar 5.44 Konfigurasi *apache2*

4. Melihat daftar aplikasi *firewall* pada *apache2*

```
pi@mulapi:~ $ sudo ufw app list
```

Gambar 5.45 Lihat daftar aplikasi *firewall* pada *apache*

5. Melihat info aplikasi *firewall* pada *apache2* secara lengkap

```
pi@mulapi:~ $ sudo ufw app info "Apache Full"
```

Gambar 5.46 Lihat *apache full*

6. Melakukan instalasi *mysql-server*

```
pi@mulapi:~ $ sudo apt-get install mysql-server
```

Gambar 5.47 Instalasi *mysql-server*

7. Melakukan instalasi *php*

```
pi@mulapi:~ $ sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

Gambar 5.48 Instalasi *PHP*

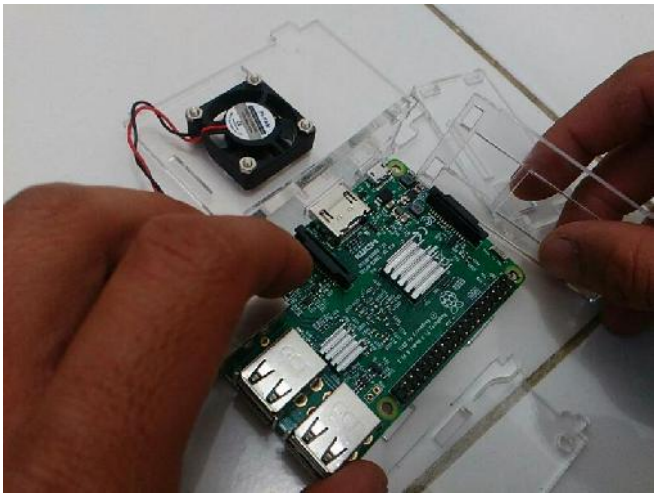
5.2 Instalasi Perangkat Keras

Proses instalasi perangkat keras pada penelitian ini terdiri dari beberapa bagian. Proses instalasi perangkat keras yang akan digunakan dalam penelitian ini akan dijelaskan masing-masing bagian. Adapun beberapa proses instalasi perangkat keras yang akan digunakan adalah sebagai berikut:

1. Instalasi *raspberry pi* pada *case*
2. Intsalasi *usb webcam*
3. Instalasi *rainbow cable*
4. Instalasi *GPIO adapter expansion board*
5. Instalasi *servo motor*
6. Instalasi seluruh peralatan ke kotak simulasi

5.2.1 Instalasi *Raspberry pi* pada *Case*

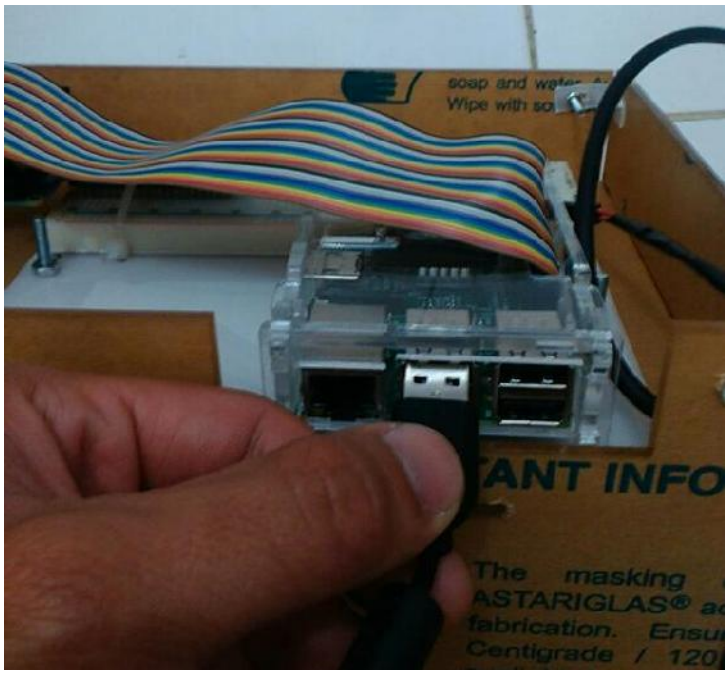
Banyak jenis dan pilihan *case* untuk *raspberry pi* dengan tujuan untuk melindungi perangkat dari kemungkinan buruk yang bisa terjadi, seperti tersiram air, kontak langsung dengan perangkat elektornik atau logam yang dapat menyebabkan korsleting sampai melindungi perangkat agar tetap bersih. Bahkan *case official raspberry pi* tersedia yang berbentuk presisi dengan perangkat. Berikut proses instalasi perangkat *raspberry pi* ke dalam *acrylic case for raspberry pi* 3 model B yang dilengkapi dengan *pi fan* dapat dilihat pada gambar 5.49 di bawah ini.



Gambar 5.49 Instalasi *raspberry pi* pada *acrylic case*

5.2.2 Instalasi *USB Webcam*

Dalam penelitian ini, peneliti menggunakan *USB Webcam* yang digunakan untuk menangkap gambar wajah sebagai objek penlitian dengan merk Logitech model C170 yang memiliki spesifikasi 5*megapixel* dengan resolusi hingga 1024 x 768 *pixel.* Spesifikasi dari kamera yang dipilih sudah memenuhi spesifikasi ayng dibutuhkan dalam pemelitian. Instalasi perangkat kamera dapat dilakukan dengan mudah yaitu cukup menghubungkan kabel *USB* pada kamera ke salah *USB port* pada *raspberry pi.* Berikut merupakan proses instalasi *USB Camera Logitech C170* pada *port USB* seperti yang ditunjukkan pada gambar 5.50 di bawah ini.



Gambar 5.50 Instalasi *USB camera*

5.2.3 Instalasi *Rainbow Cable*

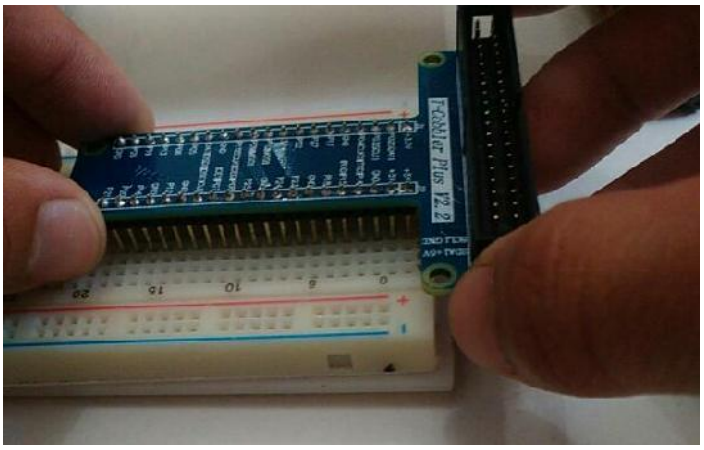
Rainbow Cable pada penelitian digunakan untuk menghubungkan antara *GPIO 40 PIN* pada *raspberry pi* dengan *GPIO breadboard* dan berfungsi sebagai pengganti kabel *jumper.* Hal ini juga berfungsi agar *GPIO PIN* pada *raspberry pi* tetap awet karena tidak terlalu sering mencabut kabel *jumper* dari *GPIO pin raspberry pi* secara langsung melainkan cukup mencabut atau memasang kabel *jumper* pada *GPIO breadboard.* Tetapi cara ini membutuhkan peralatan lagi yaitu

GPIO adapter expansion board. Proses instalasinya dapat dilihat pada gambar 5.51 di bawah ini.

Gambar 5.51 Instalasi *Rainbow Cable*

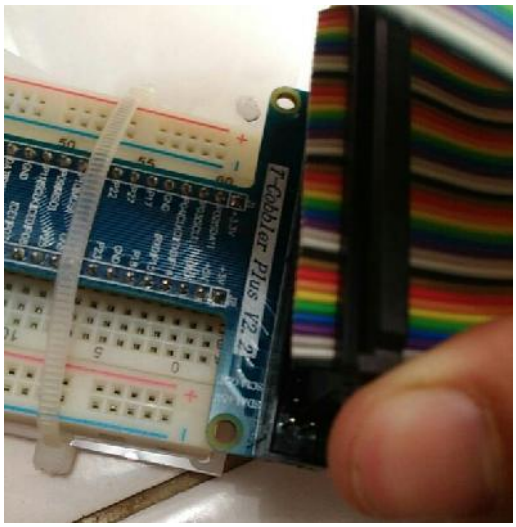
5.2.4 Instalasi *GPIO Adapter Expansion Board*

Setelah *rainbow cable* terpasang pada *raspberty pi* maka dilanjutkan dengan memasang *rainbow cable* pada *GPIO adapter expansion board*. Perangkat ini difungsikan sebagai konektor antara *rainbow cable* dengan *GPIO breadboard.* Instalasi dari perangkat *rainbow cable* dan *GPIO breadboard* dengan *GPIO adapter expansion board* sebagai konektor dapat dilihat pada gambar 5.52 di bawah ini.



Gambar 5.52 Instalasi *GPIO Adapter Expansion Board*

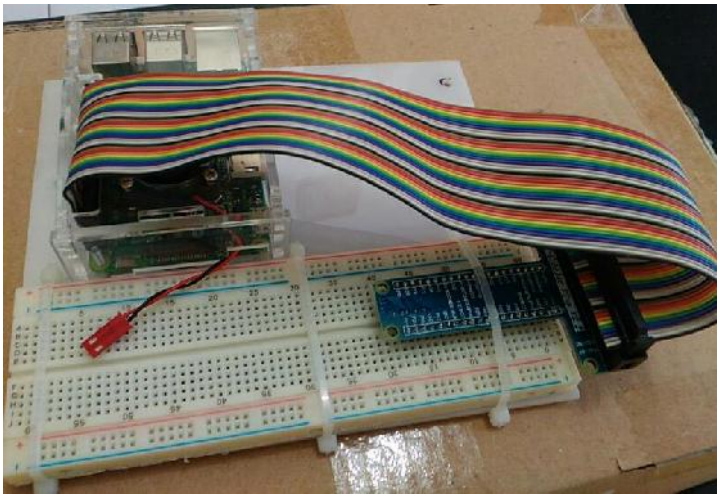
Berikut merupakan proses instalasi *rainbow cable* ke *GPIO adapter expansion board* yang dapat dilihat pada gambar 5.46 di bawah ini.



Gambar 5.53 Instalasi *Rainbow Cable* pada *Adapter Expansion*

5.2.5 Instalasi Seluruh Perngkat ke Kotak Simulasi

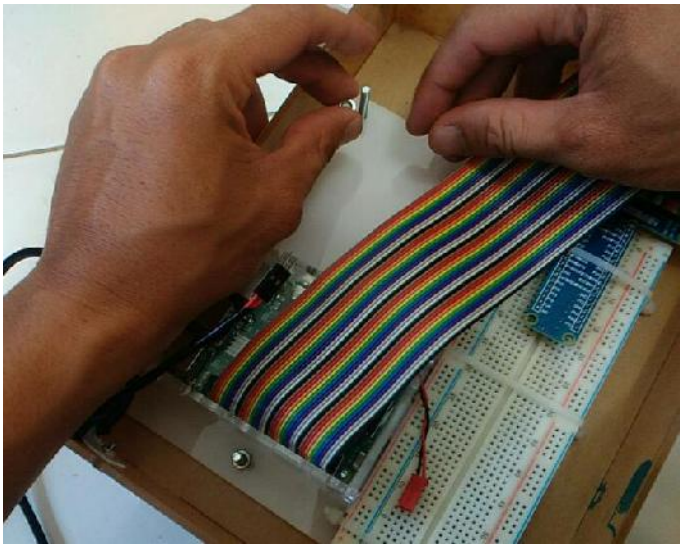
Setelah semua perangkat terpasang, yaitu meliputi perangkat *raspberry pi, rainbow cable, adapter expansion, GPIO breadboard* kemudian menyatukannya pada sebuah papan sebelum dipasang pada kotak simulasi. Hal itu bertujuan agar proses konfigurasi perangkat lebih mudah dilakukan sebelum dipasang pada kotak simulasi. Pemasangan alat pada selembar papan dapat dilihat pada gambar 5.54 di bawah ini.



Gambar 5.54 Instalasi Semua Perangkat ke Papan Komponen

5.2.6 Instalasi Semua Perangkat ke Dalam Kotak Simulasi

Ketika semua perangkat pada papan sudah terpasang, kemudian akan dipasang ke dalam kotak simulasi bagian bawah. Dengan 3 mur dan baut sebagai pengait antara papan peralatan dan kotak simulasi. Proses instalasi dapat dilihat pada gambar 5.55 di bawah ini.



Gambar 5.55 Instalasi Seluruh Perangkat ke Kotak Simulasi

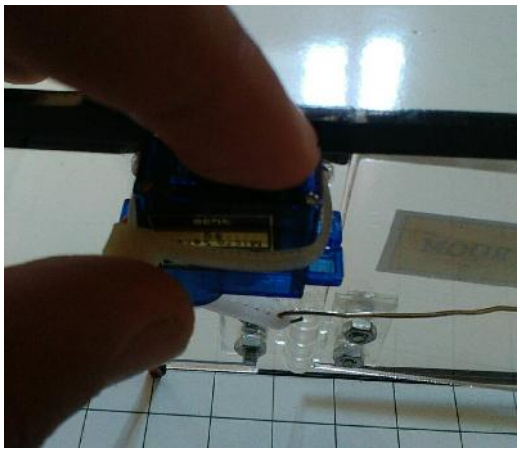
5.2.7 Instalasi *Servo Motor*

Setelah Semua perangkat terpasang ke dalam kotak simulasi, langkah instalasi terakhir yaitu memasang *servo motor* pada pintu kotak simulasi. *Servo motor* yang terpasang berfungsi untuk menggerakkan daun pintu setelah menerima perintah dari *raspberry pi* sebagai hasil dari pengenalan wajah oleh kamera.

Servo motor yang digunakan pada peneitian ini adalah *Tower Pro micro servo 99 model SG90.* Spesifikasi dari Tower Pro *micro servo* model SG90 adalah sebagai berikut:

1. Torsi maksimum / *stall torque* = 1,2 kg.cm
2. Rentang sudut putaran 180°
3. Catu daya operasional = 4 ~ 7,2 *Volt DC*
4. Kecepatan pada 4,8 *VDC* tanpa beban = 0,12 detik per 60° (57,6 *rpm*)
5. Kecepatan operasi pada 4.8V = 0.12 *sec*/60 *degrees*

Proses instalasi *servo motor* pada kotak simulasi untuk menggerakkan daun pintu dapat dilihat pada gambar 5.56 di bawah ini.



Gambar 5.56 Instalasi *Servo Motor*

5.3 Perakitan Kotak Simulasi Ruang *Server*

Setelah instalasi perangkat keras maupun perangkat lunak yang akan digunakan dalam penelitian ini selesai disiapkan, tahap yang tidak kalah penting yaitu tahap perakitan kotak simulasi yang dibuat sebagai simulasi ruang *server* dan sebagai tempat dimana semua komponen perangkat keras terpasang.

Adapun hasil dari perakitan kotak simulasi dapat dilihat pada gambar 5.57 di bawah ini.

Gambar 5.57 Hasil perakitan kotak simulasi

5.4 Implementasi Perangkat Keras

Dalam penelitian ini, perangkat keras diimplementasikan sesuai dengan skema konfigurasi yang telah dirancang. Perangkat *raspberry pi, camera* dan perangkat keras pendukung yang digunakan pada penelitian ini dihubungkan dengan *laptop* dan melakukan *remote desktop connection* dari *laptop.* Perangkat pendukung yang lain yaitu penyangga dagu dan *tripod* kamera yang digunakan dengan tujuan mendapatkan keseragaman citra yang akan diolah. Adapun konfigurasi perangkat keras pada implementasinya dapat dilihat pada gambar 5.58 di bawah ini.

