



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**PELACAKAN DAN PENGENALAN BANYAK WAJAH
MENGUNAKAN *FEATURE* HASIL
PEMBELAJARAN *CONVOLUTIONAL NEURAL
NETWORK* DAN *KLASIFIKASI SUPPORT VECTOR
MACHINE***

KUKUH RILO PAMBUDI

05111440000178

Dosen Pembimbing

Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018



TUGAS AKHIR - KI141502

**PELACAkan DAN PENGENALAN BANYAK WAJAH
MENGUNAKAN *FEATURE* HASIL
PEMBELAJARAN *CONVOLUTIONAL NEURAL
NETWORK* DAN KLASIFIKASI *SUPPORT VECTOR
MACHINE***

**KUKUH RILO PAMBUDI
05111440000178**

Dosen Pembimbing I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Dosen Pembimbing II
-

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2018**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**MULTIPLE FACE TRACKING AND RECOGNITION
USING CONVOLUTIONAL NEURAL NETWORK
LEARNING FEATURES AND SUPPORT VECTOR
MACHINE CLASSIFICATION**

**KUKUH RILO PAMBUDI
0511144000178**

**Supervisor I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Supervisor II
-**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND COMMUNICATION
Institut Teknologi Sepuluh Nopember
Surabaya, 2018**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**PELACAKAN DAN PENGENALAN BANYAK WAJAH
MENGUNAKAN *FEATURE* HASIL PEMBELAJARAN
CONVOLUTIONAL NEURAL NETWORK DAN
*SUPPORT VECTOR MACHINE***

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

KUKUH RILO PAMBUDI

NRP: 05111440000178

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Eng. Nanik Suciati, S.Kom. M.Kom.
NIP. 197104281994122001



(Pembimbing 1)

**SURABAYA
JULI, 2018**

[Halaman ini sengaja dikosongkan]

**PELACAKAN DAN PENGENALAN BANYAK WAJAH
MENGUNAKAN *FEATURE* HASIL PEMBELAJARAN
CONVOLUTIONAL NEURAL NETWORK DAN
*KLASIFIKASI SUPPORT VECTOR MACHINE***

Nama Mahasiswa : Kukuh Rilo Pambudi
NRP : 05111440000178
Jurusan : Informatika, FTIK ITS
Dosen Pembimbing 1 : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
Dosen Pembimbing 2 : -

Abstrak

Sistem pengenalan wajah adalah suatu sistem yang berguna untuk mengenali wajah seseorang melalui citra wajah. Sementara itu, video merupakan suatu multimedia yang menggabungkan citra secara berurutan sehingga membentuk citra yang seolah-olah bergerak. Sistem pengenalan wajah berbasis video merupakan sistem pengenalan wajah menggunakan citra-citra berurutan yang terdapat pada video.

Tugas akhir ini mengembangkan sistem untuk mengenali seseorang secara bersamaan pada suatu citra pada video. Data video yang digunakan merupakan hasil rekaman dari kamera yang kemudian akan diproses dan diberikan hasil pengenalannya. Sistem ini terdiri dari tiga tahapan utama, yaitu (1) tahap deteksi wajah, (2) tahap ekstraksi fitur, dan (3) tahap pengenalan dan pelacakan banyak wajah. Tahap deteksi wajah bertujuan untuk mendapatkan wajah-wajah yang terdapat pada suatu video menggunakan algoritma SeetaFace. Tahap ekstraksi fitur bertujuan memperoleh fitur yang dapat mewakili citra wajah menggunakan Visual Geometry Group - Face (VGG-Face). Tahap pengenalan dan pelacakan adalah tahap pengambilan kesimpulan pada wajah menggunakan Support Vector Machine (SVM), Multiple Face Recognition (MFR), Multiple Face Tracking (MFT).

Uji coba dilakukan pada 3 video. Tiap video terdiri dari banyak wajah yang berbeda-beda jumlahnya. Data video tersebut memiliki jumlah wajah 4, 6, dan 8. Uji coba pengenalan dan pelacakan banyak wajah pada masing-masing data video menghasilkan akurasi terbaik 97.82%, 95.61%, dan 99.01%.

Kata kunci: deteksi wajah, pengenalan banyak wajah, pelacakan banyak wajah, Seeta Face Detection, VGG Face, Support Vector Machine.

MULTIPLE FACE TRACKING AND RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK LEARNING FEATURES AND SUPPORT VECTOR MACHINE CLASSIFICATION

Student Name : Kukuh Rilo Pambudi
Registration Number : 05111440000178
Department : Informatics, FTIK ITS
First Supervisor : Dr. Eng. Nanik Suciati, S.Kom.,M.Kom
Second Supervisor : -

Abstract

Face recognition system is a system for the recognize of a person's face through face image. Meanwhile, the video is multimedia that is used to generate the imagery that is moving. A video-based face recognition system is a face recognition system using the consecutive images on a video.

This final project develops a system for recognizing a person simultaneously on an image on a video. Video data used is the recording of the camera which will then be processed and given the results of the recognition. The system consists of three main stages: (1) the phase of face detection, (2) the feature extraction stage, and (3) the recognition and tracking stage of many faces. Face detection stage aims to get the faces contained in a video using SeetaFace algorithm. The feature extraction stage aims to obtain features that can represent facial images using Visual Geometry Group - Face (VGG-Face). The recognition and tracking stage is the conclusion step on the face using Support Vector Machine (SVM), Multiple Face Recognition (MFR), and Multiple Face Tracking (MFT).

Testing using 3 videos. Each video consists of many different faces. The video data has a face count of 4, 6, and 8. Best Accuracy of multiple face recognition and tracking trials on each of the video data is 97.82%, 95.61%, and 99.01%.

Keywords: face detection, multiple face recognition, multiple face tracking, Seeta Face Detection, VGG face, Support Vector Machine.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Pengenalan dan Pelacakan Banyak Wajah Menggunakan *Feature Hasil Pembelajaran Convolutional Neural Network dan Klasifikasi Support Vector Machine***”.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian deteksi wajah dan pengenalan.

Dalam perancangan, pengerjaan dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Dr.Eng. Nanik Suciati, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Diana Purwitasari, S.Kom., M.Sc. yang telah memberi banyak masukan dan nasihat pada penulis dalam mengerjakan tugas akhir.
3. Orang tua penulis Bapak Wardaya dan Ibu Eny Budiarti yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
4. Kakak kandung (Yudho) dan seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.
5. Teman – teman seperjuangan Antonius Kevin, Jeffry Nasri, Luqman Ahmad dan Hendri Febriansyah yang senantiasa berbagi pengalaman hidup, suka dan duka selama masa perkuliahan di Teknik Informatika, ITS.

6. Teman-teman Admin Laboratorium KCV dan seluruh Sahabat KCV yang telah banyak berbagi pengetahuan, pengalaman, dan inspirasi.
7. Teman – teman Aku Pintar (Luqman, Kevin, Mas Divi, Mba Ekky, Mba Rere, Mas Iyus, Mas Gilang, Mas Pebri, Mas Ali) yang senantiasa menemani, memberi dukungan, semangat dan inspirasi selama pengerjaan Tugas Akhir ini.
8. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi	3
1.7 Sistematika Laporan.....	4
BAB II DASAR TEORI	7
2.1 SeetaFace Detection	7
2.2 VGG-Face	8
2.3 <i>Support Vector Machine (SVM)</i>	12
2.4 <i>Multiple Face Recognition (MFR)</i>	13
2.5 <i>Multiple Face Tracking (MFT)</i>	14
2.6 <i>Kombinasi Multiple Face Tracking dan Multiple Face Recognition</i>	16
BAB III PERANCANGAN	17
3.1 Perancangan Sistem	17
3.2 Perancangan Data.....	19
3.3 Perancangan Proses.....	21
3.3.1 Deteksi Wajah	21
3.3.2 <i>Feature Extraction</i>	23
3.3.3 Pengenalan dan Pelacakan Banyak Wajah	25
BAB IV IMPLEMENTASI	29

4.1	Lingkungan Implementasi.....	29
4.1.1	Perangkat Keras	29
4.1.2	Perangkat Lunak	29
4.2	Implementasi Deteksi Wajah	29
4.3	Implementasi <i>Feature Extraction</i>	30
4.4	Implementasi <i>Multiple Face Recognition</i> dan <i>Multiple Face Tracking</i>	31
BAB V UJI COBA DAN EVALUASI.....		35
5.1	Lingkungan Uji Coba.....	35
5.2	Data Uji Coba.....	35
5.3	Skenario Uji Coba	36
5.4	Uji Coba Jumlah Kelas Wajah pada Pengenalan dan Pelacakan Banyak Wajah	36
5.5	Uji Coba Layer VGG-Face pada Pengenalan dan Pelacakan Banyak Wajah	37
5.6	Evaluasi Uji Coba Jumlah Kelas Wajah pada Pengenalan dan Pelacakan Banyak Wajah.....	39
5.7	Evaluasi Uji Coba Jumlah Layer VGG-Face pada Pengenalan dan Pelacakan Banyak Wajah	40
BAB VI KESIMPULAN DAN SARAN		40
6.1	Kesimpulan	43
6.2	Saran.....	43
LAMPIRAN		45
DAFTAR PUSTAKA		55
BIODATA PENULIS		57

DAFTAR GAMBAR

Gambar 2.1 Funnel-Structured Cascade <i>Framework</i> [2]	8
Gambar 2.2 Wajah Tokoh Masyarakat [3]	9
Gambar 2.3 Tahapan Proses SVM [5].....	12
Gambar 3.1 Diagram Alir dari Sistem Pengenalan dan Pelacakan Banyak Wajah	18
Gambar 3.2 Diagram Alir Deteksi Wajah	21
Gambar 3.3 Wajah Tidak Terdeteksi	22
Gambar 3.4 Wajah Terdeteksi Semua	23
Gambar 3.5 Diagram Alir <i>Feature Extraction</i> VGG-Face.....	25
Gambar 3.6 Proses <i>Multiple Face Tracking</i>	27
Gambar 3.7 Wajah Dikenali.....	27
Gambar 3.8 Diagram Alir Pelacakan dan Pengenalan Banyak Wajah	28
Gambar 5.1 Wajah Salah Pengenalan	41

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Data Wajah Berskala Besar [3]	9
Tabel 2.2 Arsitektur Layer VGG 16 [3]	10
Tabel 3.1 Video Data Latih	20
Tabel 3.2 Video Data Uji	20
Tabel 3.3 Hasil Deteksi Wajah pada Tiap Video Uji	23
Tabel 3.4 Jumlah <i>Feature</i> Layer VGG-Face	24
Tabel 5.1 Detail Data Uji Coba Video	35
Tabel 5.2 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 4	36
Tabel 5.3 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 6	37
Tabel 5.4 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 8	37
Tabel 5.5 Hasil Uji Coba Layer VGG-Face pada Video Wajah 4	38
Tabel 5.6 Hasil Uji Coba Layer VGG-Face pada Video Wajah 6	38
Tabel 5.7 Hasil Uji Coba Layer VGG-Face pada Video Wajah 8	39

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Tahap Deteksi Wajah.....	30
Kode Sumber 4.2 Implementasi Tahap <i>Feature Extraction</i>	31
Kode Sumber 4.3 Implementasi Tahap MFR dan MFT.....	33

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Sebagai topik penelitian yang telah berjalan lama di computer vision, pengenalan wajah telah dipelajari secara intensif dan berbagai metode telah diusulkan. Pengenalan wajah adalah metode untuk mengidentifikasi seseorang dengan menggunakan wajah orang tersebut. Pengembangan sistem pengenalan wajah masih menjadi tantangan besar di dunia computer vision. Hal ini disebabkan wajah akan mempunyai banyak variasi bila direpresentasikan dalam citra. Tingkat pencahayaan yang berbeda akan menyebabkan perbedaan nilai pada citra wajah yang akan meningkatkan variasi dari suatu wajah. Selain tingkat pencahayaan, ekspresi dari wajah juga akan mempengaruhi komposisi data pada citra. Sebagai contoh, wajah seseorang yang sedang tertawa akan memiliki representasi citra yang berbeda bila dibandingkan dengan wajah seseorang yang sedang sedih. Posisi wajah juga mempengaruhi variasi dari wajah seseorang, sebagai contohnya wajah menengok ke kanan, wajah menengok ke kiri, wajah menunduk, dan wajah mengadiah akan menghasilkan representasi citra yang berbeda.

