

**LAPORAN TUGAS AKHIR**  
**PENERAPAN SISTEM ABSENSI *SMARTCAM FACE***  
***DETECTOR* BERBASIS *DEEP LEARNING***

Proposal Tugas Akhir ini diajukan untuk Melengkapi Syarat Ujian Akhir  
Program Studi D4 Teknik Telekomunikasi Jurusan Teknik Elektro  
Politeknik Negeri Semarang



Disusun Oleh:  
**AFRIZAL SALFAREL      4.39.18.0.01**

**PROGRAM STUDI D4 TEKNIK TELEKOMUNIKASI**  
**JURUSAN TEKNIK ELEKTRO**  
**POLITEKNIK NEGERI SEMARANG**  
**2022**

**LAPORAN TUGAS AKHIR**  
**PENERAPAN SISTEM ABSENSI *SMARTCAM FACE***  
***DETECTOR* BERBASIS *DEEP LEARNING***

Proposal Tugas Akhir ini diajukan untuk Melengkapi Syarat Ujian Akhir  
Program Studi D4 Teknik Telekomunikasi Jurusan Teknik Elektro  
Politeknik Negeri Semarang



Disusun Oleh:  
**AFRIZAL SALFAREL      4.39.18.0.01**

**PROGRAM STUDI D4 TEKNIK TELEKOMUNIKASI**  
**JURUSAN TEKNIK ELEKTRO**  
**POLITEKNIK NEGERI SEMARANG**  
**2022**

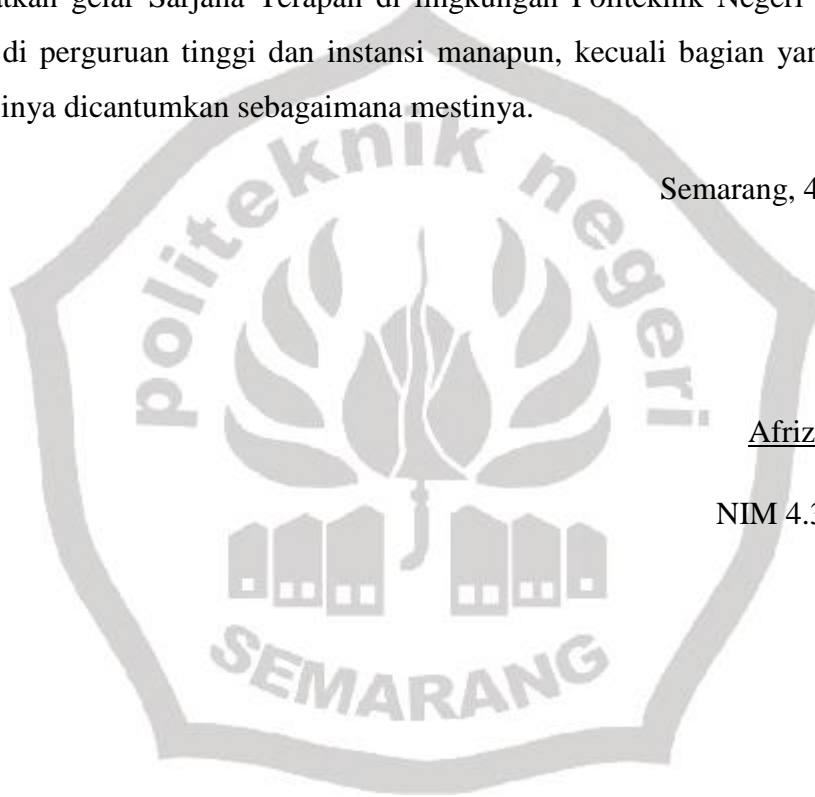
## PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir dengan judul **“Penerapan Sistem Absensi *Smartcam Face Detector* Berbasis *Deep Learning*”** yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Terapan pada Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Semarang, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari Tugas Akhir yang sudah di publikasikan dan atau pernah dipakai untuk mendapatkan gelar Sarjana Terapan di lingkungan Politeknik Negeri Semarang maupun di perguruan tinggi dan instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Semarang, 4 Juli 2022

Afrizal Salfarel

NIM 4.39.18.0.01



## PERNYATAAN PERSETUJUAN

Tugas akhir/skripsi dengan judul “**PENERAPAN SISTEM ABSENSI SMARTCAM FACE DETECTOR BERBASIS DEEP LEARNING**” dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Terapan pada Program Studi Sarjana Terapan Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Semarang dan disetujui untuk diajukan dalam sidang tugas akhir/skripsi.

Semarang, 4 Juli 2022

Pembimbing I

Pembimbing II

AGUS ROCHADI, H., ST., M.M.

ARIF NURSYAHID, H., Drs., M.T.

NIP. 19631251991031001

NIP. 196107171986031001

Mengetahui,

Ketua Program Studi S.Tr Telekomunikasi

ARI SRIYANTO N, S.T., M.T., M.Sc.

NIP. 197409042005011001

## HALAMAN PENGESAHAN

Tugas akhir/skripsi dengan judul **“PENERAPAN SISTEM ABSENSI SMARTCAM FACE DETECTOR BERBASIS DEEP LEARNING”**. Telah dipertahankan dalam ujian wawancara dan diterima sebagai syarat untuk menjadi Sarjana Terapan pada Program Studi Sarjana Terapan Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Semarang pada tanggal Juli 2022.

Tim Penguji

Penguji I,

Penguji II,

Penguji III,

NIP.

NIP.

NIP.

Ketua,

Sekretaris,

NIP.

NIP.

Mengesahkan,

Ketua Jurusan Teknik Elektro

Yusnan Badruzzaman, S.T., M.Eng

NIP. 197210271999031002

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir dengan judul “PENERAPAN SISTEM ABSENSI *SMARTCAM FACE DETECTOR* BERBASIS *DEEP LEARNING*” tepat pada waktunya. Tugas Akhir disusun untuk memenuhi salah satu persyaratan untuk menyelesaikan Program Studi D4 Teknik Telekomunikasi Jurusan Teknik Elektro di Politeknik Negeri Semarang.

Penulis sadar bahwa tugas akhir tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Supriyadi, MT. selaku Direktur Politeknik Negeri Semarang.
2. Bapak Yusnan Badruzzaman, S.T.,M.Eng. selaku Ketua Jurusan Teknik Elektro.
3. Bapak Ari Sriyanto Nugroho S.T., M.T., M.Sc. selaku Kepala Program Studi D4 Teknik Telekomunikasi
4. Agus Rochadi. H., ST., M.M. Sekaligus selaku Pembimbing I yang telah memberikan bimbingan dalam penyusunan tugas akhir.
5. Arif Nursyahid, H., Drs., M.T. selaku pembimbing II yang telah memberikan bimbingan dalam penyusunan tugas akhir.
6. Dosen-dosen D4 Teknik Telekomunikasi yang telah memberikan ilmu dan membimbing selama menempuh studi di Politeknik Negeri Semarang.
7. Orang tua yang selalu memberikan doa, dukungan dan motivasi hingga terselesaikannya tugas akhir ini.
8. Teman-Teman Teknik Elektronika yang selalu memberikan semangat dan kebersamaannya dalam proses penyusunan tugas akhir.

Semarang, 4 Juli 2022

Afrizal Salfarel

## **ABSTRAK**

*Afrizal Salfarel. "Penerapan Sistem Absensi Smartcam Face Detector Berbasis Deep Learning", Tugas Akhir Sarjana Terapan Teknik Telekomunikasi Jurusan Teknik Elektro Politeknik Negeri Semarang, dibawah bimbingan Agus Rochadi. H., ST., M.M. dan Arif Nursyahid, H., Drs., M.T., 4 Juli 2022, 51 Halaman.*

*Gambar atau citra memiliki karakteristik unik yang tidak dimiliki oleh data teks. Teknologi pengolahan citra sangat penting sebagai bentuk informasi visual, yang telah banyak diaplikasikan dalam sistem keamanan data. Sistem Pengenalan wajah merupakan hasil aplikasi pengolahan citra (Image processing). Dalam implementasi pengenalan wajah, digunakan sebuah kamera untuk menangkap wajah pengguna, lalu dibandingkan dengan wajah pengguna yang sebelumnya yang telah disimpan di dalam database. Pada penelitian ini digunakan algoritma haar cascade clasifier agar komputer dapat mengenali dan mengidentifikasi suatu wajah. Wajah Percobaan yang diidentifikasi ditangkap kamera berdasarkan wajah uji yang telah disimpan sebelumnya. Pada tahap awal wajah yang akan diidentifikasi dan wajah uji akan dimasukkan ke tahap preprocessing. Pencahayaannya yang terang akan sangat mendukung proses pemiripan wajah dibandingkan cahaya normal dan cahaya redup.*

*Kata kunci: Pengenalan wajah, wajah uji, identifikasi, algoritma haar cascade clasifier*

## **ABSTRACT**

*Afrizal Salfarel. " Penerapan Sistem Absensi Smartcam Face Detector Berbasis Deep Learning ", The Final Assignment of Applied Bachelor of Telecommunication Engineering Department of Electrical Engineering Semarang State Polytechnic, under the guidance of bimbingan Agus Rochadi. H., ST., M.M. and Arif Nursyahid, H., Drs., M.T., 4 July 2022 , 51 Pages.*