Gambar 5.58 Implementasi perangkat keras

5.5 Implementasi Perangkat Lunak

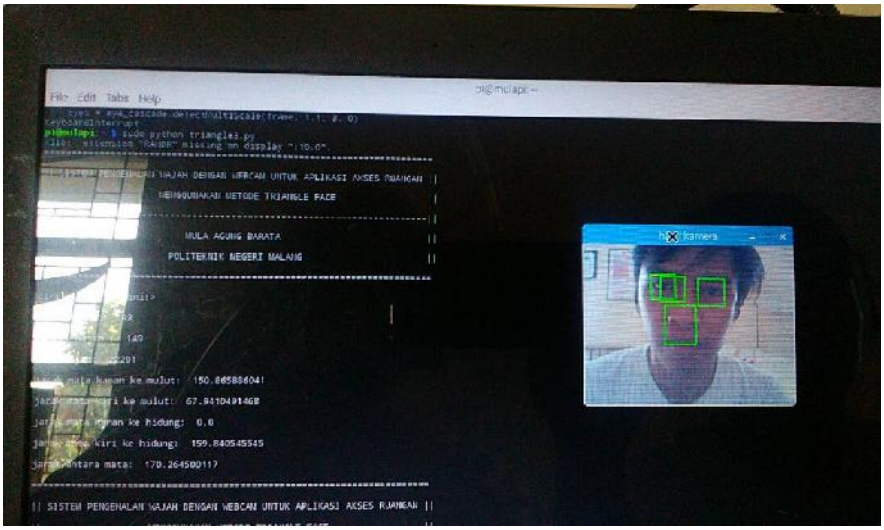
Implementasi perangkat lunak dalam pengembangan sistem ini, aplikasi dikembangkan dengan integrasi perangkat keras dengan pengolahan citra menggunakan *terminal linux* yang berbasis *command line* dengan bahasa pemograman *python.*

Adapun implementasi penggunaan perangkat lunak pada penelitian ini adalah sebagai berikut:

1. Jika *user* menekan tombol *Enter Room* pada *switch* maka akan mengeksekusi sistem yang ada pada aplikasi yang digunakan untuk mendeteksi serta mengenali wajah yang secara otomatis berintegrasi

dengan kamera yang dihubungkan dengan *raspberry pi.* Hasil pengukuran jarak antar fitur wajah akan diambil sebanyak 10x lalu disimpan ke dalam *database.*

2. Data yang telah disimpan akan digunakan sebagai data pembanding dengan citra wajah yang dikenali untuk membuka pintu ruang *server.* Jika data hasil pengukuran cocok dengan yang ada pada *database* maka pintu ruang akan terbuka otomatis. Hasil pengukuran jarak antar fitur wajah dapat dilihat pada gambar 5.59 di bawah ini.



Gambar 5.59 Pendeteksian dan pengukuran wajah

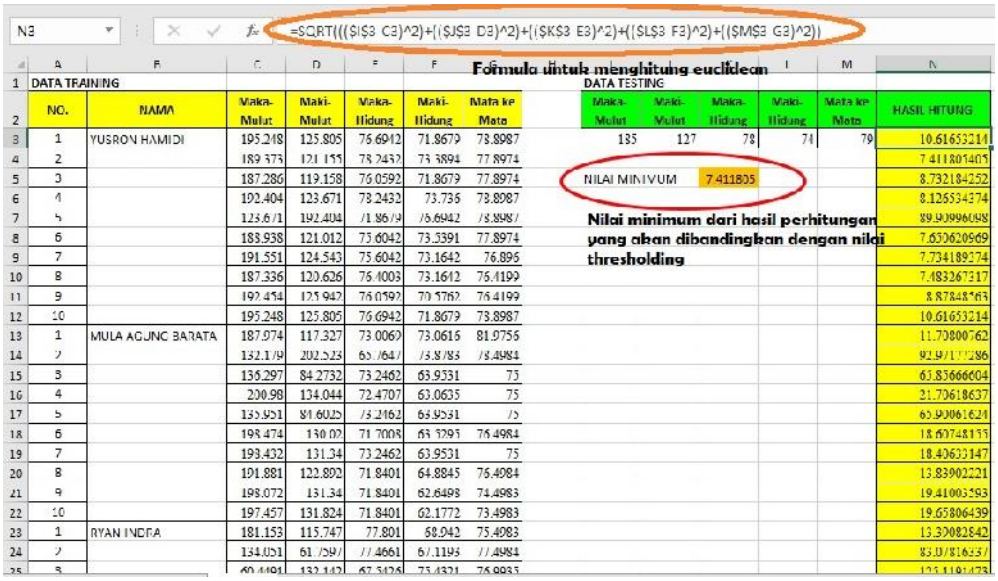
5.6 Pencocokan Hasil Pengukuran Nilai Citra Wajah dengan *Database*

Dalam sub bab ini dijelaskan bagaimana perhitungan sistem dalam melakukan pencocokan hasil pengukuran citra wajah dengan data hasil pengukuran citra wajah yang sebelumnya sudah pernah diambil. Langkah yang dilakukan adalah sebagai berikut:

1. Setiap kali melakukan *testing* pada sistem, masing-masing nilai fitur wajah yang ada dalam data *testing* dikurangi dengan masing-masing nilai fitur wajah yang dalam data *training*.
2. Hasil pengurangan pada masing-masing fitur wajah dikuadratkan.
3. Menjumlahkan seluruh hasil pengurangan data *testing* dan data *training.*
4. Hasil penjumlahan dari seluruh nilai fitur kemudian diakar.

5. Setelah mendapatkan hasil dari operasi akar, lalu menentukan nilai minimum atau terkecil dari semua nilai yang telah diolah dan membandingkannya dengan nilai *thresholding* pada aplikasi.

Untuk memudahkan dalam pemahaman mengenai tahapan pencocokan hasil pengukuran citra wajah antara data *training* dan data *testing*, penulis menjelaskan dengan contoh data hasil pengukuran yang diolah menggunakan *Microsoft office excel.* Gambar 5.60 di bawah ini menjelaskan mengenai pencocokan data *training* dan *testing* dalam perhitungan hasil pengukuran citra wajah pada *Microsoft offce excel.*



Gambar 5.60 Pencocokan Kemiripan Data *Training* dan *Testing*

Sesuai dengan langkah yang sudah dijelaskan sebelumnya, data *testing* yang baru ambil kemudian diolah dengan data *training* yang sebelumnya sudah disimpan pada *database.* Formula yang digunakan dalam menghitung *Euclidean* sudah ditunjukkan pada gambar 5.60 di atas dan menghasilkan nilai yang kemudian dicari nilai terkecilnya untuk dibandingkan dengan nilai *thresholding.*

BAB VI PENGUJIAN DAN PEMBAHASAN

Sebagai cara untuk mengetahui kinerja dari sistem yang dirancang dan dibuat, maka peru dilakukan pengujian aplikasi yang meliputi pengujian perangkat keras, pendeteksian wajah, pengujian tahap perhitungan jarak antar fitur wajah dan pembahasan keseluruhan sistem.

6.1 Pengujian Fungsional

Pengujian ini dimaksudkan untuk menguji kinerja dari perangkat keras dan perangkat lunak yang digunakan dalam penelitian. Pengujian ini bersifat fungsional yaitu melakukan pengujian masing-masing fungsi perangkat keras dan integrasinya dengan perangkat lunak yang digunakan. Adapun hasil dari pengujian fungsional dapat dilihat pada table 6.1 di bawah ini.

Tabel 6.1 Pengujian Fungsional

No.	Nama Pengujian	Hasil Pengujian
1.	Pengujian sistem operasi *minicomputer raspberry pi*	Berhasil
2.	*OpenCV* pada *raspberry pi*	Berhasil
3.	Koneksi *raspberry pi* dengan *laptop*	Berhasil
4.	Koneksi *raspberry pi* dengan *web camera*	Berhasil
5.	*Rainbow cable, adapter expansion* dan *breadboard*	Berhasil
6.	*Koneksi Servo motor*	Berhasil
7.	*Switch* (Tombol *register user* dan *enter room*)	Berhasil
8.	Koneksi *database* dengan aplikasi	Berhasil
9.	Pendeteksian wajah dan fitur wajah	Berhasil
10.	Pengenalan citra wajah	Berhasil

6.2 Pengujian Pendeteksian Fitur Wajah

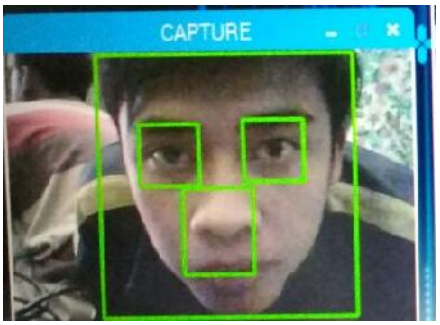
Dalam tahapan ini merupakan tahapan pengujian aplikasi dalam mendeteksi fitur wajah yang akan dihitung nilai jarak antar mata, hidung dan mulut. Adapun peralatan yang digunakan dalam tahap pengujian ini adalah sebagai berikut:

1. *Personal computer / laptop*
2. Rangkaian perangkat *minicomputer raspberry pi*

3. *Web camera*
4. Penyangga dagu
5. *Tripod* kamera

Langkah yang dilakukan dalam tahapan ini adalah menghubungkan seluruh perangkat keras yang digunakan dan menjalankannya. Jika seluruh perangkat siap digunakan kemudian memasang kamera pada *tripod* kamera dan menata penyangga dagu lurus dengan *tripod* kamera sesuai jarak yang sudah ditentukan yaitu 40 cm. Menjalankan aplikasi sistem keamanan pintu ruang *server* dengan menekan tombol *Register User* untuk mendaftarkan *user* baru untuk mendapatkan hak akses ruang dan memilih tombol *EnterRoom* untuk mengakses ruangan. *Register User* ini merupakan tahapan dalam mengambil *data training* yang akan disimpan pada *database,* lalu *EnterRoom* merupakan tahapan pengenalan citra wajah yang akan dicocokkan dengan citra yang sudah disimpan pada *database.* Hal ini merupakan proses klasifikasi dan pencocokan antara *data training* dengan *data testing* apakah sistem yang dibuat sudah akurat dan sesuai dengan rancangan yang sudah direncanakan. Setelah menjalankan aplikasi dan aplikasi sudah mendeteksi keberadaan wajah beserta fitur wajah seperti mata, hidung dan mulut maka kita tekan tombol spasi pada *keyboard,* secara otomatis sistem akan menghitung jarak antar fitur wajah pada citra wajah yang telah dideteksi. Secara otomatis hasil dari pengukuran jarak antar fitur wajah disimpan ke dalam *database.*

Dalam pengujian dalam tahap ini diperoleh hasil seperti yang ditunjukkan pada gambar 6.1 di bawah ini yang mana seluruh fitur wajah seperti mata, hidung dan mulut sudah terdeteksi dengan ditandai dengan kotak mata, kotak hidung dan mulut.



Gambar 6.1 Pendeteksian Fitur Wajah

6.3 Pengujian Pengukuran Jarak Antar Fitur Wajah

Tahap ini difungsikan untuk mengetahui seberapa sukses perangkat lunak ini dalam mengukur jarak antar fitur yang telah diidentifikasi. Adapun peralatan ang digunakan dalam pengujian ini adalah :

1. *Personal computer / laptop*
2. Rangkaian perangkat *minicomputer raspberry pi*
3. *Web camera*
4. Penyangga dagu
5. *Tripod* kamera

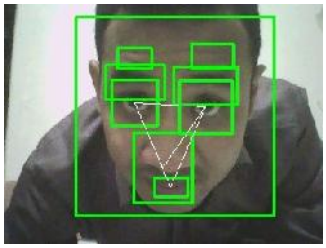