Aplikasi sistem pengenalan wajah dapat memberikan berbagai manfaat bagi kehidupan manusia. Wajah dapat digunakan sebagai *biometric authentication*. *Biometric authentication* adalah proses keamanan yang bergantung pada karakteristik unik biologis suatu individu [1]. Wajah merupakan salah satu karakteristik unik yang dimiliki manusia yang dapat membedakan antara orang satu dengan orang yang lainnya. Oleh karena itu, sistem pengenalan wajah memungkinkan untuk diterapkan di berbagai bidang yang

membutuhkan pengenalan individu. Sebagai contohnya, sistem pengenalan wajah dapat digunakan untuk mengawasi suatu lokasi dengan bantuan kamera pengawas. Sistem akan mengenali setiap orang yang ada pada lokasi tersebut dan menyimpan data tersebut, data yang dapat berguna di kemudian hari.

Pengenalan wajah yang ada saat ini kebanyakan hanya bisa mengenali satu orang saja, sedangkan pada dunia nyata citra ataupun video yang ada kebanyakan berisi lebih dari satu orang. Maka dari itu, diperlukan pengenalan wajah yang dapat mengenali banyak wajah sekaligus melalui citra ataupun video.

Dalam tugas akhir ini, akan dikembangkan sistem pengenalan banyak wajah pada video. Sistem ini dapat dimanfaatkan untuk berbagai keperluan, sebagai contohnya mencatat orang-orang yang sedang ada pada suatu ruangan. Sistem ini dalam melakukan deteksi wajah akan menggunakan algoritma SeetaFace untuk mendapatkan wajah-wajah pada suatu *frame*. Pada *feature extraction*, sistem ini akan menggunakan model Convolutional Neural Network (CNN) yang telah dilatih dengan data wajah bernama VGG-Face. Proses klasifikasi akan menggunakan algoritma Support Vector Machine (SVM) yang dilatih dengan cara one-vs-rest, keluaran dari SVM tiap wajah pada *frame* akan diproses secara bersama agar mendapatkan akurasi yang tinggi dibandingkan dengan pengenalan wajah secara sendiri-sendiri.. Diharapkan sistem yang dibangun dapat melakukan klasifikasi wajah dengan akurasi yang tinggi.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mendeteksi banyak wajah menggunakan pendeteksi wajah SeetaFace pada suatu citra wajah?
2. Bagaimana melakukan *feature extraction* menggunakan VGG-Face?
3. Bagaimana melakukan pelacakan dan pengenalan banyak wajah menggunakan *Support Vector Machine*?

4. Bagaimana melakukan uji performa perangkat lunak yang dibangun?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Implementasi sistem menggunakan Python 3.6.
2. Data uji dan latih adalah video RGB yang direkam menggunakan kamera *handphone* Samsung S9+ dengan resolusi panjang 1920 piksel dan lebar 1080 piksel.

1.4 Tujuan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini adalah membangun sistem untuk melakukan pengenalan dan pelacakan banyak wajah pada video.

1.5 Manfaat Tugas Akhir

Manfaat yang diperoleh dari pembuatan tugas akhir ini adalah :

1. Menghasilkan sistem yang dapat mengenali banyak wajah pada video.
2. Menghasilkan laporan sistem pengenalan banyak wajah yang dapat digunakan sebagai referensi studi pengenalan banyak wajah selanjutnya.
3. Sistem pengenalan banyak wajah yang dihasilkan dapat digunakan untuk mengenali orang-orang yang sedang ada pada suatu ruangan.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada studi literatur, dilakukan pengumpulan data dan studi terhadap sejumlah referensi yang diperlukan dalam pengerjaan tugas akhir. Referensi tersebut didapatkan dari beberapa artikel yang dipublikasikan oleh jurnal. Selain dari artikel, studi literatur juga dilakukan melalui pencarian referensi dari internet yang membahas mengenai informasi yang dibutuhkan.

2. Analisis dan Desain Perangkat Lunak

Pada tahap ini dilakukan analisis dan perancangan perangkat lunak. Tahap analisis merupakan penentuan kebutuhan dan ruang lingkup sistem yang akan dibangun. Tahap perancangan merupakan pendefinisian sistem, penentuan data yang digunakan dan proses – proses yang dilakukan sistem.

3. Implementasi Perangkat Lunak

Sistem pelacakan dan pengenalan banyak wajah akan dibuat dengan bahasa pemrograman Python menggunakan kakas bantu IDE Pycharm 2017.3.1 (*Community Edition*) pada platform *desktop*.

4. Uji Coba dan Evaluasi

Dalam tahap ini, dilakukan pengujian parameter-parameter yang dibutuhkan pada proses pengenalan. Hasil pengenalan dari banyak wajah dibandingkan dengan *ground truth* berupa wajah-wajah terpotong yang telah diberi label secara manual untuk dihitung performanya dengan menghitung nilai *accuracy*.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian sebagai berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan dan manfaat dari pembuatan tugas akhir. Selain itu

permasalahan, batasan masalah, metodologi yang digunakan dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

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

Bab III Perancangan

Bab ini berisi tentang perancangan desain sistem deteksi penyakit diabetes makula edema.

Bab IV Implementasi

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

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

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

[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

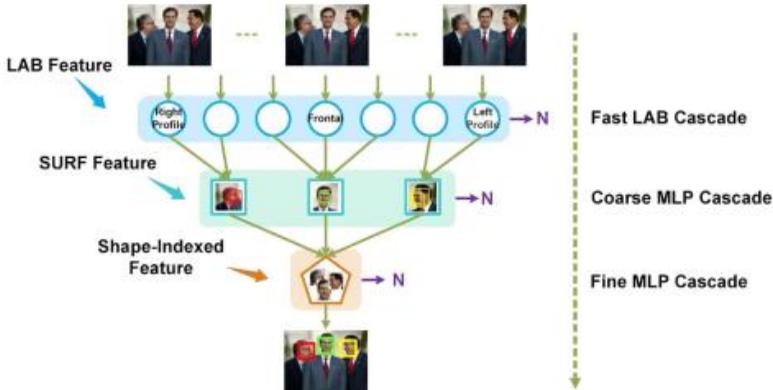
Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan.

2.1 SeetaFace Detection

SeetaFace Detection adalah algoritman yang dirancang untuk mendeteksi wajah dalam berbagai sudut pandang secara cepat dan akurat pada suatu citra [2]. SeetaFace menggunakan sebuah *funnel-structured (FuSt) multi-view face detector* untuk melakukan proses deteksi. Pada **Gambar 2.1**, Fust Detector memiliki struktur yang lebih luas pada bagian atas dan semakin sempit pada bagian bawah. Struktur ini berguna untuk mengurangi kandidat *window* yang bukan wajah. Pada proses awal, beberapa *classifier* bekerja secara parallel untuk mengurangi kandidat *window* yang bukan wajah. *Classifier* dapat bekerja sangat cepat namun masih secara kasar, yang berarti masih banyak kandidat *window* yang bukan wajah masih lolos dari proses ini. Pada tahap berikutnya, pemilihan wajah akan semakin ketat dengan menaikkan waktu proses dan kemampuan diskriminatif pada deteksi wajah untuk memverifikasikan kandidat *window* yang tersisa sehingga akan mengurangi jumlah dari *window* yang bukan wajah pada tahap sebelumnya. Dengan mengumpulkan kandidat *window* yang bertahan dari proses-proses sebelumnya, sebuah *unified multilayer perceptron (MLP) cascade* diterapkan sehingga dapat memberikan hasil deteksi wajah yang mampu mendeteksi wajah dari berbagai sudut pandang.

FuSt Detector terdiri dari tiga tahap, yakni Fast LAB Cascade Classifier, Coarse MLP Cascade Classifier, dan Fine MLP Cascade Classifier. Seiring berjalannya waktu tahap akan semakin tinggi akurasi dan tingkat komputasinya. Pada tahap awal, citra input akan di-*scan* dengan *window* yang bergeser melewati

keseluruhan citra input, kemudian setiap window akan masuk pada tahap demi tahap. Fast LAB Cascade bertujuan menghilangkan window yang bukan wajah secara cepat. Coarse MLP Cascade Classifier menyeleksi lagi kandidat window dengan kompleksitas yang tidak terlalu tinggi. Pada tahap akhir, Unified Fine MLP Cascade Classifier menentukan wajah secara akurat.



Gambar 2.1 Funnel-Structured Cascade *Framework* [2]

2.2 VGG-Face

Convolutional Neural Networks (CNNs) telah merubah komunitas terkait visi computer karena *CNN* dapat menghasilkan akurasi yang tinggi pada berbagai aplikasi bidang citra. Salah satu syarat penting *CNN* adalah data latih yang besar.

Pada dunia pengenalan wajah, data berskala besar terkait wajah yang ada sekarang dikembangkan oleh perusahaan-perusahaan besar seperti *Google* dan *Facebook* sehingga data wajah tersebut digunakan untuk kepentingan pribadi perusahaan, sebagai contohnya adalah metode pengenalan wajah yang dibuat oleh *Google* dilatih dengan 200 juta citra dengan delapan juta identitas [3], data latih ini sangat lebih besar dari segala dataset wajah yang diterbitkan publik seperti pada **Tabel 2.1**.

Tabel 2.1 Data Wajah Berskala Besar [3]

Dataset	Jumlah Identitas	Jumlah Gambar
LFW	5749	13233
WDRef	2995	99773
CelebFaces	10177	202599
Ours	2622	2.6 Juta
FaceBook	4030	4.4 Juta
Google	8 Juta	200 Juta

VGG-Face merupakan *Convolutional Neural Network* yang dilatih dengan data wajah berskala besar berjumlah 2.6 juta dengan 2622 identitas. Data wajah ini terdiri dari wajah-wajah tokoh masyarakat yang didapat dengan berdasarkan jenis kelamin dan peringkat popularitas dari *Internet Movie Data Base (IMDB)*.



Gambar 2.2 Wajah Tokoh Masyarakat [3]

VGG-Face memiliki arsitektur berbasis VGG16 terdiri dari 11 blok. Lima blok pertama merupakan convolutional layer, sedangkan tiga blok sisanya adalah Fully Connected (FC) layer. Detail arsitektur VGG-Face dapat dilihat di **Tabel 2.2**.

