

**SISTEM PENDETEKSI ORANG TIDAK DIKENAL
MENGUNAKAN METODE PENGENALAN WAJAH
PADA KAMERA PENGAWAS**

*Stranger Detection System Using Face Recognition Method
on Surveillance Camera*

TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan

Diploma Tiga Program Studi Teknik Informatika di Jurusan

Teknik Komputer dan Informatika

Oleh:

ADI MAULANA TRIADI

NIM: 151511002

ALI QORNAN JAISYURRAHMAN

NIM: 151511007

IKHSAN HARI WIJAYANTO

NIM: 151511013



POLITEKNIK NEGERI BANDUNG

2018

**SISTEM PENDETEKSI ORANG TIDAK DIKENAL
MENGUNAKAN METODE PENGENALAN WAJAH PADA
KAMERA PENGAWAS**

*Stranger Detection System using Face Recognition Method
on Surveillance Camera*

Oleh:

ADI MAULANA TRIADI

NIM: 151511002

ALI QORNAN JAISYURRAHMAN

NIM: 151511007

IKHSAN HARI WIJAYANTO

NIM: 151511013

Menyetujui

Bandung, 13 Agustus 2018

Pembimbing I

Pembimbing II



Yudi Widhiyasa, S.Si., M.T.
NIP. 197407182001121002



Nurjannah Syakrani, Dra., M.T., Dr.
NIP. 196112131992012001

Ketua Jurusan Teknik Komputer dan Informatika


Drs. Eddy Bambang Soewono, M.Kom.
NIP. 196101141992021001

**SISTEM PENDETEKSI ORANG TIDAK DIKENAL
MENGUNAKAN METODE PENGENALAN WAJAH PADA
KAMERA PENGAWAS**

*Detection System of Stranger with Face Recognition Method
on Surveillance Camera*

Oleh:

ADI MAULANA TRIADI NIM: 151511002

ALI QORNAN JAISYURRAHMAN NIM: 151511007

IKHSAN HARI WIJAYANTO NIM: 151511013

Tugas Akhir ini telah disidangkan pada tanggal 1 Agustus 2018
sesuai dengan ketentuan.

Tim Pengaji:

Ketua : Suprihanto, BSEE., M.Sc
NIP. 196303161995121001

Anggota : Jonner Hutahacan, BSET., M.Info.Sys
NIP. 196210211993031002






Nama : Adi Maulana Triadi

NIM : 151511002

Tempat, Tanggal Lahir : Bandung, 15 Agustus 1997
SD Lulus Tahun : 2009 dari SDN Mohammad Toha 2
SLTP Lulus Tahun : 2012 dari SMP Negeri 11 Bandung
SLTA Lulus Tahun : 2015 dari SMA Negeri 17 Bandung

Prestasi yang pernah dicapai :

- Juara 1 Windows App Development Caution 6 Tahun 2015
- Best Presentator Windows App Development Caution 6 Tahun 2015
- Juara 2 Software Development National Competition Technocorner UGM Tahun 2017
- Juara Harapan Expo Kewirausahaan Mahasiswa Indonesia bidang Teknologi Tahun 2017



Nama : Ali Qornan Jaisyurrahman

NIM : 151511007

Tempat, Tanggal Lahir : Bandung, 29 November 1997

SD Lulus Tahun : 2009 dari MIN 2 Margasari

SLTP Lulus Tahun : 2012 dari SMP Negeri 51 Bandung

SLTA Lulus Tahun : 2015 dari SMK Negeri 4 Bandung

Prestasi yang pernah dicapai :

- Nominator INAICTA 2014
- Top 75 Apps Hackathon BestAppsID
- Juara 2 (Runner Up) - Anugerah Motekar Unpad
- Terbaik 3 Hackathon IWIC 9
- Juara 3 Liga Digital Indonesia 2016
- 2nd Winner IWIC 10
- Winner IBM Bluemix Challenge - Bantu Ibu dengan Teknologi
- Juara 3 LKS IT Software Bandung
- Juara 2 kompetisi SDC Technocorner 2017
- Finalis Hackathon UMN 2016
- Juara 1 Samsung Indonesia Next Apps (INA) 4.0 Kategori Samsung SDK
- 2nd Runner up Hackday
- Developer Circles Community Challenge - Improve your craft 3rd place
- Finalis BTN Mortgtech Hackathon



Nama : Ikhsan Hari Wijayanto

NIM : 151511013

Tempat, Tanggal Lahir : Indramayu, 26 Juni 1998

SD Lulus Tahun : 2009 dari SDN Juntinyuat 3

SLTP Lulus Tahun : 2012 dari SMP Negeri Karangampel

SLTA Lulus Tahun : 2015 dari SMA Negeri 1 Sindang

Prestasi yang pernah dicapai :

- Juara 2 Olimpiade Sains nasional bidang komputer tingkat kabupaten Tahun 2014
- Juara 5 ICT Competition bidang komputer tingkat Nasional di Kabupaten Rembang Tahun 2014
- Juara 1 Windows App Development Caution 6 Tahun 2015
- Juara 2 Software Development National Competition Technocorner UGM Tahun 2017
- Juara 1 Hackaton Facebook Developer Circle Bandung Tahun 2017

PERNYATAAN PENULIS

Dengan ini menyatakan bahwa laporan Tugas Akhir dengan judul Sistem Pendeteksi Orang tidak Dikenal menggunakan Metode Pengenalan Wajah pada Kamera Pengawas adalah karya ilmiah yang bebas dari unsur tindakan plagiarisme, dan sesuai dengan ketentuan tata tulis yang berlaku.

Apabila dikemudian hari ditemukan adanya unsur plagiarisme, maka hasil penelitian dari Tugas Akhir ini dicabut dan bersedia menerima sanksi sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini dibuat dengan sesungguhnya dalam keadaan sadar sepenuhnya.

Bandung, 13 Agustus 2018



Adi Maulana Triadi
NIM: 151511002



SISTEM PENDETEKSI ORANG TIDAK DIKENAL MENGUNAKAN METODE PENGENALAN WAJAH PADA KAMERA PENGAWAS

TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan

Diploma Tiga Program Studi Teknik Informatika di Jurusan
Teknik Komputer dan Informatika

Oleh:

ADI MAULANA TRIADI NIM: 151511002
ALI QORNAN JAISYURRAHMAN NIM: 151511007
IKHSAN HARI WIJAYANTO NIM: 151511013



Match Overview

6%

1	www.npmjs.com	1%
2	repository.lts.ac.id	<1%
3	www.w8qqq.org	<1%
4	www.geeker.co.nz	<1%
5	id.123dok.com	<1%
6	ja.scribd.com	<1%
7	www.kajianpustaka.com	<1%
8	www.scribd.com	<1%
9	research.gold.ac.uk	<1%

ABSTRAK

Dalam beberapa tahun terakhir, orang-orang mulai menggunakan berbagai cara untuk meningkatkan pengawasan terhadap orang yang tidak dikenal yang masuk ke lingkungan rumah. Salah satu proses pengaktualisasian dalam peningkatan pengawasan adalah dengan memasang sistem kamera pengawas atau CCTV. Sistem kamera pengawas saat ini belum optimal karena sistem belum mampu untuk mendeteksi dan mengenali wajah yang terekam oleh kamera secara otomatis sehingga masih membutuhkan seorang manusia yang bertindak sebagai pengawas aktivitas yang terekam pada kamera. Kendalanya adalah pengawas tidak bisa melakukan pengawasan terus-menerus secara langsung. Maka dari itu, dibutuhkan sebuah aplikasi komputer yang dapat dilatih untuk menangani permasalahan tersebut. Untuk itu, pemanfaatan salah satu cabang ilmu dari *Artificial Intelligence* yaitu *computer vision* tentang *object detection* dan *object recognition* sangatlah diperlukan. Pada tugas akhir ini, penulis membuat suatu sistem untuk mengoptimalkan sistem kamera pengawas agar bisa mengidentifikasi setiap wajah orang yang tertangkap kamera secara otomatis dengan konsep pengenalan wajah. Ketika sistem telah berhasil mengidentifikasi adanya orang yang tidak dikenali, maka akan melakukan pengiriman pemberitahuan kepada pemilik rumah melalui media chatbot. Dengan dibuatnya sistem ini, pemilik rumah atau pengawas rumah tidak perlu mengawasi terus-menerus secara langsung karena sistem telah menangani hal tersebut.

Kata kunci: CCTV, *Artificial Intelligence*, pengenalan wajah, *chatbot*.

ABSTRACT

In recent years, people have begun to use various methods to improve surveillance of unknown people who enter the home environment. One of the actualization processes in improving supervision is by installing a surveillance camera system or CCTV. The current surveillance camera system is not optimal because the system has not been able to detect and recognize faces that recorded by the camera automatically so that it still needs a human who acts as the supervisor of the activity recorded on the camera. The problem is the supervisors cannot carry out continuous monitoring directly. Therefore, it needs a computer application that can be trained to handle these problems. For that reason, the uses of a branch of science from Artificial Intelligence, computer vision, object detection and object recognition is needed. In this final project, the author makes a system to optimize the surveillance camera system so that it can identify every face of a person who is caught on camera automatically with the concept of face recognition. When the system has successfully identified a person who is not recognized, it will send a notification to the homeowner via the chatbot media. With the creation of this system, homeowners or home supervisors do not need to monitor continuously directly because the system has handled it.

Keyword: CCTV, Artificial Intelligence, Face recognition, chatbot.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan hidayah-Nya kita masih diberikan keselamatan dan kesehatan. Sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Sistem Pendeteksi Orang tidak Dikenal dengan Menggunakan Metode Pengenalan Wajah pada Kamera Pengawas” ini. Pada kesempatan kali ini penulis mengucapkan terima kasih kepada:

1. Allah SWT yang selalu memberikan kemudahan dan kelancaran dalam proses penyelesaian tugas akhir.
2. Orang tua dan keluarga yang selalu memberika do’a dan dukungan lainnya selama penulis menyelesaikan tugas akhir ini.
3. Eddy Bambang Soewono, Drs., M.Kom., selaku Ketua Departemen Teknik Informatika atas arahan dan dukungannya..
4. Irawan Thamrin, Ir., M.T., selaku Ketua Program Studi D III Teknik Informatik atas bimbingan dan dukungannya.
5. Yudi Widhiyasana, S.Si., M.T., dan Nurjannah Syakrani, Dra., M.T., Dr. Selaku pembimbing I dan pembimbing II yang telah bersedia memberikan bimbingan selama proses pengerjaan tugas akhir ini
6. Suprihanto, BSEE., M.Sc., dan Jonner Hutahaeon, BSET., M.Info.Sys. selaku dosen penguji yang telah bersedia menguji dan memberi masukan terhadap tugas akhir yang penulis buat.
7. Seluruh dosen dan staf Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung.
8. Rekan-rekan Jurusan Teknik Komputer dan Informatika POLBAN angkatan 2015 yang telah berjuang bersama-sama dalam menyelesaikan tugas akhir ini.
9. Keluarga dan sanak saudara atas penyaluran semangat dan kasih sayang yang tiada henti kepada penulis.

Penulis menyelesaikan tugas akhir ini sebaik mungkin walaupun masih banyak kekurangan dan kesalahan di dalamnya, sehingga diharapkan adanya kritik dan

saran yang membangun dari pembaca. Semoga tugas akhir ini bisa bermanfaat bagi pembaca di lain waktu.

Bandung, 2018

Kelompok Tugas Akhir 102

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR LAMPIRAN	ix
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR ISTILAH	xiv
DAFTAR SINGKATAN DAN LAMBANG	xv
DAFTAR RUMUS	xvi
BAB I PENDAHULUAN	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	2
I.3 Tujuan	2
I.4 Ruang Lingkup	3
I.5 Rumusan Masalah	3
BAB II TINJAUAN PUSTAKA	4
II.1. Karya Ilmiah Sejenis Sebelumnya	4
II.2 Dasar Teori	5
II.2.1 Pengolahan Citra Digital	6
II.2.2 Histogram of Oriented Gradients	6
II.2.3 <i>Support Vector Machine</i>	8
II.2.4 <i>Machine Learning</i> dan <i>Deep Learning</i>	8
II.2.5 Euclidean Distance	8

II.2.6 <i>Face Recogniton</i>	9
II.2.7 Chatbot	10
II.2.8 <i>Internet of Things</i>	11
BAB III METODOLOGI PENELITIAN.....	12
III.1 Pendefinisian Masalah.....	13
III.2 Studi Literatur.....	13
III.3 Eksplorasi Teknologi.....	14
III.4 Pengembangan Perangkat Lunak	15
BAB IV REQUIREMENT ANALYSIS AND DEFINITION	21
IV.1 Analisis Sistem Sekarang	21
IV.1.1 Kamera CCTV Analog	21
IV.2 Analisis Sistem To Be	24
IV.2.1 Domain Sistem	25
IV.2.2 Analisis Arsitektur Sistem	27
IV.3 Analisis Akuisisi Citra	30
IV.3.1 Posisi Kamera	30
IV.3.2 Resolusi Kamera.....	31
IV.3.3 Pemilihan <i>Frame</i>	33
IV.4 Analisis Metode	34
IV.4.1 Metode Pendeteksian Wajah	34
IV.4.2 Metode Memproyeksikan Wajah (Transformasi).....	38
IV.4.3 Metode <i>Encoding</i> pada Wajah.....	40
IV.4.4 Metode Perbandingan Wajah.....	42
IV.5 Analisis Pengiriman Pemberitahuan	43
IV.5.1 Kondisi yang Harus Terpenuhi.....	43
IV.5.2 Konten yang Dikirimkan	44

IV.5.3 Interval Pengiriman	44
IV.6 Analisis Perangkat Lunak.....	45
IV.6.1 Operating System (for Raspberry).....	45
IV.6.2 Bahasa Pemrograman dan <i>Library</i>	45
IV.7 Analisis Perangkat Keras	47
IV.7.1 Menjalankan Proses <i>Deep Learning</i> dan <i>Image Processing</i>	48
IV.7.2 Perangkat Akuisisi Data	48
IV.7.3 Data Transfer	50
IV.7.4 Data Storage (MicroSD)	50
IV.8 Analisis Komunikasi data.....	50
IV.8.1 Interaksi <i>Service</i> Penampil Data dengan Messenger Platform.....	51
IV.9 Analisa Data	54
IV.9.1 <i>Database</i>	54
IV.9.2 Data yang Akan Diproses	54
BAB V SYSTEM AND SOFTWARE DESIGN	55
V.1 Perancangan Aktivitas pada Sistem.....	55
V.2 Perancangan Proses Alur Kerja Sistem	56
V.3 Perancangan <i>Logical</i> Arsitektur	59
V.4 Perancangan <i>Deployment</i> Sistem	61
V.5 Perancangan Class Model.....	63
V.6 Perancangan Interaksi Antar Objek	70
V.6.1 Sequence Diagram Akuisisi Citra.....	70
V.6.2 Sequence Diagram Mendeteksi Wajah	71
V.6.3 Sequence Diagram Mengenali Wajah.....	73
V.6.4 Sequence Diagram Mengirimkan Pemberitahuan	74
V.6.5 Sequence Diagram Menambah Data Orang yang Dikenal	77

V.6.6 Sequence Diagram Melihat Daftar Orang yang Dikenal	78
V.6.7 Sequence Diagram Mengubah Data Orang Dikenal	80
V.6.8 Sequence Diagram Menghapus Data Orang Dikenal	82
V.7 Perancangan Tampilan	83
V.7.1 <i>User Interface</i> Tambah Data ke Daftar Orang Dikenal	84
V.7.2 <i>User Interface</i> Lihat Daftar Orang Dikenal	85
V.7.3 <i>User Interface</i> Mengapus Data pada Daftar Orang Dikenal.....	86
V.7.4 <i>User Interface</i> Mengubah Data pada Daftar Orang Dikenal	86
V.7.5 User Interface Menampilkan Pemberitahuan.....	87
BAB VI IMPLEMENTASI	89
VI.1 Keterangan Requirement yang Diimplementasikan.....	89
VI.2 Hasil Implementasi.....	91
BAB VII PENGUJIAN	100
VII.1 Hasil Pengujian	100
BAB VII KESIMPULAN DAN SARAN	105
VIII.1 Kesimpulan.....	105
VIII.2 Saran	105
DAFTAR PUSTAKA	107

DAFTAR LAMPIRAN

LAMPIRAN A-1 <i>SOFTWARE REQUIREMENT SPECIFICATION</i>	109
LAMPIRAN A-2 <i>TEST PLAN</i>	110
LAMPIRAN A-3 <i>TEST DESIGN SPECIFICATION</i>	111
LAMPIRAN A-4 <i>TEST CASE SPECIFICATION</i>	112
LAMPIRAN A-5 <i>TEST PROCEDURE SPECIFICATION</i>	113

DAFTAR GAMBAR

Gambar II.1 Proses pengolahan citra	6
Gambar II.2 HOG <i>face pattern</i>	7
Gambar II.3 Representasi citra wajah menjadi vektor n dimensi	9
Gambar II.4 Triplet loss learning	10
Gambar III.1 Diagram metodologi penyelesaian masalah	12
Gambar III.2 Diagram pendefinisian masalah	13
Gambar III.3 Diagram studi literatur	14
Gambar III.4 Diagram eksplorasi teknologi.....	15
Gambar III.5 Diagram pengembangan perangkat lunak	15
Gambar III.6 Diagram analisis pada perancangan perangkat lunak	16
Gambar III.7 Diagram perancangan pada pengembangan perangkat lunak	17
Gambar III.8 Implementasi pada pengembangan perangkat lunak.....	19
Gambar III.9 Diagram pengujian pada pengembangan perangkat lunak.....	20
Gambar IV.1 Komponen CCTV analog.....	22
Gambar IV.2 Cara kerja sistem CCTV analog.....	23
Gambar IV.3 Cara kerja sistem <i>to be</i>	26
Gambar IV.4 Arsitektur IP kamera	28
Gambar IV.5 Arsitektur Raspberry Pi dan chatbot	29
Gambar IV.6 Penempatan posisi kamera	31
Gambar IV.7 Pendeteksian wajah dengan resolusi 640 x 480p	32
Gambar IV.8 Pendeteksian wajah dengan resolusi 1280 x 720p	32
Gambar IV.9 Proses pengenalan wajah	34
Gambar IV.10 Metode HOG	35
Gambar IV.11 Ekstraksi fitur HOG	35
Gambar IV.12 Citra yang terdeteksi HOG.....	37
Gambar IV.13 Citra wajah yang tidak terdeteksi oleh HOG	37
Gambar IV.14 <i>Face Landmark</i>	38
Gambar IV.15 Penempatan <i>face landmark</i>	39
Gambar IV.16 Citra wajah setelah transformasi affine.....	40
Gambar IV.17 Hasil representasi wajah ke dalam bentuk vektor 128D	42

Gambar IV.18 Struktur <i>event messages</i>	51
Gambar V.1 <i>Activity</i> diagram sistem	55
Gambar V.2 State Machine Diagram	57
Gambar V.3 <i>Logical</i> arsitektur.....	59
Gambar V.4 <i>Deployment</i> diagram	61
Gambar V.5 <i>Class</i> diagram.....	63
Gambar V.6 SD akuisisi citra	70
Gambar V.7 SD mendeteksi wajah	71
Gambar V.8 SD Mengenali wajah	73
Gambar V.9 Sequence diagram mengirimkan pemberitahuan	75
Gambar V.10 Sequence diagram menambah data orang yang dikenal.....	77
Gambar V.11 Sequence diagram melihat daftar orang yang dikenal.....	79
Gambar V.12 SD Mengubah data orang dikenal	80
Gambar V.13 Sequence diagram menghapus data orang dikenal.....	82

DAFTAR TABEL

Tabel II.1 Karya ilmiah sejenis sebelumnya	4
Tabel III.1 Simbol diagram metodologi penyelesaian masalah	12
Tabel IV.1 Komponen CCTV analog	21
Tabel IV.2 Kekurangan dan solusi yang ditawarkan	24
Tabel IV.3 Kekurangan dan kelebihan arsitektur pertama.....	28
Tabel IV.4 Kelebihan dan kekurangan arsitektur Raspberry Pi dan Chatbot	30
Tabel IV.5 Hasil perbandingan resolusi.....	31
Tabel IV.6 Method <code>get_frontal_face_detector()</code>	36
Tabel IV.7 Method <code>compute_face_descriptor</code>	41
Tabel IV.8 Perbandingan sintaks Curl dan Python	46
Tabel IV.9 Spesifikasi Raspberry Pi Model B	47
Tabel IV.10 Spesifikasi Raspberry Pi NoIR Camera module v.2.....	49
Tabel IV.11 Deskripsi objek event webhook.....	52
Tabel IV.12 Deskripsi objek entry	52
Tabel IV.13 Deskripsi objek event entry messaging	52
Tabel V.1 <i>State</i> tabel.....	57
Tabel V.2 Class <code>Messenger_Platform_Controller</code>	64
Tabel V.3 Class <code>Camera</code>	64
Tabel V.4 Class <code>Face_Recognizer</code>	65
Tabel V.5 Class <code>Bot_Util</code>	66
Tabel V.6 <i>Class</i> <code>White_List_Manager</code>	67
Tabel V.7 Class <code>Model_Face</code>	67
Tabel V.8 Class <code>Face Detection Model</code>	68
Tabel V.9 Class <code>Shape predictor model</code>	69
Tabel V.10 Class <code>Face Recognition Model</code>	69
Tabel V.11 SD akuisisi citra	71
Tabel V.12 Penjelasan SD mendeteksi wajah.....	72
Tabel V.13 Penjelasan SD mengenali wajah	73
Tabel V.14 Penjelasan sequence diagram mengirimkan pemberitahuan.....	75
Tabel V. 15 Penjelasan SD menambah data orang dikenal	77
Tabel V.16 Penjelasan sequence diagram melihat daftar orang dikenal.....	79

Tabel V.17 Penjelasan sequence diagram mengubah data orang dikenal.....	81
Tabel V.18 Sequence diagram menghapus data orang dikenal.....	82
Tabel V.19 Penjelasan UI tambah data ke daftar orang dikenal	84
Tabel V.20 Penjelasan UI lihat daftar orang dikenal	85
Tabel V.21 Penjelasan UI menghapus data orang yang dikenal	86
Tabel V.22 Penjelasan UI mengubah data orang yang dikenal	87
Tabel V.23 Penjelasan UI menampilkan pemberitahuan	87
Tabel VI.1 <i>Requirement</i> yang diimplementasikan.....	89
Tabel VI.2 Implementasi 001	91
Tabel VI.3 Implementasi 002.....	91
Tabel VI.4 Implementasi 003.....	92
Tabel VI.5 Implementasi 004.....	93
Tabel VI.6 Implementasi 005.....	94
Tabel VI.7 Implementasi 006.....	95
Tabel VI.8 Implementasi 007.....	96
Tabel VI.9 Implementasi 008.....	97
Tabel VI.10 Implementasi 009.....	98
Tabel VII.1 Hasil pengujian.....	100

DAFTAR ISTILAH

<i>Encoding</i>	: Proses konversi informasi dari suatu sumber (objek) menjadi data, yang selanjutnya dikirimkan ke penerima atau pengamat.
<i>Whitelist</i>	: Daftar dari orang yang dikenal
<i>Blacklist</i>	: Daftar dari orang yang tidak dikenal
<i>Database</i>	: Kumpulan informasi yang disimpan pada suatu komputer secara sistematis
<i>Pretrained</i>	: Sebuah model yang telah dilatih terlebih dahulu
<i>Library</i>	: Kumpulan program atau fungsi yang telah dicompile atau diinterpretasi di dalam bahasa pemrograman tertentu
<i>Open Source</i>	: Jenis perangkat lunak yang sumbernya terbuka dan dapat diakses secara bebas

DAFTAR SINGKATAN DAN LAMBANG

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
CCTV	<i>Closed Circuit Television</i>	1
AI	<i>Artificial Intelligence</i>	1
PIR	<i>Passive Infrared</i>	4
LBP	<i>Local Binary Pattern</i>	4
HOG	<i>Histogram of Oriented Gradients</i>	6
SVM	<i>Support Vector Machine</i>	7
LFW	<i>Labeled Faces in the Wild</i>	9
IM	<i>Instant Messaging</i>	10
REST	<i>Representational State Transfer</i>	11
API	<i>Application Program Interface</i>	11
IOT	<i>Internet of Things</i>	11
SRS	<i>Software Requirement Specification</i>	16
HTTPS	<i>Hypertext Transfer Protocol Secure</i>	46

DAFTAR RUMUS

Rumus Gradient Gx (II.1).....	7
Rumus Gradient Gy (II.2).....	7
Rumus magtitude dari gradient (II.3).....	7
<i>Angle</i> dari gradient (II.4).....	7
Euclidean Distance (II.5).....	8
Rumus <i>Triplet Loss</i> (II.6).....	9
Rumus Euclidean Distance (IV.1).....	42

BAB I

PENDAHULUAN

I.1 Latar Belakang

Dalam beberapa tahun terakhir, orang-orang mulai menggunakan berbagai cara untuk meningkatkan pengawasan terhadap orang yang tidak dikenal yang masuk ke lingkungan rumah. Salah satunya adalah menggunakan kamera CCTV. Kamera CCTV ini digunakan sebagai cara untuk mengawasi lingkungan rumah seseorang dari orang yang tidak dikenal ataupun hal-hal mencurigakan lainnya.

Pengawasan dengan menggunakan kamera CCTV saat ini membutuhkan seseorang yang bertugas mengawasi secara langsung dan terus-menerus melalui monitor untuk mengetahui apakah ada orang yang terekam kamera CCTV adalah orang yang pemilik rumah kenali atau tidak.

Kekurangan pengawasan rumah dengan kamera CCTV yang saat ini sering digunakan ialah belum dapat memberikan pengawasan yang optimal mulai dari belum dapatnya mendeteksi wajah secara otomatis hingga mengenali apakah wajah tersebut dikenal atau tidak. Maka dari itu, pemilik atau petugas harus selalu memantau secara langsung dan terus menerus segala aktifitas yang terekam untuk dapat mengetahui apakah ada orang yang dikenal atau tidak yang terekam oleh kamera pengawas. Terkadang pengawas sering lalai atau tidak berada pada tempat pengawasan sehingga mengakibatkan adanya beberapa kejadian yang luput dari pengawasan. Dengan demikian, ketidakmampuan seseorang dalam mengawasi kamera CCTV secara langsung dan terus menerus menjadi faktor utama belum optimalnya pengawasan yang ada pada CCTV.

Saat ini, telah hadir sebuah teknologi berupa kecerdasan buatan atau biasa disebut *Artificial Intelligence* (AI). Teknologi ini dikembangkan untuk membantu dalam melakukan sesuatu yang sulit untuk dilakukan oleh manusia. AI sudah berkembang sangat pesat dan dapat ditambahkan ke berbagai sistem. Salah satu cabang dari AI

yaitu *Deep Learning* dan *Computer Vision* dapat dimanfaatkan khususnya pada penggunaan *object detection* dan *object recognition* [1].

Saat ini dibutuhkan penambahan baru untuk mengoptimalkan cara kerja pengawasan rumah menggunakan kamera pengawas dalam mendeteksi dan mengenali wajah seseorang secara otomatis. Salah satu teknologi yang dapat diterapkan adalah *automatic object detection* dan *automatic object recognition* dengan menambah satu komponen baru yaitu single board komputer. Setiap citra yang ditangkap oleh kamera akan diolah menggunakan metode *object detection* dan *object recognition*. Objek yang dideteksi ini berupa wajah manusia. Selain dari itu, dibutuhkan juga sebuah media untuk mengirimkan informasi hasil pengolahan tersebut secara otomatis.

Maka pada tugas akhir dikembangkan sebuah sistem pendeteksi orang tidak dikenal menggunakan metode pengenalan wajah pada kamera pengawas. Sistem ini nantinya akan mengirimkan informasi hasil pengolahan secara otomatis dengan media chatbot kepada pemilik rumah ketika ditemukan ada orang tak dikenal. Diharapkan sistem ini dapat memudahkan penggunaanya dalam mengawasi rumah tanpa perlu mengawasi secara langsung.

I.2 Rumusan Masalah

Berdasarkan latar belakang, maka perumusan masalah adalah sebagai berikut.

1. Tidak efektifnya pengawasan rumah yang ada saat ini.
2. Butuh biaya yang besar untuk mempekerjakan petugas untuk mengawasi.

I.3 Tujuan

Tujuan dikembangkannya sistem ini adalah:

1. Memudahkan pemilik rumah dalam melakukan pengawasan rumah karena tidak perlu mengawasi secara langsung dan terus menerus.
2. Memberi kemudahan pada pemberian informasi kepada pemilik rumah ketika ada orang tidak dikenal.
3. Membuat purwarupa sistem mendeteksi orang tidak dikenal yang mempunyai fitur pengenalan wajah.

I.4 Ruang Lingkup

Adapun ruang lingkup dari sistem yang dikembangkan yaitu :

1. Sistem dapat melakukan pendeteksian dan pengenalan wajah
2. Kamera pengawas hanya digunakan dalam ruang lingkup rumahan
3. Sistem dapat mengenali wajah dari citra wajah yang terdeteksi menggunakan data yang tersedia pada *database*.
4. Hasil pengolahan akan dikirim melalui media chatbot berupa pemberitahuan keberadaan orang yang tidak dikenali sistem kepada *smartphone* pengguna
5. Pengolahan data yang digunakan sistem akan dilakukan dengan media chatbot

I.5 Rumusan Masalah

Batasan masalah dari sistem ini, yaitu:

1. Wajah yang dideteksi haruslah menghadap tegak lurus terhadap kamera.
2. Citra wajah yang didaftarkan untuk disimpan ke dalam *database* haruslah citra wajah yang menghadap tegak lurus terhadap kamera.
3. Tegak lurus yang dimaksud adalah posisi wajah yang menghadap ke depan kamera secara sejajar.
4. Citra wajah yang dapat ditambahkan sebagai data baru orang yang dikenal hanyalah citra yang terakuisisi oleh kamera pengawas

BAB II

TINJAUAN PUSTAKA

II.1. Karya Ilmiah Sejenis Sebelumnya

Karya ilmiah sejenis sebelumnya mengenai pengenalan wajah dideskripsikan pada Tabel II.1 yang menjadi acuan dalam pengembangan sistem.

Tabel II.1 Karya ilmiah sejenis sebelumnya

No	Tahun	Judul Karya Ilmiah dan Penulis	Obyek	Metode	Variabel
1	2016	<i>Notification System Based on Face Detection and Recognition: A Novel Approach</i> (A. A. Al-Bakeri., A. A. Basuhail)	Wajah pemilik salah satu rumah	<i>Principle Component Analysis</i> (PCA)	<i>Face Recognition, Face Detection, SMTP</i>
2	2016	<i>Motion Detection Camera Security System with Email Notifications and Live Streaming Using Raspberry Pi</i> (Zafar, Sundas., Carranza, Aparicio)	Sistem keamanan		<i>Motion Detecting, Email, Notification, Live Streaming.</i>
3	2017	<i>Smart Surveillance System using Raspberry Pi and Face Recognition</i>	Wajah manusia dengan wajah manusia pada database	<i>Local Binary Pattern</i> (LBP)	<i>Motion, human face, PIR Sensor</i>

Bahasan yang berkaitan dengan rujukan pada Tabel II.1 dijelaskan secara ringkas sebagai berikut.