Hasil pengujian jarak antar fitur wajah ini ditampilkan pada gambar 6.2 di bawah ini.

Gambar 6.2 Hasil Pengukuran Jarak Antar Fitur Wajah

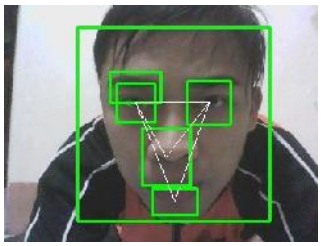
Pengambilan citra wajah oleh kamera dengan jarak pengambilan 40 cm antar kamera dan objek yang diambil akan menghasilkan ukuran jarak antar fitur wajah ini juga dilakukan selama 10 kali dengan jumlah 5 orang yang diambil citra wajahnya. Dalam pencocokan hasil pengukuran citra antara *data training* dan *data*

testing menggunakan nilai *thresholding* sebesar 230 *pixel*. Nilai *thresholding* sangat berpengaruh dengan akurasi sistem yang dijalanan, semakin kecil nilai *thresholding* yang digunakan maka semakin akurat hasil dari pencocokan sistem yang membandingkan antara *data training* dan *data testing*, tetapi jika nilai *thresholding* yang digunakan terkecil maka sistem akan mengalami kesalahan dalam penncocokan nilai ukuran fitur wajah. Sistem tidak mengenali wajah yang dideteksi dari proses *testing* sistem meskipun data *training* dari wajah yang dideteksi sudah diambil karena nilai *thresholding* yang terlalu kecil. Adapun hasil pengukuran citra wajah pada masing-masing citra wajah yang diambil dengan jarak pengambilan citra wajah dan nilai *thresholding* yang telah ditentukan di atas adalah sebagai berikut:

Tabel 6.2 Hasil Pengukuran Wajah Yusron Hamidi

Nama	**Ma-Ka Mulut**	**Ma-Ki Mulut**	**Ma-Ka Hidung**	**Ma-Ki Hidung**	**Mata ke Mata**
Yusron Hamidi 	195.248	125.805	76.6942	71.8679	78.8987
	189.373	121.155	78.2432	73.3894	77.8974
	187.286	119.158	76.0592	71.8679	77.8974
	192.404	123.671	78.2432	73.736	78.8987
	123.671	192.404	71.8679	76.6942	78.8987
	188.938	121.012	75.6042	73.5391	77.8974
	191.551	124.543	75.6042	73.1642	76.896
	187.336	120.626	76.4003	73.1642	76.4199
	192.454	125.942	76.0592	70.5762	76.4199
	195.248	125.805	76.6942	71.8679	78.8987

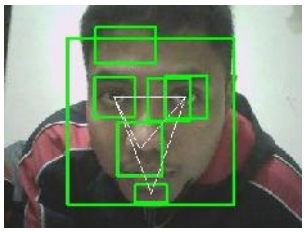
Tabel 6.3 Hasil Pengukuran Wajah Mula Agung Barata

Nama	Ma-Ka Mulut	Ma-Ki Mulut	Ma-Ka Hidung	Ma-Ki Hidung	Mata ke Mata
Mula Agung Barata 	187.974	117.327	73.0069	73.0616	81.9756
	132.179	202.523	65.7647	73.8783	78.4984
	136.297	84.2732	73.2462	63.9531	75
	200.98	134.044	72.4707	63.0635	75
	135.951	84.6025	73.2462	63.9531	75
	198.474	130.02	71.7008	63.5295	76.4984
	198.432	131.34	73.2462	63.9531	75
	191.881	122.892	71.8401	64.8845	76.4984
	198.072	131.34	71.8401	62.6498	74.4983
	197.457	131.824	71.8401	62.1772	73.4983

Tabel 6.4 Hasil Pengukuran Wajah Ryan Indra

Nama	Ma-Ka Mulut	Ma-Ki Mulut	Ma-Ka Hidung	Ma-Ki Hidung	Mata ke Mata
Ryan Indra	181.153	115.747	77.801	68.942	75.4983
	134.051	61.7597	77.4661	67.1193	77.4984
	60.4491	132.142	67.5426	75.4321	76.9935
	164.007	95.4531	77.4661	66.2118	77
	91.7232	123.889	66.2118	75.2861	76.9935
	179.343	112.131	76.8375	67.1193	76.4984
	179.097	111.791	74.8131	67.9779	76.9935
	82.7242	122.063	66.6483	74.2024	76
	87.4741	122.798	65.7419	74.6525	76.4984
	121.24	83.2565	74.8131	67.5352	77.4984

Tabel 6.5 Hasil Pengukuran Afnan Afif

Nama	Ma-Ka Mulut	Ma-Ki Mulut	Ma-Ka Hidung	Ma-Ki Hidung	Mata ke Mata
Afnan Afif 	169.103	102.508	73	61.7171	76
	170.404	102.508	72.9178	61.7171	77.4984
	174.401	106.755	74.411	62.6099	77
	180.201	113.924	76.2168	66.1891	76.4984
	177.003	108.596	76.8375	66.6483	78.4984
	177.337	109.778	76.8375	65.7647	77.9936
	178.306	109.222	76.8375	67.1193	79.4984
	178.598	111.11	77.0065	67.1193	77.9936
	178.598	111.746	77.0065	66.242	77.4855
	184.397	117.645	77.0065	67.1193	77.9936

6.4 Pengujian Aplikasi Pengenalan Wajah

Pengujian ini dimaksudkan untuk menguji perangkat lunak tahap pengenalan wajah dan untuk mengetahui seberapa akurat metode yang digunakan pada pengembangan sistem ini. Adapun peralatan ang digunakan dalam pengujian ini adalah :

1. *Personal computer / laptop*
2. Rangkaian perangkat *minicomputer raspberry pi*
3. *Web camera*
4. Penyangga dagu
5. *Tripod* kamera

Adapun hasil dari pengujian aplikasi yang menunjukkan kebenaran dan kesalahan sistem dalam mengenali wajah dengan jarak pengambilan citra wajah 40 cm yang dibandingkan dengan pengamatan manual menurut data yang dimasukkan dengan penentuan nilai *thresholding* 230 *pixel* pada proses *testing*

dari hasil *training* dengan jumlah 4 *user* dan 5 kali jumlah pengujian untuk setiap *user* dapat dilihat pada tabel 6.6 berikut ini.

Tabel 6.6 Hasil Pengujian Aplikasi

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Afnan Afif	Salah
3.	Ryan Indra	Ryan Indra	Benar
4.	Ryan Indra	Afnan Afif	Salah
5.	Ryan Indra	Ryan Indra	Benar
6.	Yusron Hamidi	Tidak dikenali	Salah
7.	Yusron Hamidi	Yusron Hamidi	Benar
8.	Yusron Hamidi	Yusron Hamidi	Benar
9.	Yusron Hamidi	Yusron Hamidi	Benar
10.	Yusron Hamidi	Yusron Hamidi	Benar
11.	Mula Agung Barata	Mula Agung Barata	Benar
12.	Mula Agung Barata	Afnan Afif	Salah
13.	Mula Agung Barata	Mula Agung Barata	Benar
14..	Mula Agung Barata	Mula Agung Barata	Benar
15.	Mula Agung Barata	Afnan Afif	Salah
16.	Afnan Afif	Afnan Afif	Benar
17.	Afnan Afif	Afnan Afif	Benar
18.	Afnan Afif	Afnan Afif	Benar
19.	Afnan Afif	Afnan Afif	Benar
20.	Afnan Afif	Afnan Afif	Benar

Hasil pengujian dari aplikasi yang dibuat dapat dihitung tingkat keakuratannya dalam prosentase dengan dimasukkan pada persamaan berikut ini:

$$\text{Keakuratan} = \frac{\textit{Hasil Benar}}{\textit{Banyaknya Pengujian}} \times 100\%$$

$$\text{Keakuratan} = \frac{15}{20} \times 100\%$$

Keakuratan = 75%

Kesalahan = 100% - Keakuratan

Kesalahan = 100% - 75%

Kesalahan = 25%

6.5 Variasi Pengujian Aplikasi Pengenalan Wajah

Setelah melakukan pengujian sistem yang telah dibuat dengan menentukan jarak pengambilan citra wajah 40 cm dan nilai *thresholding* 230 *pixel*, pada bagian ini ditunjukkan hasil dari beberapa variasi pengujian aplikasi pengenalan wajah dengan penentuan jarak pengambilan citra wajah yang berbeda serta penetuan nilai *thresholding* yang berbeda dari nilai *thresholding* yang digunakan sebelumnya. Sehingga pengujian dilakukan dengan beberapa variasi jarak pengambilan citra wajah dan nila *thresholding* sebesar 230 *pixel*, kedua pengujian dilakukan dengan jarak pengambilan citra wajah 40 cm dan pengujian dengan beberapa variasi nilai *thresholding.*

6.5.1 Pengujian dengan Variasi Jarak Pengambilan Citra Wajah

Pengujian aplikasi ini diuji dengan beberapa variasi jarak pengambilan citra wajah yang ditunjukkan pada masing-masing tabel. Bentuk pengujian sama dengan pengujian sebelumnya. Sistem diuji dengan beberapa variasi jarak pengambilan dan nilai *thresholding* sebesar 230 *pixel*. Dari hasil pengujian dengan beberapa variasi jarak pengambilan citra wajah dapat diambil kesimpulan mengenai tingkat keakurasian sistem yang dibuat. Sehingga dapat mengetahui berapa jarak yang mendekati titik optimum yang kemudian dihitung nilai jarak Euclidean dari masing-masing data hasil pengukuran jarak antar fitur wajah antara