Tabel 2.2 Arsitektur Layer VGG 16 [3]

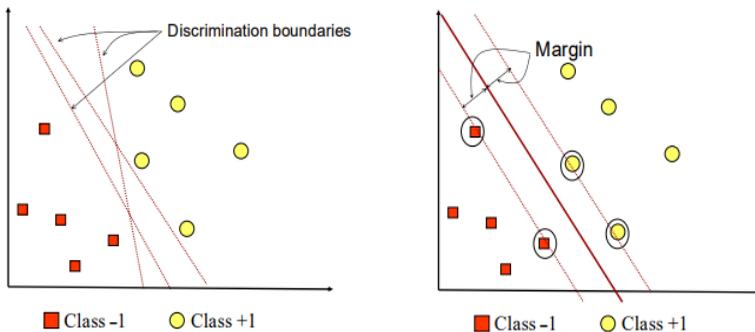
layer	0	1	2	3	4
type	input	conv	relu	conv	relu
name	-	conv1_1	relu1_1	conv1_2	pool1
support	-	3	1	3	1
filt dim	-	3	-	64	-
num filts	-	64	-	64	-
stride	-	1	1	1	1
pad	-	1	0	1	0
layer	5	6	7	8	9
type	mpool	conv	relu	conv	relu
name	pool1	conv2_1	relu2_1	conv2_2	relu2_2
support	2	3	1	3	1
filt dim	-	64	-	128	-
num filts	-	128	-	128	-
stride	2	1	1	1	1
pad	0	1	0	1	0
layer	10	11	12	13	14
type	mpool	conv	relu	conv	relu
name	pool2	conv3_1	relu3_1	conv3_2	relu3_2
support	2	3	1	3	1
filt dim	-	128	-	256	-
num filts	-	256	-	256	-
stride	2	1	1	1	1
pad	0	1	0	1	0
layer	15	16	17	18	19
type	conv	relu	mpool	conv	relu

name	conv3_3	relu3_3	pool3	conv4_1	relu4_1
support	3	1	2	3	1
filt dim	256	-	-	256	-
num filts	256	-	-	512	-
stride	1	1	2	1	1
pad	1	0	0	1	0
layer	20	21	22	23	24
type	conv	relu	conv	relu	mpool
name	conv4_2	relu4_2	conv4_3	relu4_3	pool4
support	3	1	3	1	2
filt dim	512	-	512	-	-
num filts	512	-	512	-	-
stride	1	1	1	1	2
pad	1	0	1	0	0
layer	25	26	27	28	29
type	conv	relu	conv	relu	conv
name	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3
support	3	1	3	1	3
filt dim	512	-	512	-	512
num filts	512	-	512	-	512
stride	1	1	1	1	1
pad	1	0	1	0	1
layer	30	31	32	33	34
type	relu	mpool	conv	relu	conv
name	relu5_3	pool5	fc6	relu6	fc7
support	1	2	7	1	1
filt dim	-	-	512	-	4096

num filts	-	-	4096	-	4096
stride	1	2	1	1	1
pad	0	0	1	0	0
layer	35	36	37		
type	relu	conv	softmax		
name	relu7	fc8	prob		
support	1	1	1		
filt dim	-	4096	-		
num filts	-	2622	-		
stride	1	1	1		
pad	0	0	0		

2.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperlane* terbaik yang memisahkan dua buah *class* pada *input space* [5].



Gambar 2.3 Tahapan Proses SVM [5]

Gambar 2.2a menunjukkan sekumpulan data yang terdiri dari dua *class* yaitu *class* : -1 dan +1. *Class* -1 direpresentasikan dengan kotak berwarna merah, sedangkan *class* +1 direpresentasikan dengan lingkaran berwarna kuning. Problem klasifikasi yang harus dilakukan adalah dengan menemukan *hyperplane* yang memisahkan antara dua *class* tersebut.

Hyperplane terbaik yang memisahkan kedua kelas dihitung dengan mengukur *margin hyperplane*, dan mencari titik maksimal. *Margin* adalah jarak terdekat *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *class*. Garis solid yang terdapat pada gambar 2b adalah *hyperplane* terbaik, yaitu yang terletak tepat ditengah-tengah kedua *class*. Sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*.

2.4 Multiple Face Recognition (MFR)

Banyak wajah pada video akan dikenali secara bersama-sama dengan harapan pengenalan dari banyak wajah akan lebih akurat daripada pengenalan secara sendiri-sendiri, dengan memanfaatkan informasi *temporal* dan *spatial* yang terdapat pada video [4].

Anggap $O^{(t)}$ sebagai kumpulan dari wajah yang terdeteksi pada *frame* t , dan $o_i^{(t)} \in \mathbb{R}^d$ merupakan ke- i dari $O^{(t)}$. Anggap D adalah *database* dari wajah, dan ukurannya dinotasikan $|D|$, yang berukuran tetap untuk semua *frame* dari video. Kecuali, menambahkan “*unknown*” wajah pada $|D|$. $|D|$ *face classifier* dilatih secara *offline*, dan *classifier* ke- j yang dinotasikan f_j adalah untuk mengenali orang ke- j pada database D . Anggap $B^{(t)}$ adalah *binary matrix* dari kemungkinan dari wajah terdeteksi di $O^{(t)}$ ke D . Sehingga, $b_{i,j}^{(t)} = 1$ jika wajah $o_i^{(t)}$ terdeteksi sebagai D_j , dan sebaliknya $b_{i,j}^{(t)} = 0$. Kemudian ditambahkan *matching cost matrix* $C^{(t)}$ untuk mengevaluasi rasionalitas dari deteksi, dimana $c_{i,j}^{(t)} \geq 0$ adalah usaha dari deteksi dari $o_i^{(t)}$ ke D_j . Nilai dari $c_{i,j}^{(t)}$ akan kecil bila wajah yang terdeteksi $o_i^{(t)}$ kemungkinan besar adalah D_j pada *database*. Karenanya *matching cost* dapat didefinisikan sebagai rumus (2.1)

$$c_{i,j}^{(t)} = -\ln f_j(o_i^{(t)}) \quad (2.1)$$

dimana $f_j(o_i^{(t)})$ adalah *confidence score* dari *classifier* ke- j dari wajah ke- j yang terdeteksi. *MFR* kemudian dapat dihitung dengan menggunakan formula (2.2)

$$\min J_1 = \sum_{i=1}^{|O^{(t)}|} \sum_{j=1}^{|D|} b_{i,j}^{(t)} c_{i,j}^{(t)} = \text{tr}(C^{(t)T} B^{(t)}) \quad (2.2)$$

dimana $B^{(t)} \in B^{(|O^{(t)}| \times |D|)}$, $B \in \{0,1\}$.

Pada prakteknya, wajah yang terdeteksi mungkin tidak dideteksi ke lebih dari satu subyek di D . *Identity exclusivity* dapat dilakukan dengan membuat $B^{(t)}$ menjadi rumus (2.3)

$$\sum_{j=0}^{|D|} b_{i,j}^{(t)} = 1 \quad (2.3)$$

Di sisi lain, *identity exclusivity* juga mengharuskan subyek yang terdaftar di *database* tidak boleh muncul di wajah yang terdeteksi lebih dari satu kali. Sehingga jumlah dari kolom pada $B^{(t)}$ tidak boleh lebih besar dari 1, kecuali untuk kolom terakhir yang mewakili "*unknown face*" yang dapat dideteksi lebih dari satu kali, maka $B^{(t)}$ menjadi seperti rumus (2.4)

$$B^{(t)T} \mathbf{1} \leq \begin{pmatrix} 1 \\ N_u \end{pmatrix} \quad (2.4)$$

dimana N_u adalah nilai maksimal dari wajah *unknown* pada *frame* video, dan $\mathbf{1}$ adalah vector kolom yang memiliki nilai semuanya 1.

2.5 Multiple Face Tracking (MFT)

Pada prakteknya pengenalan wajah kemungkinan akan bekerja secara tidak akurat dengan kondisi pengambilan video yang jelek termasuk banyaknya pose pada subyek [4]. Pada

skenario ini, MFT dapat mengatasi masalah tersebut dengan menggunakan *temporal consistency* pada seluruh *frame* video. *Temporal consistency* tidak hanya menghitung *appearance inconsistency* tetapi juga *motion smoothness* dari seluruh *frame* yang berurutan. MFT dapat dihitung dengan menggunakan formula (2.5) untuk memecahkan *binary linking matrix* $L^{(t, t-1)}$.

$$\begin{aligned} \min J_2 &= \sum_{q=1}^{|o^{(t)}|} \sum_{p=1}^{|o^{(t-1)}|} l_{q,p}^{(t, t-1)} w_{q,p}^{(t, t-1)} \\ &= \text{tr}(W^{(t, t-1)} L^{(t, t-1)}) \end{aligned} \quad (2.5)$$

dimana setiap nilai *Boolean* $l_{q,p}^{(t, t-1)}$ dari $L^{(t, t-1)}$ bernilai 1 jika wajah yang terdeteksi ke- p pada *frame* $t - 1$ berhubungan dengan wajah ke- q pada *frame* t . dan sebaliknya. $w_{q,p}^{(t, t-1)}$ dari matriks bobot $W^{(t, t-1)}$ adalah biaya hubungan antara wajah terdeteksi ke- p pada *frame* $t - 1$ dengan wajah ke- q pada *frame* t , yang dihitung berdasarkan *motion smoothness* dan *appearance inconsistency* dengan rumus (2.6).

$$\begin{aligned} w_{q,p}^{(t, t-1)} &= (d_M(o_q^t, o_p^{t-1}) \\ &\quad + \beta d_A(o_q^t, o_p^{t-1})) d_I(b_q^{(t)}, b_p^{(t-1)}) \end{aligned} \quad (2.6)$$

dimana $d_M(., .)$ menghitung *motion smoothness* antara wajah terdeteksi pada setiap *frame* yang berurutan $d_A(., .)$ menghitung *appearance consistency* dengan matriks jarak, $d_I(., .)$ menghitung *Hamming distance* antar dua vektor identitas dan β adalah parameter real positif. Vektor identitas $b_q^{(t)}$ dan $b_p^{(t-1)}$ adalah baris vektor ke- q dan ke- p dari matriks *binary* $B^{(t)}$ dan $B^{(t-1)}$. Parameter dari pelacakan pergerakan wajah berupa lokasi 2D *spatial*, lebar dan panjang dari wajah yang terdeteksi.

2.6 Kombinasi *Multiple Face Tracking* dan *Multiple Face Recognition*

Penggabungan MFR dan MFT untuk memecahkan $B^{(t)}$ dan $L^{(t,t-1)}$ dapat diformulasikan menjadi rumus (2.7).

$$\min_{B^{(t)}, L^{(t,t-1)}} \text{tr}(C^{(t)T} B^{(t)}) + \xi \cdot \text{tr}(W^{(t,t-1)T} L^{(t,t-1)}) \quad (2.7)$$

untuk $\xi > 0$. Seperti yang diobservasi pada rumus (2.6), matriks bobot $W^{(t,t-1)}$ sendiri adalah fungsi dari $B^{(t)}$ dan $B^{(t-1)}$, karenanya rumus (7) akan dikalikan dengan dia sendiri dengan sendirinya, yang dapat disederhanakan menjadi alternatif optimasi proses:

i) Dengan $L^{(t,t-1)}$ bernilai tetap, B^t menjadi rumus (2.8)

$$\min_{B^t} \text{tr}((C^t + \xi \hat{c}^{(t,t-1)})^T B^{(t)}) \quad (2.8)$$

Dimana nilai $\hat{c}_{j,k}^{(t,t-1)}$ dari matriks $\hat{c}^{(t,t-1)}$ didefinisikan dengan rumus (9):

$$\begin{aligned} \hat{c}_{q,k}^{(t,t-1)} = & \sum_{j=1}^{|O^{(t-1)}|} (1 - 2b_{j,k}^{(t-1)}) l_{q,j}^{(t,t-1)} \\ & \times (d_M(o_q^{(t)}, o_j^{(t-1)}) \\ & + \beta d_A(o_q^{(t)}, o_j^{(t-1)})) \end{aligned} \quad (2.9)$$

ii) Dengan $B^{(t)}$ tetap, $L^{(t,t-1)}$ dapat diselesaikan dengan rumus (2.5). Memberikan solusi MFR dan MFT pada *frame* video sebelumnya, algoritma yang diajukan secara terus menerus akan melakukan optimisasi antara pelacakan dan pengenalan (sampai *convergence* dicapai) untuk mendapatkan solusi optimal MFR dari *frame* sekarang.