*An image has unique characteristics that are not owned by a text data. Image processing technology is very important as a form of visual information, which has been widely applied in data security systems. Face recognition system is a result of image processing application. In the implementation of face recognition, a camera used to capture the user's face. Then it is compared with the previous user's face that had been stored in the database. This study implements eigenface algorithm that sets a computer to recognize and identify a face. The identified faces are captured by a camera based on previously saved faces. Both face image to be identified and experiment face image are put to the face preprocessing stage. Bright light supports facial resemblance better in compared to normal light or dim light.*

*Keywords: face recognition, face testing, identification, haar cascade classifier algorithm*



## DAFTAR ISI

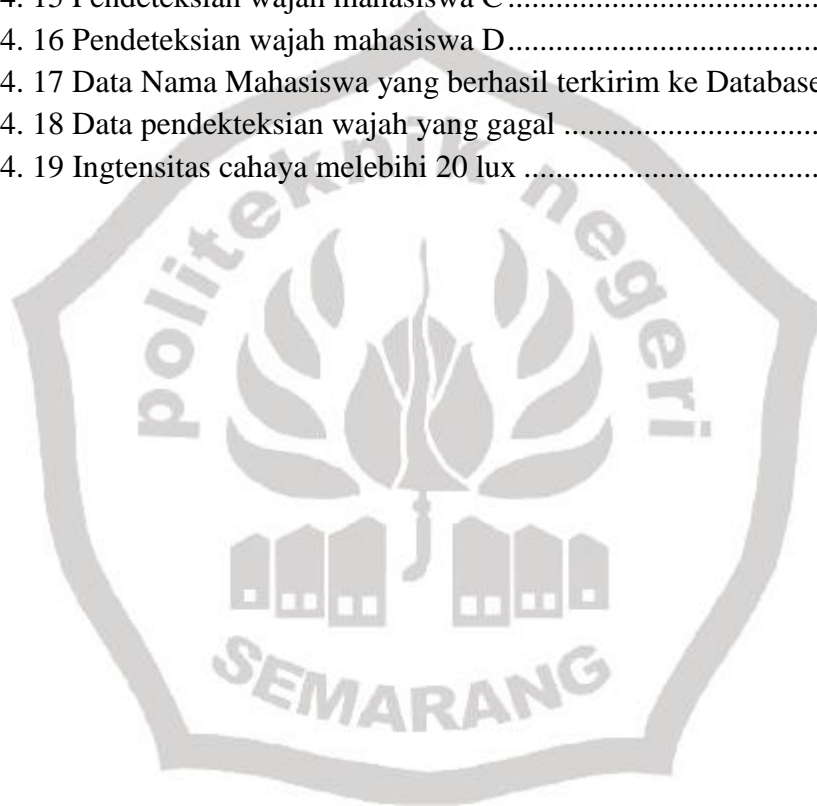
JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
PERNYATAAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR .....	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Perumusan Masalah</b> .....	2
<b>1.3 Tujuan dan Manfaat</b> .....	2
<b>1.4 Pembatasan Masalah</b> .....	2
<b>1.5 Sistematika Penulisan</b> .....	3
BAB II TINJAUAN PUSTAKA .....	4
<b>2.1 Tinjauan Pustaka</b> .....	4
<b>2.2 Landasan Teori</b> .....	5
<b>2.2.1 Deep Learning</b> .....	5
<b>2.2.2 Open CV</b> .....	7
<b>2.2.3 Visual Studio Code</b> .....	8
<b>2.2.5 Algoritma Haar Cascade</b> .....	9
<b>2.2.6 Webcam</b> .....	11
<b>2.2.7 Python</b> .....	12
<b>2.2.8 Citra Digital</b> .....	12
<b>2.2.9 Firebase</b> .....	14
BAB III KEGIATAN PELAKSANAAN .....	15
<b>3.1 Tahap Perancangan</b> .....	15
<b>3.1.1 Blok Diagram Sistem</b> .....	15
<b>3.1.2 Cara Kerja Sistem Keseluruhan</b> .....	16

3.2 Tahap Pembuatan .....	22
3.2.1 Flowchart .....	32
3.3 Perancangan Analisa Data .....	33
BAB IV ANALISIS DAN PEMBAHASAN .....	35
4.1 Hasil Rancangan Alat .....	35
4.1.1 Hasil Rancangan <i>Hardware</i> .....	35
4.1.2 Hasil Rancangan <i>Software</i> .....	36
4.2 Hasil Pengujian Fungsi Sistem .....	37
4.2.1 Data sampel wajah mahasiswa yang akan diambil untuk dikirim ke <i>Database</i> .....	38
4.2.2 Data Pendeteksian Wajah Mahasiswa .....	41
4.3 Hasil Pengujian dan Analisis .....	44
4.3.1 Hasil Pengujian Kecepatan pengambilan wajah dan jarak wajah dengan kamera .....	45
4.3.2 Hasil Pengujian Intensitas cahaya .....	45
4.3.3 Hasil Pengujian kemiripan wajah dengan sampel wajah .....	46
4.4 Analisa Hasil Pengujian .....	48
4.4.1 Analisa Pengujian Kecepatan pengambilan wajah dan jarak wajah dengan kamera .....	48
4.4.2 Analisa Pengujian Intensitas Cahaya .....	48
4.4.3 Analisa Pengujian Kemiripan wajah dengan sampel wajah .....	48
BAB V KESIMPULAN .....	48
5.1 Kesimpulan .....	48
5.2 Saran .....	49
DAFTAR PUSTAKA .....	50
LAMPIRAN .....	52

## DAFTAR GAMBAR

Gambar 2. 1 Deep Learning .....	5
Gambar 2. 2 Open CV .....	7
Gambar 2. 3 Visual Studio Code.....	8
Gambar 2. 4 Image Procecssing .....	8
Gambar 2. 5 Contoh Implementasi Algoritma Haar Cascade Classifier .....	9
Gambar 2. 6 Implementasi Haar Pada Wajah .....	10
Gambar 2. 7 Skema Pendeteksi Objek .....	11
Gambar 2. 8 Webcam Module .....	11
Gambar 2. 9 Python .....	12
Gambar 2. 10 Citra dua fungsi variabel .....	12
Gambar 2. 11 Firebase .....	14
Gambar 3. 1 Perancangan Sistem Absensi.....	15
Gambar 3. 2 Scan wajah secara vertical.....	16
Gambar 3. 3 Kelompok Pelabelan wajah pada file.....	17
Gambar 3. 4 Codebook pada Visual Code .....	17
Gambar 3. 5 Diagram Alir Citra Digital .....	18
Gambar 3. 6 Identifikasi Objek Wajah .....	19
Gambar 3. 7 Codebook Algoritma Aplikasi Dekstop .....	22
Gambar 3. 8 Tampilan Portal Absensi .....	24
Gambar 3. 9 Library OpenCv untuk Absensi Wajah .....	25
Gambar 3. 10 Library Phyton untuk Absensi Wajah .....	25
Gambar 3. 11 Library Firebase Data Absen.....	26
Gambar 3. 12 Portal Training Absen .....	26
Gambar 3. 13 Data training wajah pada file .....	27
Gambar 3. 14 Data masuk ke database .....	27
Gambar 3. 15 Menjalankan aplikasi melalui cmd.....	27
Gambar 3. 16 Menampilkan portal yang sudah dijalankan melalui cmd.....	28
Gambar 3. 17 Menjalankan kamera melalui cmd .....	28
Gambar 3. 18 Identifikasi objek wajah dan nama.....	29
Gambar 3. 19 Codebook library untuk firebase .....	29
Gambar 3. 20 Halaman firebase untuk pengiriman data.....	30
Gambar 3. 21 21 Portal absensi kamera.....	30
Gambar 3. 22 Identifikasi wajah dan nama untuk absen .....	31
Gambar 3. 23 Notifikasi bahwa data akan dikirim ke firebase .....	31
Gambar 3. 24 Flowchart pengenalan wajah.....	32
Gambar 3. 25 Flowchart training wajah.....	33
Gambar 4. 1 Rancangan Hardware .....	35
Gambar 4. 2 Aplikasi desktop absensi smartcam.....	36

Gambar 4. 3 Rancangan Software portal absensi.....	37
Gambar 4. 4 Input data sampel mahasiswa A .....	38
Gambar 4. 5 Pengiriman data sampel mahasiswa A ke Database.....	38
Gambar 4. 7 Input data sampel mahasiswa B .....	39
Gambar 4. 8 Pengiriman data sampel mahasiswa B ke Database.....	39
Gambar 4. 9 Input data sampel mahasiswa C .....	40
Gambar 4. 10 Pengiriman data sampel mahasiswa C ke Database.....	40
Gambar 4. 11 Input data sampel mahasiswa D .....	41
Gambar 4. 12 Pengiriman data sampel mahasiswa D ke Database.....	41
Gambar 4. 13 Pendeteksian wajah mahasiswa A.....	42
Gambar 4. 14 Pendeteksian wajah mahasiswa B .....	42
Gambar 4. 15 Pendeteksian wajah mahasiswa C .....	43
Gambar 4. 16 Pendeteksian wajah mahasiswa D.....	43
Gambar 4. 17 Data Nama Mahasiswa yang berhasil terkirim ke Database .....	44
Gambar 4. 18 Data pendekteksian wajah yang gagal .....	46
Gambar 4. 19 Ingtensitas cahaya melebihi 20 lux .....	46