1. Karya ilmiah yang berjudul “*Notification System Base on Face Detection and Recognition: A Novel Approach*” menjelaskan tentang bagaimana kamera mendeteksi dan mengenali wajah dari citra yang diakuisisi serta mengirimkan sebuah pemberitahuan melalui e-mail kepada pemilik rumah. Metode yang

digunakan adalah metode PCA. Pada metode ini, wajah haruslah tegak lurus dengan kamera secara langsung agar bisa terdeteksi sebagai wajah dan bisa dikenali. Pengiriman pemberitahuan melalui e-mail dilakukan dengan menggunakan SMTP.

2. Karya ilmiah selanjutnya yang berjudul “*Motion Detection Camera Security System with Email Notifications and Live Streaming Using Raspberry Pi*” menjelaskan bagaimana membuat sebuah kamera yang diintegrasikan dengan *single board computer* yaitu Raspberry Pi dan mengirimkan sebuah pemberitahuan melalui e-mail dengan SSMTP. Raspberry Pi digunakan karena harganya yang relatif murah yaitu sebesar 29 dollars dibandingkan dengan *single camera* yang harganya 100 dollars. Harga tersebut bahkan telah mampu untuk membuat sistem yang dapat mengirimkan pemberitahuan melalui e-mail dan *live streaming*.
3. *Paper* yang berjudul “*Smart Surveillance System using Raspberry Pi and Face Recognition*” menjelaskan tentang pembuatan sebuah sistem pengawasan yang cerdas, cerdas yang dimaksud adalah ketika kamera yang pada mulanya hanya merekam aktivitas saja, akan berubah menjadi kamera yang dapat mengenali dan mendeteksi wajah serta mengirimkan pemberitahuan melalui sebuah aplikasi bernama Pushetta. Dalam rujukan ini dijelaskan bahwa untuk mendapatkan hasil pengenalan yang baik, perlu adanya wajah yang telah disimpan sebanyak 70 - 80 gambar dengan kondisi yang berbeda-beda, mulai dari posisi wajah yang diambil, ekspresi, dan pencahayaan. Sistem yang dibuat ini menggunakan Raspberry Pi, kamera 8mp, dan PIR sensor agar biaya yang dikeluarkan lebih murah dan menggunakan metode LBP dalam pengenalan wajahnya.

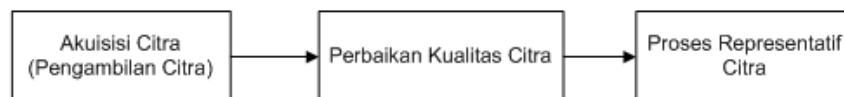
II.2 Dasar Teori

Teori yang menjadi acuan dalam pengembangan sistem pendeteksi orang tidak dikenal dibahas lebih lanjut pada subbab berikut.

II.2.1 Pengolahan Citra Digital

Citra didefinisikan sebagai fungsi dari dua variabel misalnya $a(x,y)$ dimana a sendiri sebagai amplitu (misalnya kecerahan) citra pada kordinat (x,y) [2]. Pada pengolahan citra, gambar menjadi komponen utama. Namun, seringkali gambar yang kita dapat berkualitas baik atau buruk, kualitas buruk disebabkan oleh pengaruh cahaya, atau terdapat derau (*noise*), terlalu kontras, kurang tajam (*blur*), dan lain sebagainya yang menyebabkan hilangnya informasi yang ingin disampaikan. Sehingga dibutuhkan sebuah proses untuk memperbaiki citra agar informasi yang disampaikan tidak akan hilang.

Pengolahan citra merupakan proses mengolah piksel-piksel di dalam citra digital yang bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Proses pengolahan citra secara berurutan ditunjukkan pada Gambar II.1.

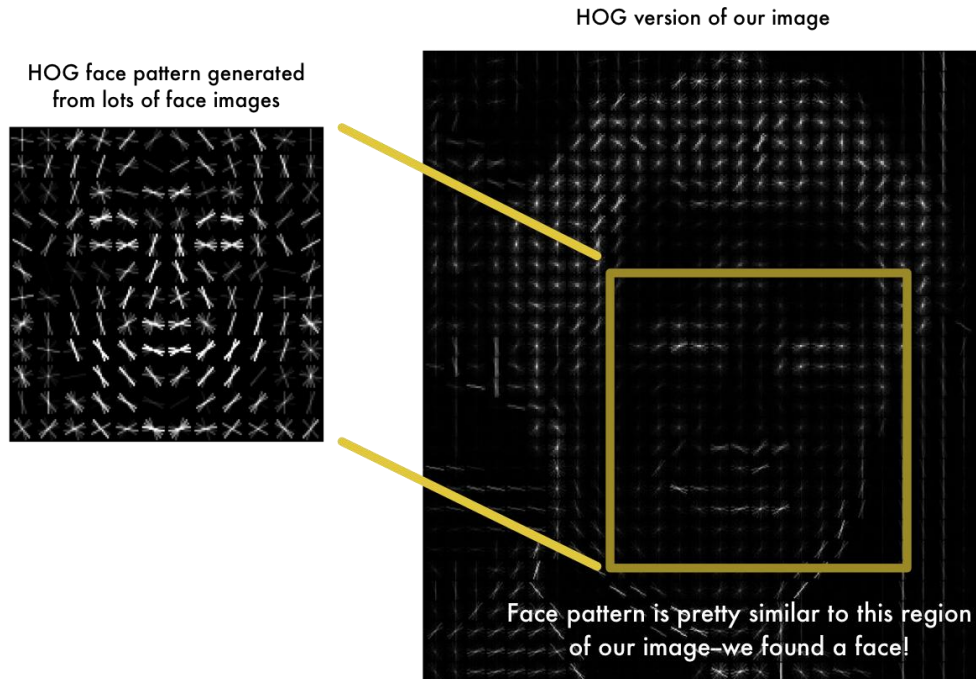


Gambar II.1 Proses pengolahan citra

II.2.2 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) adalah teknik ekstraksi fitur yang menghitung gradien berorientasi dari sebuah gambar yang terdeteksi oleh detektor gradien [3].

Tahap awal pada metode HOG ini adalah melakukan normalisasi terhadap gamma dan warna. Setelah itu adalah tahap membanding piksel citra tertentu dengan piksel citra sekitarnya yang bertujuan untuk membanding kegelapan antara satu piksel citra dengan yang lainnya, lalu nantinya akan digambarkan sebuah panah dengan arah ke gambar dengan piksel yang lebih gelap, dan dilakukan berulang terhadap setiap piksel, sehingga nantinya akan menghasilkan seperti pada Gambar II.2.



Gambar II.2 HOG *face pattern*

Proses selanjutnya adalah pembagian citra ke dalam *cell* dan membentuk histogram dari tiap *cell*. Setelah itu dilakukan normalisasi terhadap setiap blok histogram. Setelah mendapatkan HOG pada pendeteksian citra selanjutnya melakukan pengklasifikasian dengan menggunakan SVM untuk mendapatkan hasil apakah citra tersebut merupakan wajah manusia atau bukan wajah manusia.

Rumus matematika yang mendeskripsikan HOG dituliskan pada rumus sebagai berikut:

$$G_x = I(x + 1, y) - I(x - 1, y) \quad (\text{II.1})$$

$$G_y = I(x, y + 1) - I(x, y - 1) \quad (\text{II.2})$$

dimana $I(x, y)$ adalah intensitas dari pixel pada posisi (x, y) serta G_x dan G_y adalah komponen horizontal dan vertical dari gradient, dituliskan sebagai berikut:

$$M(x, y) = \sqrt{G_x^2 + G_y^2} \quad (\text{II.3})$$

$$\theta(x, y) = \tan^{-1} \frac{G_y}{G_x} \quad (\text{II.4})$$

Dimana $M(x, y)$ adalah magnitude dari gradient, $\theta(x, y)$ adalah *angle* dari gradient pada lokasi yang diberikan.

II.2.3 Support Vector Machine

Support Vector Machine (SVM) adalah metode pengambilan keputusan yang menentukan garis pemisah antara dua kelas.

Karena karakteristik SVM classifier ini adalah linear classifier, pada tugas akhir ini SVM classifier digunakan untuk mengambil keputusan dalam proses pendeteksian wajah pada metode HOG untuk mengklasifikasikan apakah dalam sebuah citra yang tertangkap kamera terdapat citra wajah atau tidak.

II.2.4 Machine Learning dan Deep Learning

Machine learning adalah sebuah algoritma khusus yang dapat memberi tahu sesuatu tanpa harus diprogram secara berulang kali oleh manusia. Agar bisa mencapai itu semua, *machine learning* membutuhkan data latih sehingga nantinya komputer dapat belajar berdasarkan data latih tersebut agar bisa mengeluarkan *output* yang sesuai. Pada *machine learning* ada dua macam kategori utama yaitu *supervised learning* dan *unsupervised learning*.

Deep learning adalah bagian dari *machine learning* yang berfokus pada pembelajaran representasi data. Secara struktur, *deep learning* menggunakan konsep jaringan syaraf tiruan. Pada *deep learning* struktur tersebut lebih kompleks dan menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam.

II.2.5 Euclidean Distance

Euclidean distance adalah perhitungan jarak di antara dua titik pada ruang euclidean. Dalam koordinat kartesius, jika sebuah titik $p = (p_1, p_2, \dots, p_n)$ dan $q = (q_1, q_2, \dots, q_n)$ adalah dua titik dalam Euclidean n -space. Maka rumus yang diberikan untuk menghitung jarak dari p ke q ditunjukkan pada rumus (II.5).

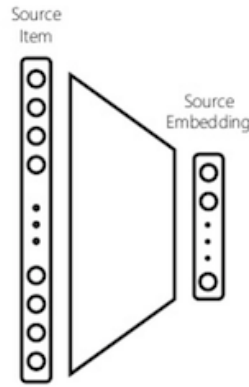
$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + \dots + (p_n - q_n)^2} \quad (\text{II.5})$$

Pada pengembangan sistem ini, euclidean distance digunakan untuk menghitung perbandingan titik antara titik pada wajah yang ada dalam *database* dengan titik yang dihasilkan dari hasil *encoding* wajah yang tertangkap oleh kamera pengawas.

II.2.6 Face Recogniton

Face recognition (pengenalan wajah) adalah salah satu aplikasi biometrik untuk mengidentifikasi secara otomatis seseorang di depan kamera. Proses ini merupakan salah satu penerapan dari *Computer Vision* (Visi computer) menggunakan proses pengolahan citra digital.

Metode yang digunakan untuk pengenalan wajah adalah dengan *deep Learning* yang dikembangkan oleh Davis E King. Metode ini sudah terbukti memiliki akurasi sebesar 99.38 % dengan dataset *Labeled Faces in the Wild* (LFW) [4] .

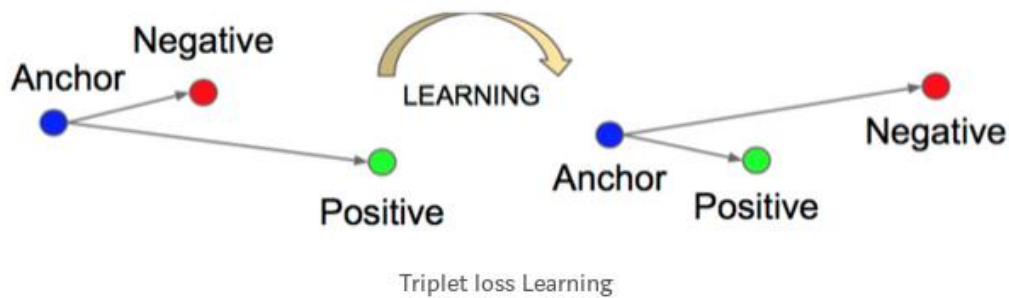


Gambar II.3 Representasi citra wajah menjadi vektor n dimensi

Proses pengenalan wajah modern secara konseptual akan merepresentasikan setiap citra wajah menjadi vektor n dimensi. Gambaran dari representasi dari citra wajah menjadi vektor n dimensi ditunjukkan pada Gambar II.3. Vektor ini berisi 128 dimensi yang berisi jarak dan nilai dari setiap citra wajah. Pada *paper* pengenalan wajah yang ditulis oleh Google FaceNet pada 2015 dilakukan sebuah metode bernama “triplet loss” yang dapat meminimalisir jarak antar vektor pada data contoh positif dan memaksimalkan jarak antar vektor pada data contoh negatif. Berikut adalah rumus dari “Triplet Loss”.

$$\sum_i^N [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]_+ \quad (\text{II.6})$$

Berdasarkan rumus (II.6) maka gambaran dari “Triplet Loss Learning” ditunjukkan pada Gambar II.4.



Gambar II.4 Triplet loss learning

Karena yang dihasilkan berupa vektor, maka menjadi masuk akal ketika melakukan penghitungan jarak antar vektor. Hasil representasi akan menghasilkan nilai minimum jika jarak antar vektor wajah ini dihasilkan dari orang yang sama dan akan menghasilkan nilai maksimum jika jarak antar vektor dihasilkan dari representasi orang yang berbeda.

II.2.7 Chatbot

Chatbot atau chatterbot merupakan sebutan sederhana dari *Intelligent Conversational Agents* [5]. Chatbot telah merubah cara manusia berkomunikasi pada hal penjualan, pemasaran, layanan pelanggan dan lainnya. Selama beberapa tahun terakhir, cepatnya penyebaran teknologi platform *Instant Messaging* atau IM (seperti Facebook Messenger, WhatsApp, Line, dan lain-lain.) telah merubah bagaimana orang berinteraksi.

Messenger adalah salah satu platform IM yang dikembangkan oleh perusahaan media sosial Facebook. Selain digunakan untuk percakapan, platform ini menyediakan seperangkat alat untuk membangun bot pada Facebook page.

Pada sistem yang dikembangkan, chatbot berperan sebagai media interaksi antara pengguna dan sistem. Chatbot hanya akan memanfaatkan fitur messaging dan webview yang tersedia di aplikasi messenger untuk mengolah interaksi yang dilakukan oleh pengguna (pemilik rumah / penjaga) pada Facebook page yang telah disiapkan. Pengguna dapat menerima pemberitahuan ketika adanya orang yang tidak dikenal dan mengolah data orang yang dikenal melalui percakapan dengan chatbot.

II.2.8 Internet of Things

Internet of Things atau IOT adalah sebuah konsep yang bertujuan untuk memungkinkan sebuah alat dapat terhubung secara terus-menerus, kapan saja, di berbagai tempat, dengan sesuatu atau seseorang menggunakan sebuah penghubung atau jaringan tertentu [6]. IOT tidak hanya sebatas hubungan antar komputer karena pada jaman sekarang IOT telah berinovasi dengan masuk ke dalam kategori *smartphone*, peralatan rumah, mainan, kamera, alat kesehatan, maupun alat industri. Biasanya alat yang menerapkan konsep IOT berkomunikasi melalui REST API. Sistem yang dikembangkan menerapkan konsep IOT, dengan komponen sebagai berikut.

1. *Single Board Computer*

Single board computer ini digunakan untuk melakukan operasi pendeteksian wajah, pengenalan wajah, manajemen data, dan mengirimkan *request* pada chatbot untuk mengirimkan sebuah pemberitahuan kepada pengguna atau pemilik rumah.

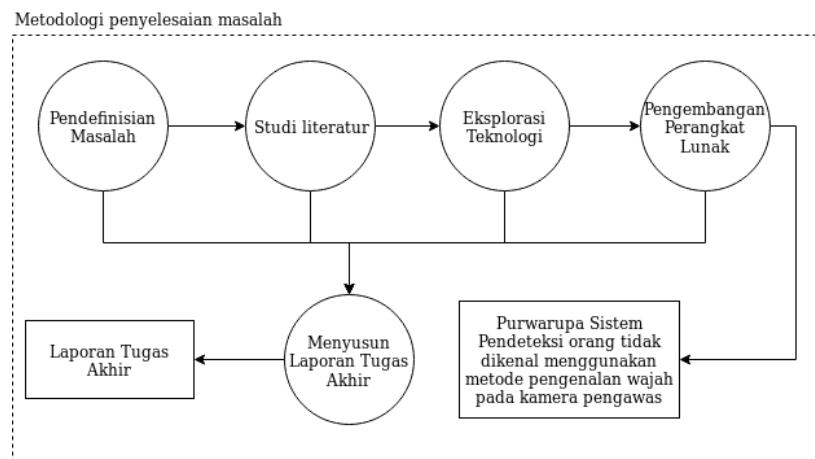
2. Kamera Pengawas

Kamera pengawas ini digunakan untuk merekam setiap aktivitas yang terjadi dan menampilkannya dalam sebuah monitor. Pada sistem yang dikembangkan, hasil rekaman dari kamera pengawas akan diolah pada *single board computer*.

BAB III

METODOLOGI PENELITIAN

Metodologi yang digunakan dalam penyelesaian tugas akhir ini ditunjukkan pada Gambar III.1.



Gambar III.1 Diagram metodologi penyelesaian masalah

Penjelasan mengenai komponen yang ada pada Gambar III.1 dijelaskan pada Tabel III.1.

Tabel III.1 Simbol diagram metodologi penyelesaian masalah

Simbol	Keterangan
.....	Ruang lingkup proses
○	Proses
□	Masukan atau keluaran proses
→	Iterasi selanjutnya

Terdapat lima proses utama yang dikerjakan dan hasil akhirnya berupa purwarupa sistem dan laporan tugas akhir.

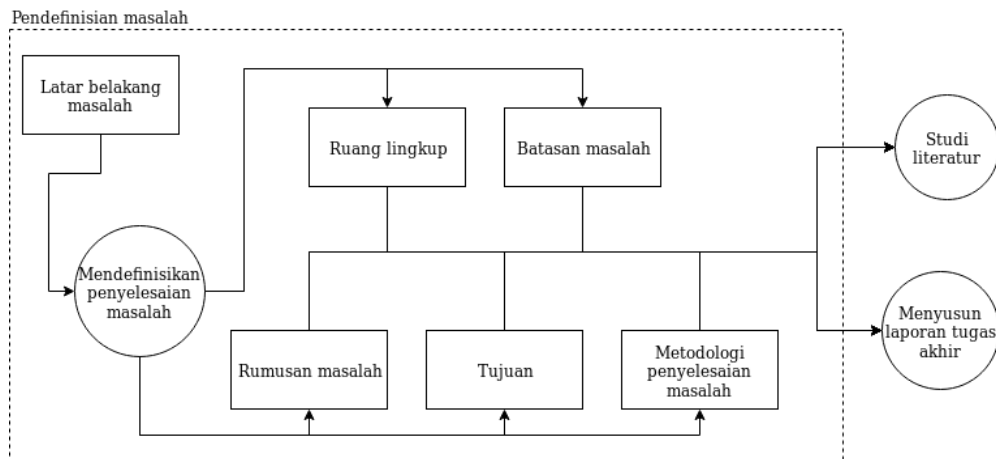
Purwarupa yang dibangun hanya mencakup beberapa hal, yaitu:

1. Sistem dapat mengakuisisi citra melalui kamera pengawas
2. Sistem dapat mendeteksi dan mengenali wajah secara otomatis
3. Sistem dapat mengirimkan pemberitahuan melalui chatbot ketika ada orang yang tidak dikenal tertangkap kamera
4. Dan sistem dapat melakukan manajemen data melalui chatbot dengan proses tambah, lihat, hapus, dan ubah data orang yang dikenal

Penjelasan mengenai metodologi penyelesaian masalah ini dibahas lebih detail pada subbab III.1.

III.1 Pendefinisian Masalah

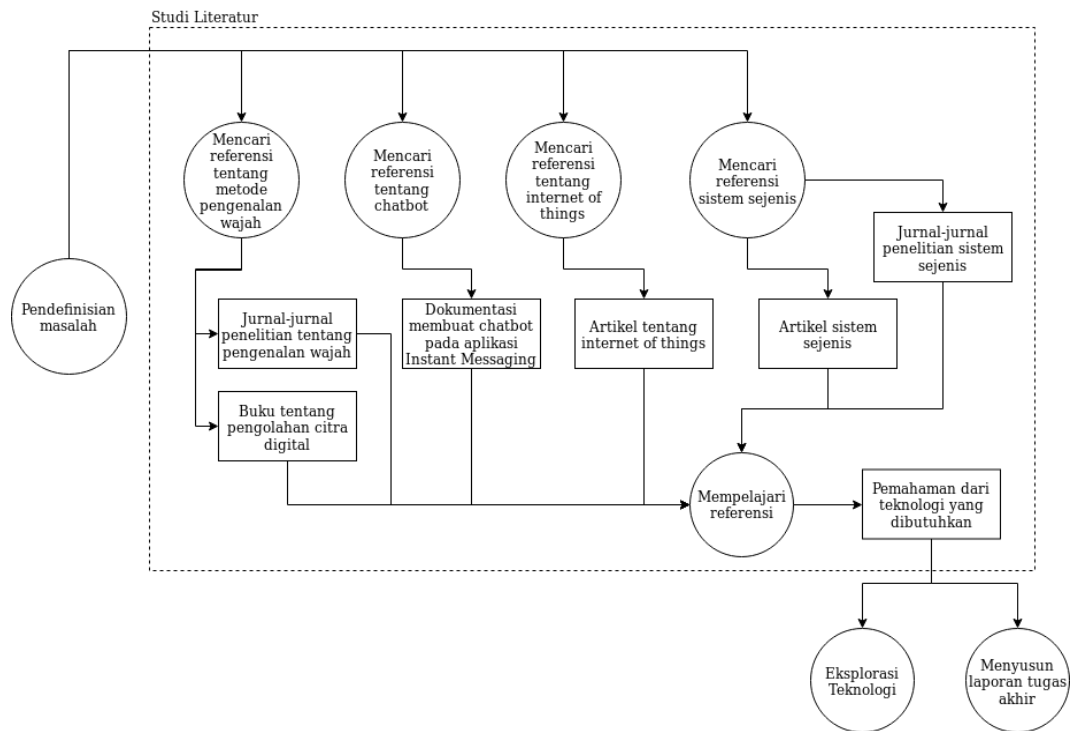
Pada tahap ini, pendefinisian masalah yang dilakukan menghasilkan sebuah *output* berupa rumusan masalah, ruang lingkup, batasan masalah, dan tujuan. Gambar III.2 menunjukkan proses yang dilakukan.



Gambar III.2 Diagram pendefinisian masalah

III.2 Studi Literatur

Studi literatur ditunjukkan pada Gambar III.3. Proses yang berkaitan dengan studi literatur meliputi mencari referensi tentang metode pengenalan wajah, chatbot, *internet of things* (IOT), dan referensi dari sistem sejenis.



Gambar III.3 Diagram studi literatur

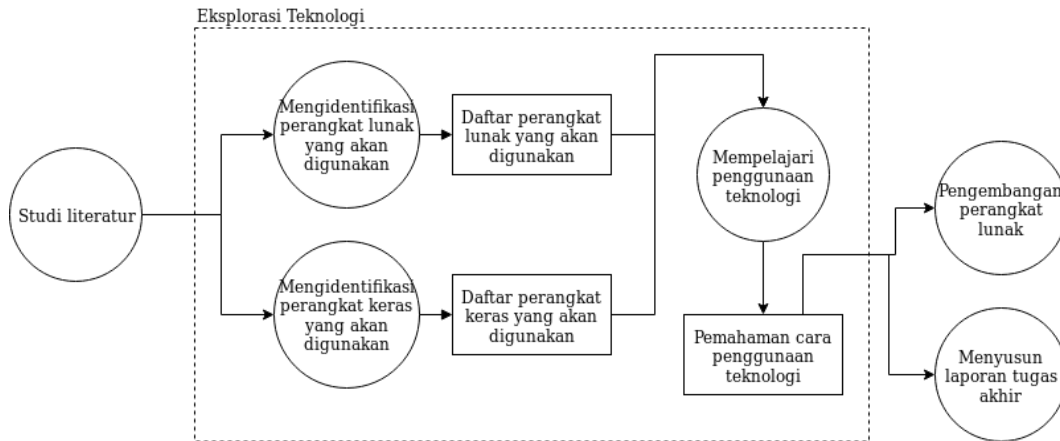
Proses pencarian referensi metode pengenalan wajah mendapatkan hasil berupa metode yang dapat digunakan bernama DeepFace yang menggunakan *Deep Learning*. Untuk mengimplementasikan metode ini dibutuhkan beberapa langkah mulai dari pendeteksian wajah, proyeksi dan transformasi wajah, representasi wajah, hingga perbandingan wajah.

Mengenai pencarian referensi tentang chatbot, hasil yang didapatkan adalah dokumentasi penggunaan chatbot pada *instant messaging* (IM).

Pencarian referensi mengenai IOT hasilnya berupa pemanfaatan IOT terhadap sistem yang dikembangkan. Proses selanjutnya mengenai pencarian sistem sejenis mendapatkan referensi mengenai penggunaan Raspberry Pi yang dapat digunakan sebagai *single board computer* pada sistem yang dikembangkan dengan Raspberry Camera sebagai perangkat keras untuk mengakuisisi citra.

III.3 Eksplorasi Teknologi

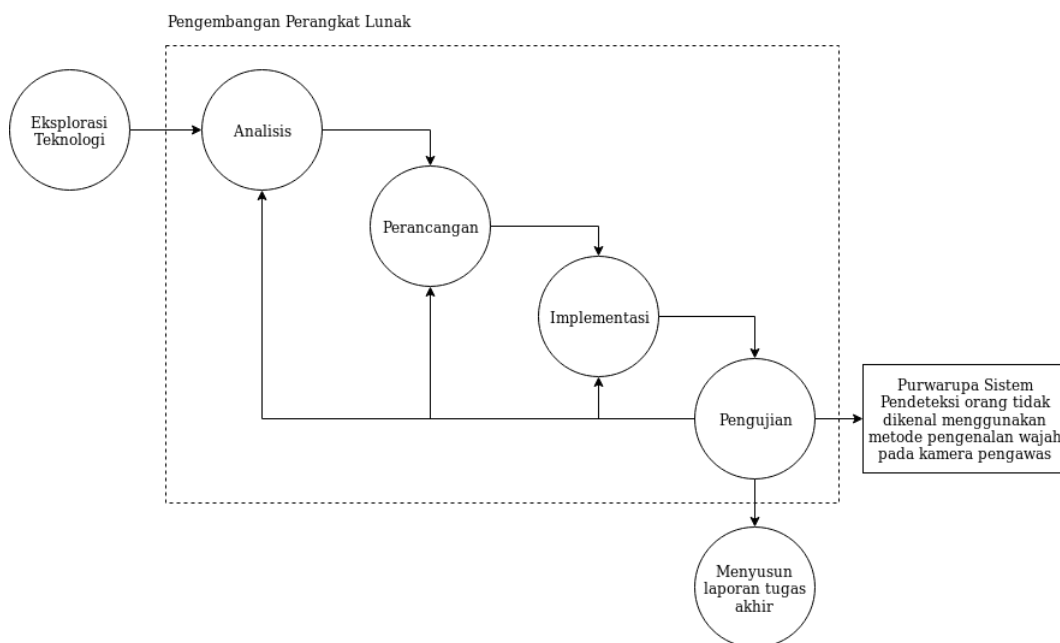
Proses eksplorasi teknologi dilakukan untuk mengetahui perangkat keras dan lunak yang cocok digunakan serta menentukan arsitektur sistem yang akan dibangun. Proses ini ditunjukkan pada Gambar III.4.



Gambar III.4 Diagram eksplorasi teknologi

III.4 Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak yang diterapkan adalah model *waterfall* yang ditunjukkan pada Gambar III.5.

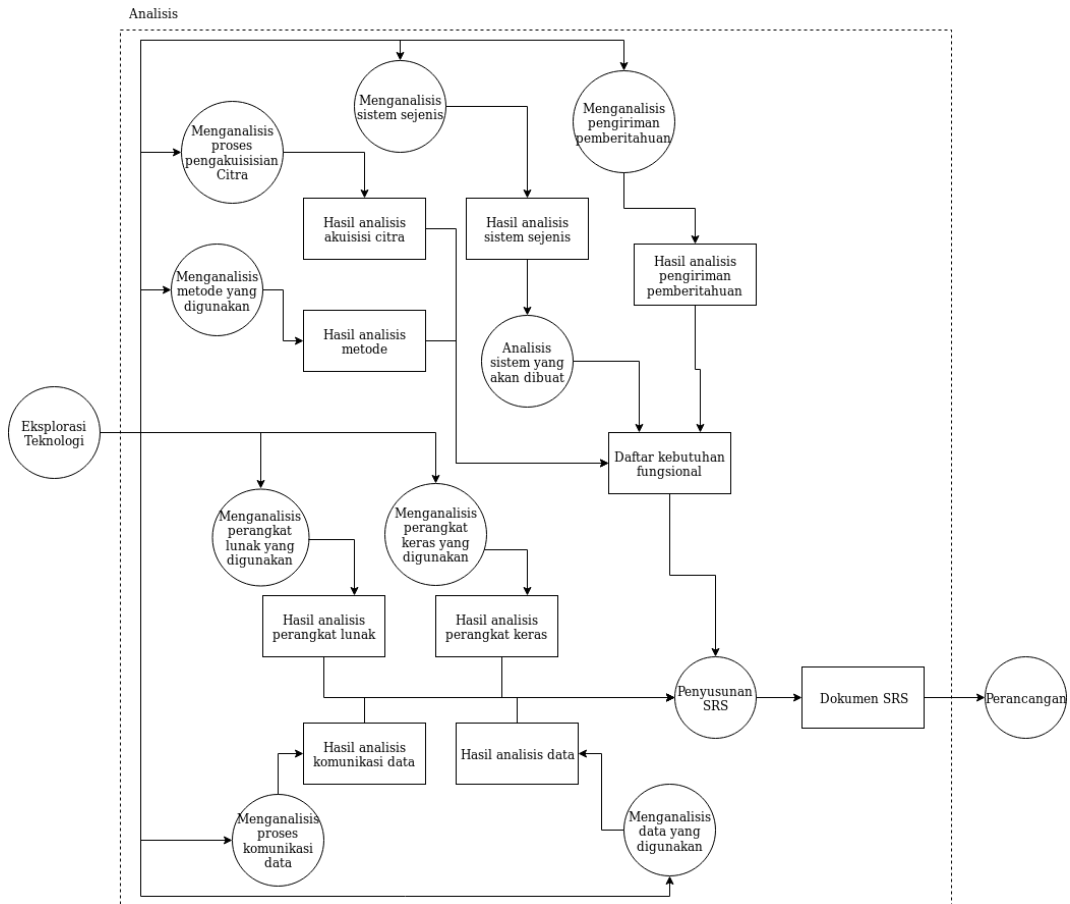


Gambar III.5 Diagram pengembangan perangkat lunak

Pada model ini, terdapat empat tahap yang dilakukan, yaitu sebagai berikut.