BAB III PERANCANGAN

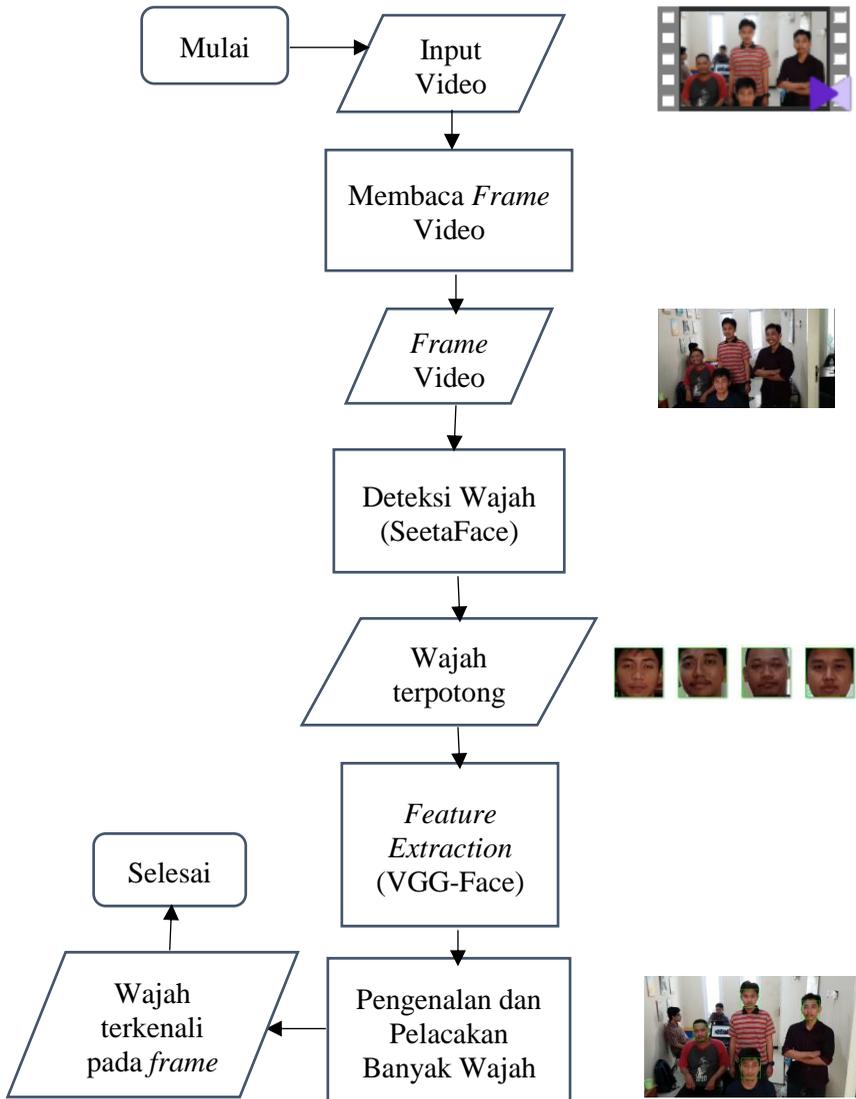
Pada bab ini diuraikan mengenai perancangan aplikasi agar dapat mencapai tujuan dari tugas akhir. Perancangan yang dibuat meliputi perancangan sistem, data dan proses.

3.1 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses pengenalan dan pelacakan banyak wajah secara keseluruhan. Agar dapat menggunakan program pengenalan dan pelacakan banyak wajah, pengguna harus memiliki video yang berisi banyak wajah sebagai data masukan. Data masukan diperoleh dari pengambilan video banyak wajah menggunakan kamera.

Sistem pengenalan wajah pada tugas akhir ini memiliki tiga proses utama. Proses pertama adalah deteksi wajah pada tiap *frame* video agar mendapatkan wajah-wajah yang ada pada *frame*. Proses kedua adalah *feature extraction* dari tiap wajah dalam *frame* sehingga mendapatkan nilai-nilai yang dapat mewakili dari suatu wajah. Proses terakhir adalah pelacakan dan pengenalan pengenalan banyak wajah agar mendapatkan wajah-wajah yang ada pada *frame*. Diagram alir dari sistem ditunjukkan pada **Gambar 3.1**.

Proses deteksi wajah adalah proses untuk menemukan wajah-wajah yang ada pada tiap *frame* pada video. Proses ini meliputi merubah *frame* pada video dari RGB (*Red, Green, Blue*) menjadi *grayscale*, mendeteksi wajah pada *frame grayscale*, pemberian indeks pada tiap wajah, mendapatkan titik-titik koordinat dari wajah pada *frame*, memotong wajah berdasarkan titik-titik koordinat dari *frame* dan merubah ukuran tiap wajah menjadi satu ukuran citra agar dapat diproses pada tahap berikutnya. Pendeteksi wajah yang digunakan pada sistem pengenalan dan pelacakan banyak wajah adalah pendeteksi wajah SeetaFace.



Gambar 3.1 Diagram Alir dari Sistem Pengenalan dan Pelacakan Banyak Wajah

Proses *feature extraction* adalah proses untuk mendapatkan nilai-nilai yang dapat mewakili suatu wajah. *Feature extraction* dilakukan dengan menggunakan VGG-Face, yaitu *convolutional neural networks (CNN)* yang telah dilatih menggunakan wajah-wajah dari 2622 orang dengan 375 wajah tiap orang. *Feature Ekstraksi* yang dilakukan diambil dari *convolution layer*.

Proses pelacakan dan pengenalan wajah adalah proses untuk mendapatkan identitas dari tiap wajah-wajah yang terdapat pada *frame*. Pelacakan wajah akan membandingkan wajah-wajah pada *frame* yang sedang diproses dengan wajah-wajah pada *frame* sebelumnya untuk mencari wajah terdekat dengan menghitung *motion smoothness*, *appearance consistency*, dan *Hamming distance*. Pengenalan wajah-wajah akan dilakukan secara bersama-sama menggunakan fungsi objektif dan algoritma *Support Vector Machine (SVM)* dengan cara pelatihan *classifier one vs rest*.

3.2 Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data video yang digunakan pada tugas akhir akan dibagi menjadi dua jenis yaitu video data latih dan video data uji. Video latih dan video uji diambil dengan menggunakan kamera *smartphone* Samsung S9+ dengan menggunakan resolusi 1920x1080 piksel dan diambil secara *landscape*.

Video latih merupakan video yang akan digunakan untuk melatih *classifier support vector machine* pada sistem pengenalan dan pelacakan banyak wajah. Video uji berjumlah 12 video yang masing-masing video hanya terdapat satu orang dan terdiri dari identitas yang berbeda-beda. Detail dari data video latih ditunjukkan dengan **Tabel 3.1**. Video latih diambil dengan cara menggerakkan kamera mengelilingi wajah dari orang untuk mendapatkan variasi wajah yang besar. Hal ini dilakukan dikarenakan dalam pengenalan wajah diperlukan variasi wajah yang besar tiap identitas agar mendapatkan akurasi yang tinggi pada saat pengenalan wajah.

Tabel 3.1 Video Data Latih

No.	Video	Durasi	Frame per Second (fps)
1	Mas Gilang	27 detik	60
2	Luqman	32 detik	60
3	Mas Pebri	31 detik	60
4	Mas Ius	36 detik	60
5	Jeffry	34 detik	60
6	Mas Divi	35 detik	60
7	Mas Badai	33 detik	60
8	Mas Ali	39 detik	60
9	Galang	31 detik	30
10	Hendri	30 detik	30
11	Mbak Ekky	30 detik	30
12	Kukuh	32 detik	30

Video uji adalah video yang digunakan untuk menguji akurasi dari sistem pengenalan dan pelacakan banyak wajah. Video uji berjumlah 3 video dengan masing-masing video berisi 4, 6, dan 8 orang. Detail dari data video uji ditunjukkan dengan **Tabel 3.2**. Video data uji diambil dengan cara menggerakkan kamera selama perekaman yang bertujuan untuk mendapatkan variasi wajah yang banyak tiap orang. Kamera yang bergerak juga dimaksudkan untuk menguji sistem pengenalan banyak wajah dalam hal mengenali variasi wajah-wajah.

Tabel 3.2 Video Data Uji

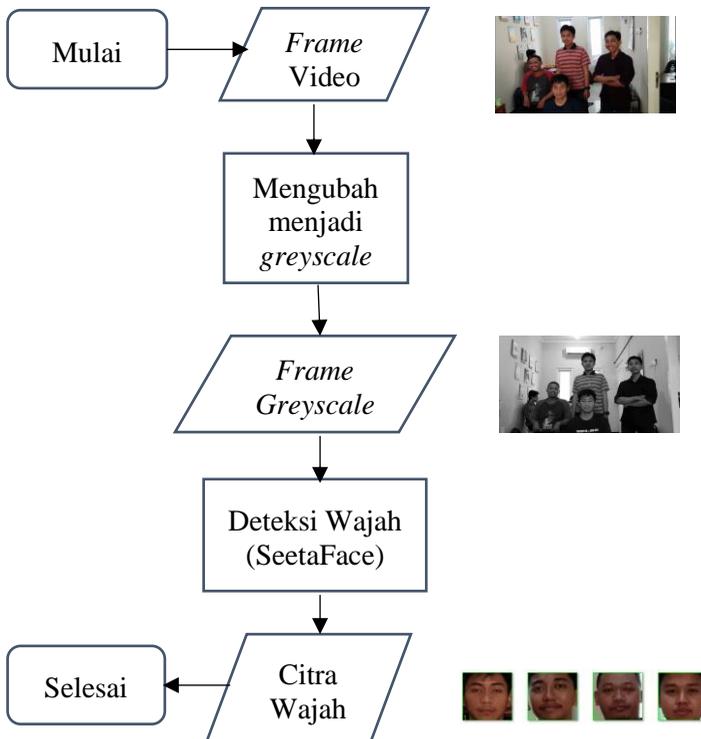
No.	Jumlah Orang	Durasi	Frame per Second (fps)	Frame skip	Jumlah frame
1	4	37 detik	60	5	376
2	6	43 detik	60	5	437
3	8	42 detik	60	5	425

3.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada sistem pengenalan dan pelacakan banyak wajah. Bagian dari setiap proses utama sistem dapat dilihat pada **Gambar 3.1**

3.3.1 Deteksi Wajah

Deteksi wajah yang dilakukan bertujuan untuk mendapatkan wajah-wajah dari tiap *frame* dalam video. Sebelum dilakukan proses deteksi wajah dilakukan pembacaan *frame* pada video. *Frame* video kemudian akan dirubah menjadi *greyscale* terlebih



Gambar 3.2 Diagram Alir Deteksi Wajah

dahulu. Selanjutnya, algoritma SeetaFace diterapkan pada *frame* tersebut. Algoritma SeetaFace menghasilkan wajah-wajah yang telah dipotong dari *frame*. Diagram alir dari proses ditunjukkan pada **Gambar 3.2**.