## DAFTAR TABEL

Tabel 3. 1 Pengukuran jarak antar kamera dan kecepatan pengambilan wajah. .	33
Tabel 3. 2 Pengukuran intensitas cahaya .....	34
Tabel 3. 3 Presentase kemiripan dan pengamatan .....	34
Tabel 4. 1 Parameter data analisa.....	44
Tabel 4. 2 Pengukuran jarak antar kamera dan kecepatan pengambilan wajah....	45
Tabel 4. 3 Pengukuran intensitas cahaya .....	45
Tabel 4. 4 Presentase kemiripan dan pengamatan .....	47



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan ilmu dan teknologi saat ini telah membawa manusia kepada kehidupan yang lebih baik. Banyak sekali manfaat dan kemudahan yang telah dihasilkan dengan adanya perkembangan teknologi. Namun selain memberi berbagai manfaat, adanya perkembangan teknologi juga memberikan dampak negatif. Misalnya dampak dari segi kejujuran pekerja dan mahasiswa dengan meningkatnya tingkat kejahatan atau penipuan dan semakin majunya teknologi menjadikan keamanan menjadi suatu hal yang sangat penting. Hal ini menyebabkan kita harus berusaha mengurangi atau mencegah tindakan kriminalitas tersebut sehingga menimbulkan rasa kepercayaan khususnya pada lingkungan tempat kerja ataupun kampus. Untuk mengatasi masalah tersebut kami membuat sistem yang sederhana yaitu Penerapan Sistem Absensi *Smartcam Face Detector* Berbasis *Deep Learning* untuk para pekerja atau mahasiswa yang akan melakukan absensi di perusahaan atau kampus, Sistem ini menggunakan penerapan deep learning atau tiruan mesin otak manusia dengan *OpenCV python* yang diberi algoritma untuk mengidentifikasi wajah yang nantinya ada pada aplikasi desktop yang kemudian tersambung dengan *Webcam* kemudian disambungkan dengan sebuah web yang nantinya akan terhubung ke Wi-Fi. Web tersebut berisikan data berupa nama dan waktu (hari, tanggal, dan jam) dengan menunjukkan wajah ke *Webcam* lalu otomatis langsung teridentifikasi bahwa sudah ter absensi dan itu nanti bisa dibuktikan dengan web yang muncul pada PC atau laptop. Smartcam ini dapat mencegah terjadinya tindak kebohongan pekerja atau mahasiswa yang biasanya hanya titip absen melalui teman dan sangat mempermudah dosen atau atasan untuk melihat data-data kehadiran pekerja atau mahasiswa.

Pada era sekarang, telah banyak sekali alat yang digunakan untuk absensi, salah satunya adalah sistem fingerprint. Oleh karena itu, pada tugas akhir ini penulis membuat suatu alat absensi yang berbeda yaitu sistem smartcam. Menggunakan

*Webcam* yang dipasang di dekat pintu ruangan beserta pc atau laptop untuk mengetahui output absen yang diketahui melalui web. Dengan adanya perangkat ini, manusia mengarahkan wajahnya ke *Webcam* dan otomatis langsung terabsen.

### 1.2 Perumusan Masalah

Berdasarkan uraian latar belakang yang telah disampaikan, perumusan masalah yang akan dibahas adalah :

1. Bagaimana membuat sebuah sistem teknologi berbasis informasi yang mampu menciptakan absensi yang valid?
2. Bagaimana cara membuat sebuah alat absensi agar akurat dalam mengidentifikasi objek?

### 1.3 Tujuan dan Manfaat

Adapun tujuan yang diperoleh dalam pembuatan Tugas Akhir ini adalah:

1. Menciptakan sistem *smartcam* dalam kegiatan absensi untuk para pekerja dan mahasiswa
2. Membuat aplikasi desktop untuk absensi
3. Merancang sistem absensi menggunakan *Python* dan *OpenCV*
4. Membuat alat absensi menggunakan *Deep Learning*

### 1.4 Pembatasan Masalah

Dalam pembatasan masalah pembuatan tugas akhir ini agar pembahasan menjadi terarah dan mempunyai maksud dan tujuan yang jelas maka pembatasan masalah yang akan dibahas adalah :

1. Sistem absensi ini dikendalikan oleh Program *Python OpenCV*
2. Penggunaan *Deep Learning* dengan Membuat Algoritma *Haar Cascade* pada *OpenCV*
3. Mengidentifikasi wajah dengan modul *Webcam* yang terhubung dengan *OpenCv*
4. Penulis fokus merancang dan mengidentifikasi alat yang akan dibuat
5. Penulis tidak akan membahas tentang *Website* dan *Database*



### **1.5 Sistematika Penulisan**

Dalam penyusunan tugas akhir ini dilakukan pengelompokan menurut isi dalam beberapa bab. Bagian yang dapat berdiri sendiri dipisahkan dengan bagian yang lain dan ditempatkan dalam bab tersendiri dengan maksud mempermudah pemahaman. Adapun sistematika penulisannya adalah sebagai berikut:

#### **BAB I : PENDAHULUAN**

Bab I Pendahuluan berisi tentang latar belakang, perumusan masalah, manfaat, tujuan yang diharapkan, batasan masalah dan sistematika penulisan laporan tugas akhir.

#### **BAB II : TINJAUAN PUSTAKA**

Bab II Tinjauan Pustaka berisi tentang penjelasan mengenai teori-teori penunjang yang dijadikan landasan dalam mengerjakan tugas akhir.

#### **BAB III : KEGIATAN PELAKSANAAN**

Bab III Kegiatan Pelaksanaan berisi tentang tahap perancangan, tahap pembuatan, dan cara kerja alat yang dibuat.

#### **BAB IV : ANALISIS DAN PEMBAHASAN**

Bab IV Analisis dan Pembahasan berisi tentang pembahasan hasil percobaan, serta pembahasan analisa hasil percobaan.

#### **BAB V : KESIMPULAN**

Bab V Kesimpulan berisi tentang kesimpulan secara keseluruhan dari hasil analisis dan saran dalam rangka memperbaiki dan menyempurnakan tugas akhir.

#### **DAFTAR PUSTAKA**

Daftar Pustaka berisi sumber-sumber, jurnal, studi pustaka, yang penulis cantumkan dalam proses penyelesaian tugas akhir ini.

#### **LAMPIRAN**

Lampiran berisi data atau pelengkap atau hasil olahan yang menunjang penulisan laporan tugas akhir tetapi tidak dicantumkan di dalam isi tugas akhir, karena akan mengganggu kesinambungan penulisan.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Dalam pembuatan Tugas Akhir yang berjudul “Penerapan Sistem Absensi *Smartcam Face Detector* Berbasis *Deep Learning*” ada sejumlah penelitian yang dilakukan dengan membandingkan hasil-hasil penelitian yang telah ada sebelumnya yang berkaitan dengan judul Tugas Akhir kami.

- [1] Yessi Novamonika Eksiroka mahasiswa D3 Teknik Telekomunikasi di Universitas Telkom Bandung yang meneliti aplikasi pengenalan wajah otomatis menggunakan *OpenCV* android yang menghubungkan pada database untuk membantu sistem pencatat kehadiran agar tidak sulit untuk mencatat manual. Metode pengenalan yang dipakai adalah metode *Eigenface* dengan algoritma *Haar Cascade* yang mampu mendeteksi dengan cepat dan realtime sebuah benda termasuk wajah manusia
- [2] Rodan Hilmi Dawwas mahasiswa D3 Teknik Telekomunikasi di Universitas Telkom Bandung yang meneliti sistem informasi untuk absensi berbasis *image processing* dalam bentuk website. *Image processing* yang diterapkan yaitu dengan menggunakan metode pengenalan wajah atau *face recognition*. Pada sistem informasi ini terhubung dengan aplikasi pengenalan wajah yang menggunakan *webcam* untuk melakukan perekaman wajah. Data kehadiran di *database* yang dikirimkan oleh aplikasi pengenalan wajah akan ditampilkan pada sistem informasi absensi.
- [3] Mohammad Andre Mutalibov mahasiswa S1 Teknik Telekomunikasi di Universitas Telkom Bandung yang meneliti sistem *face recognition* yang akan dikembangkan dan diimplementasikan pada sistem absensi. *face recognition* dinilai mampu mengurangi tindak kecurangan pada sistem absensi yang digunakan pada saat ini. Metode atau cara yang dilakukan dalam melaksanakan tugas akhir ini menggunakan *OpenCV* sebagai pustaka perangkat lunak program face recognition, kamera *Raspberry Pi* sebagai penangkap gambar, dan sebuah platform *Internet of Things* bernama *ThingSpeak*. Hal-hal tersebut

nantinya akan dihubungkan satu sama lain untuk menciptakan sebuah sistem absensi dengan pengenalan wajah, dan akan diukur performansinya dengan parameter *Quality of Service (QoS)*

Berdasarkan penelitian yang telah ada, maka diusulkan suatu alat yang berjudul “Penerapan Sistem Absensi *Smartcam Face Detector* Berbasis *Deep Learning*”. Alat ini merupakan sistem absensi berbasis Deep Learning yang sama halnya menggunakan *Python* dan *OpenCV* sebagai perangkat lunak yang bisa mengidentifikasi wajah atau *face recognition* yang akan dikembangkan untuk kegiatan absensi wajah, perbedaannya disini dari beberapa penelitian ada yang menggunakan android sebagai outputnya, untuk output yang saya buat adalah melalui database dan menggunakan bahasa algoritma, serta *Webcam* sebagai modulnya.