data *testing* dan data *training.* Adapun hasil dari pengujian dengan variasi jarak pengambilan citra wajah dapat dilihat pada tabel 6.7 di bawah ini.

Tabel 6.7 Hasil pengujian dengan jarak pengambilan citra wajah 30 cm

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Afnan Afif	Salah
3	Ryan Indra	Ryan Indra	Benar
4	Ryan Indra	Afnan Afif	Salah
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Ryan Indra	Salah
7	Yusron Hamidi	Yusron Hamidi	Benar
8	Yusron Hamidi	Mula Agung Barata	Salah
9	Yusron Hamidi	Ryan Indra	Salah
10	Yusron Hamidi	Ryan Indra	Salah
11	Mula Agung Barata	Mula Agung Barata	Benar
12	Mula Agung Barata	Ryan Indra	Salah
13	Mula Agung Barata	Ryan Indra	Salah
14	Mula Agung Barata	Mula Agung Barata	Benar
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Ryan Indra	Salah
17	Afnan Afif	Rya Indra	Salah
18	Afnan Afif	Afnan Afif	Benar
19	Afnan Afif	Afnan Afif	Benar
20	Afnan Afif	Afnan Afif	Benar

Dari hasil di atas dapat disimpulkan bahwa wajah wajah milik Ryan Indra yang memiliki ukuran paling besar yang sering muncul dalam variasi jarak pengambilan citra wajah 30 cm karena masing-masing data memiliki nilai yang lebih besar dari variasi jarak pengambilan citra wajah 40 cm.

Selanjutnya untuk pengujian aplikasi dengan variasi jarak pengambilan citra wajah dengan jarak 50 cm dana nilai *thresholding* yang sama dengan nilai sebelumnya ditunjukkan pada tabel 6.8 di bawah ini.

Tabel 6.8 Hasil pengujian dengan jarak pengambilan citra wajah 50 cm

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Afnan Afif	Salah
3	Ryan Indra	Ryan Indra	Benar
4	Ryan Indra	Afnan Afif	Salah
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Ryan Indra	Salah
7	Yusron Hamidi	Yusron Hamidi	Benar
8	Yusron Hamidi	Mula Agung Barata	Salah
9	Yusron Hamidi	Ryan Indra	Salah
10	Yusron Hamidi	Ryan Indra	Salah
11	Mula Agung Barata	Mula Agung Barata	Benar
12	Mula Agung Barata	Ryan Indra	Salah
13	Mula Agung Barata	Ryan Indra	Salah
14	Mula Agung Barata	Mula Agung Barata	Benar
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Ryan Indra	Salah
17	Afnan Afif	Rya Indra	Salah
18	Afnan Afif	Afnan Afif	Benar
19	Afnan Afif	Afnan Afif	Benar
20	Afnan Afif	Afnan Afif	Benar

Dari hasil pengujian di atas dapat disimpulkan bahwa prosentase kebenaran sistem dalam pengenalan citra wajah adalah 50%. Kesalahan sistem dalam pengenelan citra wajah adalah 50% dari jarak pengambilan citra wajah 50 cm.

Berikutnya untuk pengujian aplikasi dengan variasi jarak pengambilan citra wajah dengan jarak 60 cm dana nilai *thresholding* yang sama dengan nilai sebelumnya yang ditunjukkan pada tabel 6.9 di bawah ini.

Tabel 6.9 Hasil pengujian dengan jarak pengambilan citra wajah 60 cm

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Tidak dikenali	Tidak dikenali
3	Ryan Indra	Tidak dikenali	Tidak dikenali
4	Ryan Indra	Tidak dikenali	Tidak dikenali
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Tidak dikenali	Tidak dikenali
7	Yusron Hamidi	Tidak dikenali	Tidak dikenali
8	Yusron Hamidi	Tidak dikenali	Tidak dikenali
9	Yusron Hamidi	Tidak dikenali	Tidak dikenali
10	Yusron Hamidi	Ryan Indra	Salah
11	Mula Agung Barata	Tidak dikenali	Tidak dikenali
12	Mula Agung Barata	Tidak dikenali	Tidak dikenali
13	Mula Agung Barata	Ryan Indra	Salah
14	Mula Agung Barata	Tidak dikenali	Tidak dikenali
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Tidak dikenali	Tidak dikenali
17	Afnan Afif	Rya Indra	Salah
18	Afnan Afif	Tidak dikenali	Tidak dikenali
19	Afnan Afif	Tidak dikenali	Tidak dikenali
20	Afnan Afif	Tidak dikenali	Tidak dikenali

Dari hasil pengujian di atas dapat dilihat bahwa sistem hanya 2 kali melakukan kebenaran dalam pengenalan wajah citra wajah, 4 kali melakukan kesalahan pengenalan citra wajah dan 14 kali sistem tidak dapat mengenali citra wajah. Dapat disimpulkan bahwa jarak 60 cm dalam pengambilan citra wajah memiliki tingkat keakuratan yang rendah.

6.5.2 Pengujian Aplikasi dengan Variasi Nilai *Thresholding*

Pengujian aplikasi dengan variasi nilai *thresholding* di sini dilakukan untuk mencari nilai *thresholding* yang paling optimal untuk membandingkan antara data *training* dan data *testing* dalam pengembangan sistem yang dibuat. Jarak pengambilan citra wajah yang ditentukan adalah 40 cm sesuai dengan jarak pengambilan wajah dalam pengujian awal.

Tabel 6.10 Hasil pengujian aplikasi dengan nilai *thresholding* 190 *pixel*

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Ryan Indra	Benar
3	Ryan Indra	Afnan Afif	Salah
4	Ryan Indra	Tidak dikenali	Tidak dikenali
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Mula Agung Barata	Salah
7	Yusron Hamidi	Tidak dikenali	Tidak dikenali
8	Yusron Hamidi	Yusron Hamidi	Tidak dikenali
9	Yusron Hamidi	Tidak dikenali	Tidak dikenali
10	Yusron Hamidi	Yusron Hamidi	Salah
11	Mula Agung Barata	Tidak dikenali	Tidak dikenali
12	Mula Agung Barata	Tidak dikenali	Tidak dikenali
13	Mula Agung Barata	Yusron Hamidi	Salah
14	Mula Agung Barata	Tidak dikenali	Tidak dikenali
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Tidak dikenali	Tidak dikenali
17	Afnan Afif	Rya Indra	Salah
18	Afnan Afif	Tidak dikenali	Tidak dikenali
19	Afnan Afif	Tidak dikenali	Tidak dikenali
20	Afnan Afif	Tidak dikenali	Tidak dikenali

Dari hasil pengujian di atas dengan viariasi nilai *thresholding* 190 *pixel* adalah sistem mendapatkan 3 kali kebenaran, 6 kali salah dan 11 kali tidak mengenali wajah.

Dari hasil pengujian dengan viariasi nilai *thresholding* 220 *pixel* adalah sistem mendapatkan 13 kali benar dan 7 kali salah mengenali wajah. Kesmpulannya nilai *thresholding* sudah mendekati tingkat keakurasian yang baik dan sudah tidak terjadi kinerja sistem yang tidak mengenali wajah seperti yang ditunjukkan pada table 6.11 di bawah ini.

Tabel 6.11 Hasil pengujian aplikasi dengan nilai *thresholding* 220 *pixel*

No.	**Pengamatan Manual**	**Pengamatan Sistem**	**Hasil Uji**
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Afnan Afif	Salah
3	Ryan Indra	Ryan Indra	Benar
4	Ryan Indra	Afnan Afif	Salah
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Tidak dikenali	Salah
7	Yusron Hamidi	Yusron Hamidi	Benar
8	Yusron Hamidi	Yusron Hamidi	Benar
9	Yusron Hamidi	Yusron Hamidi	Benar
10	Yusron Hamidi	Mula Agung Barat	Salah
11	Mula Agung Barata	Yusron Hamidi	Salah
12	Mula Agung Barata	Afnan Afif	Salah
13	Mula Agung Barata	Mula Agung Barata	Benar
14	Mula Agung Barata	Mula Agung Barata	Benar
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Afnan Afif	Benar
17	Afnan Afif	Afnan Afif	Benar
18	Afnan Afif	Afnan Afif	Benar
19	Afnan Afif	Afnan Afif	Benar
20	Afnan Afif	Afnan Afif	Benar

Dari hasil pengujian dengan viariasi nilai *thresholding* 240 *pixel* adalah sistem mendapatkan 12 kali benar dan 8 kali salah mengenali wajah. Kesmpulannya nilai *thresholding* sudah mendekati tingkat keakurasian yang baik. Tetapi sistem masih mendapatkan keakurasian yang lebih baik pada penentuan nilai *thresholding* 230 *pixel* seperti yang ditunjukkan pada table 6.12 di bawah ini.