Deteksi wajah dilakukan pada tiap video uji, jumlah seluruh wajah yang terdeteksi tiap *frame* pada video akan digunakan sebagai penentuan akurasi pengenalan wajah. Pendeteksi wajah SeetaFace akan mendeteksi wajah dengan besar panjang dan lebar wajah minimal 180. Bila terdapat wajah dengan besar dibawah 180 maka pendeteksi wajah SeetaFace tidak akan mendeteksi wajah tersebut sebagai wajah. Tiap video uji akan memiliki jumlah wajah yang terdeteksi yang berbeda-beda tergantung dengan kondisi video. Kondisi video yang dapat mempengaruhi pendeteksi wajah SeetaFace adalah besarnya wajah-wajah yang terdapat dalam video tiap *frame* dan kualitas dari video. Wajah-wajah yang terdeteksi akan diberi kotak warna hijau disekitarnya dan bila tidak terdeteksi maka disekitar wajah tidak akan diberi apa-apa seperti pada **Gambar 3.3** dan **Gambar 3.4**. Jumlah seluruh wajah yang terdeteksi pada tiap video uji dijelaskan pada **Tabel 3.3**. Pada saat



Gambar 3.3 Wajah Tidak Terdeteksi

deteksi wajah dilakukan *frame skip* sebesar 5 *frame* bertujuan untuk mengurangi banyaknya wajah yang akan diproses tanpa mengurangi variasi wajah yang terdeteksi pada tiap video uji.



Gambar 3.4 Wajah Terdeteksi Semua

Tabel 3.3 Hasil Deteksi Wajah pada Tiap Video Uji

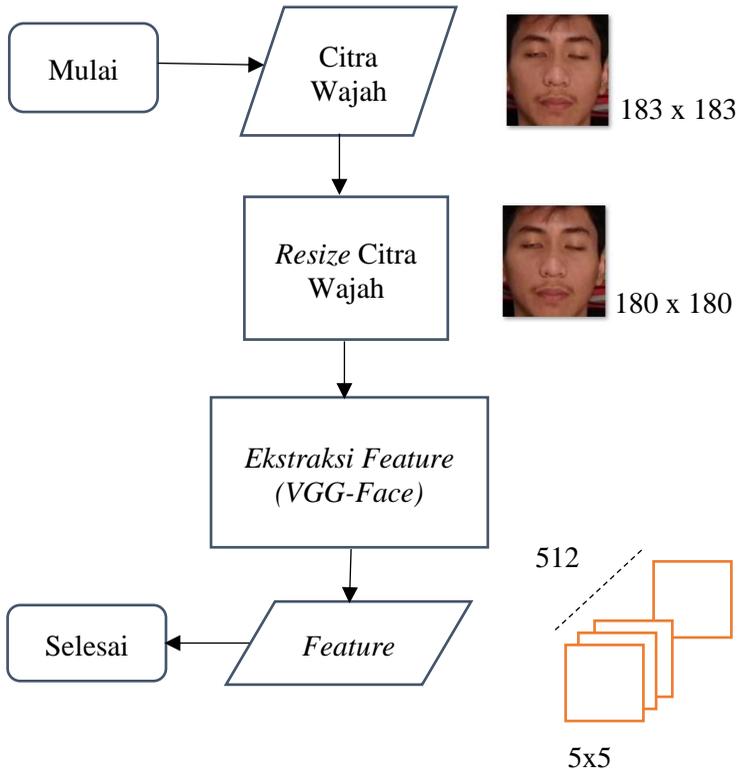
Video Uji	Jumlah Seluruh Wajah	Jumlah Wajah Terdeteksi	<i>Frame skip</i>	Akurasi
Jumlah Orang 4	1504	1327	5	88.23%
Jumlah Orang 6	2622	2050	5	78.19%
Jumlah Orang 8	3400	3145	5	92.5%

3.3.2 Feature Extraction

Feature extraction dimaksudkan untuk menentukan nilai-nilai yang dapat mewakili suatu wajah. *Feature extraction* yang dilakukan menggunakan lima *convolution layer* dari VGG-Face. Citra wajah yang akan di dapatkan *feature* nya *resize* terlebih dahulu agar menjadi satu ukuran yaitu 180x180 piksel, hal ini dilakukan agar wajah-wajah yang akan diproses dapat memiliki besar *feature* yang sama. Besarnya *feature* keluaran akan bergantung pada *layer* VGG-Face yang digunakan dan besar dari ukuran wajah. VGG-Face menggunakan arsitektur *Convolutional Neural Networks (CNN)* VGG16 yang terdiri dari 8 *layer* dengan 5 *convolutional layer* untuk *feature extraction* yaitu “pool1”, “pool2”, “pool3”, “pool4”, “pool5” dan 3 *layer* terakhir berupa *fully connected layer* yang digunakan untuk proses klasifikasi yaitu “fc6”, “fc7”, dan “fc8”. *Layer* VGG-Face yang digunakan sebagai *feature extraction* pada tugas akhir ini terdiri dari “pool1”, “pool2”, “pool3”, “pool4”, dan “pool5”. Besar dari *feature* masing-masing ditunjukkan pada tabel **Tabel 3.4**. Diagram alir dari proses ditunjukkan pada **Gambar 3.5**.

Tabel 3.4 Jumlah Feature Layer VGG-Face

No	Layer	Besar Feature
1	pool1	90x90x64
2	pool2	45x45x128
3	pool3	22x22x256
4	pool4	11x11x512
5	pool5	5x5x512



Gambar 3.5 Diagram Alir *Feature Extraction* VGG-Face

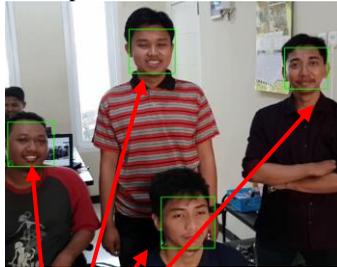
3.3.3 Pengenalan dan Pelacakan Banyak Wajah

Pengenalan dan pelacakan dilakukan setelah mendapatkan *feature* dari masing—masing wajah yang selanjutnya diproses oleh *classifier support vector machine* untuk mendapatkan *cost* penempatan pada *classifier* dari tiap wajah. *Classifier support vector machine* dilatih dengan cara *one vs rest*. *Classifier support vector machine* hasil latih menggunakan *one vs rest* akan berjumlah sebanyak n kelas data latih wajah yang digunakan untuk

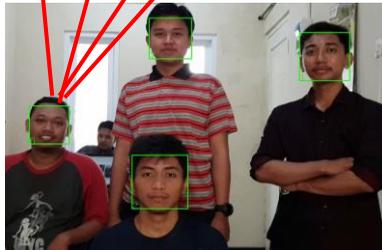
melatih *classifier support vector machine* berupa *classifier* ke-*i* sampai *classifier* ke-*n*. Tiap *classifier support vector machine* akan dilatih dengan hanya terdiri dari dua kelas yaitu benar dan salah. Kelas benar adalah wajah yang merupakan kelas ke-*i* dari data latih, sedangkan kelas yang salah adalah semua kelas wajah yang bukan dari kelas ke-*i*. Hasil dari *classifier support vector machine* sebenarnya sudah dapat digunakan untuk mengenali wajah namun agar hasil pengenalan wajah semakin baik tiap wajah harus diproses secara bersama-sama, sehingga hasil dari *classifier support vector machine* tidak digunakan untuk menentukan pengenalan wajah namun masih harus diproses kembali secara bersama dengan fungsi objektif *multiple face recognition* dan *multiple face tracking*.

Multiple face recognition akan mencari solusi optimal dari tiap *cost classifier* dengan penambahan beberapa *constraint* yaitu wajah terdeteksi tidak boleh muncul lebih dari satu kali pada tiap *frame* dan wajah terdeteksi juga tidak boleh memiliki lebih dari satu identitas. Penggunaan fungsi objektif *multiple face recognition* dapat membantu mengenali wajah pada tiap *frame* menjadi lebih baik dengan menambahkan *identity exclusivity* yaitu wajah yang dikenali pada suatu *frame* tidak boleh memiliki kelas yang sama dengan wajah yang lain pada *frame* yang sama. *Multiple face tracking* mencari wajah yang terdekat atau mirip dari wajah yang terdeteksi pada *frame* yang sedang diproses dengan wajah-wajah yang terdeteksi pada *frame* sebelumnya dengan menghitung *motion smoothness*, *appearance consistency*, dan *hamming distance* seperti pada **Gambar 3.6**. *Motion smoothness* dihitung dengan dengan membandingkan besar wajah pada *frame* sekarang dengan besar wajah-wajah pada *frame* sebelumnya, *appearance consistency* dihitung menggunakan *euclidean distance*. Hasil dari *multiple face recognition* dan *multiple face tracking* bersama-sama digunakan untuk pengenalan wajah. Wajah yang dikenali pada *frame* ditandai dengan anotasi berupa nomor berwarna hijau yang terdapat pada bagian kiri bawah dari wajah seperti pada **Error!**

Reference source not found.. Diagram alir dari proses ditunjukkan pada **Gambar 3.8**.

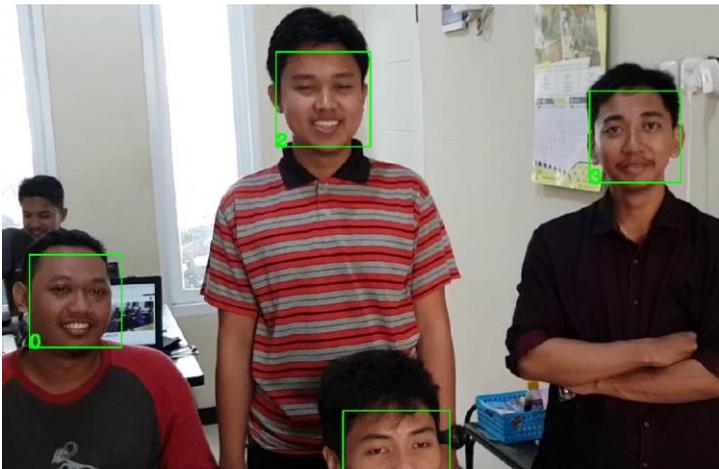


Frame t-1
(sebelum)

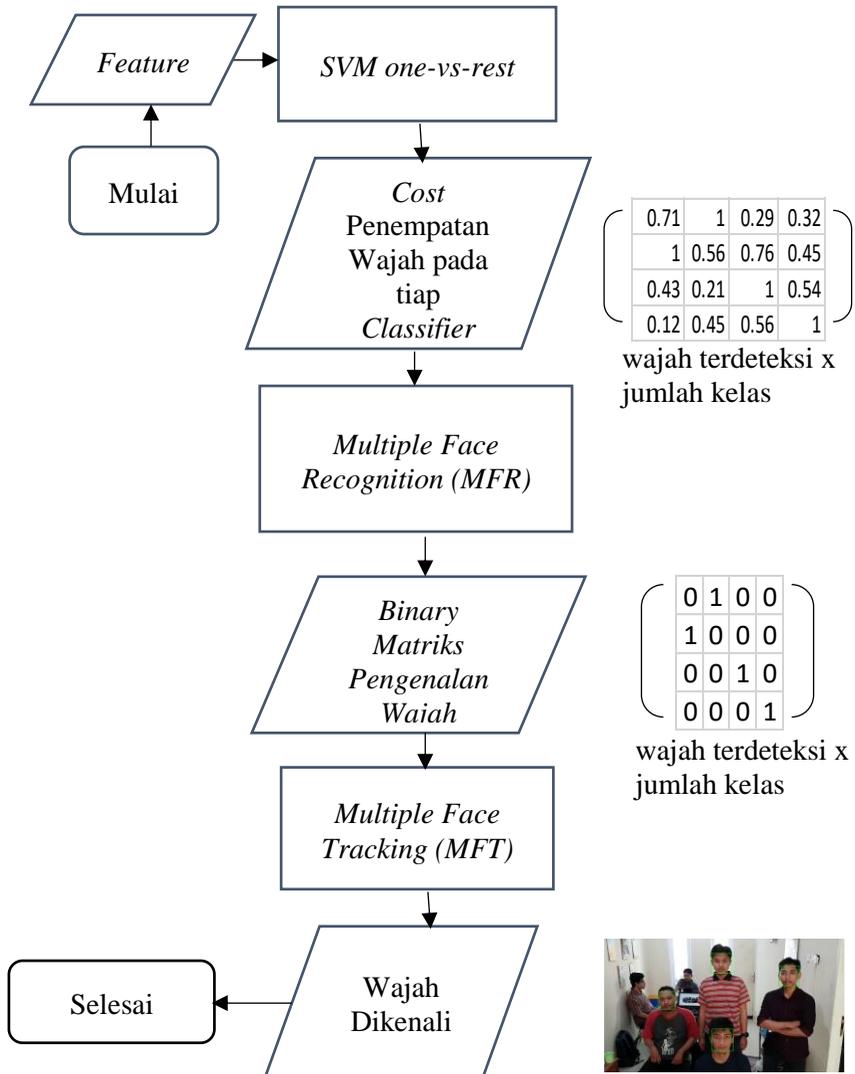


Frame t
(sekarang)

Gambar 3.6 Proses *Multiple Face Tracking*



Gambar 3.7 Wajah Dikenali



Gambar 3.8 Diagram Alir Pelacakan dan Pengenalan Banyak Wajah

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi sistem dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam aplikasi. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Objek video yang akan diolah pada implementasi tugas akhir ini adalah video dengan banyak wajah. Objek video adalah video tipe *Moving Picture Experts Group – 4 Part 14* (mp4).