## 2.2 Landasan Teori

### 2.2.1 Deep Learning



Gambar 2. 1 Deep Learning

Sumber: <https://markey.id/blog/technology/machine-learning-adalah>

*Deep Learning* dapat menyelesaikan masalah yang ada pada unsupervised *Machine Learning* dikarenakan cara kerja metode ini termasuk di dalamnya feature extraction. Yaitu kemampuan komputer untuk dapat menganalisa fitur yang ada di dalam data tanpa harus didefinisikan secara langsung.

*Deep learning* terdiri dari beberapa jaringan saraf tiruan yang saling berhubungan. Berikut ini adalah beberapa algoritmanya.

a. *Convolutional Neural Network (CNN)*

CNN terdiri dari banyak layer untuk memproses dan mengekstrak fitur dari data. Ia biasanya digunakan untuk memproses gambar dan mendeteksi objek. Saat ini, CNN banyak digunakan untuk mengidentifikasi citra satelit, citra medis, dan mendeteksi anomali.

b. *Recurrent Neural Network (RNN)*

*Recurrent Neural Networks (RNN)* merupakan salah satu bentuk arsitektur *Artificial Neural Networks (ANN)* yang dirancang khusus untuk memproses data yang bersambung/ berurutan (*sequential data*). RNN biasanya digunakan untuk menyelesaikan permasalahan data historis atau *time series*, contohnya data ramalan cuaca. Selain itu, RNN juga dapat diimplementasikan pada bidang *natural language understanding* (pemahaman bahasa alami), misalnya translasi bahasa.

c. *Long Short Term Memory Network (LSTM)*

LSTM merupakan tipe *Recurrent Neural Network* yang dapat mempelajari data historis atau *time series*. Ia merupakan algoritma *deep learning* yang kompleks dan dapat mempelajari informasi jangka panjang dengan sangat baik. LSTM sangat powerful untuk menyelesaikan berbagai permasalahan kompleks seperti *speech recognition*, *speech to text application*, komposisi musik, dan pengembangan di bidang farmasi.

d. *Self Organizing Maps (SOM)*

Jenis terakhir adalah *Self Organizing Maps* atau SOM. Algoritma ini mampu membuat visualisasi data secara mandiri. SOM diciptakan untuk membantu penggunaanya dalam memahami data dan informasi berdimensi tinggi.

### 2.2.2 Open CV

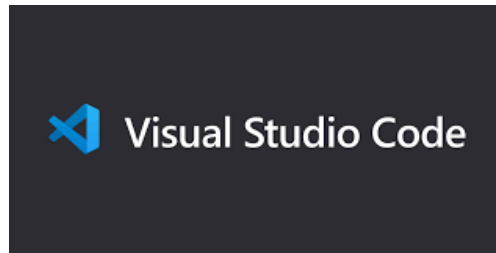


Gambar 2. 2 Open CV

Sumber : <https://keeponrockin.hatenablog.com/entry/2016/12/04/155543>

*OpenCV (Open Source Computer Vision Library)* merupakan sebuah program yang dipergunakan secara bebas dan bersifat *open source*. Program ini ditujukan untuk pengolahan citra dinamis secara riil time dan memiliki lebih dari 2500 algoritma yang sudah teroptimasi. Algoritma tersebut bisa digunakan untuk mendeteksi dan mengenali wajah, mengenali objek, mengikuti jejak kamera, dan lain-lain. Pada kasus ini, *OpenCV* digunakan untuk mendeteksi dan mengenali wajah. Secara teori *OpenCV* digunakan seperti meniru cara kerja sistem visual manusia yaitu dengan melihat objek melalui "penglihatan/mata" dan citra pada objek tersebut diteruskan ke otak untuk memproses sehingga mengerti objek apa yang tampak pada pandangan mata manusia. *OpenCV* merupakan salah satu cabang *Artificial intelligent* (kecerdasan buatan) yang digunakan untuk pengembangan atau analisis isi suatu gambar. Menghitung jumlah obyek dalam sebuah wadah menggunakan cara yang manual membuang waktu cukup lama, terutama jika obyek tersebut bergerak atau makhluk hidup. Kesalahan dalam menghitung akan sering terjadi dan dapat merugikan salah satu pihak, maka dibutuhkan aplikasi yang dapat menghitung jumlah obyek bergerak lebih mudah dan menghemat waktu. Dengan memanfaatkan teknologi Android, user dapat menggunakan aplikasi dimana saja. Program ini ditunjukkan pada penjual ikan atau nelayan yang menjual ikan dalam jumlah yang besar dan menghitung ikan satu persatu bukanlah pilihan, mereka pun menjualnya dengan mempertimbangkan berat dan memperkirakan perkilonya berisi sekian ikan hingga tidak adanya kepastian berapa jumlah ikan yang dibeli. Kesalahan dalam menghitung dapat merugikan salah satu pihak dari penjual maupun pembeli.

### 2.2.3 Visual Studio Code



Gambar 2. 3 Visual Studio Code

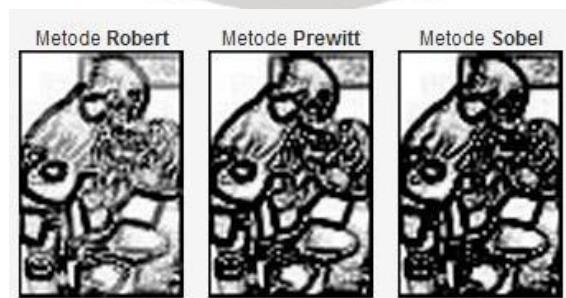
Sumber : <https://www.redbubble.com/i/mask/Visual-Studio-Code-Icon-Dark-Mode-Edition-by-zombieoumy/42146187.6BSL1>

*Visual Code Studio* adalah sebuah code editor gratis yang bisa dijalankan di perangkat desktop berbasis *Windows*, *Linux*, dan *MacOS*. Code editor ini dikembangkan oleh salah satu raksasa teknologi dunia, Microsoft.

*Visual Code* adalah software editor yang powerful, tapi tetap ringan ketika digunakan. Ia bisa dipakai untuk membuat dan mengedit source code berbagai bahasa pemrograman. Misalnya, seperti *JavaScript*, *TypeScript*, dan *Node.js*.

Bahkan, *Visual Code Studio* juga kompatibel dengan bahasa dan runtime environment lain, seperti *PHP*, *Python*, *Java*, dan *.NET*. Hal ini berkat ekosistemnya yang luas dan ketersediaan extension yang melimpah.

### 2.2.4 Image Processing



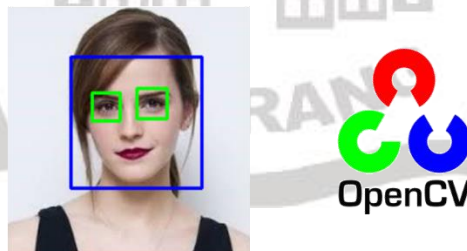
Gambar 2. 4 Image Procecssing

Sumber: <https://www.futureit.ch/image-processing-with-ai-ml-dl/>

*Image processing* adalah suatu bentuk pengolahan atau pemrosesan sinyal dengan input berupa gambar (*image*) dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu. *Image processing* dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar.

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak. Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi kontinyu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi |

### 2.2.5 Algoritma Haar Cascade



Gambar 2. 5 Contoh Implementasi Algoritma Haar Cascade Classifier

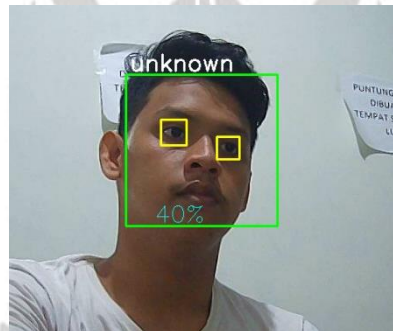
Sumber : <https://cristinamanoila28.medium.com/face-recognition-using-opencv-haar-cascade-classifications-1d42a7a5897>

Algoritma *Haar Cascade Classifier* adalah salah satu algoritma yang digunakan untuk mendeteksi sebuah wajah. Algoritma tersebut mampu mendeteksi dengan cepat dan realtime sebuah benda termasuk wajah manusia.