Tabel 6.12 Hasil pengujian aplikasi dengan nilai *thresholding* 240 *pixel*

No.	Pengamatan Manual	Pengamatan Sistem	Hasil Uji
1.	Ryan Indra	Ryan Indra	Benar
2.	Ryan Indra	Afnan Afif	Salah
3	Ryan Indra	Ryan Indra	Benar
4	Ryan Indra	Afnan Afif	Salah
5	Ryan Indra	Ryan Indra	Benar
6	Yusron Hamidi	Ryan Indra	Salah
7	Yusron Hamidi	Yusron Hamidi	Benar
8	Yusron Hamidi	Yusron Hamidi	Benar
9	Yusron Hamidi	Yusron Hamidi	Benar
10	Yusron Hamidi	Yusron Hamidi	Benar
11	Mula Agung Barata	Mula Agung Barata	Benar
12	Mula Agung Barata	Afnan Afif	Salah
13	Mula Agung Barata	Mula Agung Barata	Benar
14	Mula Agung Barata	Yusron Hamidi	Salah
15	Mula Agung Barata	Afnan Afif	Salah
16	Afnan Afif	Ryan Indra	Salah
17	Afnan Afif	Ryan Indra	Salah
18	Afnan Afif	Afnan Afif	Benar
19	Afnan Afif	Afnan Afif	Benar
20	Afnan Afif	Afnan Afif	Benar

BAB VII PENUTUP

Pada bab ini membahas mengenai hasil yang dijelaskan pada kesimpulan dari sistem yang dibuat. Setelah melakukan perencanaan, perancangan dan implementasi *minicomputer raspberry pi* untuk sistem keamanan pintu ruang *server*, kemudian dilakukan proses pengujian dan analisa maka dapat diambil kesimpulan dan saran sebagai berikut:

7.1 Kesimpulan

Hasil uji coba dari implementasi *minicomputer raspberry pi* untuk sistem keamanan pintu ruang *server* dapat ditarik kesimpulan sebagai berikut:

1. Implementasi *minicomputer raspberry pi* untuk penerapan pengolahan citra dirasa belum bisa optimal dalam kinerjanya dilihat dari *CPU Usage* ketika aplikasi *computer vision* dijalankan, mengingat spesifikasi *raspberry pi* kurang seimbang dengan spesifikasi yang dibutuhkan untuk melakukan pengolahan citra.
2. Sistem pengenalan wajah dapat dilakukan dengan metode *triangle face* dengan membandingkan jarak antar fitur wajah.
3. Pada sistem yang dibuat, proses pendeteksian fitur wajah seperti kedua mata, hidung dan mulut sangat sulit dilakukan karena kinerja perangkat keras sangat berat.
4. Sistem yang dibuat sulit mendapatkan nilai jarak yang sama persis dalam satu objek wajah, hal ini dikarenakan sulitnya proses pendeteksian fitur wajah yang sesuai.
5. Intensitas cahaya sangat mempengaruhi nilai jarak antar fitur yang diambil, sehingga data *training* dan *testing* dalam pengambilannya harus memiliki intensitas cahaya yang sama.
6. Sistem pengenalan wajah dengan metode *triangle face* ini memiliki keakuratan 75% dan terdapat 25% kesalahan.
7. Sistem yang dibuat cukup aman untuk diimplementasikan untuk sistem keamanan pintu ruang *server*.

7.2 Saran

Kesimpulan yang didapat dari pengujian sistem yang dibuat menghasilkan beberapa saran yang dapat dimanfaatkan sebagai referensi untuk pengembangan penelitian mengenai pengolahan citra. Adapun saran yang diberikan peneliti adalah sebagai berikut:

1. *Mini raspberry pi* lebih cocok digunakan sebagai *trigger* yang menghubungan antara *too*l untuk *processing* dengan *server* aplikasi yang dikembangkan.
2. Seharusnya pengambilan citra tidak terpaku pada penyangga dagu dan jarak pengambilan untuk proses deteksi, pengambilan citra dengan metode *triangle face* dirasa sangat kaku dan kurang fleksibel, pengambilan citra dilakukan dengan meletakkan dagu pada penyangga dagu dan dibatasi oleh jarak pengambilan wajah,
3. Sebaiknya kamera yang digunakan memiliki resolusi minimal 8 megapiksel agar menghasilkan kualitas data *training* yang baik karena hal itu menentukan keakurasian sistem dalam mengenali wajah.

DAFTAR PUSTAKA

[1] Muhammad Miftah, Aripin. “Pengamanan Laptop Menggunakan Pengenalan Wajah Berbasis Triangle Face”, Journal of Applied Intelligent System, Vol.1, pp. 22-35, Feb. 2016.

[2] Rasyid Sindu Prihantono**,** Ary Mahzaruddin Shiddiqi., S.Kom., M.Comp.Sc., Hudan Studiawan, S.Kom.,M.Kom., “Rancang Bangun Sistem Keamanan dan Pengenalan Objek dalam Ruangan Sebagai Pengganti CCTV dengan Menggunakan Raspberry Pi”, JURNAL TEKNIK POMITS, Vol. 2, pp. 2301-9271, 2013.

[3] Therzian Richard Perkasa, Helmy Widyantara, “Pauladie Susanto. Rancang Bangun Pendeteksi Image Subtraction Pada Single Board Computer (SBC)”, Journal of Control and Network Systems, Vol. 3, pp. 90-97, 2014.

[4] Fadli Sirait, Yoserizal, “Pemanfaatan Raspberry Pi Sebagai Processor Pada Pendeteksian Dan Pengenalan Pola Wajah”, Vol. 7, pp. 2086-9479, Sept. 2016.

[5] Apriyana, Delta Sri Maharani, Shinta Puspitasari, Renni Angreni. “Perbandingan Model Sobel, Metode Prewitt dan Metode Robert Untuk Deteksi Tepi Objek Pada Aplikasi Pengenalan Bentuk Berbasis Citra Digital”.

[6] Dinata, Andi.2017. *Physical Computing* dengan *Raspberry Pi*. Jakarta: Elex Media Komputindo.

LAMPIRAN

Lampiran 1. Biodata Penulis

CURRICULUM VITAE

IDENTITAS PRIBADI

Nama Lengkap	**Mula Agung Barata**
Tempat, Tgl Lahir	**Blora, 11 April 1993**
Jenis Kelamin	**Laki-laki**
Hobi	**Komputer dan musik**
Status	**Belum Menikah**
Agama	**Islam**
Email	**Mula.ab26@gmail.com**
Kewarganegaraan	**Indonesia**
Alamat	**Jl. Pusri no.14A RT. 01/10 Balun Kandangdoro, Cepu Kab. Blora – Jawa Tengah**

FORMAL EDUCATION

Nama Sekolah	Kota	Mulai	Selesai	Keahlian	Hasil
SD Negeri 10 Cepu	Cepu Jawa Tengah	1999	2005	-	NUN: 40,75
SMP Negeri 3 Cepu	Cepu Jawa Tengah	2005	2008	-	NUN: 29,45
SMK Negeri 1 Cepu	Cepu Jawa Tengah	2008	2011	Administrasi Perkantoran	NUN: 33,75
Politeknik Negeri Malang	Malang Jawa Timur	2012	2014	Diploma II Manajemen Informatika	GPA: 3,51
Politeknik Negeri Malang	Malang Jawa Timur	2014	Sekarang	Diploma IV Manajemen Informatika	GPA: 3,48

Lampiran 2. Surat Permohonan Ijin Obeservasi

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
Jl. Soekarno Hatta No.9 Malang 65141
Telp (0341) 404424 – 404425 Fax (0341) 404420
Laman://www.polinema.ac.id

Nomor : 2485 /PL2.1/PM/2017
Perihal : **Observasi Data**

09 MAR 2017

Yth. **Kepala**
PPSDM MIGAS CEPU
Jl.Sorogo No.1 Cepu Jawa Tengah
di tempat

Dengan hormat,

Bersama ini kami mohon bantuan Bapak/Ibu, agar dapat memberi kesempatan kepada mahasiswa kami Jurusan Teknologi Informasi Program Studi Teknik Informatika untuk dapat melakukan observasi di perusahaan/instansi yang Bapak/Ibu pimpin untuk kepentingan permohonan ijin observasi di bagian bengkel jaringan komputer dan telekomunikasi.

Adapun nama mahasiswa tersebut adalah sebagai berikut :

NO	NAMA	NIM
1.	Mula Agung Barata	1441183002

Observasi tersebut menurut rencana akan dilaksanakan pada tanggal tanggal 20 Maret 2017.
Demikian, atas perhatian dan kerjasamanya disampaikan terima kasih.

a.n.Direktur
Pembantu Direktur I,

Dr. Drs. Ludfi Djajanto, MBA
NIP. 196204211988031 003

Tembusan Yth. :
1. **Ketua Jurusan Teknologi Informasi**
2. **Ketua Program Studi Teknik Informatika**

FRM.RIF.01.20.00

Lampiran 3. Source code penyimpanan hasil pengukuran citra wajah