Dalam implementasi algoritma pengolahan video tersebut, digunakan perangkat-perangkat sebagai berikut:

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah laptop bertipe MSI GS60 6QE.

Spesifikasi dari PC yang digunakan pada tugas akhir ini adalah memiliki prosesor Intel Core i7 6700HQ dengan kecepatan 2,6 GHz dan *Random Access Memory* (RAM) untuk proses menjalankan program sebesar 24,00 GB.

4.1.2 Perangkat Lunak

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat lunak menggunakan *software* PyCharm 2017.3.1. Penggunaan PyCharm didukung dengan *library* yaitu *pyseeta*, *sklearn*, *keras*, *keras_ygg*, *scipy*, *pulp*, *opencv*, *numpy*. Dokumentasi hasil uji coba dilakukan dengan menggunakan *Microsoft Excel*.

4.2 Implementasi Deteksi Wajah

Tahap deteksi wajah pada tugas akhir ini terdiri dari tiga tahap yaitu melakukan *resize* pada *frame* masukan, mengubah menjadi citra *greyscale* dan melakukan deteksi wajah

menggunakan SeetaFace. Implementasi deteksi wajah ditunjukkan pada **Kode Sumber 4.1**.

Sebelum proses deteksi wajah, pendeteksi wajah harus di buat terlebih dahulu pada baris 1. Pada baris 2 pendeteksi wajah ditentukan besar minimal wajah untuk dapat terdeteksi. Pada baris 3 mengubah *frame* menjadi *frame greyscale* dilakukan dengan mengubah *frame* berwarna RGB menjadi citra *grayscale*. Pada baris 4 melakukan iterasi tiap wajah pada satu *frame*. Pada baris 5 memotong *frame* untuk mendapatkan citra wajah. Pada baris 6 wajah yang telah terpotong akan dirubah ukurannya agar menjadi satu ukuran.

1	<code>detector = Detector()</code>
2	<code>detector.set_min_face_size(input_size)</code>
3	<code>image_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)</code>
4	<code>for i, face in enumerate(faces):</code>
5	<code>crop_img = frame[face.top: face.bottom, face.left: face.right]</code>
6	<code>resized_image = cv2.resize(crop_img, (input_size, input_size))</code>

Kode Sumber 4.1 Implementasi Tahap Deteksi Wajah

4.3 Implementasi *Feature Extraction*

Tahap *feature extraction* digunakan untuk mendapatkan nilai-nilai yang dapat mewakili suatu wajah. Tahap ini berguna untuk mereduksi besar input wajah agar mengurangi kompleksitas proses dari sistem secara keseluruhan.

Implementasi *feature extraction* ditunjukkan pada **Kode Sumber 4.2**. Citra wajah yang telah terpotong dirubah menjadi *array* pada baris 1. *Array* wajah terpotong ditambah dimensinya pada baris 2. Pada tahap 3 implementasi *feature extraction* dari *array* wajah terpotong menggunakan VGG-Face. Pada tahap 4 mengubah *array* dari hasil *feature extraction* menjadi satu dimensi.

1	<code>x = image.img_to_array(img)</code>
---	--

2	<code>x = np.expand_dims(x, axis=0)</code>
3	<code>vgg_features = VGGFace(include_top=False, input_shape=(input_size, input_size, 3), pooling='max')</code>
4	<code>return features[0].ravel()</code>

Kode Sumber 4.2 Implementasi Tahap *Feature Extraction*

4.4 Implementasi *Multiple Face Recognition* dan *Multiple Face Tracking*

Multiple Face Recognition dan *Multiple Face Tracking* berguna untuk mengambil keputusan dalam pengenalan wajah pada tiap *frame*. Implementasi *multiple face recognition* dan *multiple face tracking* ditunjukkan pada **Kode Sumber 4.3**. Pertama dilakukan proses training *classifier*. Data latih diambil dari video wajah tiap kelas yang telah direkam sebelumnya. Pada proses selanjutnya dilakukan proses pengujian menggunakan data video dari banyak wajah. Pada proses terakhir dilakukan pengambilan keputusan menggunakan *multiple face recognition* dan *multiple face tracking* dengan parameter penentu yaitu *cost* dari penempatan wajah-wajah pada tiap *frame* pada *classifier* dengan *support vector machine*.

Pada baris 1 dilakukan inisialisasi yang nantinya digunakan untuk memecahkan fungsi objektif. Pada baris 5 sampai 10 dilakukan pembuatan *variable* untuk menyimpan nilai MFR dan penambahan *constraint*. Pada baris 10 – 15 dilakukan pembuatan *variable* untuk MFT. Pada baris 17 dideklarasikan masalah fungsi objektif dari MFR dan MFT. Baris 18 sampai 21 penambahan *constraint* untuk MFR yaitu wajah terdeteksi tidak boleh muncul lebih dari satu kali pada tiap *frame* dan wajah terdeteksi juga tidak boleh memiliki lebih dari satu identitas wajah. Baris 30 – 40 digunakan untuk mendapatkan *class* dari hasil fungsi objektif.

1	<code>mfr_mft_problem = pulp.LpProblem("MFR MFT Problem", pulp.LpMinimize)</code>
2	<code>RANGE = range(len(costs_t))</code>

3	<code>RANGE2 = range(len(linkingCosts_t))</code>
4	<code>mfr_b = []</code>
5	<code>for i in RANGE:</code>
6	<code> mfr_b.append(pulp.LpVariable.dicts("mfr_b" + str(i), range(D_len), cat="Integer"))</code>
7	<code> for j in mfr_b[i].keys():</code>
8	<code> mfr_b[i][j].lowBound = 0</code>
9	<code> mfr_b[i][j].upBound = 1</code>
10	<code>mft_l = []</code>
11	<code>for i in RANGE2:</code>
12	<code> mft_l.append(pulp.LpVariable.dicts("mft_l" + str(i), range(len(linkingCosts_t[i])), cat="Integer"))</code>
13	<code> for j in mft_l[i].keys():</code>
14	<code> mft_l[i][j].lowBound = 0</code>
15	<code> mft_l[i][j].upBound = 1</code>
16	<code><i>pulp.LpSolverDefault.msg=1</i></code>
17	<code>mfr_mft_problem += pulp.lpSum([mfr_b[i][j] * costs_t[i][j] for i in RANGE for j in range(D_len)]) + 0.06 * pulp.lpSum([mft_l[k][l] * LinkingCost()[k][l] for k in RANGE2 for l in range(len(linkingCosts_t[k]))])</code>
18	<code>for i in RANGE:</code>
19	<code> mfr_mft_problem += pulp.lpSum([mfr_b[i][j] for j in range(D_len)]) == 1</code>
20	<code>for i in range(D_len):</code>
21	<code> mfr_mft_problem += pulp.lpSum([mfr_b[j][i] for j in RANGE]) <= 1</code>

22	<code>print(mfr_mft_problem)</code>
23	<code>mfr_mft_problem.solve()</code>
24	<code>count = 0</code>
25	<code>count_all = 0</code>
26	<code>B_mfr = []</code>
27	<code>b_mfr = []</code>
28	<code>clas = []</code>
29	<code>for variable in mfr_mft_problem.variables():</code>
30	<code> b_mfr.append(variable.varValue)</code>
31	<code> print ("{} = {}".format(variable.name, variable.varValue))</code>
32	<code> if variable.varValue==1:</code>
33	<code> clas.append(count)</code>
34	<code> count+=1</code>
35	<code> if(count==D_len):</code>
36	<code> count=0</code>
37	<code> B_mfr.append(b_mfr)</code>
38	<code> b_mfr = []</code>
39	<code>Class.append(clas)</code>

Kode Sumber 4.3 Implementasi Tahap MFR dan MFT

[Halaman ini sengaja dikosongkan]

BAB V UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat keras adalah memiliki prosesor Intel Core i7 6700HQ dengan kecepatan 2,6 GHz dan memori untuk proses sebesar 24,00 GB. PC yang digunakan memiliki sistem operasi Windows 10.

Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan *software* PyCharm 2017.3.1. Penggunaan PyCharm didukung dengan *library* yaitu *pyseeta*, *sklearn*, *keras*, *keras_vgg*.

5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah video yang berisi banyak wajah dan diambil menggunakan kamera ponsel. Video yang digunakan sebagai data uji terdiri dari tiga video yang memiliki jumlah orang 4, 6, dan 8. Detail dari data video untuk uji coba ditunjukkan dengan **Tabel 5.1**.

Tabel 5.1 Detail Data Uji Coba Video

No.	Jumlah Orang	Durasi	Frame per Second (fps)
1	4	37 detik	60
2	6	43 detik	60
3	8	42 detik	60

Untuk menguji kebenaran dari hasil pengenalan wajah, digunakan data *ground truth*. Data *ground truth* diperoleh dengan cara memberi label pada wajah-wajah tiap *frame* secara manual dari masing-masing video.

5.3 Skenario Uji Coba

Uji coba dilakukan dengan dua tahap untuk mengetahui nilai *accuracy* terbaik dari pengenalan dan pelacakan banyak wajah. Uji coba pertama dilakukan untuk mengetahui jumlah kelas wajah yang menghasilkan nilai *accuracy* terbaik dari tiap video uji. Selanjutnya dilakukan uji coba tahap dua untuk mengetahui *layer* dari *extraction feature* VGG-Face yang dapat menghasilkan nilai akurasi tertinggi menggunakan jumlah kelas wajah dari uji coba tahap satu tiap video uji.

5.4 Uji Coba Jumlah Kelas Wajah pada Pengenalan dan Pelacakan Banyak Wajah

Percobaan ini dilakukan untuk menentukan jumlah kelas pada *classifier support vector machine* berdasarkan hasil akurasi dari pengenalan wajah. Pada uji coba ini setiap percobaan yang dilakukan menggunakan *feature* dari VGG-Face pada “*pool5*” *layer*. Jumlah kelas wajah yang memiliki akurasi terbaik akan digunakan untuk proses uji coba selanjutnya.