1. Analisis. Hal-hal yang dianalisis berupa analisis sistem yang sudah ada, proses akuisisi citra, metode yang digunakan dalam pengembangan sistem, proses pengiriman pemberitahuan, perangkat lunak dan perangkat keras yang digunakan, proses komunikasi data, serta analisis data yang digunakan dan

diolah. Hasil dari proses ini diantaranya adalah terdefinisinya *requirement*, *business rules*, dan dokumen *software requirement specification* (SRS). Tahapan lebih detilnya ditunjukkan pada Gambar III.6.



Gambar III.6 Diagram analisis pada perancangan perangkat lunak

2. Perancangan. Tahap ini dilakukan berdasarkan hasil analisis dan *Software Requirement Specification* (SRS) dan ditunjukkan pada Gambar III.7. Hal-hal yang dirancang adalah sebagai berikut.

a. Perancangan arsitektur sistem

Perancangan arsitektur sistem digambarkan dengan sebuah diagram *logical* arsitektur yang menjelaskan proses apa saja yang dilakukan pada sistem yang dibuat.

b. Alur kerja pada sistem

Alur kerja pada sistem digambarkan dengan menggunakan *activity* diagram dan *state machine* diagram menjelaskan secara detail bagaimana sistem akan bekerja dari awal sampai akhir nantinya.

c. Perancangan *class* model

Perancangan *class* model digambarkan menggunakan *class* diagram yang nantinya akan menggambarkan *class* apa saja yang dibangun, serta atribut dan *method* yang ada pada *class* tersebut.

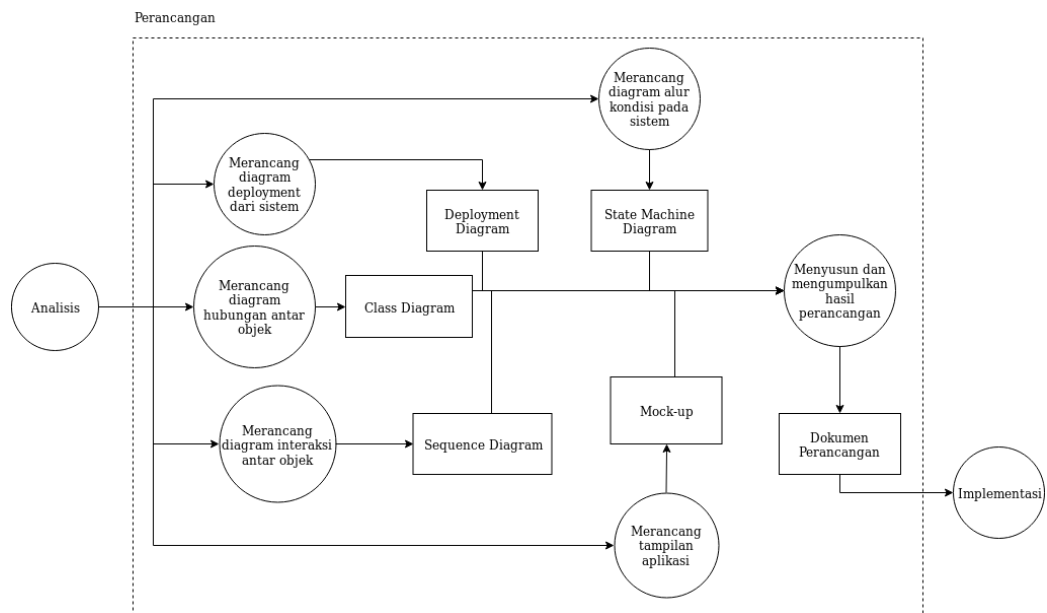
d. Interaksi antar objek

Interaksi antar objek akan digambarkan menggunakan *sequence* diagram yang bertujuan untuk menjelaskan secara detail bagaimana interaksi antar objek akan berlangsung pada sistem yang dibuat.

e. Tampilan aplikasi

Perancangan tampilan sistem akan digambarkan melalui sebuah mock-up. *Tools* yang digunakan adalah menggunakan wireframe. Tampilan yang dirancang ini meliputi.

- Tampilan chatbot ketika menampilkan pemberitahuan
- Tampilan chatbot pada fitur menambahkan data orang yang dikenal
- Tampilan chatbot pada fitur mengubah data orang yang dikenal
- Tampilan chatbot pada fitur menghapus data orang yang dikenal
- Tampilan chatbot pada fitur melihat daftar orang yang dikenal



Gambar III.7 Diagram perancangan pada pengembangan perangkat lunak

Hasil dari tahap perancangan adalah *logical* arsitektur, *deployment diagram*, *activity diagram*, *state machine diagram*, *class diagram*, *sequence diagram*, dan *mock-up* yang dijelaskan secara detil pada bab V.

3. Implementasi, tahap ini digambarkan secara detail pada . Pengembangan sistem ini menggunakan bahasa pemrograman python dan *library* OpenCV untuk pengolah citra digital dan *library* dlib untuk pengenalan wajah. Adapun sumber daya dan proses yang digunakan terbagi menjadi dua yaitu, sumber daya dan proses yang diambil dari internet secara *open source*, maksud dari *open source* ini adalah diambil dari internet secara gratis tanpa berbayar serta sumber daya atau proses yang dibuat sendiri. Berikut adalah penjelasan singkat mengenai sumber daya dan proses yang diambil dari internet secara *open source*.

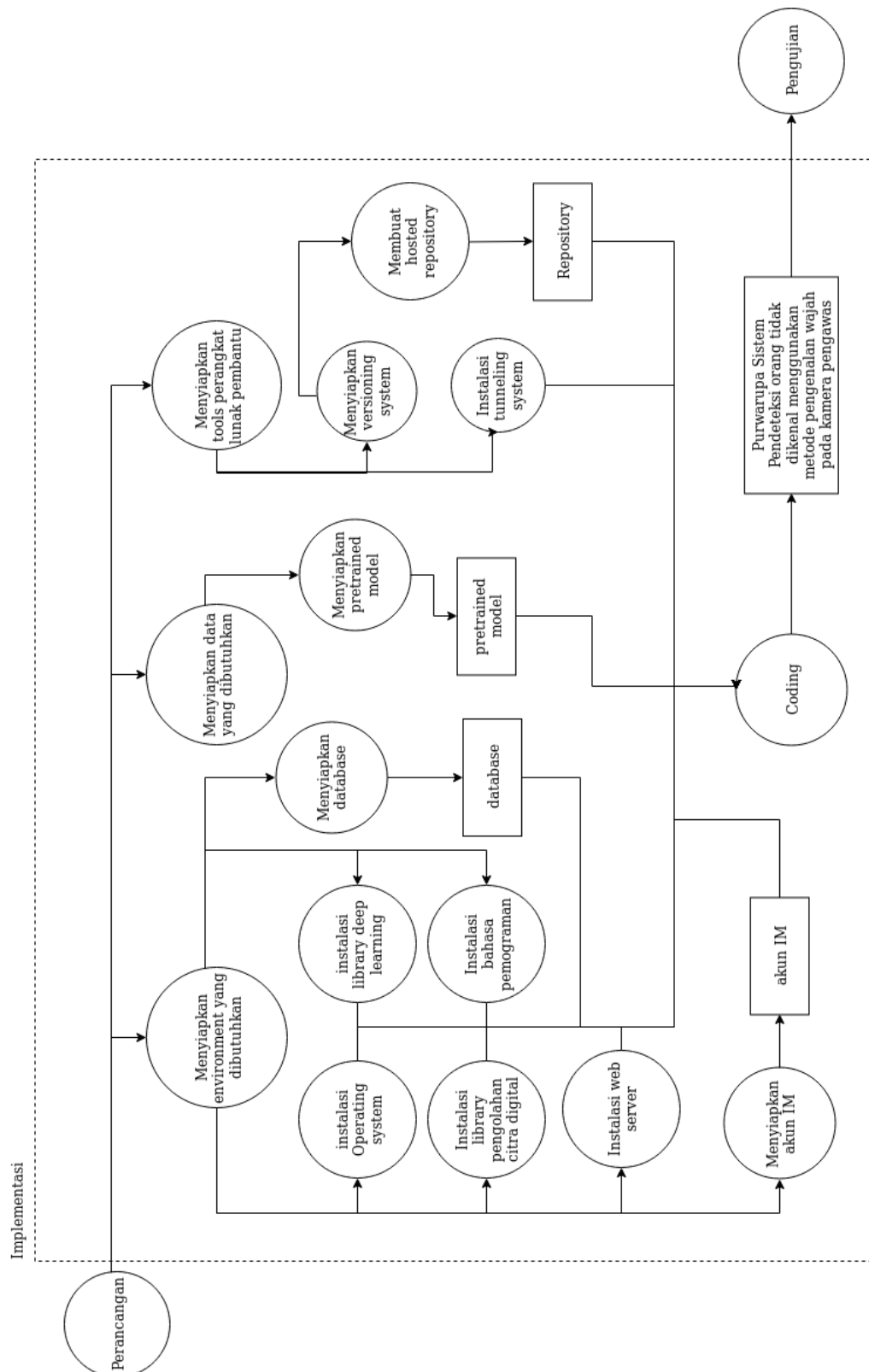
- a. *Library* pengolahan citra digital, machine learning, dan library pendukung.

Library yang dipakai untuk melakukan operasi pengolahan citra digital adalah OpenCV (Open Source Computer Vision Library). Pembuatan sistem juga memerlukan *library machine learning* yaitu *library* Dlib. Dlib adalah *toolkit* yang berisi algoritma *machine learning* dan alat lainnya untuk membuat aplikasi yang membutuhkan perhitungan kompleks dan ditujukan untuk mengatasi masalah dalam kehidupan sehari-hari. Adanya *library* pendukung untuk melakukan komputasi matematis seperti numpy dan *library* untuk membuat *web server* seperti flask.

- b. *Pretrained model*

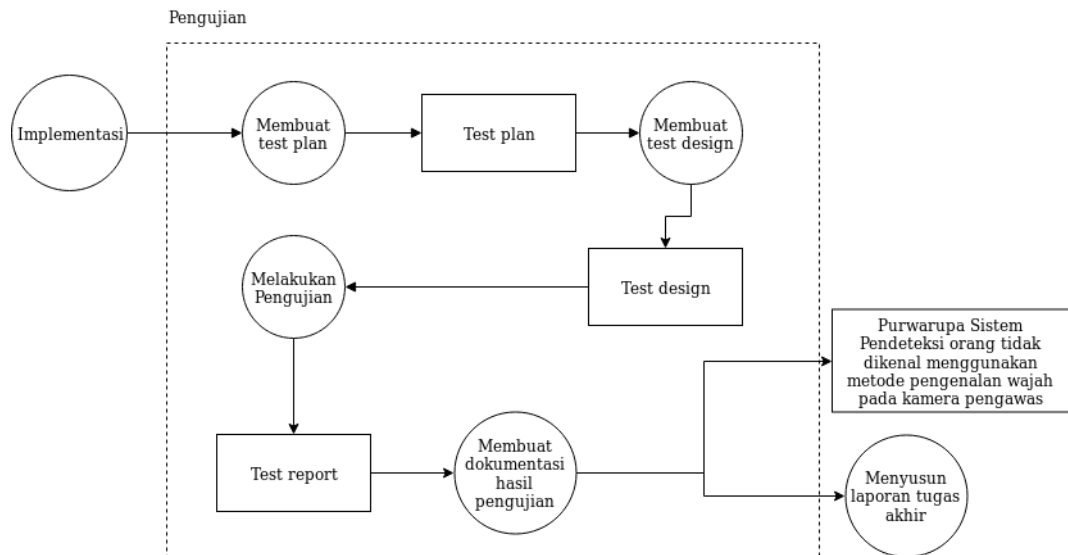
Pretrained model adalah model *machine learning* yang sudah terlebih dahulu dilatih menggunakan algoritma tertentu pada datasets tertentu sehingga memungkinkan penggunaanya untuk melakukan representasi data dengan berbagai macam tujuan. Model yang digunakan diantaranya model pendeteksian wajah, model untuk mencari *face landmark*, dan model untuk melakukan representasi wajah.

Adapun proses yang dibangun sendiri meliputi pembuatan chatbot, proses akuisisi citra dan *preprocessing*, serta integrasi proses antara proses pengenalan wajah, manajemen data, dan integrasi antar komponen. Proses implementasi secara detil ditunjukkan pada Gambar III.8.



Gambar III.8 Implementasi pada pengembangan perangkat lunak

4. Pengujian, tahap ketika sistem yang dibuat nantinya diuji apakah sudah sesuai dengan perancangan dan *requirement* yang sudah ditentukan sebelumnya. Pengujian yang dilakukan hanyalah *unit testing* dengan pendekatan *black box testing*. Untuk menunjang tahap pengujian ini, maka nantinya akan dibuat dokumen *test plan*, *test design*, *test case*, *test procedure*, dan *test report*. Tahapan pengujian digambarkan pada Gambar III.9.



Gambar III.9 Diagram pengujian pada pengembangan perangkat lunak

BAB IV

REQUIREMENT ANALYSIS AND DEFINITION

Bab ini menjelaskan tentang analisis terkait kebutuhan sistem yang akan dibangun. Analisis yang dilakukan bertujuan untuk mendapatkan gambaran secara detail dan mengidentifikasi kebutuhan yang digunakan dalam pengembangan sistem. Hal-hal yang dianalisis akan dijelaskan lebih lanjut pada subbab berikut.

IV.1 Analisis Sistem Sekarang

Analisis sistem sekarang atau yang sedang berjalan pada pengawasan rumah yaitu dengan menggunakan CCTV analog. Analisis lebih detil dijelaskan pada penjelasan sebagai berikut.

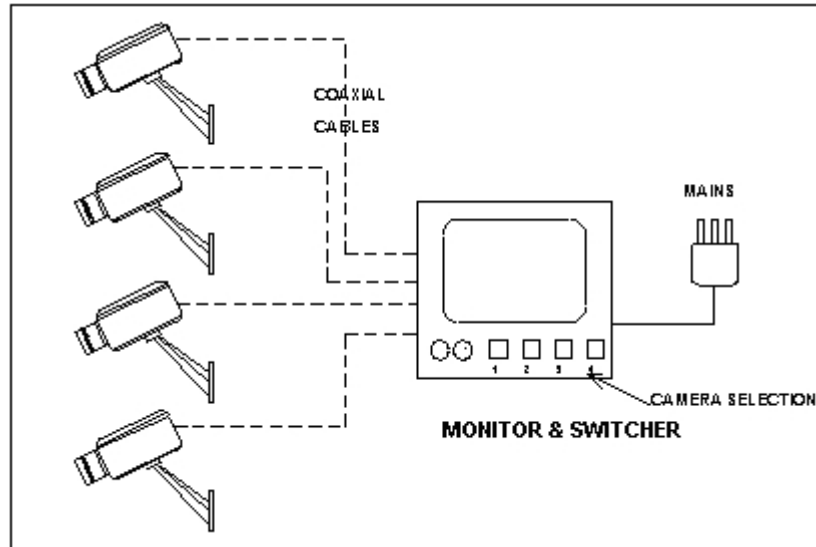
IV.1.1 Kamera CCTV Analog

CCTV analog adalah sistem kamera CCTV yang mengirimkan video dengan kabel coaxial atau kabel UTP yang ditransmisikan ke dalam DVR (Digital Video Recorder). DVR adalah perangkat penyimpanan rekaman video pada kamera CCTV. Kamera CCTV analog terdiri dari beberapa komponen utama, yang ditunjukkan pada Tabel IV.1.

Tabel IV.1 Komponen CCTV analog

Nama Komponen	Fungsi
Kamera CCTV	Merekam segala aktivitas
DVR (Digital Video Recorder)	Pengolah informasi yang dikirim oleh kamera CCTV
Storage(HDD,SDD,NAS)	Media penyimpanan yang diperlukan DVR untuk informasi dari kamera CCTV
Kabel Coaxial	Media untuk supply listrik ke kamera CCTV
Power Supply CCTV	Memberi aliran listrik untuk menghidupkan kamera CCTV

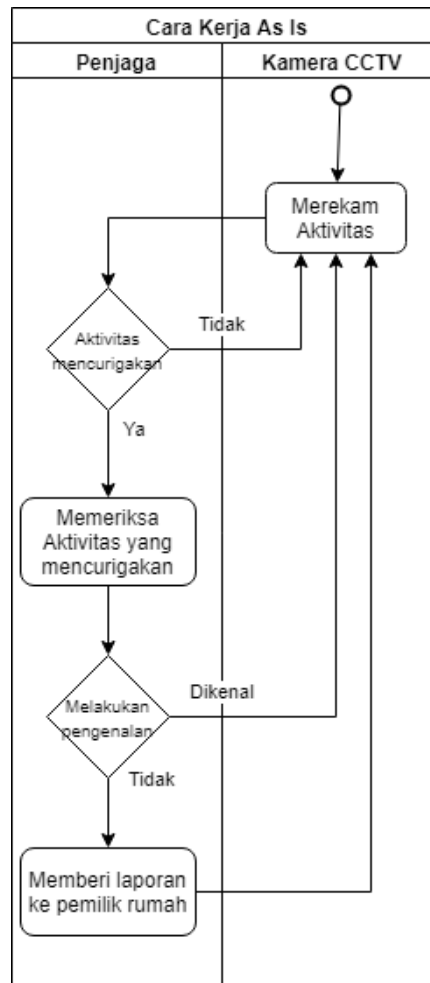
Gambar IV.1 menunjukkan gambaran untuk arsitektur dari kamera CCTV analog.



Gambar IV.1 Komponen CCTV analog

Kamera CCTV analog semuanya akan terhubung dengan menggunakan kabel coaxial yang disambungkan ke monitor yang berfungsi sebagai media untuk melihat rekaman yang dilakukan oleh kamera CCTV analog tersebut. Pada monitor tersebut disediakan sebuah fungsi untuk memilih kamera mana yang akan dilihat dan pengguna dapat mengatur kamera mana saja yang ingin dilihat rekamanannya. Rekaman ini hanya bisa dilihat melalui monitor yang tersedia.

Kamera CCTV analog ini memerlukan satu aktor utama yang bertugas untuk mengawasi semua aktivitas yang terekam. Mengawasi yang dimaksud adalah pengawas haruslah terus-menerus memantau aktivitas yang terlihat melalui layar monitor. Cara kerja pengawasan menggunakan kamera CCTV analog ini ditunjukkan pada Gambar IV.2.



Gambar IV.2 Cara kerja sistem CCTV analog

Pada mulanya kamera akan merekam aktivitas sekitar, pengawasan ini memerlukan penjaga rumah untuk mengawasi melalui monitor dan melakukan pengecekan terhadap orang yang tidak dikenal serta melakukan pengawasan terhadap aktivitas yang mencurigakan dari seseorang. Jika penjaga rumah mengidentifikasi adanya orang tidak dikenal, maka penjaga rumah harus melaporkan atau memberitahu pemilik rumah. Terdapat beberapa kelemahan dari pengawasan menggunakan kamera CCTV analog saat ini, yaitu:

1. Pengawasan yang dilakukan haruslah secara langsung dan terus-menerus melalui monitor pengawas
2. Tidak adanya fitur pengenalan wajah, membuat pengawas haruslah mengenali setiap wajah yang masuk secara manual atau menuju orang itu berada.
3. Belum adanya pemberitahuan kepada pengawas ketika pengawas sedang berada tidak di tempat atau sekitar monitor pengawasan.

4. Adanya aktivitas yang luput dari pengawasan oleh penjaga.

IV.2 Analisis Sistem To Be

Solusi yang ditawarkan pada pengembangan sistem ini ditunjukkan pada Tabel IV.2.

Tabel IV.2 Kekurangan dan solusi yang ditawarkan

No	Kekurangan Sistem Sebelumnya	Solusi	Kebutuhan Sistem
1	Pengawasan yang dilakukan haruslah secara langsung dan terus-menerus melalui monitor pengawas	Membuat sebuah fungsi pengenalan wajah pada kamera pengawas untuk mengetahui wajah yang dikenal atau tidak oleh pemilik rumah yang dapat berfungsi secara otomatis dan terus-menerus ketika kamera CCTV bekerja.	1. Sistem dapat melakukan pendeteksian dan pengenalan secara otomatis dan terus-menerus selama kamera pengawas menyala [REQ-F-U1-002] . 2. Sistem dapat menentukan wajah yang dikenal atau tidak sesuai rekomendasi pemilik rumah.
2	Tidak adanya fitur pengenalan wajah, membuat pengawas haruslah mengenali setiap wajah yang masuk secara manual atau menuju orang itu berada.		
3	Adanya aktivitas yang luput dari pengawasan oleh penjaga.		
4	Belum adanya pemberitahuan kepada pengawas ketika pengawas sedang berada tidak di tempat atau sekitar monitor pengawasan.	Membuat sebuah media untuk memberi sebuah pemberitahuan kepada pemilik rumah ketika mendapatkan wajah yang tidak dikenal	4. Sistem dapat mengirimkan pemberitahuan ketika ada orang yang tidak dikenali melalui chatbot [REQ-F-U3-001] 5. Sistem dapat mengirimkan pemberitahuan dengan konten citra wajah orang yang tidak dikenali dan waktu citra wajah yang tidak dikenali tersebut diakuisisi [REQ-F-U3-003]

Pada tugas akhir ini dikembangkan sistem yang dapat mengatasi masalah dengan solusi yang ditawarkan sesuai Tabel IV.2. Sistem yang dikembangkan adalah sistem yang dapat mendeteksi keberadaan orang yang tidak dikenal dalam sebuah citra yang tertangkap kamera pengawas menggunakan teknologi pengenalan wajah.

Sistem juga dapat mengirimkan pemberitahuan kepada pemilik rumah ketika sistem berhasil mendeteksi keberadaan orang tidak dikenal melalui media chatbot.

IV.2.1 Domain Sistem

Pada pengembangan sistem pendeteksi ini, terdapat tiga domain sistem yang akan dibuat, yaitu:

1. Aplikasi untuk akuisisi citra

Proses akuisisi citra pada sistem ini menggunakan kamera dan melalui proses *preprocessing* sebelum dilakukan proses pengenalan wajah. *Output* dari proses ini adalah sebuah citra yang belum diketahui letak posisi wajahnya.

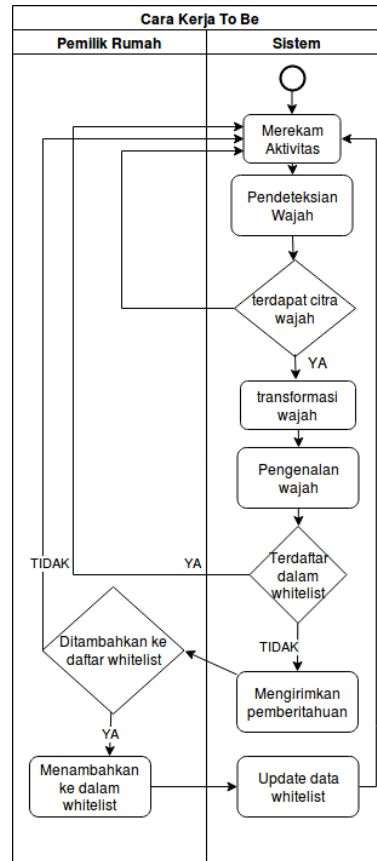
2. Aplikasi proses pengenalan wajah

Proses pengolahan citra yang sudah melalui tahap *preprocessing* akan dilakukan proses pengenalan wajah. Proses pengenalan wajah ini dimulai dengan pendeteksian wajah terlebih dahulu dari citra yang telah diakuisisi, sehingga nanti ditemukan apakah ada citra wajah atau tidak pada sebuah citra serta lokasi citra wajah tersebut berada. Setelah citra wajah ditemukan maka akan dilakukan proses pengenalan wajah yang akan memberikan *output* berupa informasi apakah citra wajah tersebut merupakan wajah dari orang yang dikenal atau tidak, dan nantinya informasi tersebut akan dikirim kepada pemilik rumah menggunakan aplikasi pengiriman.

3. Aplikasi pengolahan data dan informasi

Proses pengolahan ini diantaranya pengiriman dan manajemen informasi. Manajemen informasi yang dimaksud terdiri dari menambahkan data orang dikenal, menghapus, serta mengubahnya.

Cara kerja sistem pendeteksi orang tidak dikenal digambarkan pada Gambar IV.3
Cara kerja sistem *to be*.



Gambar IV.3 Cara kerja sistem *to be*

Tahapan alur kerja sistem pendeteksi orang tidak dikenal ini dijelaskan secara detail sebagai berikut.

1. Merekam aktivitas

Merekam aktivitas dilakukan menggunakan kamera pengawas. Proses perekaman ini dilakukan secara terus-menerus selama sistem dalam keadaan menyala. *Output* dari merekam aktivitas ini adalah tertangkapnya citra oleh kamera pengawas.

2. Pendeteksian dan transformasi citra wajah

Karena citra yang didapatkan belum diketahui apakah terdapat wajah manusia atau tidak, maka perlu dilakukan proses pendeteksian terlebih dahulu. Hanya wajah manusia yang dapat dideteksi dan diolah nantinya. Setelah terdeteksi adanya citra wajah maka dilanjutkan dengan proses transformasi citra untuk memproyeksikan wajah. Tahap *preprocessing* ini dilakukan sebelum masuk ke

proses pengenalan wajah. Metode yang digunakan dalam proses pendeteksian wajah ini dijelaskan pada subbab IV.4.1.

3. Pengenalan Wajah

Pengenalan wajah dapat berjalan jika wajah telah berhasil dideteksi. Proses pengenalan wajah ini dilakukan dengan cara membandingkan nilai dari citra wajah yang terdeteksi dengan citra wajah yang tersimpan pada *database*. Jika nilai hasil perbandingan tersebut sama dengan yang ada pada *database*, maka citra wajah tersebut telah dikenal dan melanjutkan merekam aktivitas. Jika sebaliknya, maka sistem akan memberikan sebuah pemberitahuan kepada pemilik rumah melalui aplikasi pengiriman, hasil pemberitahuan tersebut diperiksa oleh pemilik rumah apakah akan ditambahkan ke dalam *database* atau tetap pada keadaan tidak dikenal. Proses pengenalan wajah akan menghasilkan output berupa status dikenal atau tidak. Penjelasan tentang metode pengenalan wajah akan dijelaskan pada subbab IV.4.3.

4. Pengiriman informasi dan manajemen data

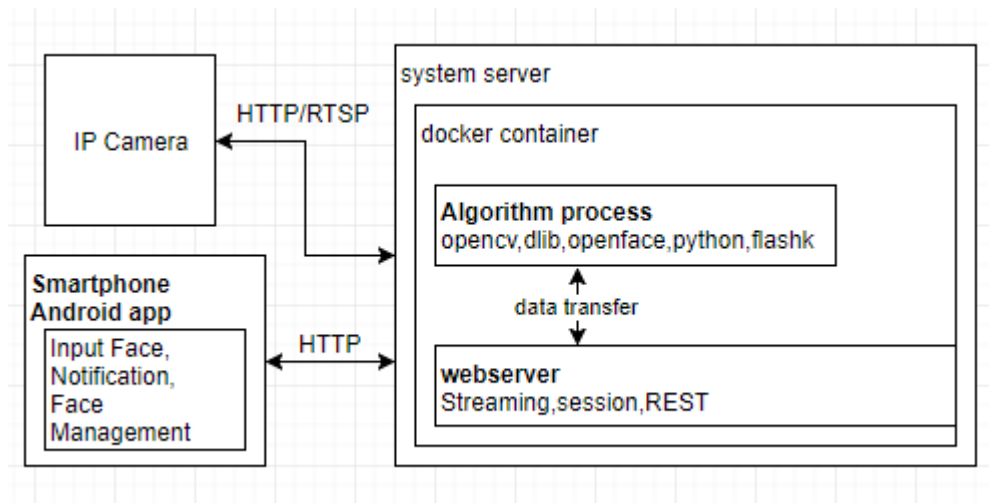
Pemberitahuan akan dikirimkan kepada pemilik rumah melalui media chatbot yang dilakukan ketika adanya wajah yang tidak dikenali terekam oleh kamera pengawas. Informasi yang dikirim ini berupa wajah yang tertangkap oleh kamera pengawas serta waktu citra tersebut didapatkan. Ketika pemberitahuan sampai kepada pemilik rumah, pemilik rumah dapat mengabaikan pemberitahuan atau memprosesnya agar terdaftar pada daftar orang yang dikenal. Aplikasi ini juga untuk melakukan manajemen data seperti menambahkan, menghapus atau mengubah daftar orang dikenal. Adapun hal-hal yang mempengaruhi pengiriman pemberitahuan akan dijelaskan lebih lanjut pada subbab IV.4.

IV.2.2 Analisis Arsitektur Sistem

Terdapat beberapa komponen yang dapat digunakan dalam pengembangan sistem pendeteksi orang tidak dikenal ini. Berikut beberapa penjelasan singkat mengenai aspek positif dan negatif dari komponen sistem yang dapat diterapkan.

1. Menggunakan IP kamera, *cloud server* dan aplikasi android

Model arsitektur yang digambarkan pada Gambar IV.4 membutuhkan data masukan berupa stream video dari perangkat IP kamera. Data video stream diakses oleh server sehingga proses pengenalan wajah dapat dilakukan pada server menggunakan *resource* perangkat keras server dan hasil pemrosesan disampaikan ke dalam aplikasi android pada *smartphone*.



Gambar IV.4 Arsitektur IP kamera

Model seperti ini dapat menghasilkan kualitas sistem yang sangat baik dan dapat diandalkan untuk *multi user*. Model ini lebih berorientasi kepada kualitas produk.

Akan tetapi, model ini juga membutuhkan modal finansial yang cukup besar, proses *development* yang kompleks dan waktu pengerjaan yang relatif lama, sehingga model sistem seperti ini tidak cocok untuk diimplementasikan pada kesempatan tugas akhir kali ini. Kekurangan dan kelebihan dari model ini ditunjukkan pada Tabel IV.3

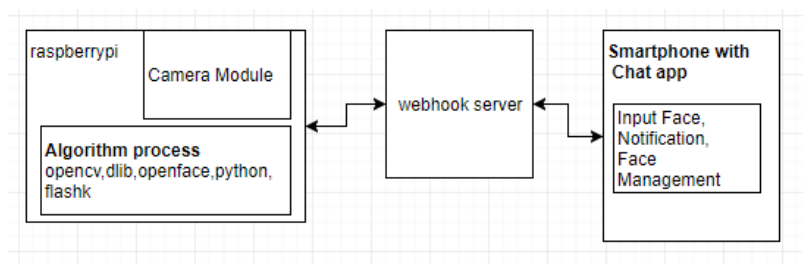
Tabel IV.3 Kekurangan dan kelebihan arsitektur pertama

Kelebihan	Kekurangan
Hasil pemrosesan akan lebih cepat karena komputer <i>server</i> mempunyai spesifikasi tinggi	Membutuhkan dana untuk menyewa <i>cloud server</i>
Notifikasi dapat lebih dikustomisasi karena menggunakan aplikasi android	Pembuatan apps android memakan waktu lama dan tidak bisa <i>cross platform</i>

Dapat melakukan <i>multi process</i> dari <i>multi input source</i> secara bersamaan	Stream video cukup memakan bandwidth (resolusi 640x480 dengan 25 fps membutuhkan bandwidth sebesar 1024 kbps)
	Tidak semua IP camera menyediakan data video stream tanpa otentifikasi / menggunakan aplikasi bawaan IP cam

2. Menggunakan Raspicam, Raspberry Pi dan Chatbot

Model yang ditunjukkan pada Gambar IV.5 menggunakan Raspberry Pi sebagai media pemrosesan data dan menggunakan modul kamera yang dipasang pada Raspberry Pi sebagai sumber data video stream. Hasil pemrosesan berupa informasi diberikan kepada chatbot pada *smartphone* melalui webhook yang menjadi media.