*Haar like Feature* merupakan metode yang lazim digunakan dalam pendeteksian obyek nama. *Haar* sendiri mengacu pada *Haar Wavelet*, sebuah fungsi matematika yang berbentuk kotak dan memiliki prinsip seperti pada fungsi Fourier. (Purwanto, Dirgantoro, & Jati, 2015). *Haar-like features* merupakan *rectangular features* (fungsi persegi), yang memberikan indikasi secara spesifik pada sebuah gambar atau image. Prinsip *Haar-like features* adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image (Viola, Paul, & Jones, 2001).

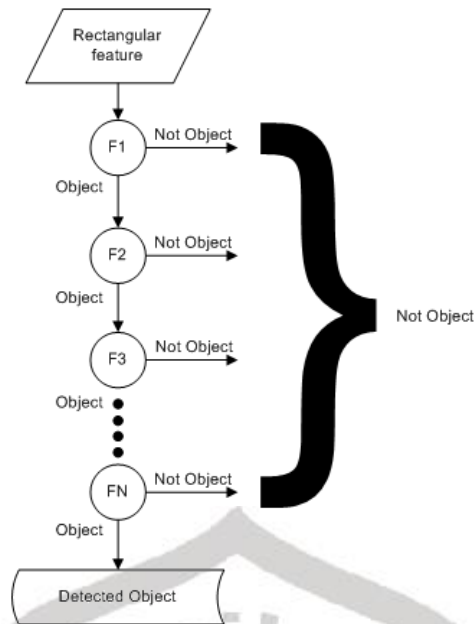
- a. Fitur persegi sederhana, disebut fitur Haar
- b. Integral image untuk pendeteksian fitur dengan cepat
- c. Metoda AdaBoost machine-learning
- d. Cascade classifier untuk mengkombinasikan banyak fitur



Gambar 2. 6 Implementasi Haar Pada Wajah

Haar like feature memproses citra wajah dalam sebuah kotak persegi dengan ukuran tertentu misalnya seperti ditunjukkan pada gambar sebagai berikut:

Di dalam kotak inilah proses filtering obyek dilakukan untuk diketahui apakah ada atau tidak obyek yang akan dideteksi. Proses filterisasi ini dilakukan secara bertingkat yang menyebabkan metode ini nantinya disebut sebagai Haar Cascade Classifier seperti ditunjukkan pada skema filter di gambar sebagai berikut:



Gambar 2. 7 Skema Pendeteksi Objek

Sumber: [https://www.researchgate.net/figure/Flowchart-for-real-time-face-detection-and-recognition\\_fig1\\_317012786](https://www.researchgate.net/figure/Flowchart-for-real-time-face-detection-and-recognition_fig1_317012786)

Hasil deteksi *Haar-like Feature* kurang akurat jika hanya menggunakan satu fungsi saja. Semakin tinggi tingkatan filter pendeteksian maka semakin tepat pula sebuah obyek dideteksi akan tetapi akan semakin lama proses pendeteksian. Pemrosesan *Haar-like feature* yang banyak tersebut diatur dalam *classifier cascade*.

### 2.2.6 Webcam



Gambar 2. 8 Webcam Module

Sumber: <https://www.ldproducts.com/logitech-c615-webcam-2-megapixel-black-usb-2.0-1-pack.html>



*Webcam* disebut pula ‘web camera’ adalah perangkat keras komputer yang berbentuk kamera digital dan dihubungkan ke laptop ataupun komputer. Kemampuannya sama seperti kamera digital lainnya, mengambil gambar, merekam video. Hanya saja, webcam dilengkapi pula dengan kemampuan merekam dan mengambil gambar secara live.

Artinya, kamu sebagai pengguna komputer bisa mengirim video dan gambar kamu secara langsung ke pengguna lain di seluruh dunia melalui jaringan internet.

### 2.2.7 Python

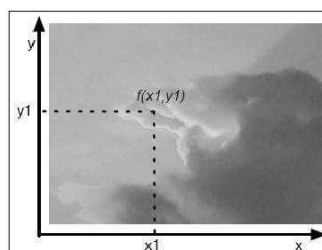


Gambar 2. 9 Python

Sumber: [https://www.vhv.rs/viewpic/hohwxbo\\_python-logo-hd-png-download/](https://www.vhv.rs/viewpic/hohwxbo_python-logo-hd-png-download/)

*Python* merupakan bahasa pemrograman yang dibuat oleh seorang keturunan Belanda yaitu Guido van Rossum. Pemrograman Bahasa *Python* ini merupakan *open source* atau *freeware*, sehingga siapapun dapat mengembangkannya. *Python* banyak digunakan untuk membuat berbagai macam program seperti CLI, GUI, Aplikasi Mobile, Web, Internet of Things, dan lain sebagainya.

### 2.2.8 Citra Digital



Gambar 2. 10 Citra dua fungsi variabel

Sumber: <https://jayateknikqu.blogspot.com/2017/07/power-law-transformation-dan-contrast.html>

Citra digital adalah suatu citra  $f(x,y)$  yang memiliki koordinat spatial, dan tingkat kecerahan yang diskrit. Citra yang terlihat merupakan cahaya yang direfleksikan dari sebuah objek. Fungsi  $f(x, y)$  dapat dilihat sebagai fungsi dengan dua unsur.

### **1. Pengenalan Wajah (Face Recognition)**

Wajah merupakan bagian dari tubuh manusia yang menjadi fokus perhatian didalam interaksi sosial, wajah memainkan peranan vital dengan menunjukkan identitas dan emosi. Kemampuan manusia untuk mengetahui seseorang dari wajahnya sangat luar biasa. Kita dapat mengenali ribuan wajah karena frekuensi interaksi yang sangat sering ataupun hanya sekilas bahkan dalam rentang waktu yang sangat lama. Bahkan kita mampu mengenali seseorang walaupun terjadi perubahan pada orang tersebut karena bertambahnya usia atau pemakaian kacamata atau perubahan gaya rambut. Oleh karena itu wajah digunakan sebagai organ dari tubuh manusia yang dijadikan indikasi pengenalan seseorang atau *face recognition*.

### **2. Konsep Pengenalan Wajah**

Pengenalan wajah adalah suatu metoda pengenalan yang berorientasi pada wajah. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : Dikenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola yang sebelumnya disimpan di dalam database. Metoda ini juga harus mampu mengenali objek bukan wajah. Perhitungan model pengenalan wajah memiliki beberapa masalah. Kesulitan muncul ketika wajah direpresentasikan dalam suatu pola yang berisi informasi unik yang membedakan dengan wajah yang lain.

### **3. Hidden Markov Models (HMM)**

*Hidden Markov models* merupakan pemodelan probabilitas suatu sistem dengan mencari parameter-parameter markov yang tidak diketahui untuk memperoleh analisis sistem tersebut. Metode *Hidden Markov Models* (HMM) mampu menangani perubahan statistik dari gambar, dengan memodelkan elemen- elemen menggunakan probabilitas.

### 2.2.9 Firebase



Gambar 2. 11 Firebase

Sumber: <https://vectorified.com/firebase-icon>

*Firebase* adalah tools database dari Google yang dapat kita gunakan untuk membantu para developer mengembangkan aplikasi mereka. Ia juga merupakan BaaS (Backend as a Service), yang dapat mempercepat pekerjaan para developer sehingga mereka dapat fokus untuk mengembangkan aplikasi tanpa effort yang besar pada back-end.

Firebase merupakan tools yang pertama kali rilis pada 2011 oleh Andrew Lee dan James Tamplin. Produk pertama Ia adalah Realtime Database yang berfungsi untuk menyimpan data dan menyinkronkannya ke beberapa user. Aplikasi ini selanjutnya berkembang menjadi berbagai layanan untuk developer aplikasi.

## BAB III

### KEGIATAN PELAKSANAAN

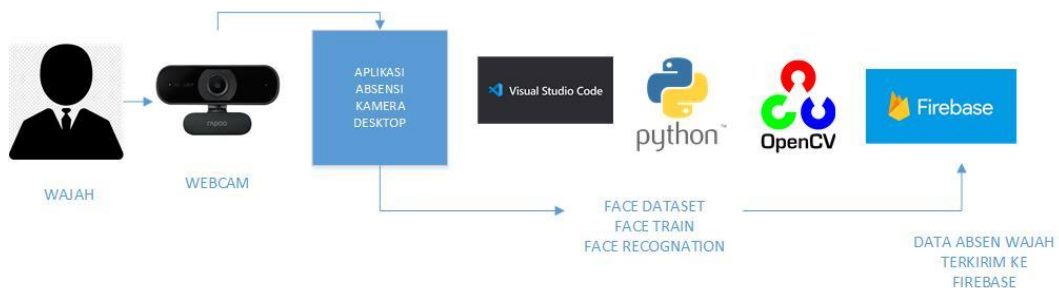
#### 3.1 Tahap Perancangan

Pada tahap perancangan Sistem Absensi *Smartcam* Berbasis *Deep Learning* ini diharapkan memiliki spesifikasi sebagai berikut :