Tabel 5.2 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 4

Video Wajah 4	Jumlah Kelas Wajah		
	4	6	8
Wajah Benar	1283	1198	1246
Waktu	93.78 detik	96.09 detik	98.29 detik
Akurasi	96.68%	90.28%	93.9%

Hasil uji coba yang dilakukan untuk video uji jumlah wajah 4 ditunjukkan pada **Tabel 5.2**. Nilai *accuracy* yang dihasilkan dari tiap jumlah kelas wajah adalah 96.68% untuk jumlah kelas wajah 4, 90.28% untuk jumlah kelas wajah 6, dan 93.9% untuk jumlah kelas wajah 8.

Tabel 5.3 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 6

Video Wajah 6	Jumlah Kelas Wajah		
	6	8	10
Wajah Benar	1953	1960	1843
Waktu	138.31 detik	138.47 detik	138.28 detik
Akurasi	95.27%	95.61%	89.9%

Pada uji coba untuk video uji jumlah wajah 6 ditunjukkan pada **Tabel 5.3**. Nilai *accuracy* yang dihasilkan dari tiap jumlah kelas wajah adalah 95.27% untuk jumlah kelas wajah 6, 95.61% untuk jumlah kelas wajah 8, dan 89.9% untuk jumlah kelas wajah 10.

Tabel 5.4 Hasil Uji Coba Jumlah Kelas Wajah pada Video Wajah Jumlah 8

Video Wajah 8	Jumlah Kelas Wajah		
	8	10	12
Wajah Benar	3114	3012	3145
Waktu	170.44 detik	174.38 detik	171.14 detik
Akurasi	99.01%	95.77%	97.23%

Hasil uji coba pada video uji jumlah wajah 8 ditunjukkan pada **Tabel 5.4**. Nilai *accuracy* yang dihasilkan dari tiap jumlah kelas wajah adalah 99.01% untuk jumlah kelas wajah 8, 95.77% untuk jumlah kelas wajah 10, dan 97.23% untuk jumlah kelas wajah 12.

5.5 Uji Coba Layer VGG-Face pada Pengenalan dan Pelacakan Banyak Wajah

Uji Coba ini dilakukan untuk mengetahui layer dari VGG-Face yang dapat memberikan nilai pengenalan wajah terbaik. Layer yang dipilih akan mempengaruhi waktu dari pengenalan wajah dan akan memberikan nilai performa pengenalan wajah yang berbeda juga.

Pengujian dilakukan dengan menghitung akurasi dari hasil pengenalan. Perhitungan akurasi dilakukan dengan cara menghitung jumlah wajah yang dikenali benar selama video uji berjalan pada sistem dibandingkan dengan jumlah wajah terdeteksi pada video uji. *Layer* yang digunakan pada percobaan terdiri dari 5 layer VGG-Face yang terdiri dari “pool1”, “pool2”, “pool3”, “pool4”, dan “pool5”.

Tabel 5.5 Hasil Uji Coba Layer VGG-Face pada Video Wajah 4

Video Wajah 4	pool1	pool2	pool3	pool4	pool5
Wajah Benar	948	1055	1233	1298	1283
Waktu	94.05 detik	95.55 detik	82.51 detik	91.30 detik	93.78 detik
Akurasi	71.439%	79.503%	92.916%	97.815%	96.684%

Hasil uji coba video jumlah orang 4 ditunjukkan pada **Tabel 5.5**. Dari tabel tersebut terlihat bahwa nilai *accuracy* yang dihasilkan oleh jumlah orang 4 adalah 71.439% pada *layer* “pool1”, 79.503% pada *layer* “pool2”, 92.916% pada *layer* “pool3”, 97.815% pada *layer* “pool4”, dan 96.684% pada *layer* “pool5”.

Tabel 5.6 Hasil Uji Coba Layer VGG-Face pada Video Wajah 6

Video Wajah 6	pool1	pool2	pool3	pool4	pool5
Wajah Benar	851	1033	1557	1893	1960
Waktu	158.55 detik	118.27 detik	117.71 detik	130.99 detik	138.47 detik
Akurasi	41.512%	50.390%	75.951%	92.341%	95.610%

Pada hasil uji coba video jumlah orang 6 ditunjukkan pada **Tabel 5.6**. Dari tabel tersebut terlihat bahwa nilai *accuracy* yang dihasilkan oleh jumlah orang 6 adalah 41.512% pada *layer* “pool1”, 50.39% pada *layer* “pool2”, 75.951% pada *layer* “pool3”, 92.341% pada *layer* “pool4”, dan 95.61% pada *layer* “pool5”.

Tabel 5.7 Hasil Uji Coba Layer VGG-Face pada Video Wajah 8

Video Wajah 8	pool1	pool2	pool3	pool4	pool5
Wajah Benar	1110	1345	3052	3107	3114
Waktu	207.34 detik	163.65 detik	149.33 detik	156.33 detik	170.44 detik
Akurasi	35.294%	42.766%	97.043%	98.792%	99.014%

Hasil uji coba video jumlah orang 8 ditunjukkan pada **Tabel 5.7**. Dari tabel tersebut terlihat bahwa nilai *accuracy* yang dihasilkan oleh jumlah orang 4 adalah 35.294% pada *layer* “pool1”, 42.766% pada *layer* “pool2”, 97.043% pada *layer* “pool3”, 98.792% pada *layer* “pool4”, dan 99.014% pada *layer* “pool5”.

5.6 Evaluasi Uji Coba Jumlah Kelas Wajah pada Pengenalan dan Pelacakan Banyak Wajah

Berdasarkan hasil uji coba penentuan jumlah kelas wajah, akurasi terbaik masing-masing video uji yaitu pada data uji jumlah wajah 4 *accuracy* tertinggi adalah 96.684% pa jumlah kelas 4, data uji jumlah wajah 6 *accuracy* tertinggi adalah 95.61% pada jumlah kelas 8, dan data uji jumlah wajah 8 *accuracy* tertinggi adalah 99.014% pada jumlah kelas 8.

Penggunaan jumlah kelas wajah yang semakin banyak memerlukan memori dan waktu komputasi yang semakin banyak juga. Secara keseluruhan penggunaan wajah yang semakin banyak akan mengurangi nilai *accuracy* dari pengenalan wajah.

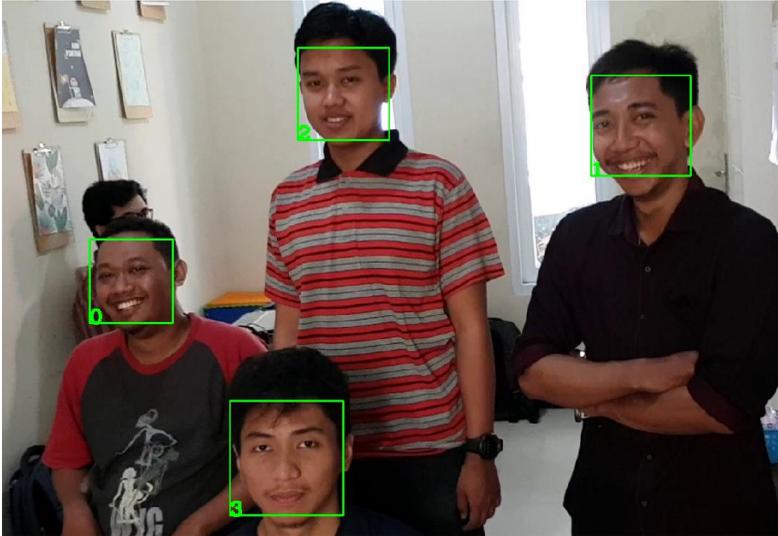
5.7 Evaluasi Uji Coba Jumlah *Layer* VGG-Face pada Pengenalan dan Pelacakan Banyak Wajah

Berdasarkan hasil uji coba penentuan *layer* VGG-Face, akurasi terbaik masing-masing video uji yaitu pada video uji jumlah wajah 4 *accuracy* tertinggi adalah 97.815% pada *layer* “pool4”, video uji jumlah wajah 6 *accuracy* tertinggi adalah 95.610% pada *layer* “pool5”, dan video uji jumlah wajah 8 *accuracy* tertinggi adalah 99.014% pada *layer* “pool5”. Pada *layer* “pool1” dan “pool2” memiliki nilai *accuracy* yang rendah hal ini terjadi karena *feature* yang dihasilkan dari *layer* tersebut masih kurang mewakili dari suatu wajah.

5.8 Evaluasi Wajah pada *Frame* Salah Pengenalan

Dalam melakukan pengenalan, sistem pengenalan dan pelacakan banyak wajah juga mengalami salah pengenalan yaitu wajah yang dikenali tidak sesuai dengan wajah yang ada pada data latih video. Hasil pengenalan yang benar dapat dilihat pada **Gambar 3.7** dan hasil pengenalan wajah yang salah dapat dilihat pada **Gambar 5.1**. Dari uji coba yang telah dilakukan, hasil pengenalan wajah dapat dipengaruhi oleh beberapa faktor yaitu besar wajah dari wajah yang dideteksi, kualitas gambar wajah yang ditangkap dan variasi wajah yang belum diwakili pada data latih. Besar wajah sangat mempengaruhi dalam pengenalan wajah dikarenakan ketika wajah memiliki ukuran yang kecil maka wajah tersebut juga akan mengurangi besar dari *feature* yang dimiliki. Karena semakin besar *feature* wajah yang dimiliki, maka semakin besar juga nilai unik dari wajah yang dapat didapatkan. Selain besar wajah, kualitas dari gambar yang didapatkan pada *frame* juga mempengaruhi proses pengenalan wajah karena ketika wajah yang didapatkan *blur* maka memiliki kemungkinan besar salah pengenalan. Faktor terakhir adalah variasi wajah, variasi wajah dapat membuat salah klasifikasi pada saat pengenalan wajah ketika variasi wajah yang diuji tidak dapat diwakili dengan wajah – wajah yang ada pada data latih. Untuk mendapatkan variasi wajah yang

besar sangat sulit dikarenakan wajah tersendiri memiliki kemungkinan variasi yang sangat banyak.



Gambar 5.1 Wajah Salah Pengenalan

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

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

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Proses deteksi wajah dapat dilakukan menggunakan algoritma SeetaFace.
2. Proses ekstraksi fitur dapat dilakukan menggunakan VGG-Face dengan menggunakan convolution layer.
3. Proses pengenalan dapat dilakukan menggunakan algoritma *multiple face recognition*, *multiple face tracking* dan *support vector machine*.
4. Uji Coba dilakukan dengan menentukan jumlah kelas dan layer VGG-Face yang memiliki akurasi terbaik. Pada video uji didapatkan nilai akurasi tertinggi yaitu untuk jumlah orang 4 dengan nilai 97.82%, video uji jumlah orang 6 dengan nilai 95.61%, dan video uji jumlah orang 8 dengan nilai 99.01%.

6.2 Saran

Saran yang dapat diberikan dalam pengujian system pengenalan dan pelacakan banyak wajah adalah sebagai berikut:

1. Penambahan jumlah data uji coba video dengan menambahkan variasi jumlah orang untuk mengetahui sejauh mana sistem

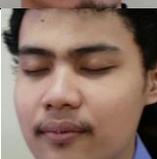
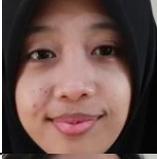
pengenalan dan pelacakan wajah mampu mengenali wajah banyak wajah.

2. Merubah perhitungan *appearance consistency* pada proses *multiple face tracking* berupa *metrics* jarak menjadi suatu *metrics learning*.