Gambar IV.5 Arsitektur Raspberry Pi dan chatbot

Penggunaan Raspberry Pi sebagai *device portable* yang mampu melakukan proses pengenalan wajah dan menyediakan data video stream dapat menghemat biaya untuk menyediakan kelengkapan perangkat keras. Akan tetapi, karena penggunaan Raspberry Pi yang menjadi *single board computer* ini hanya memiliki spesifikasi perangkat keras yang tidak terlalu besar, maka terdapat kelemahan dalam kecepatan pemrosesan data. Penggunaan Chatbot dapat menjadi alternatif dalam memenuhi kebutuhan sistem dalam interaksi dengan pengguna. Chatbot sudah tersedia dalam banyak platform seperti messenger, line atau telegram, sehingga pengguna tidak perlu menginstall aplikasi android tambahan.

Model seperti ini dinilai cocok untuk diterapkan dalam mengembangkan sistem pendeteksi orang tidak dikenal. Kelebihan dan kekurangan arsitektur ini dijelaskan pada Tabel IV.4.

Tabel IV.4 Kelebihan dan kekurangan arsitektur Raspberry Pi dan Chatbot

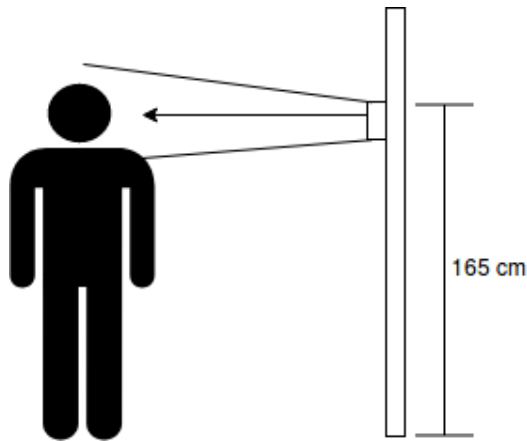
Kelebihan	Kekurangan
Pengembangan lebih sederhana karena <i>environment</i> yang digunakan untuk <i>development</i> sama, yaitu menggunakan linux	Proses pengenalan wajah akan jadi lambat.
<i>development</i> lebih murah	<i>Frame resolution</i> yang diproses akan lebih kecil karena <i>resource</i> perangkat keras terbatas
Chatbot <i>compatible</i> didalam semua operating system	
<i>Bandwith</i> yang digunakan lebih sedikit karena tidak menggunakan <i>streaming</i>	

IV.3 Analisis Akuisisi Citra

Terdapat beberapa faktor yang mempengaruhi untuk mendapatkan hasil optimal saat melakukan akuisisi citra, faktor tersebut dijelaskan secara rinci sebagai berikut.

IV.3.1 Posisi Kamera

Posisi penempatan kamera sangat menentukan hasil proses. Untuk mendapatkan proses yang optimal, posisi kamera harus sejajar dengan wajah yang akan dideteksi. Posisi penempatan kamera diharapkan dapat mendeteksi semua wajah yang tertangkap kamera. Tinggi penempatan kamera dapat disesuaikan oleh pemilik rumah, tetapi dalam kasus ini ukuran yang ditentukan adalah sesuai dengan tinggi badan rata-rata orang Indonesia yaitu 167 cm. Maka diasumsikan bahwa yang terdeteksi berkisar 167cm. Kamera dapat ditempatkan di samping pintu masuk pada rumah. Kamera dipasang di samping pintu masuk agar setiap orang yang akan masuk dapat dikenali terlebih dahulu. Ilustrasi penempatan kamera dapat dilihat pada Gambar IV.6.



Gambar IV.6 Penempatan posisi kamera

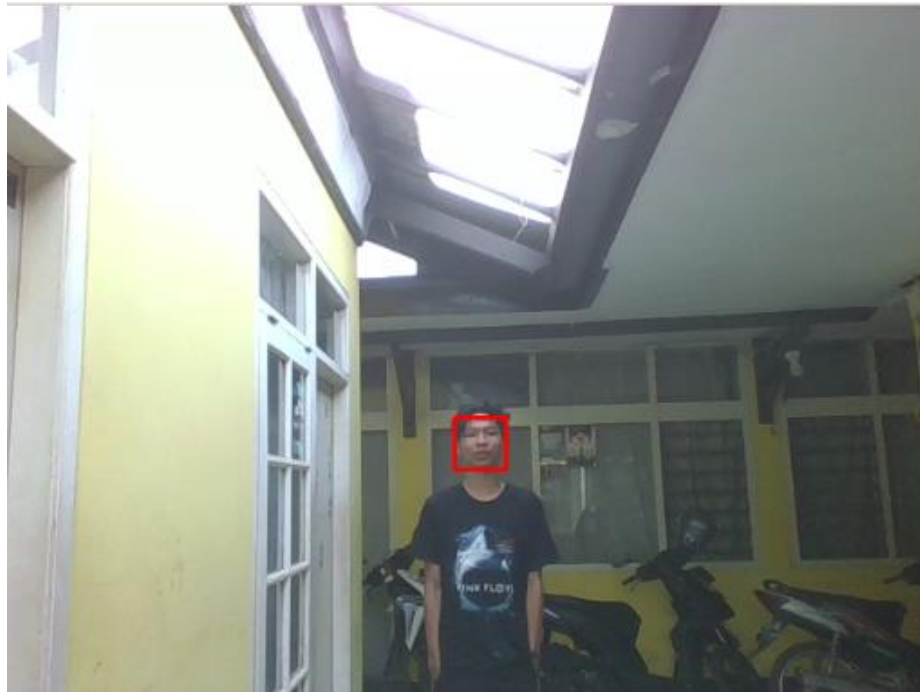
IV.3.2 Resolusi Kamera

Hasil pemrosesan sistem sangat tergantung pada resolusi citra yang ditangkap. Semakin besar resolusi yang dipakai, kamera dapat semakin akurat untuk melakukan proses pendeteksian wajah dalam jarak yang lebih jauh. Hasil percobaan proses pendeteksian wajah dalam resolusi tertentu, waktu pemrosesan, dan hasil pemrosesannya akan ditunjukkan pada Tabel IV.5.

Tabel IV.5 Hasil perbandingan resolusi

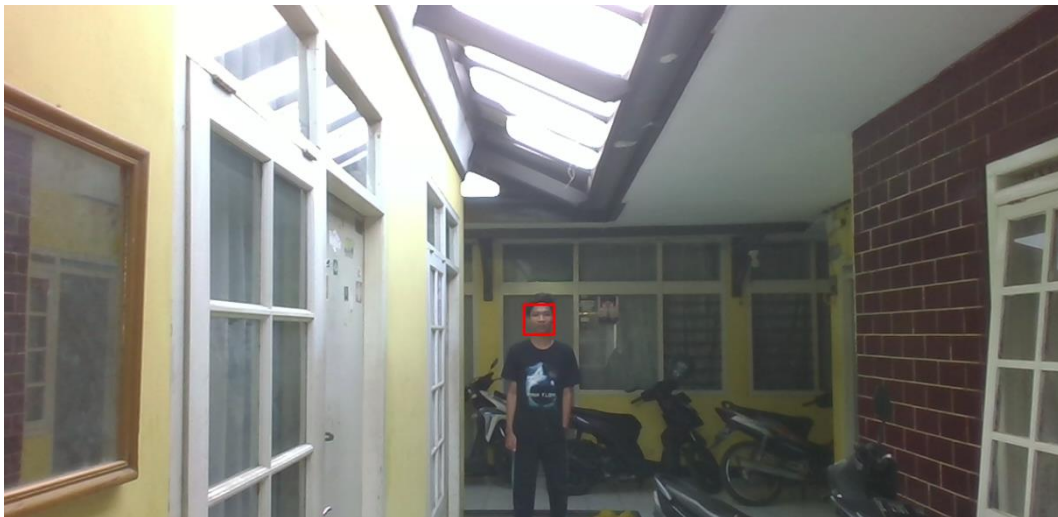
Resolusi	FPS camera	Waktu proses	Jarak maksimal deteksi
640 x 480p	30	0,377 seconds/frame	3 Meter
1080 x 720p	10	0,999 Seconds / frame	5 Meter

Gambar dari perbandingan resolusi yang diuji coba dapat dilihat pada Gambar IV.7 dan Gambar IV.8 .



Gambar IV.7 Pendeteksian wajah dengan resolusi 640 x 480p

Proses pendeteksian wajah dengan resolusi 640x480p menghasilkan jarak deteksi tiga meter dari kamera seperti yang ditunjukkan pada Gambar IV.7.



Gambar IV.8 Pendeteksian wajah dengan resolusi 1280 x 720p

Proses pendeteksian wajah dengan resolusi 1280x720p menghasilkan jarak deteksi 4,9 meter dari kamera seperti yang ditunjukkan pada Gambar IV.8.

Penggunaan resolusi 1280x720p menghasilkan jarak deteksi yang lebih jauh, sehingga sistem dapat bekerja lebih baik untuk melakukan proses pendeteksian

wajah dan pengenalan wajah. Akan tetapi, perlu waktu yang lebih lama ketika melakukan proses ini. Proses pendeteksian wajah saja diperlukan 0,99 detik untuk satu *frame*. Ketika dikenakan proses pengenalan wajah, waktu proses bertambah dua sampai tiga kali lipat karena membutuhkan waktu cukup lama pada tahap *encoding* citra. Oleh sebab itu, penggunaan resolusi 1080 x 720p dinilai kurang cocok diterapkan pada sistem yang akan dikembangkan karena sistem mempunyai batasan dari *resource* perangkat keras yang digunakan yaitu sebatas *single board computer* dengan spesifikasi minimum.

Resolusi yang dipakai dalam sistem yang dikembangkan adalah 640 x 480p. Resolusi 480p dirasa cukup untuk memenuhi kebutuhan dalam proses pendeteksian dan pengenalan wajah. Resolusi 480p hanya menggunakan 0,377 detik untuk proses pendeteksian wajah pada setiap *frame*. Resolusi ini dapat digunakan karena tidak terlalu berat dalam pemrosesan setiap frame dan juga tidak terlalu memberatkan kerja *single board computer*. Maka dari itu, sistem harus mampu mengakuisisi citra melalui kamera pengawas dengan resolusi 480p [REQ-F-U1-001]. Dampak negatif dari penggunaan resolusi 480p adalah sistem hanya bisa mendeteksi dan mengenali wajah jika wajah yang tertangkap kamera berjarak kurang dari atau sama dengan tiga meter dari kamera.

IV.3.3 Pemilihan *Frame*

Pendeteksian dan pengenalan wajah cukup berat dilakukan karena *resource* perangkat keras yang terbatas. Proses pengecekan dan pengenalan tidak mungkin dilakukan pada semua *frame* yang ada secara terus-menerus. Diperlukan *sample frame* untuk dilakukan proses deteksi dan pengenalan wajah. Metode *sample* ini didefinisikan karena beberapa alasan sebagai berikut.

Pada resolusi 640x480p, *frame per second* yang didapat dari webcam adalah 30 fps, sedangkan waktu proses yang digunakan untuk satu *frame* itu berkisar antara 0,6 - 0,8 detik. Jadi, untuk setiap enam *frame* yang ada, akan diambil satu *frame* terakhir untuk dilakukan proses pendeteksian dan pengenalan. Maka dari itu, sistem harus mampu untuk mengambil satu *frame* dari enam *frame* yang ada saat mengakuisisi citra [REQ-F-U1-005]. Pemilihan *frame* dengan hanya memproses

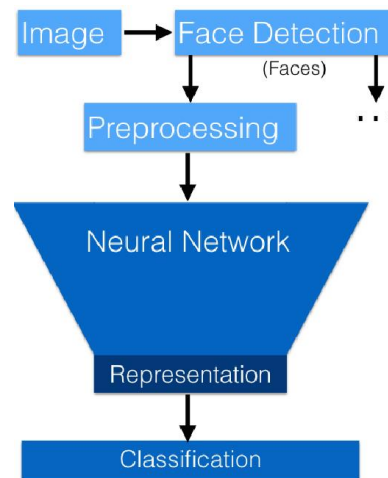
frame tertentu dapat mengurangi jeda proses yang timbul ketika sistem melakukan pendeteksian dan pengenalan wajah kepada semua *frame* yang ada.

Jeda proses yang dimaksud adalah selisih waktu antara citra yang diakuisisi kamera dengan citra yang diproses. Jeda tersebut dapat dirasakan saat program dijalankan dan *rate frame per second* yang didapatkan setelah proses pendeteksian dan pengenalan wajah menjadi drop dan menyebabkan *lagging*.

IV.4 Analisis Metode

Proses pengenalan wajah ini menggunakan metode deep learning yang dikembangkan oleh Davis E King [4]. Metode ini mempunyai tingkat akurasi sebesar 99.38 % dalam LFW datasets.

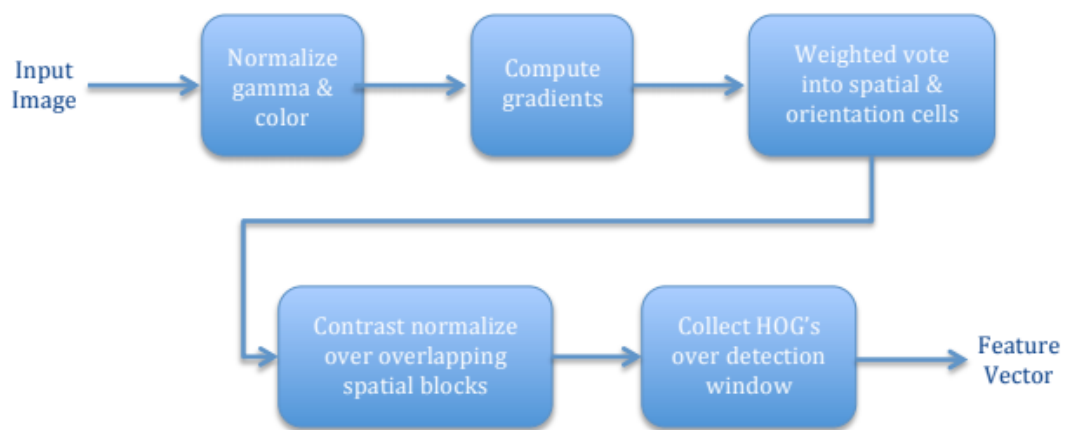
Proses pengenalan wajah dibagi menjadi empat tahap yaitu proses pendeteksian wajah, transformasi wajah, representasi wajah, dan klasifikasi wajah. Secara umum, diagram proses pengenalan wajah ditunjukkan pada Gambar IV.9.



Gambar IV.9 Proses pengenalan wajah

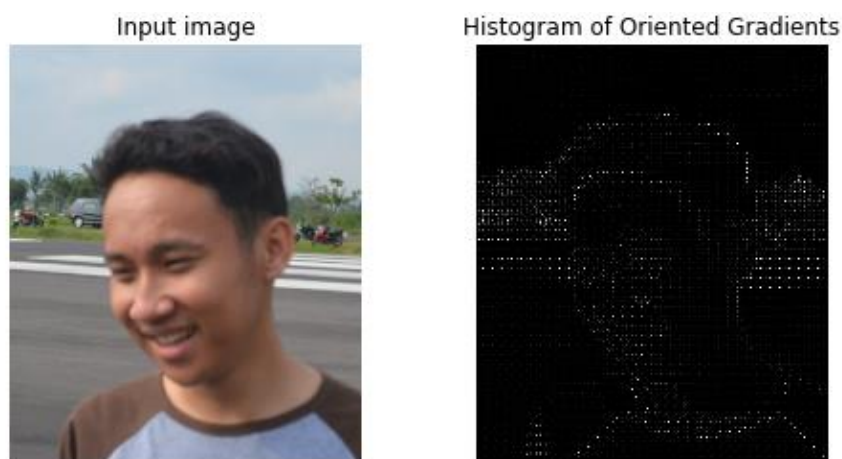
IV.4.1 Metode Pendeteksian Wajah

Proses pendeteksian wajah pada pengembangan sistem ini menggunakan metode HOG (Histogram of Oriented Gradients) ditambah dengan SVM Classifier. Secara umum, metode HOG ditunjukkan pada Gambar IV.10.



Gambar IV.10 Metode HOG

Contoh hasil ekstraksi terhadap suatu gambar dengan menggunakan fitur HOG ditunjukkan pada Gambar IV.11.



Gambar IV.11 Ekstraksi fitur HOG

Setelah HOG fitur didapatkan, maka diperlukan sebuah *classifier* untuk menentukan apakah terdapat citra wajah dari sebuah citra yang terakusisi oleh kamera. Metode pendeteksian wajah yang digunakan menggunakan Support Vector Machine (SVM) *classifier* yang terlebih dahulu dilatih menggunakan datasets dari ImageNet, AFLW, Pascal VOC, the VGG dataset, WIDER, dan face scrub. Metode HOG ini dapat melakukan pendeteksian wajah ketika wajah menghadap ke depan kamera maupun dalam posisi wajah yang tidak tegak lurus asalkan masih terlihat bentuk wajahnya. Jadi, agar bisa dilakukan pendeteksian wajah oleh metode HOG, citra wajah yang ditangkap harus terlihat jelas pada bagian alis, mata, hidung,

pipi, dan mulut. Citra input pada proses pendeteksian ini adalah citra wajah yang telah diakuisisi oleh kamera.

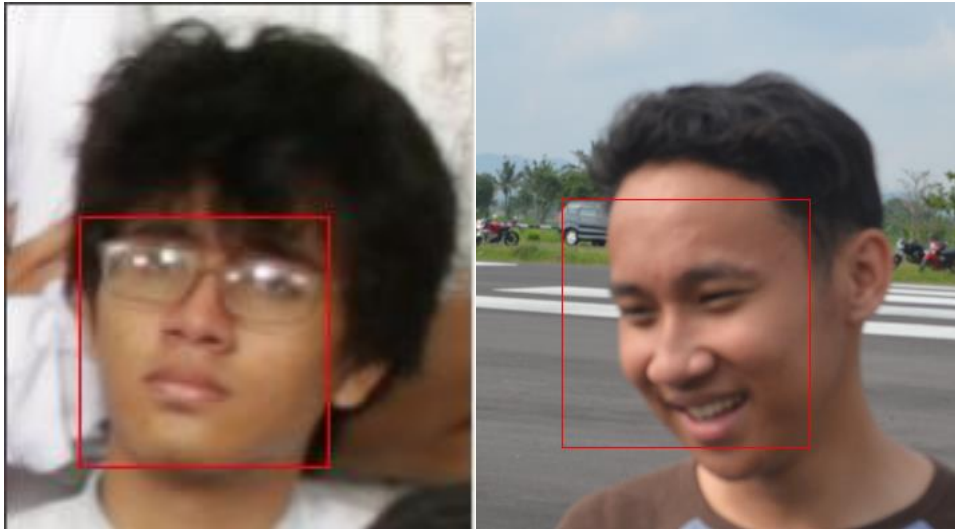
Pendeteksian wajah menggunakan HOG ini digunakan sebuah *pretrained* model “mmod_human_face_detector.dat” yang *di-invoke* ke dalam sebuah *class* dan menjalankan method `get_frontal_face_detector()` dari *class* yang berisi model tersebut. Deskripsi dari method dapat dilihat pada Tabel IV.6.

Tabel IV.6 Method `get_frontal_face_detector()`

Nama Class	dlib
Nama Method	<code>get_frontal_face_detector()</code>
Deskripsi	Method yang digunakan untuk mendeteksi wajah menggunakan HOG detector
Parameter	
Variable	Keterangan
image	array2d of pixel
Return Value	
rectangle location	empat titik margin berupa persegi panjang yang digunakan untuk membuat bounding box

Face detector yang digunakan menggunakan image pyramid, dan skema deteksi sliding window berukuran 80 x 80 pixel, sehingga *face detector* hanya dapat mendeteksi objek berupa wajah manusia dengan citra yang mempunyai ukuran perbandingan lebih dari 80 x 80 pixel.

Gambar IV.12 menunjukkan hasil pendeteksian wajah menggunakan HOG ketika wajah dengan bagian halis, mata, hidung, pipi, dan mulut terlihat jelas sedangkan pada Gambar IV.13 menunjukkan hasil pendeteksian wajah yang tidak terdeteksi oleh metode HOG.



Gambar IV.12 Citra yang terdeteksi HOG

Pada penggunaan HOG ini, terdapat batasan yang perlu diperhatikan, yaitu citra wajah yang ditangkap oleh kamera tidak boleh memakai kaca mata hitam, masker, atau penutup wajah lainnya yang dapat menutupi bagian penting seperti alis, mata, hidung, dan mulut karena citra tersebut nantinya tidak akan terdeteksi sebagai citra wajah manusia.

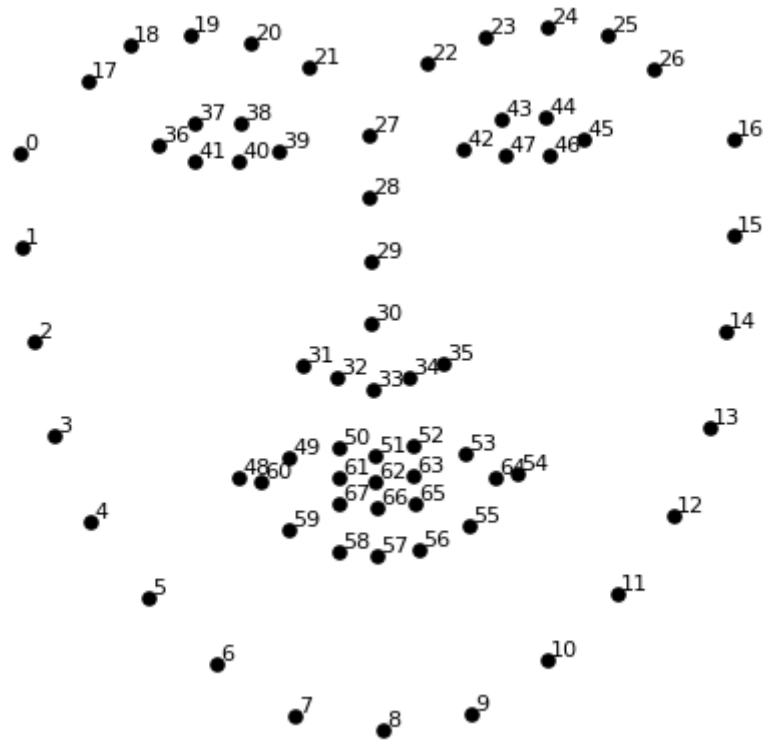


Gambar IV.13 Citra wajah yang tidak terdeteksi oleh HOG

Hasil dari proses pendeteksian wajah ini adalah letak posisi wajah tersebut. Jika citra terdeteksi sebagai wajah manusia maka akan mengembalikan koordinat empat titik yang digunakan untuk membuat *bounding box* yang menandai wajah manusia.

IV.4.2 Metode Memproyeksikan Wajah (Transformasi)

Proses transformasi ini dimulai dengan mendeteksi *landmark* pada citra wajah lalu dilanjutkan dengan transformasi affine. Proses pendeteksian *landmark* yang dimaksud adalah proses penanda pada wajah. *Landmark* pada wajah ini berjumlah 68 yang ditunjukkan pada Gambar IV.14.

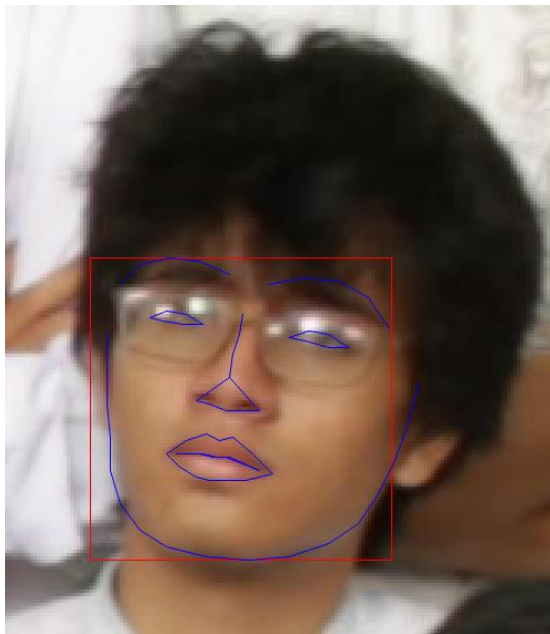


Gambar IV.14 *Face Landmark*

Proses transformasi ini sebenarnya bukanlah proses yang mandatori dalam pengenalan wajah, sesuai yang dikatakan oleh *paper* [7]. Eksperimen yang dilakukan sebelumnya adalah mencari tahu seberapa banyaknya pengaruh proses transformasi ini pada tingkat akurasi yang dihasilkan pada proses pengenalan wajah. Hasilnya adalah akurasi meningkat lebih dari lima persen. Oleh karena itu, proses proyeksi wajah ini layak dilakukan untuk mendapatkan hasil yang lebih optimal pada proses pengenalan wajah.

Pencarian *landmark* pada wajah ini menggunakan fungsi `shape_predictor` yang telah disediakan oleh *library* `dlib` dengan model bernama “`shape_predictor_68_face_landmarks.dat`”. Model ini akan mencari 68 point dari

face landmarks pada setiap wajah yang terlebih dahulu sudah dideteksi pada proses sebelumnya. Model ini akan di-*invoke* ke dalam sebuah *class* yang dibuat. Fungsi ini mengimplementasikan dari paper [8]. Dalam pencarian *landmark* pada wajah manusia, ada beberapa komponen penting dalam pencarian landmark mulai dari sudut ujung mulut dan mata, ujung hidung, dan batas wajah bawah. *Face landmark* yang digunakan telah dilatih sebelumnya pada iBUG 300-W *face landmark dataset*, sehingga tidak perlu membuat *training* untuk *landmark* dari awal. Contoh penggambaran *landmark* yang telah ditemukan ditunjukkan pada Gambar IV.15.



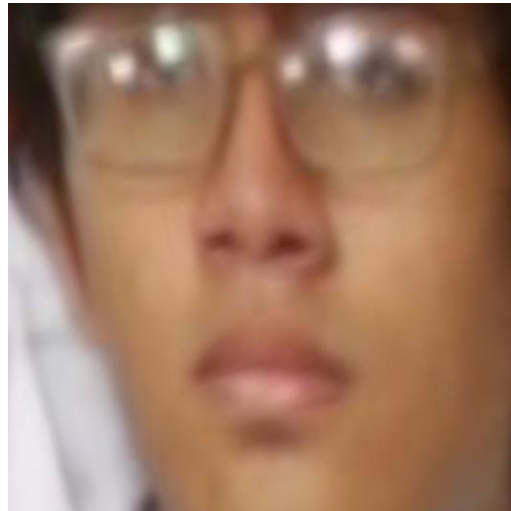
Gambar IV.15 Penempatan *face landmark*

Dari gambar Gambar IV.15, dapat disimpulkan bahwa wajah yang telah terdeteksi akan diberikan operasi *face landmark*. Maka dari itu, muncul kebutuhan fungsional yaitu sistem dapat mencari 68 *face landmark* pada wajah yang terdeteksi [REQ-F-U1-003].

Setelah 68 *face landmark* ditemukan, tahap selanjutnya adalah melakukan tranformasi wajah dengan menggunakan affine transformasi untuk membuat wajah menjadi tegak lurus. Tahap ini perlu diproyeksikan untuk membuat hidung sejajar dengan sumbu vertical lalu mata dan mulut sejajar dengan sumbu horizontal.

Wajah yang telah dicari *landmark*nya akan ditransformasi dan diposisikan hingga mendapatkan hanya bagian *landmark* nya saja, yang berarti ketika wajah tersebut

mempunyai rambut dan dahi yang terlihat ketika dideteksi, maka ketika ditransformasikan, selain yang 68 *landmark* akan hilang. Contoh citra wajah ketika telah diproyeksikan lalu diambil hanya sekitar bagian *landmark*nya saja ditunjukkan pada Gambar IV.16.



Gambar IV.16 Citra wajah setelah transformasi affine

Proses memproyeksikan wajah ini berguna untuk meningkatkan akurasi untuk proses *encoding* pada wajah nantinya, karena citra yang akan dilakukan proses *encoding* hanya wajah dan tidak perlu memberi nilai terhadap bagian citra lain. Pada proses proyeksi wajah ini, hasil yang dihasilkan ada citra wajah dengan ukuran 150x150 pixels. Maka dari itu, sistem harus mampu untuk mentransformasikan posisi wajah menjadi tegak lurus menghadap kamera [REQ-F-U1-004].

IV.4.3 Metode *Encoding* pada Wajah

Proses *encoding* adalah proses merepresentasikan sebuah citra wajah menjadi 128 dimensional vektor. Setiap citra wajah akan menghasilkan nilai yang berbeda, untuk wajah-wajah yang berasal dari orang yang sama akan menghasilkan nilai vektor yang mendekati. Proses pengukuran jarak antara nilai dua buah vektor menggunakan *euclidean distance*. Masukan pada proses *encoding* ini adalah sebuah citra yang telah berhasil dilakukan proses pendeteksian dan pencarian *face landmark*nya.

Library Dlib mempunyai *pretrained* model yang dapat digunakan publik untuk melakukan proses *encodings* / representasi ini. *Pretrained* model ini bernama

“dlib_face_recognition_resnet_model_v1.dat.bz2” yang mempunyai akurasi sebesar 99.13 % dalam standar LFW (Labeled Faces in the Wild) benchmark.

Model ini berasal dari Resnet Network dengan 29 layer konvolusi hasil modifikasi dari Resnet 34 *network* yang ditulis di dalam paper Deep Residual Learning for Image Recognition by He, Zhang, Ren, and Sunyang.

Model dilatih menggunakan tiga juta data berupa gambar wajah manusia yang berasal dari 7485 individu yang berbeda. Setelah menentukan model yang akan dipakai untuk memetakan vektor 128D, maka selanjutnya menghitung euclidean distance menggunakan method `compute_face_descriptor` pada dlib yang dijelaskan pada Tabel IV.7.