```
import numpy as np
import cv2
import math
import urllib
import urllib2

NIP = input("Masukkan NIP: ")
nama = raw_input("Masukkan Nama Anda: ")

# training file loader
face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade =
cv2.CascadeClassifier('haarcascade_eye_tree_eyeglasses.xml')
mouth_cascade =
cv2.CascadeClassifier('haarcascade_mcs_mouth.xml')
nose_cascade  =
cv2.CascadeClassifier('haarcascade_mcs_nose.xml')
# ---------------------

cap = cv2.VideoCapture(0)
cap.set(3,320)#window width 320 px
cap.set(4,240)#window height 240 px

while cap.isOpened():
    ret, frame = cap.read()
    loop = 0

    #deteksi face
    faces1 = face_cascade.detectMultiScale(frame, 1.1, 3, 0)
    for (x,y,w,h) in faces1:
          cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
          my = y
          mx = x
          mw = w
          mh = h
    #deteksi face

    #deteksi eyes
    eyes = eye_cascade.detectMultiScale(frame, 1.1, 3, 0)
    #deteksi eyes

    eyesNumber = 0 # jumlah mata terdeteksi
    arrayMata=[[0,0],[0,0]]

    for (ex,ey,ew,eh) in eyes:
```

```
    cv2.rectangle(frame,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
          if eyesNumber < 2 :
                arrayMata[eyesNumber][0] = ex + (ew/2)
                arrayMata[eyesNumber][1] = ey + (eh/2)
                eyesNumber = eyesNumber + 1