LAMPIRAN

A. Confussion Matrix Uji Coba Jumlah Kelas Wajah pada Pengenalan dan Pelacakan Banyak Wajah.

No.	Nama Kelas	Akronim	Foto
1	Mas Gilang	0	
2	Luqman	1	
3	Mas Pebri	2	
4	Mas Ius	3	
5	Jeffry	4	

6	Mas Divi	5	
7	Mas Badai	6	
8	Mas Ali	7	
9	Galang	8	
10	Hendri	9	
11	Mbak Ekky	10	
12	Kukuh	11	

Confussion Matrix Jumlah Kelas 4 pada Video Uji Jumlah 4

		<i>Predicted Class</i>			
		0	1	2	3
<i>Actual Class</i>	0	278	0	0	4
	1	0	320	9	5
	2	0	12	342	9
	3	0	5	0	343

Confussion Matrix Jumlah Kelas 6 pada Video Uji Jumlah 4

		<i>Predicted Class</i>					
		0	1	2	3	4	5
<i>Actual Class</i>	0	275	0	1	3	3	0
	1	0	323	0	8	0	3
	2	0	12	291	36	22	2
	3	0	3	1	309	35	0

Confussion Matrix Jumlah Kelas 8 pada Video Uji Jumlah 4

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	276	0	1	4	1	0	0	0
	1	0	326	0	6	1	1	0	0
	2	0	8	316	17	20	2	0	0
	3	0	5	0	328	15	0	0	0

Confussion Matrix Jumlah Kelas 8 pada Video Uji Jumlah 4

		<i>Predicted Class</i>					
		0	1	2	3	4	5
<i>Actual Class</i>	0	120	0	0	0	0	0
	1	0	353	10	1	0	4
	2	0	46	351	4	13	0
	3	2	0	2	376	0	0
	4	1	1	0	1	358	0
	5	8	0	1	0	3	395

Confussion Matrix Jumlah Kelas 6 pada Video Uji Jumlah 6

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	118	0	0	0	0	0	2	0
	1	0	366	0	1	0	1	0	0
	2	0	30	357	3	8	0	16	0
	3	2	1	1	375	0	0	1	0
	4	0	1	0	1	359	0	0	0
	5	3	0	1	0	4	385	13	1

Confussion Matrix Jumlah Kelas 8 pada Video Uji Jumlah 6

		<i>Predicted Class</i>									
		0	1	2	3	4	5	6	7	8	9
<i>Actual Class</i>	0	118	0	0	0	0	0	0	0	2	0
	1	0	359	0	1	0	4	0	0	4	0
	2	0	40	274	1	5	0	9	0	36	49
	3	1	1	1	373	1	0	0	0	1	2
	4	0	1	0	0	358	0	0	0	2	0
	5	1	0	1	0	2	361	7	1	34	0

Confussion Matrix Jumlah Kelas 10 pada Video Uji Jumlah 6

		<i>Predicted Class</i>									
		0	1	2	3	4	5	6	7	8	9
<i>Actual Class</i>	0	118	0	0	0	0	0	0	0	2	0
	1	0	359	0	1	0	4	0	0	4	0
	2	0	40	274	1	5	0	9	0	36	49
	3	1	1	1	373	1	0	0	0	1	2
	4	0	1	0	0	358	0	0	0	2	0
	5	1	0	1	0	2	361	7	1	34	0

Confussion Matrix Jumlah Kelas 8 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	383	0	0	2	1	0	0	0
	1	1	379	0	0	0	1	0	0
	2	0	17	397	1	0	0	0	0
	3	1	0	0	365	1	0	0	1
	4	0	0	0	1	406	0	0	0
	5	0	1	0	0	0	402	1	0
	6	0	0	0	0	0	1	383	0
	7	0	0	0	1	0	0	0	393

Confussion Matrix Jumlah Kelas 10 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	383	0	0	2	1	0	0	0
	1	1	379	0	0	0	1	0	0
	2	0	17	397	1	0	0	0	0
	3	1	0	0	365	1	0	0	1
	4	0	0	0	1	406	0	0	0
	5	0	1	0	0	0	402	1	0
	6	0	0	0	0	0	1	383	0
	7	0	0	0	1	0	0	0	393

Confussion Matrix Jumlah Kelas 12 pada Video Uji Jumlah 8

		<i>Predicted Class</i>											
		0	1	2	3	4	5	6	7	8	9	10	11
<i>Actual Class</i>	0	360	0	0	6	0	1	2	0	0	0	8	9
	1	1	380	0	0	0	0	0	0	0	0	0	0
	2	0	14	387	0	1	0	0	0	1	0	9	3
	3	1	0	0	363	0	1	1	0	0	0	1	1
	4	0	0	0	2	400	1	1	0	0	0	3	0
	5	0	0	1	0	0	391	0	1	0	0	11	0
	6	0	0	0	0	0	0	382	0	1	0	1	0
	7	0	0	0	1	0	1	0	395	0	1	1	1

B. Confussion Matrix Uji Coba Jumlah Layer VGG-Face pada Pengenalan dan Pelacakan Banyak Wajah.

Confussion Matrix Layer pool1 pada Video Uji Jumlah 4

		<i>Predicted Class</i>			
		0	1	2	3
<i>Actual Class</i>	0	185	11	0	86
	1	0	251	74	9
	2	44	59	259	1
	3	74	17	4	253

Confussion Matrix Layer pool2 pada Video Uji Jumlah 4

		<i>Predicted Class</i>			
		0	1	2	3
<i>Actual Class</i>	0	213	11	0	58
	1	1	288	44	1
	2	46	4	281	32
	3	46	29	0	273

Confussion Matrix Layer pool3 pada Video Uji Jumlah 4

		<i>Predicted Class</i>			
		0	1	2	3
<i>Actual Class</i>	0	273	2	7	0
	1	0	303	7	24
	2	8	21	332	2
	3	0	18	5	325

Confussion Matrix Layer pool4 pada Video Uji Jumlah 4

		<i>Predicted Class</i>			
		0	1	2	3
<i>Actual Class</i>	0	277	1	0	4
	1	1	323	2	8
	2	0	3	358	2
	3	0	6	2	340

Confussion Matrix Layer pool1 pada Video Uji Jumlah 6

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	109	1	1	5	0	0	1	3
	1	0	185	0	0	171	1	1	0
	2	87	188	3	9	3	0	113	11
	3	26	4	1	340	0	0	9	0
	4	3	3	112	7	204	0	25	7
	5	123	1	0	2	1	0	280	0

Confussion Matrix Layer pool2 pada Video Uji Jumlah 6

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	100	1	0	0	0	0	5	14
	1	0	294	0	6	63	5	0	0
	2	27	40	15	20	40	0	220	52
	3	41	0	0	318	5	0	12	4
	4	1	1	26	11	305	0	11	6
	5	206	0	0	17	1	1	182	0

Confussion Matrix Layer pool3 pada Video Uji Jumlah 6

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	119	0	0	0	0	0	1	0
	1	0	365	0	0	3	0	0	0
	2	10	33	299	1	9	0	62	0
	3	20	5	2	226	9	1	117	0
	4	0	2	0	0	357	0	2	0
	5	13	1	0	2	2	191	198	0

Confussion Matrix Layer pool4 pada Video Uji Jumlah 6

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	120	0	0	0	0	0	0	0
	1	0	367	0	0	1	0	0	0
	2	0	14	378	0	3	0	19	0
	3	1	0	2	371	0	1	5	0
	4	0	2	0	0	359	0	0	0
	5	0	0	0	1	0	298	108	0

Confussion Matrix Layer pool1 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	0	29	0	15	301	0	40	1
	1	0	302	4	11	6	58	0	0
	2	0	32	38	3	64	235	30	13
	3	13	4	27	274	0	0	3	47
	4	0	7	305	1	41	0	6	47
	5	125	16	0	18	1	54	184	6
	6	139	0	5	58	0	9	160	13
	7	133	8	1	17	0	0	0	241

Confussion Matrix Layer pool2 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	0	9	0	20	285	7	59	6
	1	0	317	0	46	3	15	0	0
	2	0	15	100	20	29	232	16	3
	3	25	29	24	244	1	1	12	32
	4	0	15	247	4	98	0	5	38
	5	184	2	0	19	2	79	114	4
	6	100	0	0	50	0	20	212	2
	7	98	2	0	2	1	0	2	295

Confussion Matrix Layer pool3 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	361	15	0	0	4	1	4	1
	1	0	375	0	4	0	2	0	0
	2	0	10	401	0	0	0	4	0
	3	6	1	0	359	1	0	1	0
	4	0	0	0	1	406	0	0	0
	5	1	5	0	0	0	379	19	0
	6	6	0	0	0	0	2	376	0
	7	2	0	0	0	0	0	3	395

Confussion Matrix Layer pool4 pada Video Uji Jumlah 8

		<i>Predicted Class</i>							
		0	1	2	3	4	5	6	7
<i>Actual Class</i>	0	366	8	0	7	1	0	2	0
	1	0	378	1	2	0	0	0	0
	2	0	7	406	0	0	1	1	0
	3	1	0	0	365	1	0	0	1
	4	0	0	0	1	406	0	0	0
	5	0	1	0	1	0	401	1	0
	6	0	0	0	0	0	0	384	0
	7	0	1	0	0	0	0	0	399

DAFTAR PUSTAKA

- [1] D. Ana, “Biometric Authentication Overview, Advantages & Disadvantages,” 12 Januari 2018. [Online]. Available: <https://heimdalsecurity.com/blog/biometric-authentication>. [Accessed 3 Juli 2018].
- [2] S. Wu, M. Kan, Z. He, S. Shan, X . Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, *Neurocomputing* 22 1, pp. 138–145, 2017
- [3] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: *British Machine Vision Conference*, 2015.
- [4] X. Zhou, K. Jin, Q. Chen, M. Xu, Y. Shang, Multiple face tracking and recognition with identity-specific localized metric learning, *Neurocomputing* 000, pp. 1–10, 2017.
- [5] A. S. Nugroho, A. B. Witarto, D. Handoko, “Support Vector Machine, Teori dan Aplikasinya dalam Bioinformatika,” 2003. . [Online]. Available: <http://asnugroho.net/papers/ikcsvm.pdf>. [Accessed 3 Januari 2018].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Kukuh Rilo Pambudi, lahir di Klaten pada tanggal 4 November 1996. Penulis telah menempuh pendidikan dari SDN Kenteng (2002-2008), SMPN 4 Purworejo (2008-2011), SMAN 7 Purworejo (2011-2014) di Purworejo. Terakhir penulis menempuh pendidikan perguruan tinggi negeri sebagai mahasiswa departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (2014-2018) di Surabaya.

Selama kuliah penulis aktif mengikuti kegiatan organisasi dan kepanitiaan. Diantaranya penulis berpartisipasi sebagai anggota Departemen Riset dan Teknologi HMTTC (2015-2016), Staf NST Schematics 2015, Staf Keluarga Muslim Informatika (2015-2016), Bendahara Umum Keluarga Muslim Informatika (2016-2017), dan Staf NST Schematics 2016.

Penulis juga mengikuti kegiatan pelatihan, diantaranya berpartisipasi sebagai peserta aktif LKMM Pra Tingkat Dasar FTIf 2014 dan peserta aktif LKMM Tingkat Dasar FTIf 2015.

Penulis memiliki bidang minat Komputasi Cerdas Visi (KCV) dengan fokus studi pada bidang pengolahan citra digital dan *data mining*. Selain itu, penulis juga memiliki ketertarikan pada perancangan dan pengembangan perangkat bergerak. Komunikasi dengan penulis dapat melalui email:

rilokukuh@gmail.com