Tabel IV.7 Method `compute_face_descriptor`

Nama Method	<code>compute_face_descriptor</code>
Deskripsi	Mengkomputasi vektor 128D dari wajah yang telah diidentifikasi.
Parameter	
Variable	Keterangan
<code>img</code>	citra wajah yang ditangkap kamera
<code>face</code>	digunakan untuk menentukan titik yang menentukan posisi dari object.
Return	
128D Vector: <code>numpy.ndarray</code>	Array 128 dimensi berisi representasi dari citra wajah

Output dari proses *encoding* ini adalah mengkonversi citra menjadi nilai array dengan 128 dimensi dari hasil per satu wajah hasil proses *encoding* yang disimpan dalam array. Hasil representasi wajah ke dalam bentuk vektor 128D ditunjukkan pada Gambar IV.17.

```

128
[-0.09143443  0.13086104  0.01314377 -0.05788449  0.0162896  0.0004133
-0.08469845 -0.09900515  0.17989591 -0.10539673  0.24560221  0.08059315
-0.21611468 -0.13486724  0.04742456  0.12056779 -0.16367504 -0.07826024
-0.11224693 -0.10610127  0.03652949  0.00634991  0.1053369  0.04300563
-0.12117682 -0.33629149 -0.06974638 -0.18218069 -0.00158561 -0.11208323
-0.09656747 -0.02059189 -0.18194009 -0.10914117  0.02073216 -0.02022123
 0.00240959 -0.00374011  0.20474009  0.02820575 -0.11632425  0.09632836
 0.01547989  0.21318351  0.28629941  0.07692295 -0.01180617 -0.09913057
 0.10386179 -0.21633521  0.0727405  0.14290071  0.08237928  0.04238795
 0.09769623 -0.18852286  0.00360183  0.08834422 -0.14143486  0.00837212
 0.00788715 -0.08102694 -0.04035502  0.03879581  0.20594744  0.09965958
-0.12292912 -0.05094444  0.13211262 -0.02900145  0.02445139  0.02434405
-0.1843133  -0.20063356 -0.22774032  0.09293826  0.37345201  0.19359812
-0.20881185  0.01955769 -0.19599997  0.02415313  0.0610562  0.00819605
-0.07174445 -0.13538508 -0.04118636  0.05282187  0.08226584  0.0320852
-0.04098896  0.21506967 -0.03382798  0.06236767  0.01853625  0.05682234
-0.15838754 -0.03170493 -0.16015242 -0.06845069  0.01404161 -0.04203653
 0.03085325  0.14781637 -0.23243298  0.05921925  0.00418686 -0.04666767
 0.02229128  0.07022517 -0.02721729 -0.03373834  0.05814214 -0.23816797
 0.24889059  0.23403467  0.02495457  0.17327933  0.07225875  0.03394284
-0.01637946 -0.02267806 -0.18229848 -0.06459417  0.06046801  0.07552323
 0.08523151  0.00671962]
0.3977155119968996

```

Gambar IV.17 Hasil representasi wajah ke dalam bentuk vektor 128D

Proses *encoding* dapat berjalan setelah proses pendeteksian dan pencarian *face landmark* dan harus mempunyai model yang dapat digunakan untuk memetakan vektor 128D. Maka dari itu, sistem harus mampu melakukan *encoding* pada citra wajah yang telah dideteksi [REQ-F-U2-001].

IV.4.4 Metode Perbandingan Wajah

Perbandingan wajah akan dilakukan satu-persatu pada setiap data vektor wajah di *database* dengan data vektor wajah yang dihasilkan dari hasil citra yang tertangkap kamera dengan menggunakan *euclidean distance*.

Rumus yang digunakan untuk menghitung *euclidean distance* antara n dimensi vektor ditunjukkan pada rumus (IV.1).

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + \dots + (p_n - q_n)^2} \quad (\text{IV.1})$$

Input untuk proses perbandingan wajah menggunakan *euclidean distance* ini adalah nilai vektor 128D yang didapatkan dari proses *encoding* sebelumnya. Nilai itu akan dibandingkan dengan nilai pada daftar wajah dikenal dengan menggunakan rumus *euclidean distance*, jika hasil kurang dari atau sama dengan *threshold*, maka wajah tersebut merupakan wajah dari orang yang sama.

Penghitungan jarak dari dua buah vektor ini membutuhkan sebuah *threshold* yang harus dicapai untuk menentukan apakah sebuah citra yang dihasilkan dan dibandingkan itu berasal dari orang yang sama atau tidak. Nilai dari *threshold* yang ditetapkan adalah 0,625. Jika hasil pengukuran mempunyai hasil akhir melebihi *threshold*, maka dapat disimpulkan bahwa itu adalah wajah yang berbeda. Nilai 0,625 didapatkan dari rata-rata *threshold* yg menghasilkan akurasi terbesar pada proses pengenalan wajah. Maka dari itu, terdapat kebutuhan untuk sistem yaitu sistem dapat membandingkan hasil *encoding* dari wajah yang terdeteksi dengan yang ada pada *database* [REQ-F-U2-002] serta sistem dapat menentukan status dikenal atau tidak dikenal sesuai hasil perbandingan dari nilai *encoding* [REQ-F-U2-003].

IV.5 Analisis Pengiriman Pemberitahuan

Pengiriman pemberitahuan mempunyai beberapa faktor yang mempengaruhi dalam pelaksanaannya yang dijelaskan pada subbab berikut ini.

IV.5.1 Kondisi yang Harus Terpenuhi

Pemberitahuan informasi akan keberadaan orang asing yang tertangkap kamera akan dikirimkan melalui Chatbot pada aplikasi messenger. Pada sistem yang dikembangkan, diasumsikan bahwa pemilik rumah telah terotentikasi pada facebook messenger API, sehingga bisa menggunakan API dari chatbot untuk menerima dan mengirim informasi yang dibutuhkan. Kondisi *single board computer* pun diasumsikan sudah terinstall dan sudah dikonfigurasi sebelumnya.

Pemberitahuan dikirimkan jika sistem berhasil mendeteksi setidaknya satu orang yang tidak dikenal pada citra yang ditangkap kamera. Ketika ada lebih dari satu orang yang terdeteksi pada proses pendeteksian wajah, sistem kemudian melakukan pengecekan apakah ada orang yang tidak dikenal pada kumpulan wajah tersebut. Jika ada, maka sistem akan tetap menganggap wajah tersebut sebagai orang yang tidak dikenal walaupun pada citra yang tertangkap terdapat juga wajah yang terlebih dahulu sudah dikenali oleh sistem. Pada pengembangan sistem ini tidak mengasumsikan pemilik rumah mengenali orang-orang yang dideteksi sebagai orang asing oleh sistem.

Proses pengenalan bisa saja mendapatkan hasil yang tidak akurat karena banyak faktor, sehingga memberikan informasi bahwa sebuah citra yang terdeteksi adalah tidak dikenal, padahal sebelumnya citra orang yang sama sudah terlebih dahulu dikenal dimasukkan ke dalam daftar orang yang dikenal. Oleh karena itu, diperlukan pengecekan ulang terhadap tiga *frame* berikutnya. Pengecekan ulang tersebut dimaksudkan untuk menghasilkan informasi yang benar-benar *valid* tentang hasil pengenalan wajah dan keputusan untuk mengirimkan informasi kepada pemilik atau tidak. Maka dari itu, sistem harus mampu untuk melakukan pengecekan ulang setiap tiga *frame* setelah citra dapat dikenali [REQ-F-U2-004].

IV.5.2 Konten yang Dikirimkan

Chatbot akan melakukan pengiriman pemberitahuan berupa informasi dengan konten tertentu. Konten tersebut berupa gambar wajah dari orang yang tidak dikenal ketika tertangkap kamera pengawas dan waktu ketika citra atau wajah tersebut ditangkap oleh kamera pengawas. Maka dari itu, sistem harus mampu menentukan waktu kapan wajah orang yang tidak dikenali itu tertangkap oleh kamera [REQ-F-U3-002] serta sistem dapat mengirimkan pemberitahuan dengan konten citra wajah orang yang tidak dikenali dan waktu citra wajah yang tidak dikenali tersebut diakuisisi [REQ-F-U3-003].

IV.5.3 Interval Pengiriman

Sebelum mengirimkan pemberitahuan, dibutuhkan interval dalam pengiriman pemberitahuan dikarenakan sistem membutuhkan waktu proses untuk melakukan pendeteksian wajah dan proses pengenalan secara terus-menerus terhadap citra yang ditangkap kamera. Interval pengiriman pemberitahuan ini berfungsi ketika sistem telah berhasil melakukan proses pengenalan terhadap wajah yang terdeteksi agar tidak mengirimkan pemberitahuan terus-menerus. Pengiriman pemberitahuan terus-menerus ini disebabkan karena sistem menangkap citra lalu mengenali dari setiap *frame* yang ditangkap oleh kamera pengawas.

Pemberitahuan ini akan dikirim ke pemilik rumah dalam interval setiap lima *frame* ketika mendeteksi dan mengenali wajah yang berbeda, interval ini dibutuhkan agar pemberitahuan tidak selalu dikirim ketika menangkap citra wajah yang sama dalam kurun waktu yang dekat.

IV.6 Analisis Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini dibahas lebih lanjut pada subbab berikut ini.

IV.6.1 Operating System (for Raspberry)

Raspberry Pi membutuhkan sebuah Arm berbasis Linux Operating System agar dapat dijalankan. Dalam pengembangan sistem ini, digunakan Raspbian Os yang sudah menyediakan antar muka untuk koneksi dengan perangkat keras eksternal seperti modul kamera. Raspbian Os juga menyediakan *precompiled* perangkat lunak aplikasi. Karena Raspbian ini berbasis Debian linux, maka semua aplikasi yang dibuat di atas linux akan *compatible* asalkan mendukung *ARM architecture*. Kebutuhan akan sistem operasi ini juga untuk mempermudah proses *development*. Proses *development* tidak akan berjalan langsung di dalam Raspberry-Pi. Proses *development* dilakukan pada perangkat lain (laptop) dengan menggunakan Linux based OS, dan hasil dari proses *development* akan *dideploy* di dalam Raspberry-Pi.

IV.6.2 Bahasa Pemrograman dan Library

Program aplikasi yang dibutuhkan harus program yang dibuat di atas bahasa pemrograman dan *library* tertentu. Bahasa pemrograman yang dipakai adalah Python menggunakan *library* utama yaitu OpenCV, Dlib, Flask dan Requests. *Library* digunakan dikarenakan mempertimbangkan hasil analisa sebagai berikut.

IV.6.2.1 Akses kedalam perangkat keras

Kamera yang digunakan untuk mengakuisisi citra memerlukan sebuah antar muka agar bisa diakses oleh program. Penggunaan *library* OpenCV pada python sudah dapat mengakses kamera secara langsung, sehingga dapat menentukan spesifikasi yang harus diterapkan pada kamera. contohnya FPS, Resolusi, effect, dll.

IV.6.2.2 Processing

Proses *deep learning* pengolahan citra menggunakan bahasa pemrograman python. Python mempunyai *support* yang baik untuk digunakan pada *single board computer*. Dalam website resmi Raspberry Pi diberikan *package* untuk mengakses perangkat seperti kamera dalam bahasa python. Secara sintaks, python lebih sederhana dibandingkan dengan bahasa pemrograman lainnya, oleh karena itu

proses *debugging* dapat lebih mudah dilakukan. Python menyediakan berbagai *package* yang mendukung untuk melakukan pengolahan citra. Salah satu *package* yang ada di dalam python adalah OpenCV yang sebenarnya ditulis dengan bahasa C dan C++, tetapi tersedia juga dalam bahasa python. Dalam library OpenCV terdapat fungsi-fungsi pengolahan citra yang dapat digunakan tanpa harus membuat fungsi-fungsi tersebut dari awal.

Library machine learning yang digunakan adalah *library Dlib*. Dlib ditulis dalam bahasa C++, tetapi Dlib menyediakan *binding* ke dalam bahasa python. Dlib menyediakan banyak fungsi yang digunakan dalam *machine learning* seperti klasifikasi, *object trainer* dan banyak lagi. Penggunaan library Dlib akan sangat membantu proses pengenalan wajah yang akan dilakukan.

IV.6.2.3 Komunikasi dengan Chatbot

Proses pengiriman pemberitahuan akan dilakukan oleh *single board computer* ke chatbot yang ada di Messenger platform melalui protokol http dan https. Komunikasi ini dilakukan dua arah menggunakan method http.POST dan http.GET. Komunikasi http pada python dapat dilakukan dengan banyak cara yaitu menggunakan curl, menggunakan library urllib, atau menggunakan *library requests*.

Dalam sistem yang dikembangkan, komunikasi http akan dilakukan menggunakan *library Requests*. *Library* tersebut dipilih karena sintaks proses komunikasi http lebih mudah dibaca dan struktur sintaksnya lebih berorientasi objek dibandingkan curl. Perbandingan sintaks tersebut ditunjukkan pada Tabel IV.8.

Tabel IV.8 Perbandingan sintaks Curl dan Python

Curl	Python Requests
curl 'http://en.wikipedia.org/' -H 'Accept-Encoding: gzip, deflate, sdch' -H 'Accept-Language: en-US,en;q=0.8' -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H 'Referer: http://www.wikipedia.org/' -H 'Connection:	import requests headers = { 'Accept-Encoding': 'gzip, deflate, sdch', 'Accept-Language': 'en-US,en;q=0.8', 'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95

keep-alive' --compressed	Safari/537.36', 'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8', 'Referer': 'http://www.wikipedia.org/', 'Connection': 'keep-alive', } response = requests.get('http://en.wikipedia.org/', headers=headers)
--------------------------	--

Python Request menyediakan banyak fitur-fitur lain selain hanya request get. Salah satunya adalah *Multipart File Uploads* yang digunakan untuk melakukan proses mengunggah gambar yang akan dilampirkan dalam proses pengiriman informasi pemberitahuan.

IV.7 Analisis Perangkat Keras

Perangkat keras yang digunakan adalah *single board computer* Raspberry Pi 3 model B dengan spesifikasi yang ditunjukkan pada Tabel IV.9.

Tabel IV.9 Spesifikasi Raspberry Pi Model B

Parameter	Spesifikasi
CPU	Quad Core Arm Cortex-A53 1.2 GHz
GPU	Broadcom VideoCore IV
SOC	Broadcom BCM2837
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4GHz 802.11n wireless
Storage	MicroSD
GPIO	40-pin header
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Perangkat keras kamera yang digunakan adalah Raspberry Pi NoIR Camera module v2. Kamera tersebut dipilih karena hasil analisis akan kebutuhan-kebutuhan berikut ini.

IV.7.1 Menjalankan Proses *Deep Learning* dan *Image Processing*

Sistem menjalankan proses pengolahan citra untuk melakukan proses pengenalan wajah. Karena proses pengolahan citra ini membutuhkan tenaga komputasi yang relatif cukup besar dan memerlukan operasi matematis yang dapat diimplementasikan dalam sebuah sistem operasi.

Proses *deep learning* adalah proses yang relatif kompleks saat pengeksesusiannya. Dibutuhkan *resource* perangkat keras dengan spesifikasi yang tinggi untuk melakukan *training* pada *neural network*. Proses *deep learning* pada Raspberry Pi akan terhambat oleh dua masalah utama, yaitu :

1. Memory yang terbatas (Raspberry Pi hanya mempunyai memory sebanyak 1 GB)
2. Kecepatan CPU yang terbatas

Dibandingkan untuk melakukan proses *training* pada Raspberry Pi, proses *deployment* dari hasil *training* neural network didalam Raspberry Pi lebih menguntungkan jika implementasikan. Proses *deployment* neural network ini juga tidak akan secepat jika dibandingkan proses *deployment* pada perangkat yang mempunyai spesifikasi perangkat keras lebih tinggi. Contohnya pada komputer yang memiliki GPU dengan spesifikasi tinggi.

IV.7.2 Perangkat Akuisisi Data

Proses akuisisi data akan ditangani oleh perangkat kamera yang terhubung pada *single board computer*. Setiap *frame* yang diambil akan diproses langsung untuk proses *preprocessing* pada *single board computer*. Kondisi cahaya pada saat proses akuisisi memberikan dampak besar terhadap citra yang dihasilkan, oleh karena itu diperlukan sebuah kamera yang dapat meminimalisir dampak yang terjadi akibat kondisi cahaya tersebut.

Kamera dengan filter NOIR (No Infrared) dapat digunakan untuk memaksimalkan hasil foto pada kondisi kurang cahaya karena tidak menggunakan filter infra merah.

Proses pengambilan citra menggunakan camera dengan sensor tipe CMOS sehingga kamera dapat lebih responsif dan dengan daya listrik yang lebih hemat. Kamera yang akan dipakai untuk proses akuisisi citra adalah Raspberry Pi NoIR Camera module v2 yang mempunyai spesifikasi perangkat keras yang ditunjukkan pada Tabel IV.10.

Tabel IV.10 Spesifikasi Raspberry Pi NoIR Camera module v.2

Parameter	Spesifikasi
Pixel Resolution	8 Megapiksel
Video mode	1080p30, 720p 60 and 640 × 480p 60/90
Linux Integration	V4L2 Driver
C Programming API	OpenMax IL and others
Sensor	Sony IMX219
Sensor Resolution	3280 x 2464 Pixels
Sensor Image Area	3.68 x 2.76 mm (4.6 mm diagonal)
Pixel size	1.12 μm x 1.12 μm
Optical size	¼ “
Focal Length	3.04 mm
Horizontal field of view	62.2 Degrees
Vertical field of view	48.8 Degrees

Kamera module v2 ini sudah diberikan *built in software* untuk keperluan pengolahan citra seperti *filter*, *effect*, *exposure* dan *white balance*. Secara *default* format keluaran citra yang dihasilkan untuk gambar adalah JPEG (accelerated),

JPEG + RAW, GIF, BMP, PNG, YUV420, RGB888 dan untuk video adalah raw, h.264.

Module kamera raspberry pi v2 ini menggunakan CSI2 (Camera serial interface) untuk koneksi pada *single board computer*. Kamera dapat diakses MMAL (Multimedia Abstraction Layer) menggunakan library Picamera. Kamera menggunakan *Low Voltage Differential Signalling* (SubLVD) 1.2 V.

IV.7.3 Data Transfer

Dibutuhkan sebuah perangkat keras yang dapat menghubungkan single board dengan jaringan luar untuk melakukan transmit dan *receive* data secara *online* dengan Messenger API menggunakan protokol HTTP/HTTPS. Karena pertukaran data dilakukan secara nirkabel, maka perangkat keras yang dibutuhkan adalah wireless radio.

Perangkat keras Raspberry Pi 3 model B sudah terdapat modul wireless lan secara *built in*. Modul ini menggunakan Broadcom BCM43438 chip yang mempunyai frekuensi 2.4 GHz dan bisa terkoneksi dengan wifi rumahan.

IV.7.4 Data Storage (MicroSD)

Perangkat penyimpanan digunakan untuk menyimpan sistem operasi, aplikasi, dan data yang akan diolah dan dikirimkan. Data tersebut akan disimpan dan diolah oleh *single board computer*. Pada Raspberry Pi data disimpan dalam sebuah media penyimpanan berjenis MicroSD yang mempunyai ukuran jauh lebih kecil dibandingkan dengan hardisk atau SSD. Oleh karena itu, dibutuhkan sebuah Micro SD dengan ukuran minimal 32 GB untuk menyimpan semua data yang diperlukan.

IV.8 Analisis Komunikasi data

Pada bagian ini akan dijelaskan bagaimana data dapat dikirim dan diterima dari *service A* ke *service B*. *Service* didefinisikan sebagai sekumpulan perangkat lunak yang dirancang untuk melakukan suatu kegiatan tertentu. Terdapat tiga *service* yang akan dibuat yaitu *service* untuk akuisisi data, *service* untuk pengolahan data, dan *service* untuk penampilan hasil keluaran data. Agar masing-masing *service* dapat melaksanakan tugasnya dengan baik maka perlu didefinisikan bagaimana *service*

yang satu dengan yang lainnya dapat berkomunikasi dengan pembahasan sebagai berikut.

IV.8.1 Interaksi *Service* Penampil Data dengan Messenger Platform

Analisis mengenai interaksi *service* penampilan data dengan messenger platform yang dijelaskan pada subbab berikut ini.

IV.8.1.1 Webhook

Interaksi yang dilakukan pengguna pada Facebook page akan diolah oleh Messenger platform dan dikirimkan hasilnya ke sistem melalui webhook. Untuk menerapkan webhook yang dapat menerima *event* webhook, perlu disiapkan perangkat lunak untuk mengolah data yang didapat yang dihosting pada server publik HTTP dengan syarat sebagai berikut.

1. Memiliki sertifikat SSL yang valid
2. Menunjang protokol komunikasi HTTPS
3. Menyediakan *port* yang menerima permintaan GET dan POST

Sistem yang dikembangkan hanya akan memanfaatkan satu dari sekian banyak *event*, yaitu *messages* yang merupakan *event webhook* yang terjadi ketika ada pesan yang diterima dari pengguna. Format *event* yang diterima adalah json dengan struktur dasar seperti yang ditunjukkan pada Gambar IV.18.

```
{
  "object": "page",
  "entry": [
    {
      "id": "<PAGE_ID>",
      "time": 1458692752478,
      "messaging": [
        {
          "sender": {
            "id": "<PSID>"
          },
          "recipient": {
            "id": "<PAGE_ID>"
          },
          ...
        }
      ]
    }
  ]
}
```

Gambar IV.18 Struktur *event messages*

Properti objek :

Tabel IV.11 Deskripsi objek event webhook

Nama objek	Tipe	Deskripsi
object	String	Isinya adalah “page” tanpa kutip
entry	Array Entry	Array berisi data <i>event</i> yang diterima

Entry

Tabel IV.12 Deskripsi objek entry

Nama objek	Tipe	Deskripsi
id	String	Kode Facebook page
time	Number	Waktu <i>Timestamp</i> (dalam satuan milidetik)
messaging	Array messaging	Array berisi data messaging. Walaupun objek ini adalah array tapi hanya akan memiliki satu elemen array saja.

Entry.messaging

Tabel IV.13 Deskripsi objek event entry messaging

Nama objek	Tipe	Deskripsi
sender.id	String	Kode akun pengirim
receipient.id	String	Kode akun penerima

Elemen objek array *messaging* akan berisi objek yang berbeda tergantung dengan *event* yang diterima.

Webhook harus merespon dengan cepat *event* webhook yang telah diterima dengan kode status 200 OK dengan waktu kurang dari sama dengan 20 detik. Jika webhook gagal memenuhi salah satu persyaratan di atas selama lebih dari 15 menit, maka peringatan 'Webhooks Failing' akan muncul di halaman 'Alerts' pada pengaturan aplikasi. Jika webhook terus-menerus gagal selama 8 jam, maka akan menerima

peringatan 'Webhooks Disabled', dan aplikasi Facebook akan berhenti berlangganan untuk menerima *event* untuk Facebook *page*.

Webhook ini berjalan di *web service* yang memiliki dua *endpoint* yaitu *endpoint* untuk menerima token verifikasi dari Messenger platform dan *endpoint* untuk menerima *event webhook*. Token verifikasi ini berfungsi sebagai keamanan.

Web *service* yang dibuat menggunakan bahasa pemrograman python dan library Flask. Flask adalah micro web *framework* yang ditulis dalam bahasa pemrograman Python dan berdasarkan Werkzeug *toolkit* dan *template engine* Jinja. *Micro* tidak berarti bahwa seluruh aplikasi web harus masuk ke dalam file Python tunggal (meskipun tentu saja bisa), juga tidak berarti bahwa Flask memiliki kekurangan dalam fungsionalitasnya. *Micro* dalam *microframework* berarti Flask hanya menyediakan fungsi utama untuk membangun web service.

IV.8.1.2 Tunnel Protocol

Protokol tunneling adalah protokol komunikasi yang memungkinkan pergerakan data yang aman dari satu jaringan ke jaringan lainnya. Maka, dapat digunakan komunikasi jaringan pribadi untuk dikirim melalui jaringan publik melalui proses yang disebut enkapsulasi.

Syarat yang harus dipenuhi Messenger Platform agar bisa menggunakan webhook yang sudah dibuat yaitu webhook dapat menunjang endpoint HTTPS. Pada sistem yang dikembangkan, webhook server disimpan di *single board computer*. Akan tetapi karena sumber daya yang terbatas, *single board computer* tidak dapat menunjang endpoint HTTPS. Oleh karena itu, sistem ini menggunakan aplikasi ketiga bernama ngrok untuk melakukan *tunneling*. Ngrok dapat membuat public HTTPS url dan menjalankan website yang ada di lokal dan gratis.

Ngrok memiliki batas waktu penggunaan per sesinya yaitu delapan jam. Cara menjalankan ngrok adalah dengan mengunduh aplikasinya ke lokal kemudian menjalankannya dengan perintah khusus.

IV.9 Analisa Data

Pengembangan sistem ini, memerlukan sebuah data untuk disimpan dan diolah. Analisa mengenai data yang digunakan dijelaskan sebagai berikut.

IV.9.1 Database

Database yang dibuat berisi nilai rata-rata dari vektor hasil proses *encoding* pada setiap nama orang. Karena data yang disimpan hanyalah sedikit, akan lebih ringan secara performa jika disimpan dalam sebuah file berformat .csv. Proses *query* pada *database* juga tidak terlalu dibutuhkan. File .csv ini akan di-load sebagai *database* pada sistem saat dijalankan.

Penggunaan csv cukup untuk memenuhi kebutuhan sistem karena tidak ada *constraint multi user* yang menjalankan sistem dengan menggunakan satu *database*. Penggunaan csv juga tidak memerlukan memori proses tambahan seperti menjalankan *service DBMS* pada komputer. Struktur yang disimpan akan menyimpan vektor hasil *encoding*, nama, waktu, dan atribut lain yang nanti akan dibutuhkan.

IV.9.2 Data yang Akan Diproses

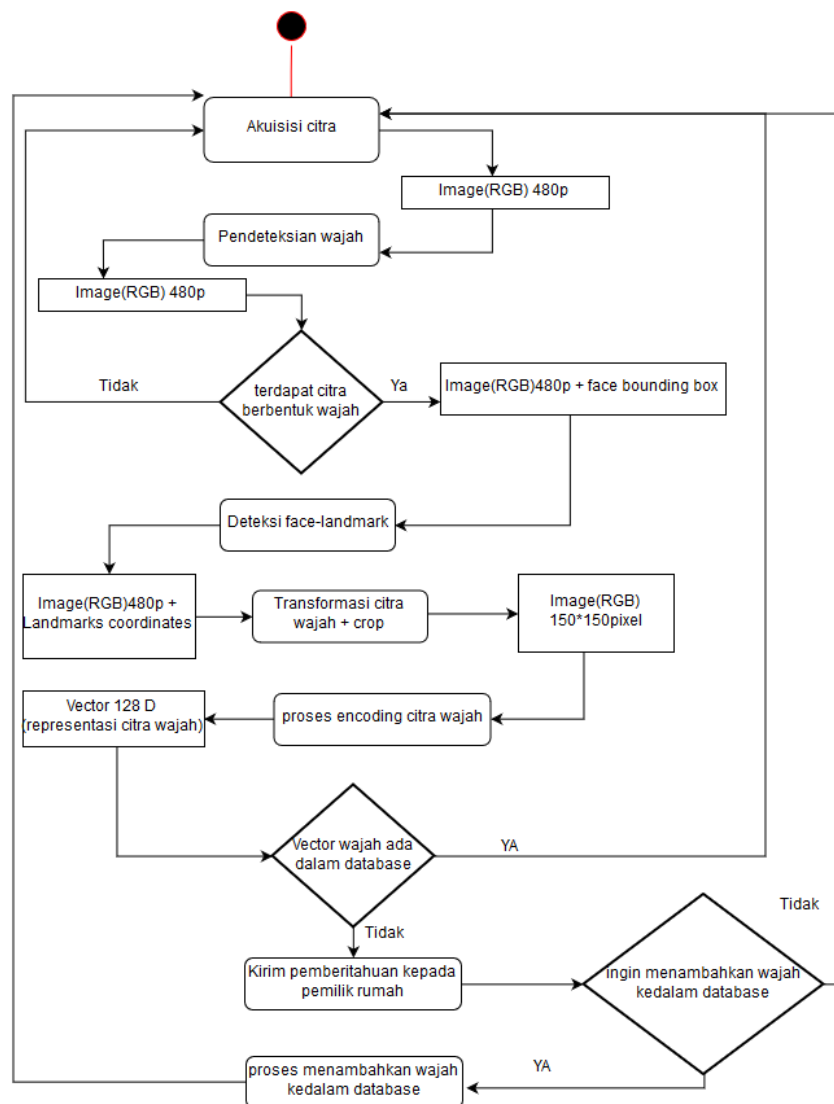
Terdapat sebuah daftar informasi berupa nama dan foto-foto dari wajah yang didefinisikan dikenal oleh pemilik rumah. Daftar orang dikenal ini dapat diolah oleh pemilik rumah berupa menambah, mengubah nama, serta menghapus data. Hanya pemilik rumah yang dapat menambah atau menghapus data orang dikenal yang tersimpan pada *database*. Pada proses penambahan data orang dikenal, terdapat beberapa ketentuan yang harus dipenuhi oleh pemilik rumah, yaitu data orang yang dikenal harus berupa nama dan foto RGB dengan ukuran minimal 150x150. Foto harus berupa wajah dengan posisi tegak lurus menghadap kamera dan tanpa penutup wajah. Jumlah foto untuk masing-masing orang yang dikenal hanya satu.

BAB V

SYSTEM AND SOFTWARE DESIGN

Bab ini membahas mengenai perancangan sistem pendeteksi orang tidak dikenal berdasarkan *requirement* yang telah didefinisikan pada dokumen SRS. Perancangan yang dibuat meliputi hal-hal berikut ini.

V.1 Perancangan Aktivitas pada Sistem



Gambar V.1 Activity diagram sistem

Proses aktivitas yang terjadi pada sistem digambar menggunakan *activity* diagram yang ditunjukkan pada Gambar V.1. Tujuan perancangan ini adalah menggambarkan aktivitas sistem yang terjadi dari awal hingga akhir.

V.2 Perancangan Proses Alur Kerja Sistem

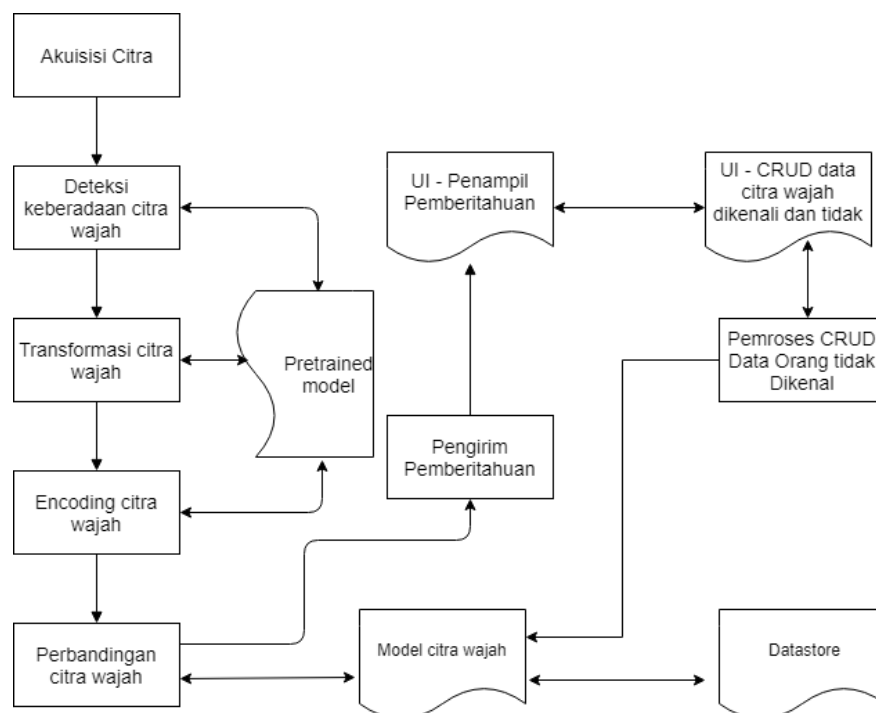
Proses alur kerja sistem digambarkan dengan menggunakan *state machine* diagram yang ditunjukkan pada Gambar V.2. Perbedaan dari perancangan pada subbab ini dengan subbab sebelumnya yaitu dengan memakai *state machine diagram* aktifitas sistem dapat digambarkan lebih jelask dengan adanya *trigger* yang mempengaruhi dari satu *state* ke *state* yang lainnya. Penjelasan mengenai *state* yang dilakukan sistem dijelaskan pada Tabel V.1. *Source target* merupakan *state* awal yang akan menuju *state* selanjutnya yaitu ditandai dengan *target state*. Perpindahan *state* tersebut dapat dipengaruhi oleh sebuah *event*, *guard*, ataupun *procedure*.