    #deteksi nose
    nose = nose_cascade.detectMultiScale(frame, 1.1, 3, 0)
    #deteksi nose

    noseNumber = 0    # jumlah hidung terdeteksi
    arrayHidung=[0,0]
    for (nx,ny,nw,nh) in nose:

    cv2.rectangle(frame,(nx,ny),(nx+nw,ny+nh),(0,255,0),2)
          if noseNumber == 0:
                arrayHidung[0] = nx + (nw/2)
                arrayHidung[1] = ny + (nh/2)
                noseNumber = noseNumber+1

    #deteksi mouth
    mouth = mouth_cascade.detectMultiScale(frame, 1.1, 3, 0)
    #deteksi mouth

    mouthNumber = 0   # jumlah mouth terdeteksi
    arrayMulut=[0,0]
    for (mx,my,mw,mh) in mouth:

    cv2.rectangle(frame,(mx,my),(mx+mw,my+mh),(0,255,0),2)
          if mouthNumber == 0:
                arrayMulut[0] = mx + (mw/2)
                arrayMulut[1] = my + (mh/2)
                mouthNumber = mouthNumber+1

    tmbl_out = cv2.waitKey(1)
    m = 0

    if tmbl_out == 32 and loop<1:
          #menyimpan foto ----------------------------------
------
          cv2.imwrite("foto1.jpg", frame)
          # menyimpan foto ---------------------------------
-------

          # membaca foto -----------------
          image = cv2.imread("foto1.jpg"[1])
          # membaca foto -----------------

          m = m+1
```

```
##perhitungan jarak mata
          perhitungan = ( ( ( arrayMata[0][0]-
arrayMata[1][0] ) * ( arrayMata[0][0]-arrayMata[1][0] ) + (
arrayMata[0][0]-arrayMata[1][0]) * (arrayMata[0][1]-
arrayMata[1][1]) ) )
          jarakMata   = math.sqrt(perhitungan)
        #perhitungan jarak antar mata

        ##perhitungan jarak mata kanan ke mulut
          perhitungan = (( ( mw/2 ) - arrayMata[1][0] ) *
((mw/2) - arrayMata[1][0])) + ( ( ( arrayMata[1][1] - ( mh -
( mh/8.5 ) ) ) * (arrayMata[1][1] - ( mh - ( mh/8.5 ) ) ) )
)
          jarakMaKaMulut = math.sqrt(perhitungan)
        ##perhitungan jarak mata kanan ke mulut

        ##perhitungan jarak mata kiri ke mulut
          perhitungan = (((mw/2) - arrayMata[0][0])*((mw/2)
- arrayMata[0][0])) + (((arrayMata[0][1] - (mh-
(mh/8.5)))*(arrayMata[0][1] - ( mh-(mh/8.5) ))))
          jarakMaKiMulut = math.sqrt(perhitungan)
        ##perhitungan jarak mata kiri ke mulut

        ##perhitungan jarak mata kanan ke hidung
          perhitungan = (((arrayHidung[0]-
arrayMata[1][0])*(arrayHidung[0]-
arrayMata[1][0]))+((arrayMata[1][1]-
arrayHidung[1])*(arrayMata[1][1]-arrayHidung[1])))
          jarakMakaHidung   = math.sqrt(perhitungan)
        ##perhitungan jarak mata kanan ke hidung

        ##perhitungan jarak mata kiri ke hidung
          perhitungan = (((arrayHidung[0]-
arrayMata[0][0])*(arrayHidung[0]-
arrayMata[0][0]))+((arrayMata[0][1]-
arrayHidung[1])*(arrayMata[0][1]-arrayHidung[1])))
          jarakMakiHidung   = math.sqrt(perhitungan)


    cv2.line(frame,(arrayMata[0][0],arrayMata[0][1]),(arrayM
ata[1][0],arrayMata[1][1]),(255,255,255),1,8,0) #gambar
jarak antar mata

    cv2.line(frame,(arrayMata[0][0],arrayMata[0][1]),(arrayH
idung[0],arrayHidung[1]),(255,255,255),1,8,0)   #gambar
jarak antar mata hidung

    cv2.line(frame,(arrayMata[1][0],arrayMata[1][1]),(arrayH
idung[0],arrayHidung[1]),(255,255,255),1,8,0)   #gambar
jarak antar mata hidung
```

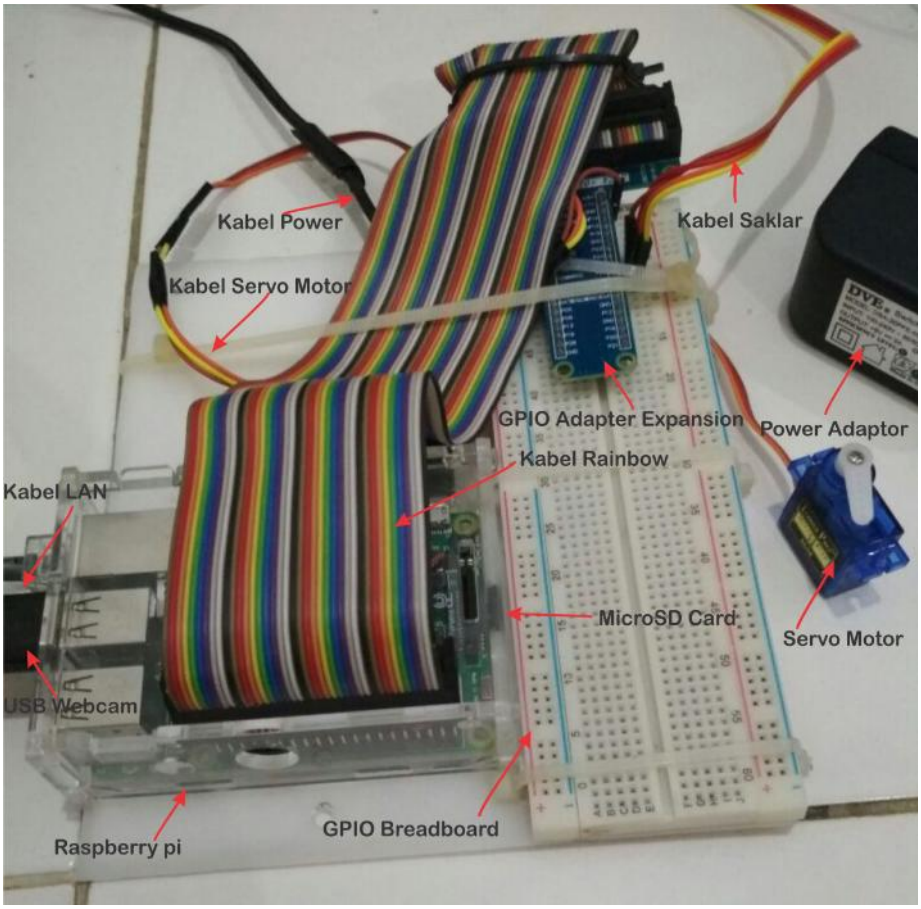
```
    cv2.line(frame,(arrayMata[][],arrayMata[][]),(arrayMulut
[],arrayMulut[]),(255,255,255),1,8,0) #gambar jarak antar
mata mulut

    cv2.line(frame,(arrayMata[][],arrayMata[][]),(arrayMulut
[],arrayMulut[]),(255,255,255),1,8,0) #gambar jarak antar
mata mulut

          cv2.imwrite("foto.jpg", frame)
          loop = loop + 1

          luasWajah=mw*mh
          data = {}
          data['nip'] = NIP
          data['nama'] = nama
          data['luas_wajah'] = luasWajah
          data['makan_mulut'] = jarakMaKaMulut
          data['makir_mulut'] = jarakMaKiMulut
          data['makan_hidung'] = jarakMakaHidung
          data['makir_hidung'] = jarakMakiHidung
          data['jarak_mata'] = jarakMata
          url_values = urllib.urlencode(data)
          #print url_values  # The order may differ.
          url =
'http://localhost/skripsi/index.php/parsingData/simpan'
          full_url = url + '?' + url_values
          urllib2.urlopen(full_url)

        #menampilkan hasil perhitungan
          print
"==========================================================="
,"\n"
          print "|| SISTEM KEAMANAN PINTU RUANG SERVER PPSDM
MIGAS CEPU ||","\n"
          print "||-----------------------------------------
------------||","\n"
          print "||                   MULA AGUNG BARATA
||","\n"
          print "||               POLITEKNIK NEGERI MALANG
||","\n"
          print
"==========================================================="
,"\n"

          print "NIP:> ",NIP,"\n"
          print "Nama:>",nama,"\n"
          print "lebar wajah: ",mw,"\n"
          print "tinggi wajah : ",mh,"\n"
          print "luas wajah: ",mw*mh,"\n"
```

```
print "jarak mata kanan ke mulut: ",jarakMaKaMulut,"\n"
          print "jarak mata kiri ke mulut:
",jarakMaKiMulut,"\n"
          print "jarak mata kanan ke hidung:
",jarakMakaHidung,"\n"
          print "jarak mata kiri ke hidung:
",jarakMakiHidung,"\n"
          print "jarak antara mata: ",jarakMata,"\n"
         #menampilkan hasil perhitungan
    cv2.imshow('CAPTURE', frame)

    if cv2.waitKey(10) & 0xFF == ord('q'):
          break

cap.release()
cv2.destroyAllWindows()
```

Lampiran 4. Source code pencocokan data *training* dan data *testing.*

```
# Use all the SQL you like
      cur.execute("SELECT * FROM login_face")
      jarak=[]
      orang = []
      x=0