1. Menggunakan bahasa *Python* dan *OpenCv* sebagai wadah untuk algoritma identifikasi objek wajah
2. Memiliki beberapa tahapan pembuatan algoritma atau kode pada aplikasi *Visual Studio Code*
3. Menggunakan Algoritma *Haar Cascade*, karena algoritma ini terkenal sebagai algoritma pintar yang mampu mengenali objek wajah
4. Mampu mengenal objek wajah tatapan antar wajah dan *webcam* untuk mengidentifikasi titik-titik objek wajah yang akan dilatih atau direkam
5. Menjalankan 4 urutan untuk menjalankan sistem, diantaranya yaitu Face dataset, Face Training, Face Recognition, dan aplikasi desktop
6. Menggunakan *webcam* untuk modul kameranya
7. Menggunakan *Firebase* untuk mengirim data absen wajah yang akan dikirimkan ke *website*

##### 3.1.1 Blok Diagram Sistem

Blok diagram perancangan Sistem Absensi



Gambar 3. 1 Perancangan Sistem Absensi

Garis besar sistem ini memiliki prinsip kerja sebagai berikut :

1. Wajah yang akan direkam usahakan dengan pencahayaan yang cukup
2. Data yang akan diambil adalah keseluruhan wajah meliputi titik retina mata, hidung dan bibir
3. Menggunakan aplikasi desktop untuk menginput data yang akan dikirim untuk masuk ke *webcam*
4. *Webcam* yang digunakan bisa support dengan aplikasi yang ada untuk melakukan penginputan algoritma pada sistem absensi ini
5. Data yang diterima akan diolah oleh *OpenCv* dan *Python* untuk identifikasi objek wajah
6. Data akan dikirim ke Firebase untuk diteruskan ke website

### 3.1.2 Cara Kerja Sistem Keseluruhan

#### 1. Training Wajah

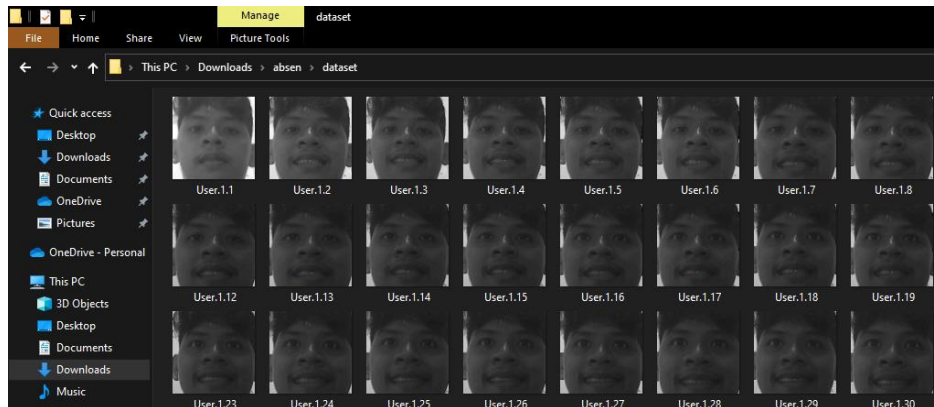


Gambar 3. 2 Scan wajah secara vertical

Pada wajah tersebut dengan lebar  $W$  dan tinggi  $H$ , dimana wajah diproses dengan cara scan secara vertical dari atas ke bawah. Prinsip kerja sistem secara keseluruhan adalah pengenalan beberapa gambar wajah yang unik dengan berbagai ekspresi yang berfungsi sebagai input dengan gambar wajah yang di simpan dalam database dengan metode training Hidden Markov Model (HMM).

Proses perancangan sistem identifikasi wajah dengan proses sebagai berikut:

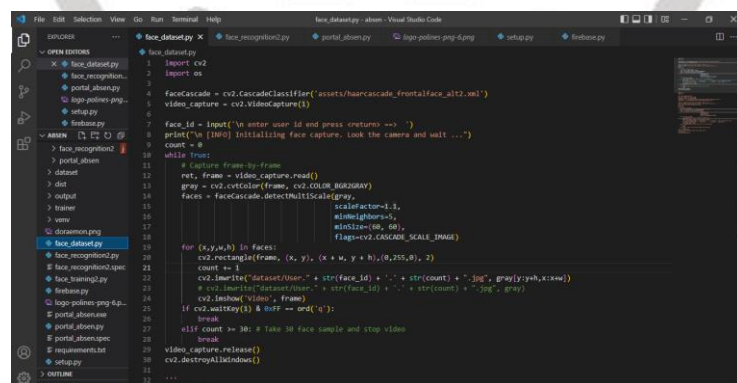
## 2. Pelabelan



Gambar 3. 3 Kelompok Pelabelan wajah pada file

Pada proses ini dilakukan pembuatan label untuk masing-masing jenis gambar wajah. Tiap-tiap gambar wajah akan dikelompokkan dalam satu label dan label tersebut diberi nama sesuai dengan nama yang dimaksud, sehingga jumlah label sama dengan jumlah gambar wajah. Nama label inilah yang nantinya akan menjadi keluaran akhir pada simulasi.

## 3. Codebook



Gambar 3. 4 Codebook pada Visual Code

Setelah proses pembuatan label dilakukan, langkah selanjutnya adalah membuat codebook dari label-label yang telah dibentuk. Pembuatan codebook ini

terlaksana ketika tombol codebook ditekan dan memilih codebook berapa yang diinginkan (32, 64, 128, 256, 512), selanjutnya menginput iterasi yang diinginkan, dan menekan tombol save untuk memulai proses pencarian codebook.

#### 4. Training database HMM

Setelah proses pembuatan codebook dilakukan, langkah selanjutnya adalah membuat training database model HMM dari label-label dan codebook-codebook yang telah terbentuk dari proses-proses sebelumnya. Untuk membuat model HMM, user harus terlebih dahulu menekan tombol Save yang terdapat dalam tampilan menu training database.

Perancangan sistem sebagai langkah awal yang sebagai mana sudah dijelaskan pada gambar pengenalan citra awal, akan tetapi dibagian ini nanti lebih pada gambaran umumnya karena rangkaiannya mencakup dari sebelum proses sampai outputnya program. Maka gambar diagram alur secara umumnya dapat dilihat seperti ini.



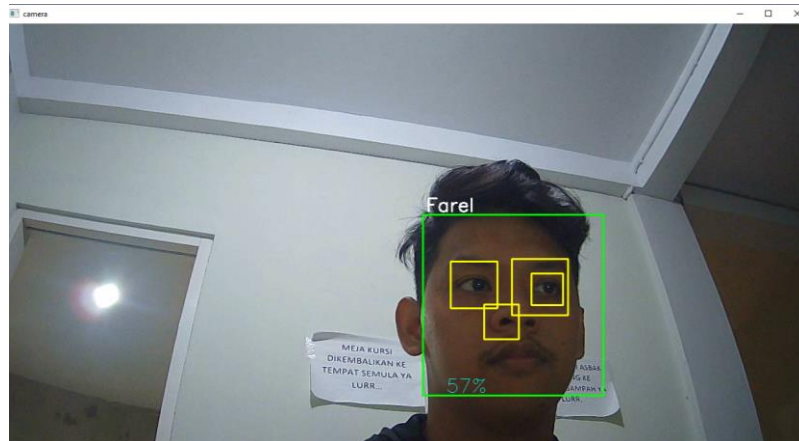
Gambar 3. 5 Diagram Alir Citra Digital

Sumber: <https://api46o.ilovepdf.com/v1/download>

Untuk proses awal yang terdiri dari Resize Citra yang bertujuan untuk pengukuran ulang citra yang selanjutnya akan diubah atau dikonvert menjadi citra Grayscale. Selain itu peneliti juga menyertakan identifikasi dengan metode Nearest Neighbor Algorithms Brute Force Metode ini melakukan pengindeksan terhadap vektor ciri sebagai langkah yang dibuat untuk mempercepat proses identifikasi.



## 5. Face Identification



Gambar 3. 6 Identifikasi Objek Wajah

Face recognition adalah sebuah teknologi yang mampu untuk mengidentifikasi dan mengkonfirmasi identitas seseorang menggunakan wajah mereka. Face recognition menjadi salah satu sistem identifikasi biometrik yang paling baik dalam mengidentifikasi seseorang dengan fitur-fitur khusus pada tubuh maupun DNA yang menjadi pembeda antara satu orang dengan orang lainnya. Menurut US Government Accountability Office, ada 4 komponen yang dibutuhkan untuk melakukan face recognition, yaitu: kamera, faceprint, Database dan terakhir Algoritme untuk membandingkan faceprint dari wajah target dengan faceprint dalam database.

Setelah komponen diatas terpenuhi, maka kita bisa mulai melakukan face recognition. Ada 3 tahapan yang biasanya dilewati, yaitu:

### 1. Detection

Sistem akan mengekstraksi pola dalam sebuah gambar lalu membandingkannya. Jika polanya sama, sistem akan mengasumsikan bahwa ada wajah dalam gambar tersebut.

### 2. Faceprint creation

Adalah ‘cetakan’ atau ‘model’ wajah. Kami akan menjelaskan detailnya di bawah.