Citra tertangkap	Citra wajah tertangkap			mencari posisi wajah manusia
Citra wajah tertangkap	Face landmark ditemukan			mencari face landmark
Face Landmark Ditemukan	Wajah ditransformasi			transformasi affine
Wajah ditransformasi	Nilai wajah hasil <i>encoding</i>	Wajah hasil transformasi		<i>encoding</i>
Nilai wajah hasil <i>encoding</i>	Status didapatkan	nilai wajah hasil <i>encoding</i>		membandingkan nilai
Status didapatkan	Camera stand by		status dikenal	
Status didapatkan	Citra wajah tidak dikenal disimpan		status tidak dikenal	Menyimpan citra wajah
Citra wajah tidak dikenal disimpan	Waktu saat mengambil citra telah disimpan			Mengambil waktu saat dikenali
Waktu saat mengambil citra telah disimpan	Pesan telah tersusun			menyusun pesan
Pesan telah tersusun	Pesan terkirim ke pemilik rumah			push messages
Pesan terkirim ke pemilik rumah	Menampilkan halaman chatting dengan chatbot	membuka aplikasi messenger		
Menampilkan halaman chatting dengan chatbot	Menampilkan daftar whitelist	mengetikkan keyword	keyword = lihat	get daftar whitelist
Menampilkan daftar whitelist	Mengubah nama whitelist yang dipilih	mengetikkan keyword	keyword = update	update whitelist
Mengubah nama whitelist yang dipilih	Menampilkan halaman chatting dengan chatbot	Menampilkan pesan data berhasil diubah		
Menampilkan daftar whitelist	Menghapus data yang dipilih	mengetikkan keyword	keyword = delete	delete whitelist
Menghapus data yang dipilih	Menampilkan halaman chatting dengan chatbot	Menampilkan pesan data berhasil dihapus		
Menampilkan daftar whitelist	Camera stand by	menutup aplikasi	no keyword	

Menampilkan halaman chatting dengan chatbot	Camera stand by	menutup aplikasi	no keyword	
Menampilkan halaman chatting dengan chatbot	Menampilkan konfigurasi untuk menambahkan data	mengetikkan keyword	keyword = tambah	
Menampilkan konfigurasi untuk menambahkan data	memberi nilai pada wajah	memilih foto dan memasukkan nama		
Memberi nilai pada wajah	Menampilkan halaman chatting dengan chatbot	Input selesai, data tersimpan pada daftar whitelist		
Menampilkan halaman chatting dengan chatbot	Camera stand by	menutup halaman chatting		

V.3 Perancangan *Logical* Arsitektur

Logical arsitektur dibuat untuk memberikan gambaran terkait arsitektur yang dibuat pada sistem pendeteksi orang tidak dikenal ini. Gambar V.3 menunjukkan diagram *logical* arsitektur dari sistem ini.



Gambar V.3 *Logical* arsitektur

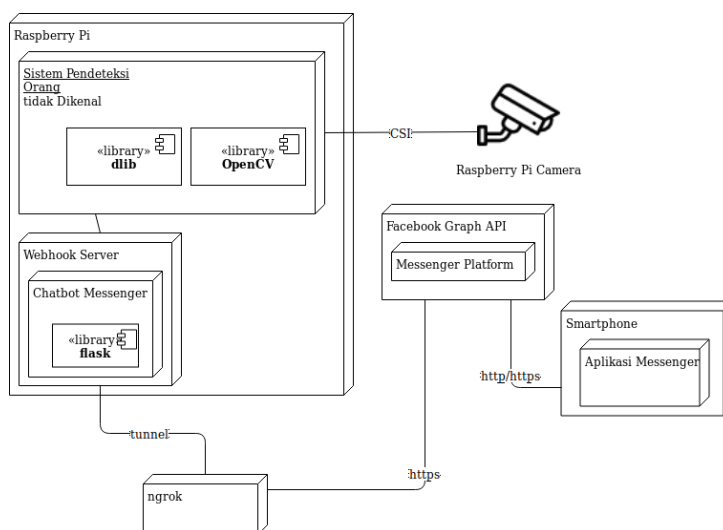
Penjelasan mengenai proses yang ada pada Gambar V.3 dibahas secara rinci sebagai berikut.

1. Akuisisi citra adalah proses ketika kamera merekam aktivitas. Output yang dihasilkan dari proses ini adalah sebuah citra yang tertangkap oleh kamera yang nantinya akan dilanjutkan dengan proses pendeteksian.
2. Deteksi keberadaan citra wajah, Input dari proses ini adalah citra yang telah diakuisisi sebelumnya, citra tersebut nantinya akan dideteksi apakah ada wajah pada citra atau tidak, jika ada wajah maka proses ini akan mengeluarkan output berupa dimana lokasi wajah tersebut berada dan ditandai dengan menggunakan *bounding box*.
3. Transformasi citra wajah adalah proses setelah citra wajah ditemukan nantinya akan dilakukan proses transformasi citra wajah. Jadi, citra wajah yang telah didapatkan akan ditransformasikan hingga nantinya menjadi tegak lurus menghadap kamera. Jadi output dari proses ini adalah, citra wajah yang telah ditransformasi yang menghadap tegak lurus ke depan.
4. *Encoding* citra wajah adalah proses mengubah sebuah citra menjadi nilai vektor 128D. Masukan dari proses ini adalah citra wajah yang telah ditransformasikan tadi akan dilakukan proses *encoding* sehingga menghasilkan nilai vektor 128D.
5. Pretrained model, adalah sebuah model *machine learning* yang telah dilatih yang memungkinkan untuk melakukan representasi data. Model yang digunakan ada tiga macam, yaitu model untuk pendeteksian wajah yang berisi *object detector*, model untuk mencari *face landmark* menggunakan *shape_predictor_68_face_landmarks*, dan yang terakhir adalah model untuk melakukan representasi citra wajah menjadi nilai vektor 128D menggunakan *dlib_face_recognition_resnet_model_v1*. Semua model ini disediakan oleh library *machine learning* *dlib*. Perbandingan citra wajah adalah proses membandingkan nilai vektor 128D dari citra wajah yang terdeteksi dengan nilai yang disimpan pada *datastore*. Nilai tersebut dibandingkan yang nantinya akan menghasilkan sebuah status dikenal atau tidak dari citra wajah jika hasil perbandingannya mendekati nilai *threshold*.

6. Pengirim pemberitahuan adalah proses yang dilakukan saat orang tak dikenal terdeteksi oleh sistem. Data yang dibutuhkan adalah gambar hasil *capture* kamera yang berisi wajah orang tak dikenal tersebut dan kapan waktu terdeteksinya.
8. Pemroses CRUD data orang tidak dikenal ini dilakukan dengan media chatbot untuk pengguna berinteraksi dengan sistem.
9. Model citra wajah adalah sebuah model dengan bentuk sebuah nilai hasil *encoding* yaitu berupa nilai vektor 128D.
10. UI Penampil Pemberitahuan adalah sebuah tampilan antarmuka yang digunakan untuk menampilkan pemberitahuan ketika sistem mendeteksi dan mengenali adanya orang yang tidak dikenal.
11. UI CRUD data citra wajah yang dikenal dan tidak adalah tampilan antarmuka untuk pemilik rumah berinteraksi dengan sistem untuk melakukan proses melihat daftar orang dikenal, menambahkan, menghapus, dan mengubah data orang yang dikenalnya.
12. *Datastore* merupakan tempat untuk menyimpan data.

V.4 Perancangan *Deployment* Sistem

Perancangan *deployment* sistem digambarkan menggunakan *deployment diagram*, bertujuan untuk menggambarkan arsitektur yang dibuat seperti pada Gambar V.4.



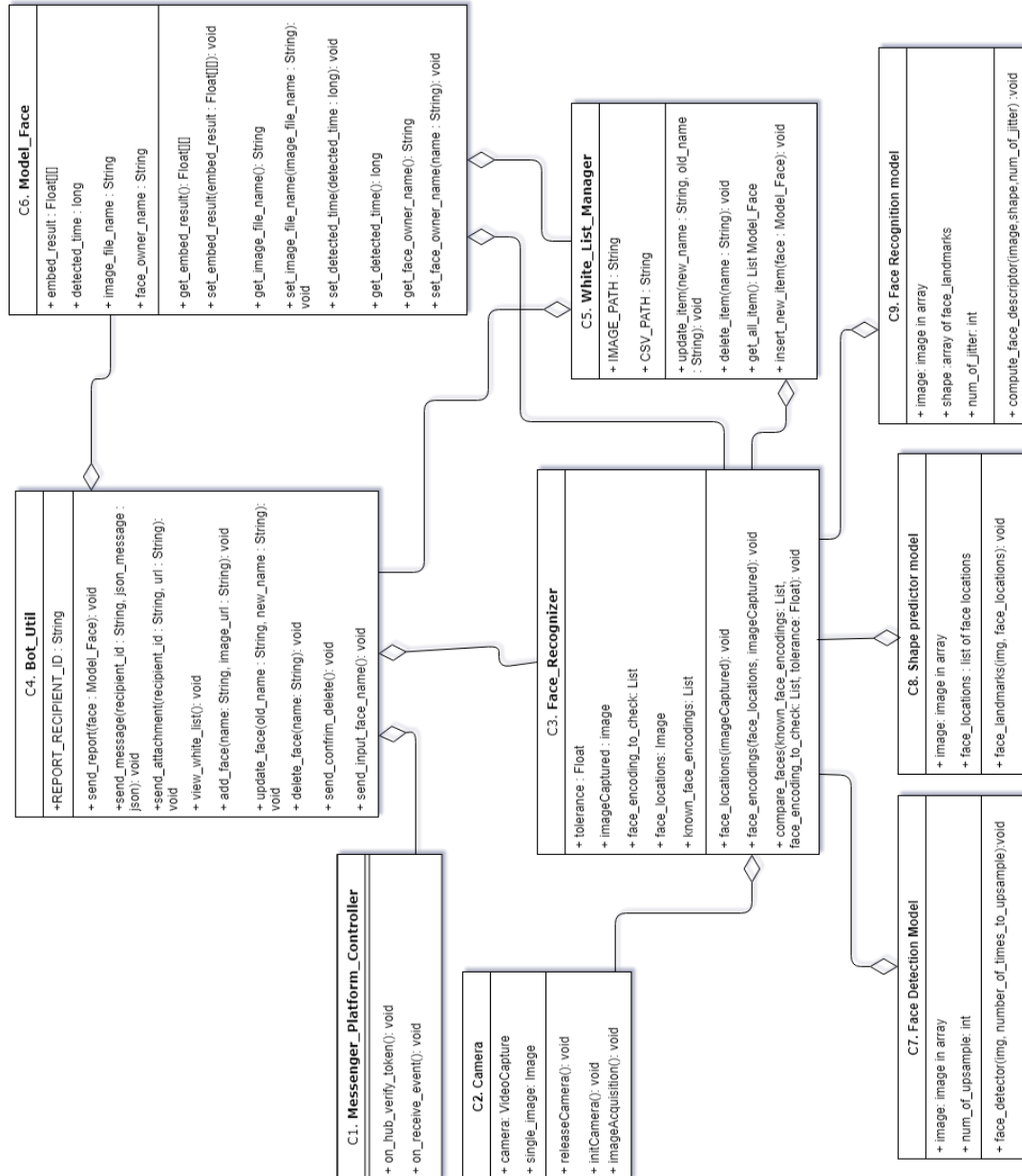
Gambar V.4 *Deployment* diagram

Penjelasan node dan komponen yang terdapat pada deployment diagram adalah sebagai berikut:

- a. Raspberry Pi 3 Model B, digunakan sebagai *single board computer* dari sistem dan tempat pengolahan untuk pengenalan wajah.
- b. Sistem Pendeteksi Orang tidak Dikenal, sistem yang dibuat penulis untuk mengenali orang yang tidak dikenal secara otomatis.
- c. Library OpenCV digunakan sebagai library pendukung untuk pengolahan citra digital.
- d. Library dlib, digunakan sebagai library pendukung untuk machine learning.
- e. Webhook Server, digunakan untuk menerima HTTP Request dari Messenger platform.
- f. Chatbot Messenger, sebuah aplikasi web service yang mengirim informasi dan menerima perintah pengguna dalam antarmuka IM Messenger.
- g. Library flask, framework yang digunakan untuk membangun web service yang dijalankan di webhook server.
- h. Raspberry Pi NoIR Camera Module v2, sebagai hardware untuk akuisisi citra. Jumlah kamera yang digunakan hanya satu buah.
- i. Messenger platform, sebagai media untuk interaksi antara pengguna dan sistem
- j. Ngrok, aplikasi yang menyediakan proses tunneling pada host HTTPS.
- k. Facebook graph API, antarmuka pemrograman pada aplikasi Facebook.
- l. Aplikasi Messenger, perangkat lunak Messenger yang diinstall pada smartphone.
- m. Smartphone, perangkat keras dengan kemampuan layaknya komputer yang digunakan pemilik rumah untuk berkomunikasi dan membantu kegiatan sehari-hari.

V.5 Perancangan Class Model

Perancangan ini bertujuan untuk menggambarkan *class-class* yang dibangun pada sistem yang digambarkan dengan *class diagram* pada Gambar V.5.



Gambar V.5 *Class diagram*

Penjelasan dari *class diagram* yang ditunjukkan mulai dari Tabel V.2.

Tabel V.2 Class Messenger_Platform_Controller

C1. Messenger_Platform_Controller			
Deskripsi	Class yang berfungsi untuk menerima event webhook dari Messenger Platform dan meneruskannya ke class Bot_Util untuk diolah.		
Atribut	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return	Deskripsi
on_hub_verify_token() ()	Public	void	method untuk memverifikasi token yang dikirimkan messenger platform setelah menyimpan endpoint baru di dasbor akun messenger bot.
on_receive_event() ()	Public	void	method yang dipanggil webhook server setiap kali ada event webhook yang dikirimkan Messenger platform. Di dalam method ini berisi fungsi utama dari bot.

Tabel V.3 Class Camera

C2. Camera			
Deskripsi	class yang berfungsi sebagai interface camera dan sistem		
Atribut	Visibility	Type Data	Deskripsi
camera	Public	VideoCapture	atribut untuk menghubungkan camera dengan sistem
single_image	Public	Image	atribut untuk citra yang diakuisisi oleh camera
Method			
Nama	Visibility	Return	Deskripsi
initCamera()	Public	void	method yang berfungsi untuk mengkonfigurasi camera
imageAcquisition()	Public	Image	method yang digunakan untuk mengakuisisi citra

releaseCamera()	Public	void	method yang digunakan untuk menonaktifkan kamera
-----------------	--------	------	--

Tabel V.4 Class Face_Recognizer

C3. Face_Recognizer			
Deskripsi	class yang berfungsi untuk mengolah citra wajah		
Atribut	Visibility	Type Data	Deskripsi
tolerance	Private	Float	Variable yang digunakan sebagai threshold untuk membandingkan citra
imageCaptured	Public	Image	Variable atau atribut untuk menampung citra
face_encoding_to_check	Public	List	Atribut yang digunakan untuk menampung nilai vector 128D dari wajah
known_face_encodings	Public	List	Atribut dari nilai vector 128D dari wajah yang dikenal
face_locations	Public	Image	Atribut untuk menampung citra yang ditemukan lokasinya
Method			
Nama	Visibility	Return	Deskripsi
face_locations(imageCaptured)	Public		Method untuk mendeteksi wajah dari citra yang tertangkap kamera
face_encodings(face_locations, imageCaptured)	Public		Memberi nilai terhadap wajah yang telah dideteksi
compare_faces(known_face_encodings: List, face_encoding_to_check: List, tolerance: Float)	Public	Boolean	Membandingkan nilai antara wajah yang diketahui nilai, wajah yang dibandingkan, sehingga mengembalikan nilai benar atau salah

Tabel V.5 Class Bot_Util

C4. Bot_Util			
Deskripsi	Class yang berfungsi untuk melakukan fungsi utama bot di sistem. Bot dapat mengirimkan pesan, mengubah data wajah, menambah data wajah baru, menghapus data wajah, dan mengirimkan report.		
Atribut	Visibility	Type Data	Deskripsi
REPORT_RECIPIENT_ID	Public	String	Konstanta id akun messenger pemilik rumah. Bot akan mengirimkan pesan ke id yang tertulis di konstanta ini.
Method			
Nama	Visibility	Return	Deskripsi
send_report(face: Face)	Public	void	method untuk mengirimkan report ke pemilik rumah
send_messages(recipient_id: String, json_messages: json)	Public	void	method untuk mengirimkan pesan ke akun messenger pemilik rumah
view_white_list()	Public	void	method untuk mengirimkan pesan yang berisi daftar whitelist ke akun messenger pemilik rumah
add_face(name: String, image_url: String)		void	method untuk bot menambah data wajah baru
update_face(old_name: String, new_name: String)		void	method untuk bot mengubah data wajah di <i>database</i>
delete_face(name: String)	public	void	method untuk bot menghapus data wajah di <i>database</i>
send_confirm_delete()	public	void	method untuk bot mengirimkan pesan konfirmasi sebelum menghapus data wajah di <i>database</i>
send_input_face_name()	public	void	method untuk bot mengirimkan pesan yang berisi meminta data nama untuk diinputkan

Tabel V.6 Class White_List_Manager

C5. White_List_Manager			
Deskripsi	Class yang berfungsi untuk melakukan operasi pengolahan <i>database</i> seperti menambah, menghapus, mengubah, dan mengambil data item wajah yang sudah dikenali sistem		
Atribut	Visibility	Type Data	Deskripsi
IMAGE_PATH	Public	String	konstanta yang berisi alamat direktori gambar
CSV_PATH	Public	String	konstanta yang memuat alamat direktori csv
Method			
Nama	Visibility	Return	Deskripsi
update_item(name: String, face:ModelFace)	Public	Void	method untuk mengubah data item wajah
delete_item(face: ModelFace)	Public	Void	method untuk menghapus data item wajah
get_all_item()	Public	List Model_Face	method untuk mendapatkan daftar item wajah
insert_new_item(face: ModelFace)	Public	Void	method untuk menambah data item wajah baru

Tabel V.7 Class Model_Face

C6. Model_Face			
Deskripsi	Class yang berfungsi untuk menampung data wajah yang telah diolah sistem.		
Atribut	Visibility	Type Data	Deskripsi
embed_result	Public	Array Float	variabel array 2 dimensi berisi nilai hasil proses <i>encoding</i>
detected_time	Public	Long	variabel yang berisi tanggal dan waktu orang tak dikenal itu terdeteksi

image_file_name	Public	String	variabel yang berisi nama berkas gambar yang ditangkap kamera.
face_owner_name	Public	String	variabel yang berisi nama orang pemilik wajah. Variabel ini berisi kosong apabila belum pemilik rumah tidak menambahkannya ke daftar orang dikenal.
Method			
Nama	Visibility	Return	Deskripsi
get_embed_result()	Public	Float[][]	method untuk mendapat nilai hasil proses <i>encoding</i>
set_embed_result(embed_result: Float[][])	Public	Void	method untuk menyimpan data hasil proses <i>encoding</i> ke variabel.
get_image_file_name()	Public	String	method untuk mendapatkan nama berkas gambar wajah
set_image_file_name(image_file_name: String)	Public	String	method untuk menyimpan data nama berkas gambar wajah ke variabel.
set_detected_time(detected_time: long)	Public	Void	method untuk menyimpan data tanggal dan waktu orang tak dikenal ini terdeteksi ke variabel.
get_detected_time()	Public	long	method untuk mendapat waktu dan tanggal terdeteksinya orang tak dikenal ini.
get_face_owner_name()	Public	String	method untuk mendapat nama pemilik wajah.
set_face_owner_name(name: String)	Public	Void	method untuk menyimpan data nama pemilik wajah ke variabel.

Tabel V.8 Class Face Detection Model

C7. Face Detection Model	
Deskripsi	Class model untuk mendeteksi wajah

Atribut	Visibility	Type Data	Deskripsi
image	public	image in array	atribut untuk menampung citra yang dideteksi
num_of_upsample	public	int	
Method			
Nama	Visibility	Return	Deskripsi
face_detector(img, num_of_upsample)	Public	Void	method untuk mendeteksi wajah pada citra yang diakuisisi

Tabel V.9 Class Shape predictor model

C8. Shape predictor model			
Deskripsi	Class model untuk mencari <i>face landmark</i> pada wajah		
Atribut	Visibility	Type Data	Deskripsi
image	public	image in array	atribut untuk menampung citra yang dideteksi
face_locations	public	List dari face locations	Atribut dari posisi letak wajah manusia
Method			
Nama	Visibility	Return	Deskripsi
face_landmark(img, face_locations)	Public	Void	Method untuk mencari 68 <i>face landmark</i> pada wajah

Tabel V.10 Class Face Recognition Model

C9. Face Recognition Model			
Deskripsi	Class model untuk pengenalan wajah		
Atribut	Visibility	Type Data	Deskripsi
image	public	image in array	atribut untuk menampung citra yang dideteksi

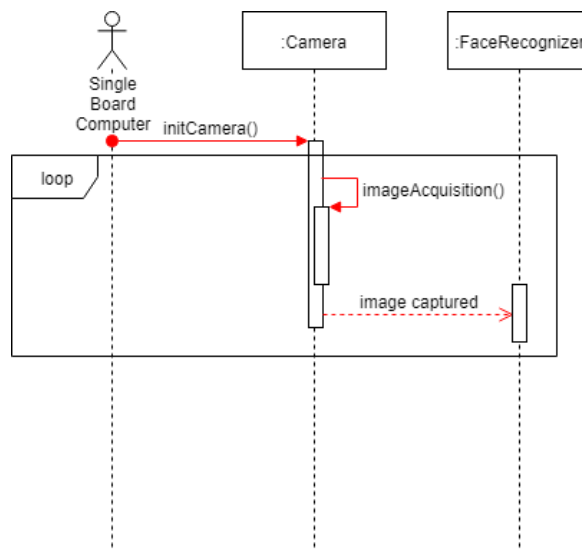
shape	public	array dari face_landmark	Array dari hasil pencarian face landmark dari wajah yang terakuisisi
num_of_jitter	public	int	
Method			
Nama	Visibility	Return	Deskripsi
compute_face_descriptor(image, shape, num_of_jitter)	public	void	method untuk menecoding citra wajah menjadi nilai vektor 128D

V.6 Perancangan Interaksi Antar Objek

Perancangan interaksi antar objek menggambarkan interaksi antar *class* berdasarkan fungsi yang dibangun dalam aplikasi. Penggambaran interaksi dengan menggunakan *sequence* diagram berdasarkan *use case* yang telah dibuat.

V.6.1 Sequence Diagram Akuisisi Citra

Sequence diagram ini menggambarkan interaksi kamera mengakuisisi citra hingga nantinya citra didapatkan.



Gambar V.6 SD akuisisi citra

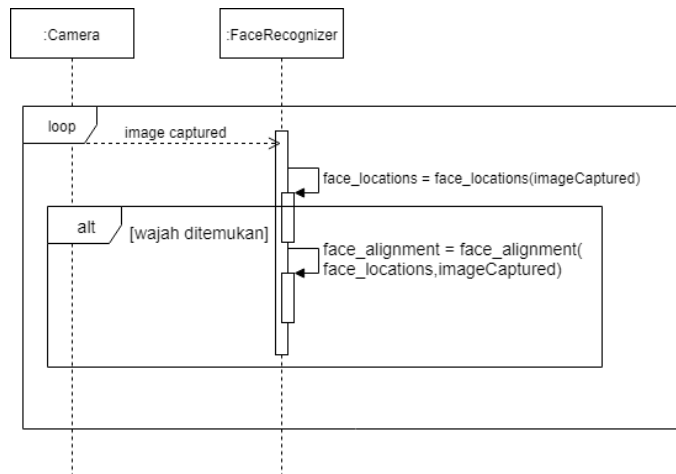
Tabel V.11 menunjukkan tentang deskripsi serta *class* dan *method* terkait yang ada pada Gambar V.6.

Tabel V.11 SD akuisisi citra

ID Sequence Diagram	SD-01
Nama Sequence Diagram	Sequence Diagram Akuisisi Citra
ID Use Case	UC-1
ID Requirement	REQ-F-U1-001
Nama Class Terkait	Camera, FaceRecognizer
Deskripsi	Sequence diagram untuk akuisisi citra yang ditangkap oleh kamera, diawali dari konfigurasi kamera dan akan mengakuisis setiap frame yang tertangkap dan mengirimkannya ke class facerecognizer
Method yang terkait	initCamera(), imageAcquisition()

V.6.2 Sequence Diagram Mendeteksi Wajah

Sequence diagram ini menggambarkan interaksi kamera dan *single board computer* ketika melakukan proses pencarian wajah pada citra yang diakuisisi oleh kamera.



Gambar V.7 SD mendeteksi wajah

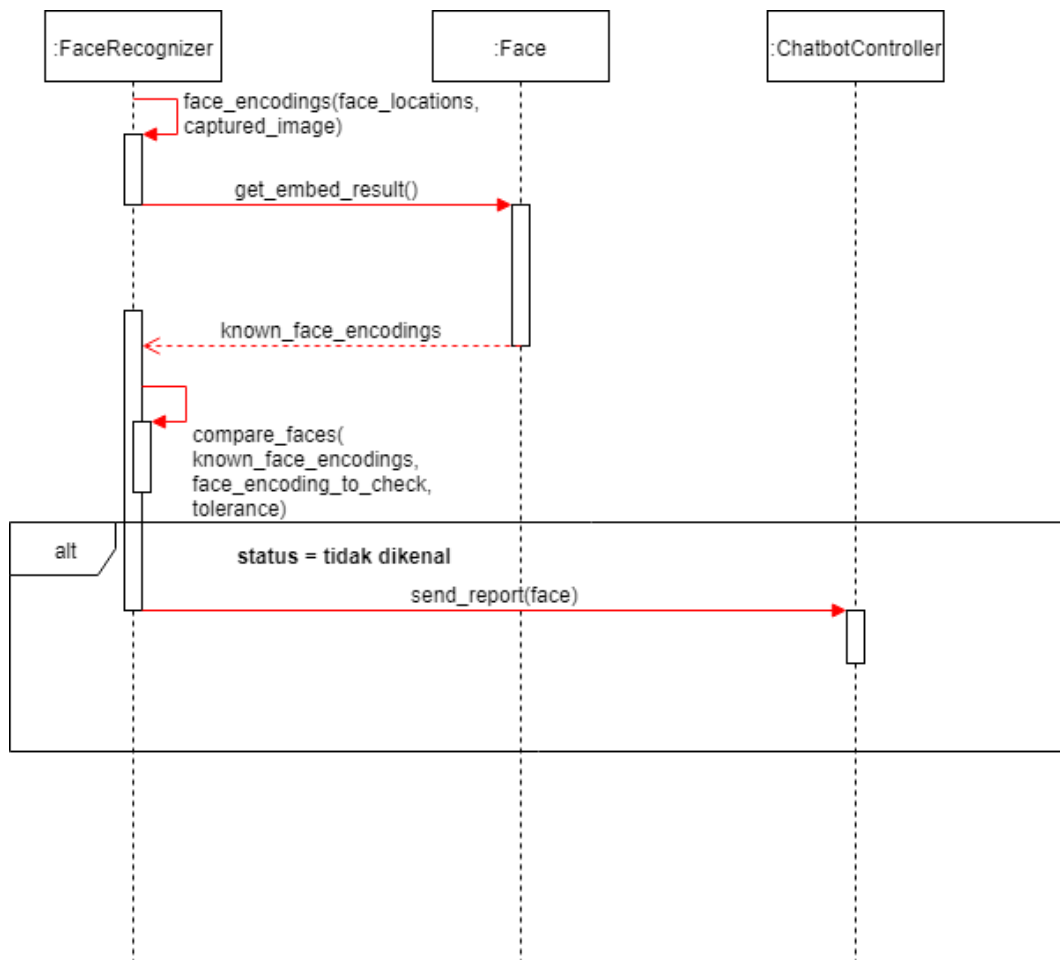
Penjelasan algoritma dan *method* yang digunakan pada Gambar V.7 dijelaskan pada Tabel V.12.

Tabel V.12 Penjelasan SD mendeteksi wajah

ID Sequence Diagram	SD-02
Nama Sequence Diagram	Sequence Diagram Mendeteksi Wajah
ID Use Case	UC-1
ID Requirement	REQ-F-U1-002, REQ-F-U1-003, REQ-F-U1-004
Nama Class Terkait	Camera, FaceRecognizer
Deskripsi	Sequence diagram untuk mendeteksi wajah dengan input berupa citra RGB yang telah diakusisi dan diproses menggunakan <code>dlib.get_frontal_face_detector()</code> dari library <code>dlib</code> untuk mencari wajah yang ada pada citra yang tertangkap dengan output berupa lokasi letak wajah jika terdeteksi
Method yang terkait	<code>face_locations(imageCaptured)</code> , <code>face_alignment(face_locations, imageCaptured)</code>
<p>Algoritma keterangan method</p> <p>Kamus data:</p> <pre> imageCaptured : image face_locations : detector :</pre> <p>FUNCTION <code>face_locations(imageCaptured)</code></p> <p>BEGIN</p> <pre> detector <- Face_Detection_Model.face_detector(imageCaptured,1) face_locations = detector(imageCaptured) face_alignment = face_alignment(face_locations, imageCaptured) return face_alignment</pre> <p>END FUNCTION</p> <p>FUNCTION <code>face_alignment(face_locations, imageCaptured)</code></p> <p>BEGIN</p> <pre> faces <- dlib.full_object_detections() sp <- face_landmarks(imageCaptured, face_locations) for detection in face_locations: faces.append(sp(imageCaptured, detection)) images = dlib.get_face_chips(imageCaptured, faces, sizes=150) return images</pre> <p>END FUNCTION</p>	

V.6.3 Sequence Diagram Mengenali Wajah

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika melakukan pengenalan wajah.



Gambar V.8 SD Mengenali wajah

Penjelasan *method* dan algoritma yang digunakan pada Gambar V.8 ditunjukkan pada Tabel V.13.