# print all the first cell of all the rows
      for row in cur.fetchall():
      orang.append(row[2])
jarak.append(math.sqrt(math.pow(row[3]-(mw*mh),2)+math.pow(row[4]-
jarakMaKiMulut,2)+math.pow(row[5]-
jarakMaKiMulut,2)+math.pow(row[6]-
jarakMakaHidung,2)+math.pow(row[7]-
jarakMakiHidung,2)+math.pow(row[8]-jarakMata,2)))

#jarak.append(math.sqrt(math.pow(row[3]-1000,2)+math.pow(row[4]-
200,2)+math.pow(row[5]-400,2)+math.pow(row[6]-
600,2)+math.pow(row[7]-800,2)+math.pow(row[8]-800,2)))
      print jarak[x]
x+=1
            db.close()
            nilaiMin = min(jarak)
            inn = jarak.index(nilaiMin)

            if(nilaiMin < 230):
                        os.system("python cobaservo.py")
```

Lampiran 5. Konfigurasi Perangkat Keras



Keterangan:

Gambar di atas menunjukkan implementasi dari konfigurasi perangkat keras yang digunakan pada penelitian ini. Semua peralatan terhubung dengan raspberry pi. Mulai dari power adaptor yang memberikan supply listrik pada raspberry pi kemudian USB camera yang terhubung dengan raspberry pi melalui port USB. Terdapat kabel LAN yang terhubung port LAN adapter pada raspberry pi yang berfungsi untuk mengkoneksikan raspberry pi dengan komputer yang digunakan dalam melakukan remote desktop. Seluruh perangkat yang terhubung pada GPIO breadboard yang meliputi motor servo dan saklar push on yang terhubung dengan kabel jumper. Untuk menghubungkan GPIO breadboard dengan raspberry pi menggunakan kabel rainbow. Kabel rainbow tidak bisa langsung terhubung dengan GPIO breadboard melainkan memerlukan GPIO adapter expansion untuk bisa terhubung dengan PIN GPIO pada raspberry pi.

Lampiran 6. Activity Control Bimbingan

Lembar Bimbingan Pembimbing 1

KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

NO SKRIPSI: 8

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu ruang server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).

Nama : MULA AGUNG BARATA **NIM** : 1441183002

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	1-3-'17	Skema alat / rasby		
2.	8-3-'17	Draft pengenalan		
3.		wajah.		
4.	14-3-'17	Metodologi		
5.		penelitian		
6.	22-3-'17	dokumentasi metode		
7.	29-3-'17	konfigurasi		
8.		hardware		
9.	7-4-'17	Dokumentasi		
10.		konfigurasi Hardware.		
11.	13-4-'17	Mulai simulasi		
12.		metode face detection		
13.	19-4-'17	Membuat draft		
14.		metode triangle face		
15.	26-4-'17	Draft metode		
16.		triangle		
17.	3-5-'17	Simulasi Capture		
18.		wajah		
19.	17-5-'17	Uji coba capture		
		wajah.		

Malang, ...
Dosen Pembimbing Skripsi,

Indra Dharma Wijaya, ST., M.MT.
NIP. 19730510 200801 1 010

KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

NO SKRIPSI: 8

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu ruang server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).

Nama : MULA AGUNG BARATA **NIM** : 1441183002

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	24-5-'17	Persiapan UAS		
2.		create dan update		
3.		data wajah		
4.	30-5-'17	Uji coba update		
5.		data User di metode		
6.	9-6-'17	Dokumentasi data		
7.		di library.		
8.	16-6-'17	Uji coba data		
9.		User di linux / database		
10.	5-7-'17	Cetak laporan.		
11.		Skripsi.		
12.	11-7-'17	Acc ujian		
13.		tahap 2		
14.				
15.				
16.				
17.				
18.				
19.				

Malang, 11-7-2017
Dosen Pembimbing Skripsi,

Indra Dharma Wijaya, ST., M.MT.
NIP. 19730510 200801 1 010

Lembar Bimbingan Pembimbing 2

KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

NO SKRIPSI: 8

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu ruang server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).

Nama : MULA AGUNG BARATA **NIM** : 1441183002

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	3-3-17	Diskusi konsep		
2.	11-3-17	Diskusi perangkat / alat yg digunakan		
3.	21-3-17	Metodologi penelitian.		
4.	29-3-17	Konfigurasi hardware		
5.	13-4-17	Rancangan total simulasi		
6.	20-4-17	Dokumentasi konfigurasi hardware		
7.	25-4-17	Konfigurasi hardware		
8.	28-4-17	Simulasi hardware		
9.	4-5-17	Rancangan tripod kamera + dagu		
10.	10-5-17	Simulasi metode face detection		
11.	22-5-17	laporan bab 1 - 3		
12.	29-5-17	draft metode triangle face		
13.	8-6-17	simulasi capture wajah		
14.	12-6-2017	rancangan GUI aplikasi		
15.	15-6-17	dokumentasi data		
16.	19-6-17	uji coba data		
17.	16-6-17	cetak laporan		
18.	5-7-17	sisum jurnal		
19.	11-7-17	Acc lahap 2		

Malang, 11-7-2017.....
Dosen Pembimbing Skripsi,

Usman Nurhasan, S.Kom, MT.
NIP. 19860923 201504 1 001

Lampiran 7. Lembar Persetujuan Mengikuti Ujian Skripsi

KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

NO SKRIPSI: 8

LEMBAR PERSETUJUAN
MENGIKUTI UJIAN SKRIPSI 2016/2017
PROGRAM STUDI TEKNIK INFORMATIKA

NAMA : MULA AGUNG BARATA **NIM / KELAS** : 1441183002 / TI-4A
JUDUL SKRIPSI : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).
PEMBIMBING : 1. Indra Dharma Wijaya, ST., M.MT. **NIP** : 19730510 200801 1 010
2. Usman Nurhasan, S.Kom, MT. **NIP** : 19860923 201504 1 001

No.	Uraian / Bab	Diselesaikan	Tanda Tangan	
			Pembimbing 1	Pembimbing 2
1.	**PENDAHULUAN**	✓		
2.	**LANDASAN TEORI**	✓		
3.	**METODOLOGI PENELITIAN**	✓		
4.	**ANALISIS DAN PERANCANGAN**	✓		
5.	**IMPLEMENTASI**	✓		
6.	**PENGUJIAN DAN PEMBAHASAN**	✓		
7.	**KESIMPULAN DAN SARAN**	✓		
8.	**BAGIAN AKHIR** - Daftar Pustaka - Lampiran (*Isi lampiran disesuaikan dengan judul laporan akhir*) - Profile Penulis (*Riwayat Penulis*)	✓		
9.	**Hardware/Software** - Didemokan di depan pembimbing	✓		
10	**Draft Makalah**	✓		

Malang, ...
Ketua Pelaksana LA & SKRIPSI 2016/2017
Program Studi Teknik Informatika

Arief Prasetyo, S.Kom., M.Kom.
NIP. 19790313 200812 1 002

DISETUJUI UNTUK DAPAT MAJU UJIAN SETELAH HASIL KARYA DINILAI LAYAK SERTA HASIL UJI SESUAI DENGAN SPESIFIKASI YANG DIRENCANAKAN

Pembimbing I	Pembimbing II
Indra Dharma W, ST., M.MT.	Usman Nurhasan, S.Kom, MT.
NIP. 19730510 200801 1 010	NIP. 19860923 201504 1 001

11-7-2017

FRM.RTI.01.49.04

Lampiran 8. Form Revisi Skripsi

Form Revisi Skripsi Penguji 1

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

No. Skripsi :8

FORM REVISI SKRIPSI

Nama Mahasiswa : MULA AGUNG BARATA **NIM** : 1441183002
Tanggal Ujian : 17/7/2017
Judul : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu ruang server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).

NO	SARAN PERBAIKAN	PARAF
	- lengkapi tentang konsep : - deteksi tepi untuk mata, hidung, dan mulut. - bagaimana cara menemukan titik tengah - representasi velocity untuk 1 wajah bagaimana? - perbedaan kemiripan euclidean dan shape bagaimana? - asal usul triangle face dari mana?	

Malang, 17-7-2017
Dosen Penguji,

(FAISAL RAHUTOMO)

FORM VERIFIKASI:
Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	FAISAL RAHUTOMO		21/7-2017
Pembimbing 1	Indra Dharma Wijaya, ST., M.MT.		24-7-2012
Pembimbing 2	Usman Nurhasan, S.Kom, MT.		24-7-2017

FRM.RTI.01.35.03

Form Revisi Skripsi Penguji 2

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122

No. Skripsi :8

FORM REVISI SKRIPSI

Nama Mahasiswa : MULA AGUNG BARATA **NIM** : 1441183002
Tanggal Ujian : 17-7-2017
Judul : Implementasi Raspberry Pi untuk rancang bangun sistem keamanan pintu ruang server dengan pengenalan wajah menggunakan metode triangle face (Studi Kasus Ruang Server PPSDM Migas Cepu).

NO	SARAN PERBAIKAN	PARAF
1)	Perhitungan kemiripan	
2)	Perhitungan titik tengah.	
3)	Cari paper pendukung	

Malang, 17-7-2017
Dosen Penguji,

(Putra Prima Arhandi)
198611032019091001

FORM VERIFIKASI:
Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	Putra Prima Arhandi		21-07-17
Pembimbing 1	Indra Dharma Wijaya, ST., M.MT.		24-7-2017
Pembimbing 2	Usman Nurhasan, S.Kom, MT.		29-7-2017

FRM.RTI.01.35.03