3. Pendekatan geometris (geometric approach)  
Mengukur jarak dan relasi special antara fitur wajah seperti titik pusat mata, bagian ujung hidung, atau garis bibir untuk mengenali wajah.
4. Pendekatan Fotometrik (photometric approach)  
Menganalisis foto dan membandingkannya dengan database untuk mengenali identitas seseorang berdasarkan statistiknya.
5. Analisis tekstur wajah (skin texture analysis)  
Memetakan lokasi unik pori-pori, garis atau bercak pada kulit yang berbeda antara seseorang dengan yang lainnya.
6. Verifikasi atau Identifikasi  
Verifikasi berbeda dengan identifikasi. Jadi, outputnya akan berbeda.
7. Verifikasi (autentikasi)  
Cara ini membandingkan input foto wajah dengan data foto pengguna yang membutuhkan autentikasi. Agar lebih mudah diingat, perbandingannya  $1 \times 1$ .
8. Identifikasi  
Perbandingan  $1 \times N$ . Jadi, input foto wajah dibandingkan dengan seluruh foto wajah yang ada dalam database untuk menemukan orang yang cocok dengan input foto tersebut.

Prinsip kerja face recognition sebenarnya sangat sederhana dan mudah dimengerti. Kamera dan kecerdasan buatan akan melakukan scan wajah terhadap orang secara mendetail. Scan ini akan menyimpan semua data mulai dari bentuk mata, rahang, bibir, mulut, hidung, ukuran wajah, dan lain – lain yang ada di dalam wajah orang tersebut. Kemudian, datanya disimpan di dalam server khusus milik perusahaan atau pemilik face recognition ini.

Berikut ini adalah cara kerja pada device face recognition system yaitu:

- a. Pendeteksian wajah – Pendeteksian wajah dilakukan dengan pengambilan foto wajah dari manusia dengan memindai foto 2D secara digital, atau bisa juga menggunakan video untuk mengambil foto wajah 3D.
- b. Penjajaran – Setelah wajah berhasil dideteksi, software akan dapat menentukan posisi, ukuran, dan sikap kepala. Pada software 3D foto wajah mampu dikenali

hingga 90 derajat, sedangkan untuk software 2D posisi kepala harus menghadap kamera paling tidak 35 derajat.

- c. Pengukuran – Selanjutnya software dapat mengukur lekukan yang ada pada wajah dengan menggunakan skala sub-milimeter (microwave) dan membuat template.
- d. Representasi – Kemudian jika template sudah jadi maka template tersebut dapat diterjemahkan kedalam sebuah kode yang unik, yang mempresentasikan setiap wajah.
- e. Pencocokan – Jika foto wajah yang telah direpresentasikan dan ketersediaan foto wajah dalam basis data sama-sama 3D, proses pencocokan dapat langsung dilakukan. Namun, saat ini masih ada tantangan untuk mencocokkan representasi 3D dengan basis data foto 2D. Teknologi baru kini tengah menjawab tantangan ini. Ketika foto wajah 3D diambil, software akan mengidentifikasi beberapa titik (biasanya tiga titik) yaitu mata bagian luar dan dalam, serta ujung hidung. Berdasarkan hasil pengukuran ini software akan mengubah gambar 3D menjadi 2D, dan membandingkannya dengan gambar wajah 2D yang sudah ada di dalam basis data.
- f. Verifikasi atau identifikasi – Verifikasi merupakan proses pencocokkan satu berbanding satu. Sedangkan identifikasi adalah perbandingan foto wajah yang diambil dengan seluruh gambar yang memiliki kemiripan dalam database.
- g. Analisis tekstur wajah – Kemajuan dalam software face recognition adalah penggunaan biometrik kulit atau keunikan tekstur kulit untuk meningkatkan akurasi hasil pencocokkan. Namun terdapat beberapa faktor yang menyebabkan proses analisis tekstur ini tidak dapat bekerja, misalnya pantulan cahaya dari kacamata atau foto wajah yang menggunakan kacamata matahari. Faktor penghambat analisis lainnya adalah rambut panjang yang menutupi bagian tengah wajah, pencahayaan yang kurang tepat (yang mengakibatkan foto wajah menjadi kelebihan atau kekurangan cahaya), serta resolusi yang rendah (foto diambil dari kejauhan).

## 3.2 Tahap Pembuatan

### A. Pembuatan Aplikasi Desktop

```

portal_absen.py
1  import sys
2  import os
3  from tkinter import *
4  from PIL import ImageTk, Image
5
6  window=Tk()
7
8  window.title("Absen Kelas")
9  window.geometry('500x500')
10
11 canvas = Canvas(window, width=175, height=175)
12 canvas.pack()
13
14 img = (Image.open("logo-polines-png-6.png"))
15
16 resized_image = img.resize((150,150), Image.ANTIALIAS)
17 new_image = ImageTk.PhotoImage(resized_image)
18
19 canvas.create_image(10,10,anchor=NW,image=new_image)
20
21 label1 = Label(window, text = "Portal Absen Mahasiswa Politeknik Negeri Semarang", font=11)
22 label1.pack(pady=5, padx=2)
23
24
25 def run():
26     os.system('face_recognition2.py')
27
28 btn = Button(window, text="Absen Sekarang !", width=45,height=2,font=11, command=run)
29 btn.pack(pady=30, padx=2)
30
31 window.mainloop()

```

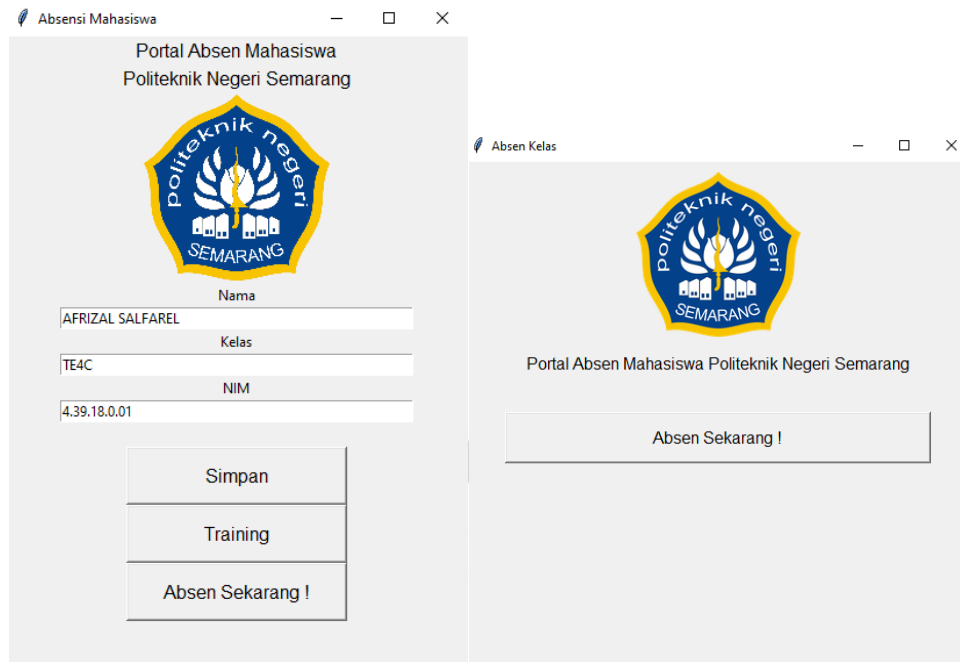
Gambar 3. 7 Codebook Algoritma Aplikasi Dekstop

Ada 2 metode utama yang user harus tau jika ingin membangun *GUI* dengan aplikasi *Python*.

1. *Tk(screenName=None, baseName=None, className='Tk', useTk=1)* : Untuk mengubah nama window, kamu dapat mengubah *classname* sesuai yang diinginkan. Kode dasar yang digunakan untuk membuat window utama aplikasi adalah *m=tkinter.Tk()* dimana *m* merupakan nama window yang diinginkan
2. *Mainloop ()* : *Mainloop ()* digunakan ketika aplikasi kamu siap dijalankan. *mainloop ()* adalah *loop* tak terbatas yang digunakan untuk menjalankan aplikasi, tunggu sampai suatu peristiwa terjadi dan proseskan peristiwa selama jendela tidak ditutup. *tkinter* juga menawarkan akses ke konfigurasi geometrik widget yang dapat mengatur widget di parent window. Ada tiga class utama dalam Geometri Class manager :
3. *method pack ()*: mengatur widget di blok sebelum menempatkan di parent widget.

4. *method grid ()*: mengatur widget di grid (struktur seperti tabel) sebelum menempatkan di parent widget.
5. *method place ()*: mengatur widget dengan menempatkannya pada posisi tertentu yang diarahkan oleh programmer.
6. *Button*: Untuk menambahkan Button pada widget kamu, kamu bisa menggunakan *syntax* seperti berikut :  
*w=Button(master, option=value)*
7. *master* adalah parameter yang digunakan untuk merepresentasikan parent windownya.
8. Ada sejumlah opsi yang digunakan untuk mengubah format Button. Jumlah opsi dapat diteruskan sebagai parameter yang dipisahkan oleh koma.
9. *Activebackground* : untuk mengatur warna background saat button berada dibawah kursor
10. *Forebackground* : untuk mengatur warna foreground saat button berada dibawah kursor
11. *Bg* : untuk mengatur warna background normal
12. *Command* : untuk memanggil function
13. *Font* : untuk mengatur font pada label button
14. *Image* : untuk menambahkan gambar pada button
15. *Width* : mengatur lebar button
16. *Height* : mengatur tinggi button

Hasil dari pembuatan Aplikasi Desktop Absensi Politeknik Negeri Semarang.