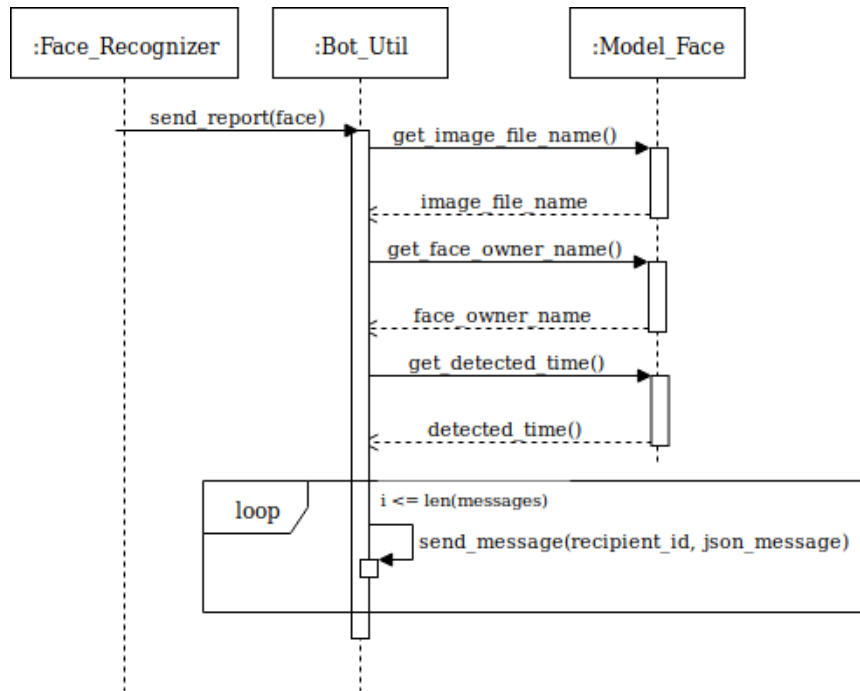
Tabel V.13 Penjelasan SD mengenali wajah

ID Sequence Diagram	SD-03
Nama Sequence Diagram	Sequence Diagram Mengenali Wajah
ID Use Case	UC-2
ID Requirement	REQ-F-U2-001, REQ-F-U2-002, REQ-F-U2-003

Nama Terkait	Class	FaceRecognizer, Face, ChatbotController
Deskripsi		Sequence diagram ini menjelaskan proses pengenalan wajah dengan input citra wajah yang telah diproyeksikan dengan proses pemberian nilai vector 128D pada citra wajah lalu dibandingkan dengan nilai yang ada pada <i>database</i> dengan menggunakan euclidean distance dan menghasilkan output apakah status dikenal atau tidak
Method terkait	yang	face_encoding(face_locations, captured_image), get_embed_result(), compare_faces(known_face_encodings, face_encoding_to_check, tolerance), push_message(receiptId, message)
Algoritma keterangan method Kamus data: <pre> known_face_encodings: Objek bertipe List dari wajah yang diketahui face_encoding_to_check: Objek bertipe List tolerance: variabel lokal bertipe float receiptId: message: facerec : objek dari class face_recognition_model_v1 //library dlib shape_predictor : objek dari class shape_predictor //library dlib </pre> FUNCTION face_encoding(face_locations, captured_image) BEGIN <pre> encoding = Face_Recognition_Model.compute_face_descriptor(captured_image, face_alignment, 1) return encoding </pre> END FUNCTION		

V.6.4 Sequence Diagram Mengirimkan Pemberitahuan

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika melakukan pengiriman pemberitahuan pada saat mendeteksi orang tidak dikenal. Class yang berkaitan pada proses mengirimkan pemberitahuan diantaranya *class* Face_Recognizer, Bot_Util, Model_Face sesuai yang ditunjukkan pada Gambar V.9.



Gambar V.9 Sequence diagram mengirimkan pemberitahuan

Penjelasan *method* dan algoritma yang digunakan pada Gambar V.9 ditunjukkan pada Tabel V.14.

Tabel V.14 Penjelasan sequence diagram mengirimkan pemberitahuan

ID Sequence Diagram	SD-03
Nama Sequence Diagram	Sequence Diagram Mengirimkan Pemberitahuan
ID Use Case	UC-3
ID Requirement	REQ-F-U3-001, REQ-F-U3-002, REQ-F-U3-003, REQ-F-U3-004, REQ-F-U3-005
Nama Class Terkait	BotUtil, Face, WhitelistManager
Deskripsi	<i>Sequence diagram</i> ini menjelaskan proses pengiriman report oleh bot kepada pemilik rumah. Langkah-langkahnya adalah menampung data waktu terdeteksi dan gambar. Proses selanjutnya adalah membuat variabel dengan tipe data Dictionary untuk merancang struktur json data yang akan dikirim ke Messenger platform. Setelah data telah disiapkan, tahap selanjutnya ialah melakukan HTTP Request Post pada

	Messenger platform untuk mengirim pesan teks, pesan bertombol, dan lampiran gambar. Proses HTTP Request tersebut sudah dikemas dalam method <code>send_attachment(recipient_id, url)</code> dan <code>send_message(recipient_id, json_message)</code>
Method yang terkait	<code>get_image_file_name()</code> , <code>get_face_owner_name()</code> , <code>push_message(recipient_id, json_message)</code>

Algoritma keterangan method

Kamus data:

`REPORT_RECIPIENT_ID`: Objek konstanta bertipe String

`IMAGE_PATH` : Objek konstanta bertipe String

PROCEDURE `send_report(face)`

BEGIN

```

timestamp <- face.get_detected_time()
image <- face.get_image_file_name()
text_chat <- {
    "recipient" : {
        "id" : REPORT_RECIPIENT_ID
    },
    "messages" : {
        "text" : "Ada orang tak dikenal terdeteksi
pada " + timestamp
    }
}
text_button = {
    "recipient" : {
        "id" : REPORT_RECIPIENT_ID
    },
    "messages" : {
        "attachment" : {
            "payload" : {
                "text" : "Apakah ingin ditambahkan ke
daftar dikenal?",
                "buttons" : [
                    {
                        "title" : "ya"
                    }
                ]
            }
        }
    }
}

```

```

send_attachments(REPORT_RECIPIENT_ID, IMAGE_PATH + image)
send_messages(REPORT_RECIPIENT_ID, json.dumps(text_chat))
send_messages(REPORT_RECIPIENT_ID, json.dumps(text_button))

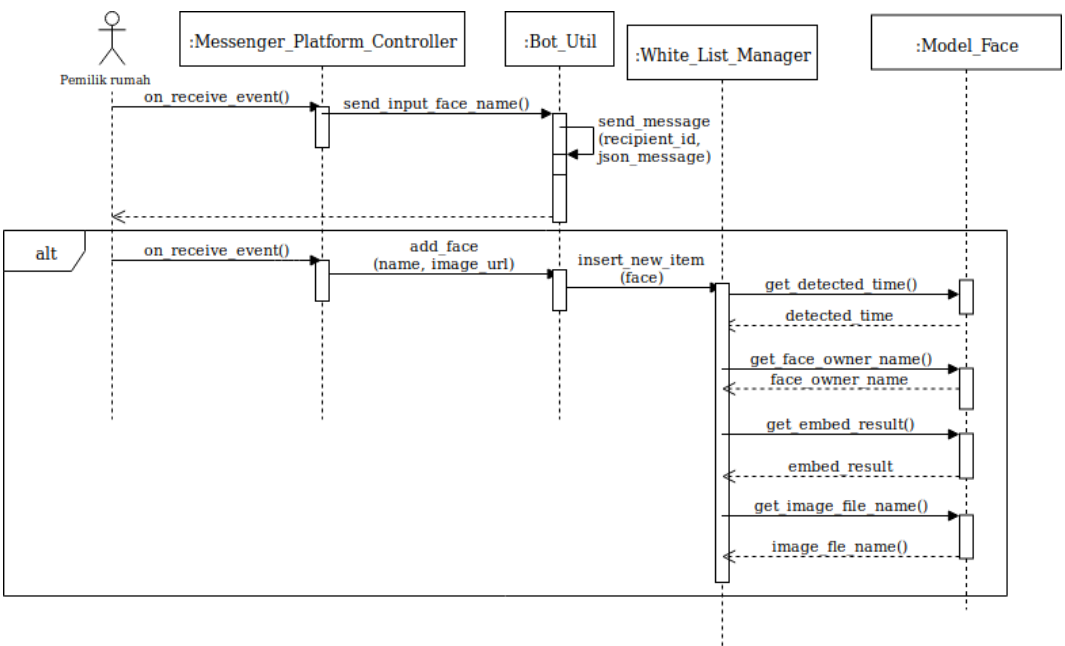
```

END PROCEDURE



V.6.5 Sequence Diagram Menambah Data Orang yang Dikenal

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika melakukan penambahan data orang yang dikenal dari pemberitahuan yang dikirimkan yang nantinya data yang ditambahkan akan disimpan pada *database*.



Gambar V.10 Sequence diagram menambah data orang yang dikenal

Penjelasan *method* dan algoritma yang digunakan pada Gambar V.10 ditunjukkan pada Tabel V. 15.

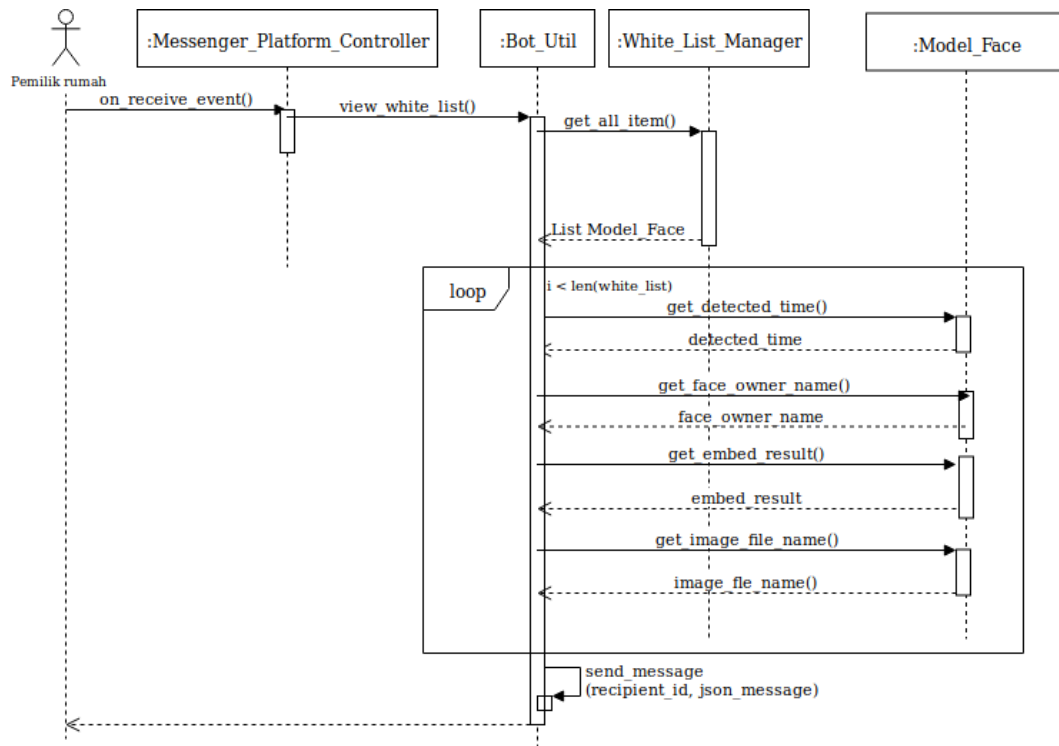
Tabel V. 15 Penjelasan SD menambah data orang dikenal

ID Sequence Diagram	SD-05
Nama Sequence Diagram	Sequence Diagram Menambah Data Orang Dikenal
ID Use Case	UC-4

ID Requirement	REQ-F-U4-001
Nama Class Terkait	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
Deskripsi	<i>Sequence diagram</i> ini menjelaskan proses penambahan data baru ke daftar orang dikenal. Setelah pemilik rumah menekan tombol “tambah” di report, Bot mengirim pesan teks yang berisi meminta agar pemilik rumah Pemilik rumah memasukkan nama untuk data baru whitelist. Setelah dimasukkan, bot akan mengolah data tersebut ke dalam format csv kemudian menyimpannya ke storage di <i>Single board computer</i> .
Method yang terkait	get_image_file_name(), get_face_owner_name(), get_detected_time(), send_message(recipient_id, json_message), on_receive_event(), send_input_face(), add_face(name, image_url), insert_new_item(face)
Algoritma keterangan method Kamus data: <pre> REPORT_RECIPIENT_ID: Objek konstanta bertipe String CSV_PATH: Objek konstanta bertipe String IMAGE_PATH: Objek konstanta bertipe String </pre> PROCEDURE insert_new_item(face) BEGIN <pre> timestamp <- face.get_detected_time() image <- face.get_image_file_name() embedded <- face.get_embed_result() name <- face.get_face_owner_name() </pre> END PROCEDURE PROCEDURE send_input_face() BEGIN <pre> text_chat <- { "recipient" : { "id" : REPORT_RECIPIENT_ID }, "messages" : { "text" : "Masukkan nama.." } } send_messages(REPORT_RECIPIENT_ID, json.dumps(text_chat)) </pre> END PROCEDURE	

V.6.6 Sequence Diagram Melihat Daftar Orang yang Dikenal

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika menjalankan perintah melihat daftar orang yang dikenal yang tersimpan pada *database*.



Gambar V.11 Sequence diagram melihat daftar orang yang dikenal

Penjelasan *method* dan algoritma yang digunakan pada Gambar V.11 ditunjukkan pada Tabel V.16.

Tabel V.16 Penjelasan sequence diagram melihat daftar orang dikenal

ID Sequence Diagram	SD-06
Nama Sequence Diagram	Sequence Diagram Melihat Daftar Orang Dikenal
ID Use Case	UC-5
ID Requirement	REQ-F-U5-001
Nama Class Terkait	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
Deskripsi	<i>Sequence diagram</i> ini menjelaskan proses untuk bot menampilkan daftar orang yang sudah dikenali. Tahapannya ialah pemilik rumah mengirim pesan ke bot dengan format “lihat whitelist”. Lalu Bot akan membalas pesan tersebut dengan Carousel yang memuat daftar orang yang sudah dikenali.
Method yang terkait	get_image_file_name(), get_face_owner_name(), get_detected_time(), send_message(recipient_id, json_message), get_all_item(), view_white_list(), on_receive_event()

Algoritma keterangan method

Kamus data:

REPORT_RECIPIENT_ID: Objek konstanta bertipe String

CSV_PATH: Objek konstanta bertipe String

IMAGE_PATH: Objek konstanta bertipe String

PROCEDURE view_white_list()

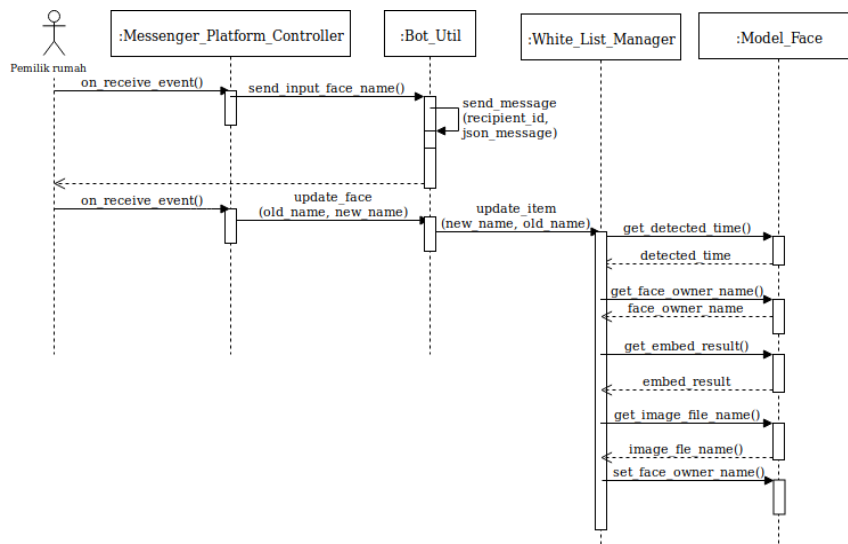
BEGIN

```
text_carousel = {  
    "recipient" : {  
        "id" : REPORT_RECIPIENT_ID  
    },  
    "messages" : {  
        "attachment":{  
            "type" : "template",  
            "payload" : {  
                "template_type" : "generic",  
                "elements" : []  
            }  
        }  
    }  
}
```

END PROCEDURE

V.6.7 Sequence Diagram Mengubah Data Orang Dikenal

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika pengguna menjalankan perintah mengubah data pada daftar orang yang dikenal.



Gambar V.12 SD Mengubah data orang dikenal

Penjelasan *method* dan algoritma yang digunakan pada Gambar V.12 ditunjukkan pada Tabel V.17.

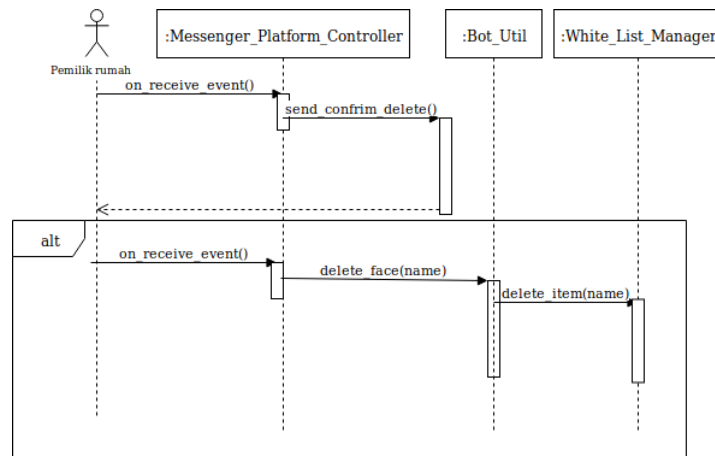
Tabel V.17 Penjelasan sequence diagram mengubah data orang dikenal

ID Sequence Diagram	SD-07
Nama Sequence Diagram	Sequence Diagram Mengubah Data Orang Dikenal
ID Use Case	UC-7
ID Requirement	REQ-F-U7-001, REQ-F-U7-002
Nama Class Terkait	BotUtil, Model_Face, White_List_Manager, Messenger_Platform_Controller
Deskripsi	<i>Sequence diagram</i> ini menjelaskan proses untuk bot mengubah data dari daftar orang yang sudah dikenali. Tahapannya setelah pemilik rumah memilih data dan menekan “ubah”, Bot mengirim pesan teks yang berisi meminta agar pemilik rumah Pemilik rumah memasukkan nama baru yang diinginkan. Setelah dimasukkan, bot akan mengolah data tersebut ke dalam format csv kemudian menyimpannya ke storage <i>Single board computer</i> .
Method yang terkait	get_image_file_name(), get_face_owner_name(), get_detected_time(), get_embed_result(), set_face_owner_name(), update_item(new_name, old_name), send_message(recipient_id, json_message), update_face(old_name, new_name), on_receive_event(), send_input_face_name()
<p>Algoritma keterangan method</p> <p>Kamus data:</p> <p>REPORT_RECIPIENT_ID: Objek konstanta bertipe String CSV_PATH: Objek konstanta bertipe String IMAGE_PATH: Objek konstanta bertipe String</p> <p>PROCEDURE update_item(new_name, old_name) BEGIN</p> <p>timestamp <- face.get_detected_time() image <- face.get_image_file_name()</p> <p>END PROCEDURE</p> <p>PROCEDURE send_input_face_name() BEGIN</p> <p>text_chat <- { "recipient" : { "id" : REPORT_RECIPIENT_ID }, "messages" : { "text" : "Masukkan nama.."</p> <p> } } send_messages(REPORT_RECIPIENT_ID, json.dumps(text_chat))</p>	

END PROCEDURE

V.6.8 Sequence Diagram Menghapus Data Orang Dikenal

Sequence diagram ini menggambarkan interaksi yang terjadi pada sistem ketika pengguna menjalankan perintah menghapus data pada daftar orang yang dikenal.



Gambar V.13 Sequence diagram menghapus data orang dikenal

Penjelasan *method* dan algoritma yang digunakan pada

Gambar V.13 ditunjukkan pada Tabel V.18.

Tabel V.18 Sequence diagram menghapus data orang dikenal

ID Sequence Diagram	SD-08
Nama Sequence Diagram	Sequence Diagram Menghapus Data Orang Dikenal
ID Use Case	UC-6
ID Requirement	REQ-F-U6-001
Nama Class Terkait	BotUtil, White_List_Manager, Messenger_Platform_Controller
Deskripsi	<i>Sequence diagram</i> ini menjelaskan proses untuk bot menghapus data dari daftar orang yang sudah dikenali. Tahapannya setelah pemilik rumah memilih data dan menekan “hapus”, Bot mengirim pesan teks dan <i>quick_reply</i> yang mengkonfirmasi apakah pemilik rumah yakin untuk menghapus data tersebut

	atau tidak. Setelah dipilih, bot akan menghapus data csv dan gambar yang dimiliki data tersebut dari storage <i>Single board computer</i> .
Method yang terkait	send_message(recipient_id, json_message), on_receive_event, delete_face(name), delete_item(name), send_confrim_delete()
<p>Algoritma keterangan method</p> <p>Kamus data:</p> <p>REPORT_RECIPIENT_ID: Objek konstanta bertipe String CSV_PATH: Objek konstanta bertipe String IMAGE_PATH: Objek konstanta bertipe String</p> <p>PROCEDURE delete_item(name) BEGIN for i to get_files(CSV_PATH):</p> <p>END PROCEDURE</p> <p>PROCEDURE send_confrim_delete() BEGI</p> <pre> text_chat <- { "recipient" : { "id" : REPORT_RECIPIENT_ID }, "messages" : { "text" : "Apakah anda yakin?", "quick_replies" : [{ "content_type": "text", "title" : "Ya, hapus", "payload" : { "status" : true } }, { "content_type": "text", "title" : "Tidak, batalkan", "payload" : { "status" : false } }] } } send_messages(REPORT_RECIPIENT_ID, json.dumps(text_chat)) </pre> <p>END PROCEDURE</p>	

V.7 Perancangan Tampilan


Perancangan tampilan mengacu pada subbab *User Interface* yang terdapat pada dokumen SRS. Tujuan dari perancangan tampilan ini adalah untuk membuat sebuah

gambaran yang nantinya digunakan untuk berkomunikasi antara pengguna dan sistem. Berikut dipaparkan setiap perancangan tampilan yang terdapat sistem pendeteksi orang yang tidak dikenal ini.

V.7.1 User Interface Tambah Data ke Daftar Orang Dikenal

Penjelasan mengenai tampilan antar muka ketika menambahkan data orang yang dikenal ditunjukkan pada Tabel V.19.

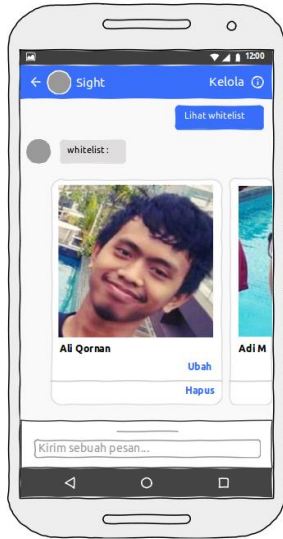
Tabel V.19 Penjelasan UI tambah data ke daftar orang dikenal

No UI	UI-1
Nama UI	Tambah data ke daftar orang dikenal
ID Requirement	REQ-F-U4-001
Precondition	Pemilik rumah meminta untuk menambah data ke whitelist
Deskripsi	User Interface berfungsi untuk pemilik rumah memberikan perintah pada chatbot messenger untuk siap menerima dan mengolah berkas gambar yang dikirimkan. Perintahnya adalah “tambah whitelist”.
Gambar	
	

V.7.2 User Interface Lihat Daftar Orang Dikenal

Penjelasan mengenai tampilan antar muka ketika melihat data orang yang dikenal ditunjukkan pada Tabel V.20.


Tabel V.20 Penjelasan UI lihat daftar orang dikenal

No UI	UI-3
Nama UI	Lihat Daftar Whitelist
ID Requirement	REQ-F-U5-001
Precondition	Pemilik rumah meminta untuk menampilkan daftar whitelist
Deskripsi	User Interface berfungsi untuk memberikan perintah agar chatbot messenger menampilkan daftar orang yang dikenal sistem. Tampilan daftar ini menggunakan bentuk templat pesan generik dan menerapkan <i>carousel</i> . Sintak perintah yang diterima adalah “lihat whitelist”. Tiap data pada daftar ini berisi foto wajah dan nama orang yang telah dikenal sistem. Selain itu, di bawah ada dua tombol yaitu “ubah” dan “hapus”.
Gambar	
	

V.7.3 User Interface Mengapus Data pada Daftar Orang Dikenal

Penjelasan mengenai tampilan antar muka ketika menghapus data dari daftar orang yang dikenal ditunjukkan pada Tabel V.21.

Tabel V.21 Penjelasan UI menghapus data orang yang dikenal

No UI	UI-3
Nama UI	Menghapus data orang yang dikenal
ID Requirement	REQ-F-U6-001
Precondition	Pemilik rumah meminta untuk menghapus data pada whitelist
Deskripsi	User Interface berfungsi untuk menampilkan konfirmasi sebelum sistem menghapus data yang dipilih dari <i>screen</i> sebelumnya. Tampilan konfirmasi ini berisi dua tombol yaitu “Ya, tidak” dan “Tidak, batalkan”. Apabila pemilik rumah menekan tombol “Ya, tidak” maka sistem akan menghapus data yang dipilih.
Gambar	
	

V.7.4 User Interface Mengubah Data pada Daftar Orang Dikenal

Penjelasan mengenai tampilan antar muka ketika mengubah data dari daftar orang yang dikenal ditunjukkan pada Tabel V.22.

Tabel V.22 Penjelasan UI mengubah data orang yang dikenal

No UI	UI-4
Nama UI	Mengubah data orang yang dikenal
ID Requirement	REQ-F-U7-001
Precondition	Pemilik rumah meminta untuk mengubah data pada whitelist
Deskripsi	User Interface berfungsi untuk memuat webview yang berisi form untuk mengubah data yang dipilih dari screen sebelumnya. Form ini berisi foto, field nama, dan tombol “ubah”. Apabila pemilik rumah menekan tombol “ubah” maka akan memberikan perintah pada sistem untuk mengubah data yang telah dipilih tersebut.
Gambar	
	

V.7.5 User Interface Menampilkan Pemberitahuan

Penjelasan mengenai tampilan antar muka ketika menampilkan pemberitahuan ditunjukkan pada Tabel V.23.

Tabel V.23 Penjelasan UI menampilkan pemberitahuan

No UI	UI-5
-------	------

Nama UI	Menampilkan Pemberitahuan
ID Requirement	REQ-F-U3-003
Precondition	Ada wajah orang tidak dikenal
Deskripsi	User Interface berfungsi untuk menampilkan pemberitahuan yang dikirim oleh <i>single board computer</i> . Pemberitahuan ini berisi foto orang yang terdeteksi dan sebuah tombol bertuliskan “tambah”.

Gambar



BAB VI

IMPLEMENTASI

Bab ini menjelaskan hasil implementasi dari perancangan yang telah dibuat. Hasil implementasi yang dilakukan dibahas pada subbab berikut.

VI.1 Keterangan Requirement yang Diimplementasikan

Berdasarkan *requirement* fungsional yang telah dituliskan pada SRS dari tiap *use case* yang ada, maka Tabel VI.1 menunjukkan daftar requirement fungsional yang diimplementasikan.

Tabel VI.1 *Requirement* yang diimplementasikan


No	Kode Requirement	Deskripsi	Interaksi	Mock-Up	Status Implementasi
1	REQ-F-U1-001	Sistem dapat melakukan akuisisi citra menggunakan kamera dengan resolusi 480p	SD-01		Sudah Diimplementasi
2	REQ-F-U1-002	Sistem mampu mendeteksi wajah manusia yang terakuisisi kamera secara otomatis dan terus-menerus selama kamera pengawas menyala	SD-02		Sudah Diimplementasi
3	REQ-F-U1-003	Sistem dapat mencari 68 face landmark pada wajah yang terdeteksi	SD-02		Sudah Diimplementasi
4	REQ-F-U1-004	Sistem dapat mentransformasikan wajah menjadi seolah-olah menghadap ke depan kamera	SD-02		Sudah Diimplementasi
5	REQ-F-U1-004	Sistem dapat mentransformasikan posisi wajah menjadi tegak lurus menghadap kamera	SD-02		Sudah Diimplementasi
6	REQ-F-U2-001	Sistem dapat melakukan <i>encoding</i> pada citra wajah yang telah ditemukan face landmarknya	SD-03		Sudah diimplementasi
7	REQ-F-U2-002	Sistem dapat membandingkan hasil <i>encoding</i> yang tertangkap kamera dengan yang ada pada <i>database</i>	SD-03		Sudah Diimplementasi

8	REQ-F-U2-003	Sistem dapat menentukan status dikenal atau tidak sesuai hasil perbandingan hasil <i>encoding</i>	SD-03		Sudah Diimplementasi
9	REQ-F-U2-004	Sistem dapat melakukan pengecekan ulang setiap tiga frame setelah citra dapat dikenali	SD-03		Sudah Diimplementasi
10	REQ-F-U3-001	Sistem dapat mengirimkan pemberitahuan ketika ada orang yang tidak dikenali melalui chatbot	SD-04		Sudah Diimplementasi
11	REQ-F-U3-002	Sistem dapat menentukan waktu kapan wajah orang yang tidak dikenali itu tertangkap oleh kamera	SD-04		Sudah Diimplementasi
12	REQ-F-U3-003	Sistem dapat mengirimkan pemberitahuan dengan konten citra wajah orang yang tidak dikenali dan waktu citra wajah yang tidak dikenali tersebut diakuisisi	SD-04	UI-5	Sudah Diimplementasi
13	REQ-F-U3-004	Sistem dapat menampilkan pemberitahuan melalui chatbot ketika ada orang tidak dikenal	SD-04		Sudah Diimplementasi
14	REQ-F-U4-001	Sistem dapat menambahkan data baru orang yang dikenal ke dalam <i>database</i>	SD-05	UI-1	Sudah Diimplementasi
15	REQ-F-U4-002	Sistem dapat melakukan <i>encoding</i> terhadap wajah yang dikirimkan oleh pemilik rumah	SD-05		Sudah Diimplementasi
16	REQ-F-U5-001	Sistem dapat menampilkan daftar orang yang dikenal sesuai dengan <i>database</i>	SD-06	UI-3	Sudah Diimplementasi
17	REQ-F-U6-001	Sistem dapat menghapus data pada whitelist sesuai yang dipilih pemilik rumah	SD-08	UI-4	Sudah Diimplementasi
18	REQ-F-U7-001	Sistem dapat mengubah nama pada whitelist	SD-07	UI-2	Sudah Diimplementasi
19	REQ-F-U7-002	Sistem dapat memperbaharui data pada <i>database</i> sesuai yang dipilih oleh pemilik rumah	SD-07	UI-2	Sudah Diimplementasi

VI.2 Hasil Implementasi

Hasil dari *requirement* yang sudah diimplementasikan dengan penjelasan singkat dan bukti berupa *screenshot*. Hasil implementasi dari setiap *requirement* yang telah diimplementasikan ditunjukkan mulai dari Tabel VI.2.

Tabel VI.2 Implementasi 001

No	Kode Requirement	Deskripsi	Class yang diimplementasi
1	REQ-F-U1-001	Sistem mampu mengakuisisi citra melalui kamera pengawas dengan resolusi 480p	C2 - Class Camera
2	REQ-F-U1-002	Sistem mampu mendeteksi wajah manusia yang terakuisisi kamera secara otomatis dan terus-menerus selama kamera pengawas menyala	C2 - Class Camera dan C3 - Class Face_Recognizer
Hasil Implementasi			
			

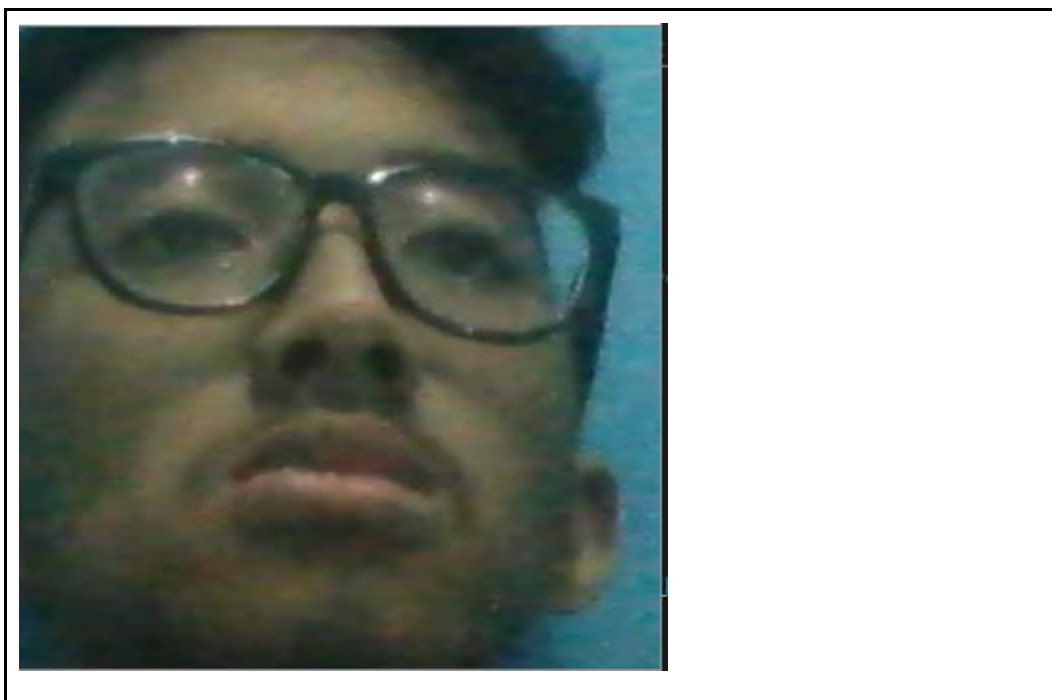
Tabel VI.3 Implementasi 002

No	Kode Requirement	Deskripsi	Class yang diimplementasi
----	------------------	-----------	---------------------------

3	REQ-F-U1-003	Sistem dapat mencari 68 <i>face landmark</i> pada wajah yang terdeteksi	C2- Class Face_Recognizer
Hasil Implementasi			
			

Tabel VI.4 Implementasi 003

No	Kode Requirement	Deskripsi	Class yang diimplementasi
4	REQ-F-U1-004	Sistem dapat mentransformasikan posisi wajah menjadi tegak lurus menghadap kamera	C2- Class Face_Recognizer
5	REQ-F-U1-005	Sistem dapat mengambil satu frame dari enam frame yang ada saat mengakuisisi citra	C2- Class Face_Recognizer
Hasil Implementasi			



Tabel VI.5 Implementasi 004

No	Kode Requirement	Deskripsi	Class yang diimplementasi
6	REQ-F-U2-001	Sistem dapat melakukan <i>encoding</i> pada citra wajah yang telah dideteksi	C2- Class Face_Recognizer
Hasil Implementasi			
Berikut adalah beberapa nilai dari jumlah keseluruhan nilai vector 128D yang berhasil digenerate dari wajah yang sama pada tabel ..			

```

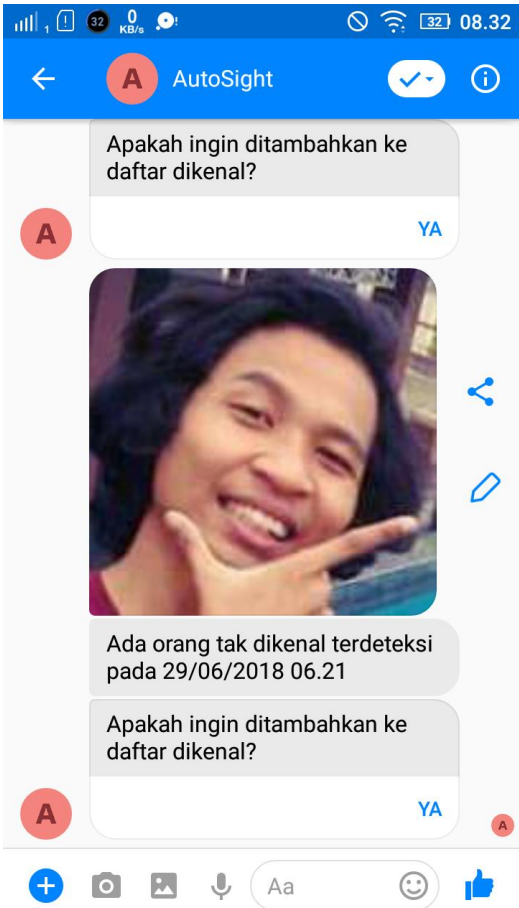
Detection 0: Left: 216 Top: 142 Right: 439 Bottom: 365
-0.126221
0.080451
0.115942
-0.0355988
-0.0531843
-0.0402167
-0.059496
-0.0433622
0.121788
-0.173156
0.208493
-0.0828206
-0.187573
-0.120866
-0.0494186
0.175344
-0.19794
-0.135614
-0.0261803
-0.0662593
0.0550397
-0.0309224
-0.0521663
0.0222901

```

Adanya keterangan left,right, top, bottom, merupakan posisi dimana wajah ditemukan dan dibawahnya adalah hasil generate citra yang didapatkan.

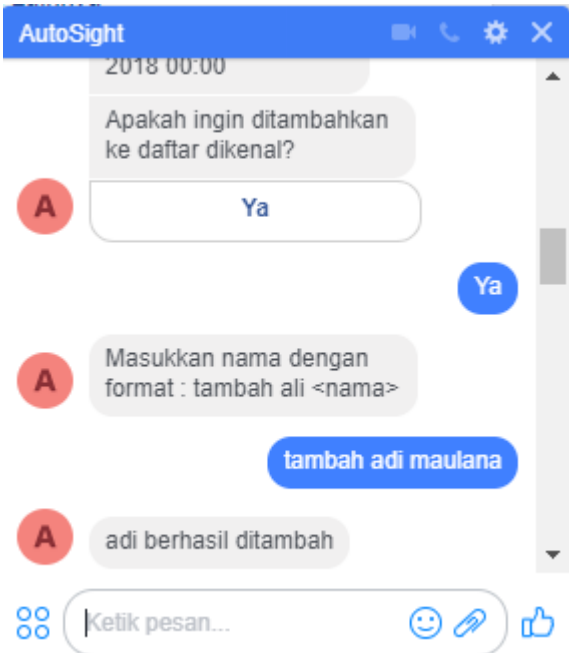
Tabel VI.6 Implementasi 005

No	Kode Requirement	Deskripsi	Class yang diimplementasi
7	REQ-F-U3-001	Sistem dapat mengirimkan pemberitahuan ketika ada orang yang tidak dikenali melalui chatbot	C4- Class Bot_Util
8	REQ-F-U3-002	Sistem dapat menentukan waktu kapan wajah orang yang tidak dikenali itu tertangkap oleh kamera	
9	REQ-F-U3-003	Sistem dapat mengirimkan pemberitahuan dengan konten citra wajah orang yang tidak dikenali dan waktu citra wajah yang tidak dikenali tersebut diakuisisi	

10	REQ-F-U3-004	Sistem dapat menampilkan pemberitahuan melalui chatbot ketika ada orang tidak dikenal	
Hasil Implementasi			
Berikut adalah screenshot hasil implelementasi requirement tersebut. Report ini tampil di aplikasi Messenger yang berjalan di smartphone Android.			
			

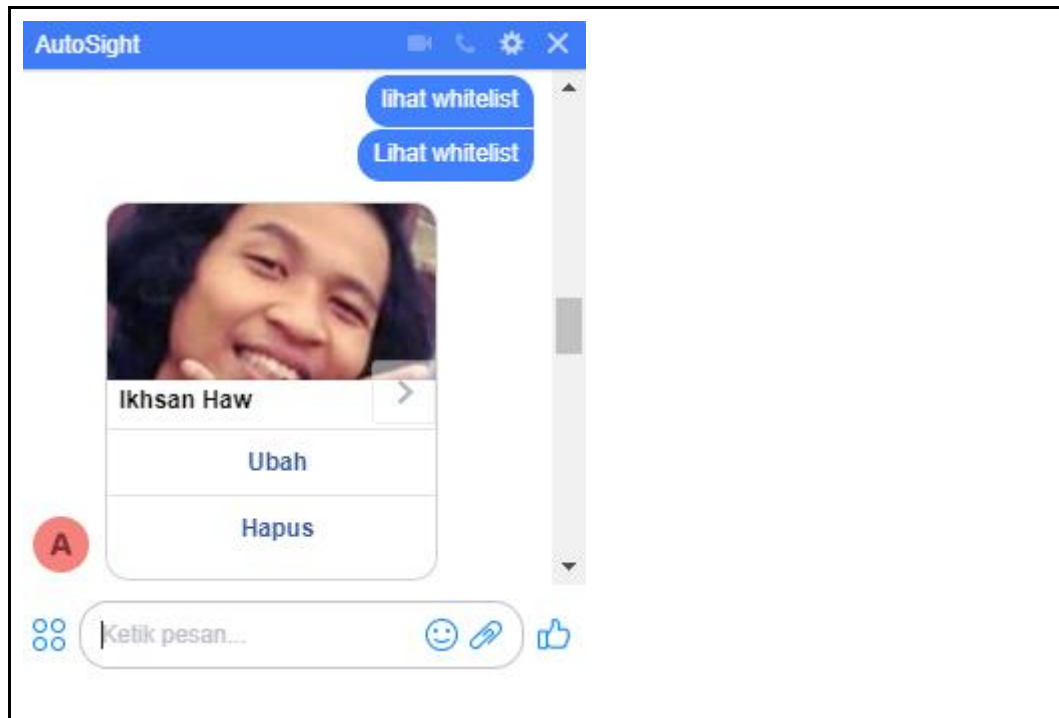
Tabel VI.7 Implementasi 006

No	Kode Requirement	Deskripsi	Class yang diimplementasi
11	REQ-F-U4-001	Sistem dapat menambahkan data baru orang yang dikenal ke dalam <i>database</i>	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
12	REQ-F-U4-002	Sistem dapat melakukan <i>encoding</i> terhadap wajah yang	

		dikirimkan oleh pemilik rumah
Hasil Implementasi		
Berikut adalah screenshot hasil implelementasi requirement tersebut. Report ini tampil di aplikasi Messenger yang berjalan di smartphone Android.		
		

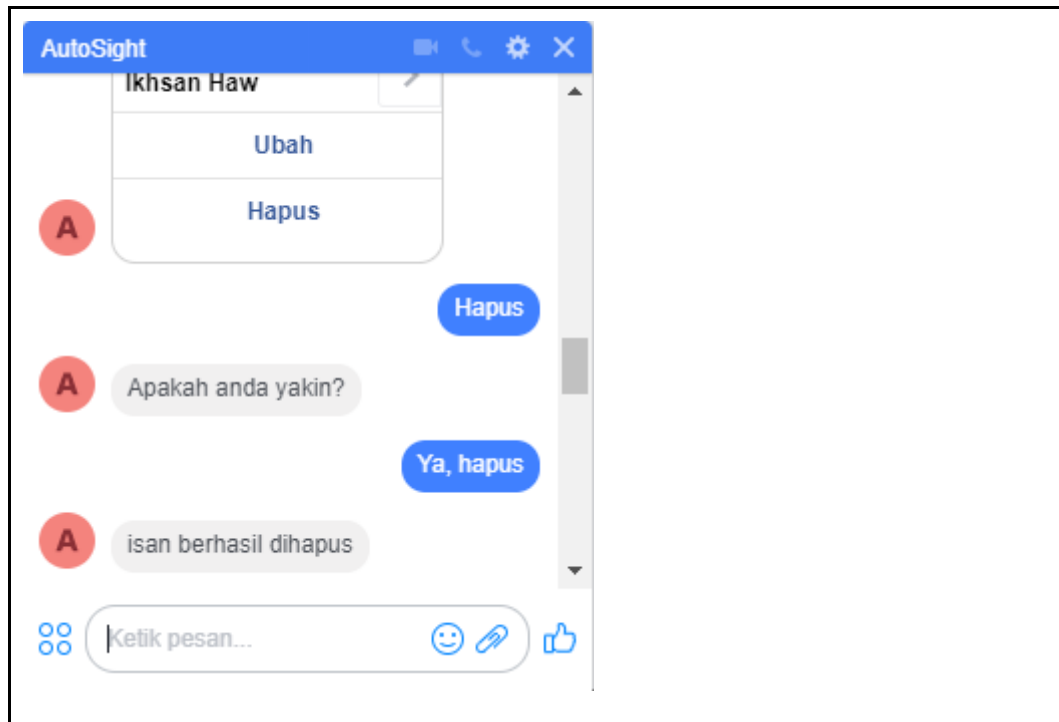
Tabel VI.8 Implementasi 007

No	Kode Requirement	Deskripsi	Class yang diimplementasi
13	REQ-F-U5-001	Sistem dapat menampilkan daftar orang yang dikenal sesuai dengan <i>database</i>	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
Hasil Implementasi			
Berikut adalah screenshot hasil implelementasi requirement tersebut. Report ini tampil di aplikasi Messenger yang berjalan di smartphone Android.			



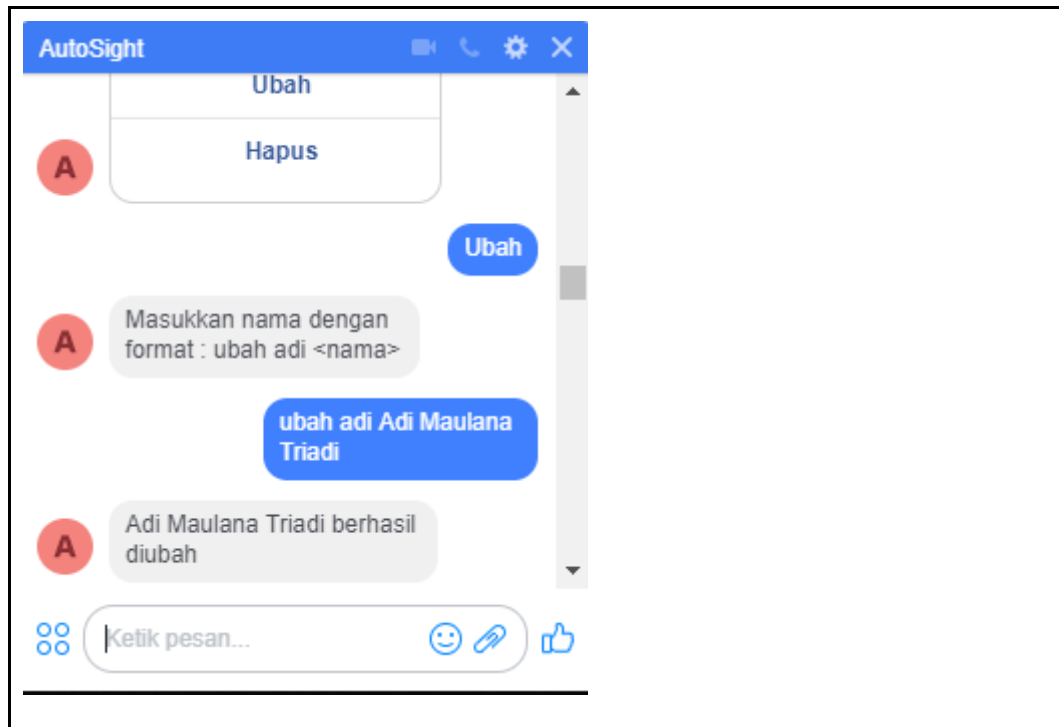
Tabel VI.9 Implementasi 008

No	Kode Requirement	Deskripsi	Class yang diimplementasi
14	REQ-F-U6-001	Sistem dapat menghapus orang yang dikenal yang dipilih oleh pemilik rumah	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
Hasil Implementasi			
Berikut adalah screenshot hasil implelementasi requirement tersebut. Report ini tampil di aplikasi Messenger yang berjalan di smartphone Android.			



Tabel VI.10 Implementasi 009

No	Kode Requirement	Deskripsi	Class yang diimplementasi
15	REQ-F-U7-001	Sistem dapat mengubah nama pada daftar orang yang dikenal	Bot_Util, Model_Face, Messenger_Platform_Controller, White_List_Manager
16	REQ-F-U7-002	Sistem dapat memperbaharui data pada <i>database</i> sesuai yang dipilih oleh pemilik rumah	
Hasil Implementasi			
Berikut adalah screenshot hasil impelementasi requirement tersebut. Report ini tampil di aplikasi Messenger yang berjalan di smartphone Android.			



BAB VII

PENGUJIAN

VII.1 Hasil Pengujian

Pengujian dilakukan untuk melihat sistem yang dibuat telah sesuai dengan kebutuhan yang telah didefinisikan sebelumnya pada dokumen SRS. Pengujian yang dilakukan hanyalah *unit testing* dengan pendekatan *black box testing*.

Rencana pengujian sebelumnya telah dituliskan pada dokumen *test plan*, *test design*, dan *test procedure*. Untuk hasil pengujian secara mendetail, dituliskan pada dokumen *test case*. Untuk melihat keterhubungan antara kebutuhan fungsional dengan *test case* yang dibuat, akan ditunjukkan pada Tabel VII.1.

Tabel VII.1 Hasil pengujian

Requirement	Test Scenario	Test Case ID	Test Case	Result
REQ-F-U1-002, REQ-F-U1-003 REQ-F-U1-004 REQ-F-U1-005, REQ-F-U1-006, REQ-F-U1-007	Pengujian pendeteksian wajah pada saat siang hari dengan citra wajah dengan input yang valid	S-01-001	Mendeteksi wajah dengan posisi wajah menghadap ke depan kamera pada siang hari. Wajah terdeteksi dan ditandai dengan adanya <i>bounding box</i> .	PASS
		S-01-002	Mendeteksi wajah dengan posisi tidak menghadap ke depan kamera / menyamping ketika terekam oleh kamera pada siang hari. Wajah tidak terdeteksi.	PASS
		S-01-003	Mendeteksi wajah dengan posisi wajah tanpa melihat ke kamera saat siang hari. Wajah terdeteksi dan memberi tanda berupa <i>bounding box</i> .	PASS
		S-01-004	Mendeteksi wajah yang berjumlah lebih dari satu saat terekam	PASS

			oleh kamera pada siang hari.	
		S-01-005	Mendeteksi wajah yang tertutup masker atau penutup lainnya saat terekam kamera pada siang hari	PASS
REQ-F-U1-002, REQ-F-U1-003 REQ-F-U1-004 REQ-F-U1-005, REQ-F-U1-006, REQ-F-U1-007	Pengujian maksimal pendeteksian pada siang hari jarak untuk wajah	S-02-001	Mendeteksi wajah pada jarak kurang dari satu meter dari kamera pada saat siang hari	PASS
		S-02-002	Mendeteksi wajah pada jarak 1 – 2 m dari kamera pada saat siang hari	PASS
		S-02-003	Mendeteksi wajah pada jarak berkisar 2 – 3m dari kamera pada saat siang hari	PASS
		S-02-004	Mendeteksi wajah pada jarak lebih dari 3m dari kamera pada saat siang hari	PASS
REQ-F-U2-001, REQ-F-U2-002, REQ-F-U2-003, REQ-F-U2-004, REQ-F-U2-005	Pengujian sistem saat mengenali wajah dengan citra wajah yang telah tersimpan pada <i>database</i>	S-03-001	Mengenali wajah terhadap orang yang telah dikenal atau telah tersimpan pada <i>database</i> dengan menghadap ke depan kamera	PASS
		S-03-002	Mengenali wajah terhadap orang yang telah dikenal atau tersimpan pada <i>database</i> dengan posisi wajah tidak melihat ke depan kamera	PASS
		S-03-003	Mengenali wajah dengan jumlah lebih dari satu terhadap orang yang telah dikenal dengan posisi menghadap ke depan terhadap kamera	PASS
		S-03-004	Mengenali wajah dengan jumlah lebih dari satu terhadap orang yang telah dikenal dengan posisi wajah tidak melihat ke depan kamera	PASS
		S-03-005	Mengenali wajah dengan input citra wajah yang berbeda dengan citra wajah	PASS

			yang disimpan pada <i>database</i> namun masih wajah orang yang sama (Pada <i>database</i> : citra wajah tanpa memakai kaca mata, yang terekam kamera dengan memakai kaca mata.)	
		S-03-006	Mengenali wajah yang belum terdaftar pada daftar orang dikenal	PASS
		S-03-007	Mengenali wajah dengan jumlah lebih dari satu orang dengan komposisi adanya wajah yang dikenal dan adanya wajah yang tidak dikenal masuk dan terekam oleh kamera secara bersamaan	PASS
REQ-F-U3-001, REQ-F-U3-002, REQ-F-U3-003, REQ-F-U3-004, REQ-F-U3-005	Pengujian pengiriman pemberitahuan saat melakukan pengenalan wajah pada citra wajah yang terekam oleh kamera	S-04-001	Pengujian pengiriman pemberitahuan saat pengenalan terhadap orang yang tidak ada dalam daftar orang yang dikenal atau <i>database</i>	PASS
		S-04-002	Pengujian pengiriman pemberitahuan saat melakukan pengenalan wajah terhadap orang yang ada dalam daftar orang dikenal	PASS
		S-04-003	Pengujian pengiriman pemberitahuan saat pengenalan terhadap orang yang dikenali dan tidak dikenali secara bersamaan	PASS
REQ-F-U5-001, REQ-F-U5-002	Pengujian pada chatbot saat mengakses fitur melihat daftar orang yang dikenal dengan inputan yang valid	S-05-001	Pengujian dengan input "Lihat whitelist". Menampilkan daftar orang yang dikenal sesuai yang tersimpan pada <i>database</i> .	PASS
		S-05-002	Pengujian dengan input "lihat whitelist". Menampilkan pesan error atau salah penulisan sintaks. Tidak memunculkan	PASS

			daftar orang yang dikenal.	
		S-05-003	Pengujian dengan input tidak sesuai sintaks. Menampilkan pesan error atau salah penulisan sintaks. Tidak memunculkan daftar orang yang dikenal.	PASS
REQ-F-U4-001, REQ-F-U4-002, REQ-F-U4-003, REQ-F-U4-004	Pengujian Menambah data orang yang dikenal dengan input nama yang valid	S-06-001	Menambahkan data orang yang dikenal dengan sintaks yang benar dan input nama secara lengkap dan valid, data tersimpan	PASS
		S-06-002	Menambahkan data orang yang dikenal dengan mengikuti sintaks yang benar namun input nama tidak valid, data tidak tersimpan	PASS
		S-06-003	Menambahkan data orang yang dikenal dengan sintaks yang benar namun tanpa memasukan nama, data tidak tersimpan	PASS
		S-06-004	Menambahkan data orang yang dikenal tanpa mengikuti sintaks, muncul pesan untuk menampilkan salah input	PASS
REQ-F-U7-001, REQ-F-U7-002, REQ-F-U7-003	Pengujian mengubah data nama orang yang dikenal dengan input valid	S-07-001	Mengubah data nama orang yang dikenal dengan nama baru yang valid, data diubah pada <i>database</i>	PASS
		S-07-002	Mengubah data nama orang yang dikenal dengan nama baru yang tidak valid, data tidak berubah	PASS
		S-07-003	Mengubah data nama orang yang dikenal dengan nama baru yang dikosongkan, data tidak berubah	PASS
REQ-F-U6-001, REQ-F-U6-002, REQ-F-U6-003, REQ-F-U6-004	Pengujian penghapusan data orang yang dikenal dari daftar orang yang dikenal	S-08-001	Memilih perintah hapus pada orang yang dipilih dari daftar orang yang dikenal, menampilkan pesan	PASS

			penghapusan berhasil.	
		S-08-002	Membatalkan perintah hapus pada orang yang dipilih dari daftar orang yang dikenal, chatbot menampilkan pesan batal menghapus.	PASS

Berdasarkan pengujian yang dilakukan sesuai dengan *test case*, sistem yang dibuat telah memenuhi *requirement* yang didefinisikan sebelumnya.

Fitur pendeteksian dan pengenalan wajah, sistem yang dibuat dapat melakukan pendeteksian dan pengenalan wajah apabila ada manusia yang terekam oleh kamera dengan jarak maksimal tiga meter, semakin dekat orang tersebut dengan kamera, maka hasil pengenalannya dapat lebih akurat, tapi pada jarak tiga meter pun sistem telah bisa mengenali sesuai dengan data orang yang dikenal dan menentukan apakah orang tersebut dikenal atau tidak. Sistem pun dapat mengenali wajah jika berjumlah lebih dari satu ketika terdeteksi.

Namun, jika adanya wajah yang menyamping ketika tertangkap kamera, maka terkadang wajah tersebut dapat dideteksi ataupun tidak, tergantung apakah bagian penting pada wajahnya terlihat atau tidak.

BAB VIII

KESIMPULAN DAN SARAN

VIII.1 Kesimpulan

Pencapaian yang berhasil didapatkan dalam pengerjaan pengembangan “Sistem Pendeteksi Orang tidak Dikenal Menggunakan Metode Pengenalan Wajah pada Kamera Pengawas” ini telah dapat mengimplementasikan beberapa fitur, yaitu:

1. Sistem sudah mampu untuk mengakuisisi citra dari kamera pengawas
2. Sistem sudah mampu mendeteksi wajah manusia dari citra yang telah diakuisisi
3. Sistem sudah mampu untuk melakukan pengenalan wajah
4. Sistem sudah mampu mengirimkan sebuah pemberitahuan ketika mendeteksi ada orang yang tidak dikenal
5. Sistem mampu melakukan manajemen data melalui chatbot.

Kesimpulan yang didapatkan adalah sistem kamera pengawas dapat diimplementasikan menggunakan metode pengenalan wajah dengan menggunakan *deep learning*. Namun, implementasi yang dilakukan hanyalah sebatas menggunakan model yang telah disediakan, bukan melatih model dari awal. Selain itu, resolusi yang dapat digunakan dengan *resource* yang terbatas ini mencapai 480p. Resolusi akan semakin bagus jika perangkat keras yang digunakan mempunyai spesifikasi yang lebih besar. Dengan *resource* yang digunakan saat ini, proses pengenalan wajah dapat dilakukan hingga mencapai jarak maksimal sejauh 3m.

VIII.2 Saran

Berikut adalah saran-saran mengenai pembuatan sistem pendeteksi orang tidak dikenal:

1. Pembuatan kamera pengawas dengan menggunakan pengenalan wajah membutuhkan *resource* perangkat keras yang tidak kecil untuk memberikan performa yang memuaskan. Sebaiknya gunakan perangkat untuk mengolah proses dengan spesifikasi komputer yang mencukupi, atau mengubah arsitektur pembuatan sistem menjadi lebih terdistribusi dalam setiap *service* yang akan

dibuat. Contohnya menggunakan *stream* kamera dan mengirimkan hasil *stream* kamera ke dalam *server* dan dilanjutkan oleh *server* untuk mengolah proses pengenalan wajah dan *server* memberikan *output* proses berupa informasi melalui media chatbot.

2. Penggunaan algoritma *deep learning* yang digunakan untuk membuat model yg digunakan untuk proses pengenalan wajah dapat mencapai akurasi yang tinggi dibandingkan menggunakan metode pengolahan citra saja seperti contohnya PCA/LDA.

Banyak hal-hal yang dapat dijadikan media untuk mengimplementasikan metode pengenalan wajah, mulai dari lingkungan pribadi sampai di ranah umum. Metode pengenalan wajah pada kamera pengawas diharapkan bisa menjadi salah satu solusi untuk pembuatan sistem pengawasan rumah yang sedang banyak dibuat.

DAFTAR PUSTAKA

- [1] S. Rusell dan P. Norvig, "Artificial Intelligence: A Modern Approach Third Edition," dalam *Artificial Intelligence: A Modern Approach Third Edition*, 2010, pp. 942-957.
- [2] P. Hidayatullah, "Pengolahan Citra Digital: Teori dan Aplikasi Nyata," dalam *Pengolahan Citra Digital: Teori dan Aplikasi Nyata*, Bandung, Informatika, 2017, pp. 1-9.
- [3] N. Dalal dan B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005.
- [4] D. King, "High Quality Face Recognition with Deep Metric Learning," 2017. [Online]. Available: <http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html>.
- [5] R. Messiahdas, N. Gupta dan D. D. Sarkar, "Power through AI and Automation with Chatbots," 2017.
- [6] K. K. Patel dan S. M. Patel, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *IJESC Volume 6 Issues No. 5*, 2016.
- [7] C. Ferrari, G. Lisanti, S. Berretti dan A. D. Bimbo, "Investigating Nuisance Factors in Face Recognition with DCNN Representation," 2017.
- [8] V. Kazemi dan J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," 2014.
- [9] "Software Engineering: A Practitioner's Approach, 7th Edition," dalam *Software Engineering: A Practitioner's Approach, 7th Edition*, 2010, p. 42.
- [10] R. N dan D. M. Z. Kurian, "Face Detection in Real Time Based on HOG," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 3, no. 4, 2014.
- [11] C. Kaundanya, O. Pathak, A. Nalawade dan S. Parode, "Smart Surveillance System using Raspberry Pi and Face Recognition," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 4, 2017.
- [12] R. Kar dan R. Haldar, "Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements," 2016.
- [13] A. Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning," 24 July 2016. [Online]. Available:

<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>.

- [14] L. Deng dan D. Yu, “Deep Learning: Methods and Applications,” *Foundations and Trends® in Signal Processing: Vol. 7: No. 3–4*, pp. 197-387, 2014.
- [15] A. A. Al-Bakeri dan A. A. Basuhail, “Notification System Based on Face Detection and Face Recognition: A Novel Approach,” *International Journal of Computer Science and Information Security (IJCSIS)*, 2016.

LAMPIRAN A-1
SOFTWARE REQUIREMENT SPECIFICATION

LAMPIRAN A-2

TEST PLAN

LAMPIRAN A-3
TEST DESIGN SPECIFICATION

LAMPIRAN A-4
TEST CASE SPECIFICATION

LAMPIRAN A-5
TEST PROCEDURE SPECIFICATION