Gambar 3. 8 Tampilan Portal Absensi

## 2. Canvas

*Canvas* digunakan untuk membuat gambar dan tata letak kompleks lainnya seperti grafik, teks dan widget. *Syntax* ialah seperti berikut :

- a.  $w = \text{Canvas}(\text{master}, \text{option}=\text{value})$
- b. *bd*: untuk mengatur lebar border dalam piksel.
- c. *bg*: untuk mengatur warna latar belakang normal.
- d. *Cursor* : untuk mengatur kursor yang digunakan di kanvas.
- e. *Highlightcolor* : untuk mengatur warna yang ditunjukkan dalam sorotan fokus.
- f. *Width* : untuk mengatur lebar widget.
- g. *Height* : untuk mengatur tinggi widget.

## B. Menambahkan Library

### 1. Menambahkan *Library OpenCv* beserta Algoritma *Haar Cascade Classifier*

```

1  import cv2
2  import os
3
4  faceCascade = cv2.CascadeClassifier('assets/haarcascade_frontalface_alt2.xml')
5  video_capture = cv2.VideoCapture(1)
6
7  face_id = input('\n enter user id end press <return> ==> ')
8  print("\n [INFO] Initializing face capture. Look the camera and wait ...")
9  count = 0
10 while True:
11     # Capture frame-by-frame
12     ret, frame = video_capture.read()
13     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
14     faces = faceCascade.detectMultiScale(gray,
15                                         scaleFactor=1.1,
16                                         minNeighbors=5,
17                                         minSize=(60, 60),
18                                         flags=cv2.CASCADE_SCALE_IMAGE)
19     for (x,y,w,h) in faces:
20         cv2.rectangle(frame, (x, y), (x + w, y + h), (0,255,0), 2)
21         count += 1
22         cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])
23         # cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray)
24         cv2.imshow('Video', frame)
25         if cv2.waitKey(1) & 0xFF == ord('q'):
26             break
27         elif count >= 30: # Take 30 face sample and stop video
28             break
29     video_capture.release()
30     cv2.destroyAllWindows()
31
32 ...

```

Gambar 3. 9 Library OpenCv untuk Absensi Wajah

### 2. Menambahkan *Library Phyton*

```

1  import cv2
2  import numpy as np
3  import os
4  from datetime import datetime
5
6  import tkinter as tki # untuk tampilan GUI
7  from tkinter import Frame # untuk frame pada GUI
8  import tkinter.messagebox as tkMessageBox # message box
9
10 import firebase_admin
11 from firebase_admin import credentials
12 from firebase_admin import db
13
14 cred = credentials.Certificate('assets/serviceAccountKey.json')
15
16 # Initialize the app with a custom auth variable, limiting the server's access
17 firebase_admin.initialize_app(cred, {
18     'databaseURL': 'https://absensi-camera-default-rtdb.firebaseio.com/'
19 })
20
21 # The app only has access as defined in the Security Rules
22 ref = db.reference('/absensi')
23
24 recognizer = cv2.face.LBPHFaceRecognizer_create()
25 recognizer.read('trainer/trainer.yml')
26 cascadePath = "assets/haarcascade_frontalface_alt2.xml"
27 # cascadePath = "assets/haarcascade_frontalface_default.xml"
28 eyeDetector = cv2.CascadeClassifier('assets/haarcascade_eye.xml')
29 smileCascade = cv2.CascadeClassifier('assets/haarcascade_smile.xml')
30 faceCascade = cv2.CascadeClassifier(cascadePath)
31 font = cv2.FONT_HERSHEY_SIMPLEX

```

Gambar 3. 10 Library Phyton untuk Absensi Wajah

### 3. Menambahkan *Library Firebase*

```

1  import firebase_admin
2  from firebase_admin import credentials
3  from firebase_admin import db
4
5  # Fetch the service account key JSON file contents
6  cred = credentials.Certificate('assets/serviceAccountKey.json')
7
8  # Initialize the app with a custom auth variable, limiting the server's access
9  firebase_admin.initialize_app(cred, {
10 |   'databaseURL': 'https://absensi-camera-default-rtdb.firebaseio.com/'
11 | })
12
13 # The app only has access as defined in the Security Rules
14 ref = db.reference('/absensi')
15
16 # posts_ref = ref.child('22092020')
17 # print(posts_ref.get())
18
19 ref.push({'nama': 'farel', 'nim': 13.34})

```


Gambar 3. 11 Library Firebase Data Absen

Kemudian setelah sudah ada *library* di masing-masing program, saya coba jalankan melalui *command prompt*.

### 4. Selanjutnya, saya coba training wajah dengan aplikasi desktop yang saya buat

Absensi Mahasiswa

Portal Absen Mahasiswa  
Politeknik Negeri Semarang



Nama  
AFRIZAL SALFAREL

Kelas  
TE4C

NIM  
4.39.18.0.01

Simpan

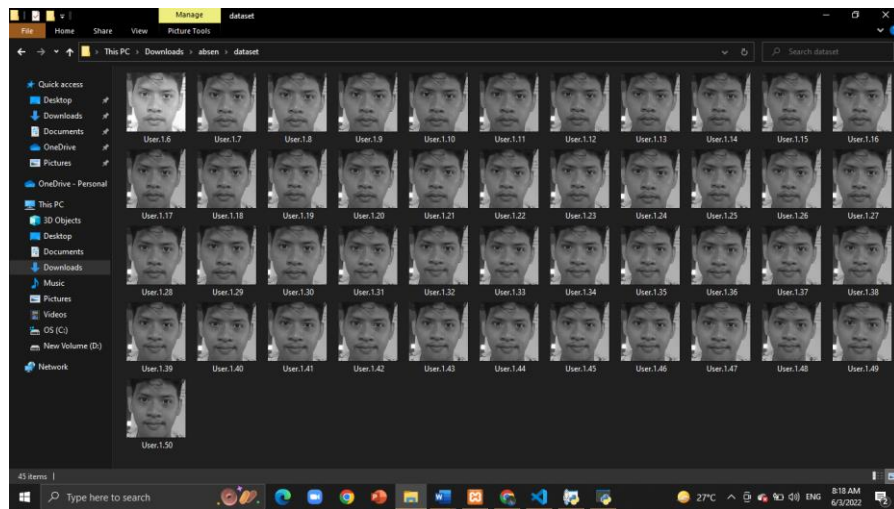
Training

Absen Sekarang !

Gambar 3. 12 Portal Training Absen



Lalu kita isi, Nama, Kelas dan NIM, setelah itu klik simpan, maka wajah kita akan di training atau di rekam serta tersimpan seperti gambar berikut:



Gambar 3. 13 Data training wajah pada file

5. Kemudian data training akan masuk ke *database* untuk direkam datanya untuk pengolahan di input absen sekarang untuk bisa dikirim datanya ke *firebase*

	id	nama	kelas	nim	created_at
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1	AFRIZAL SALFAREL	TE4C	4.39.18.0.01	2022-06-03 08:16:53
<input type="checkbox"/> Pilih Semua    Dengan pilihan: <input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus <input type="checkbox"/> Ekspor					

Gambar 3. 14 Data masuk ke database

6. Menjalankan Aplikasi dekstop Absensi menggunakan *command prompt*

```

Command Prompt
Microsoft Windows [Version 10.0.19043.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd Downloads

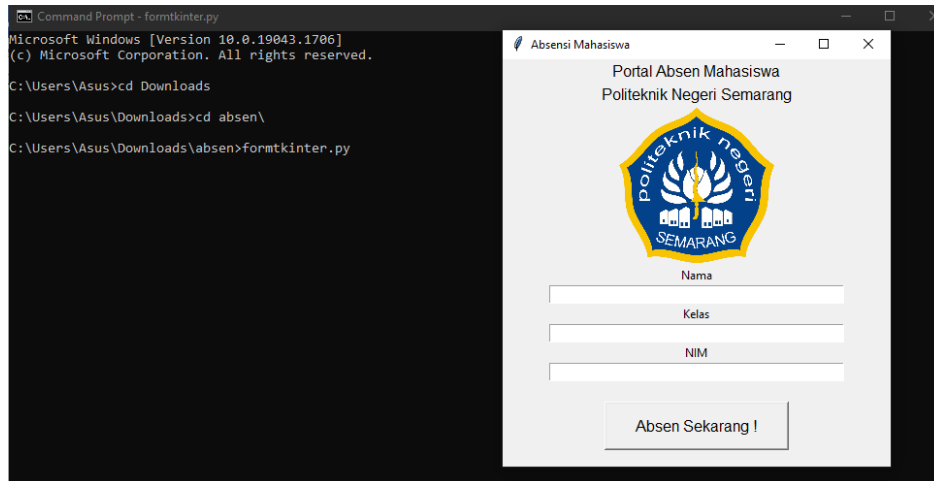
C:\Users\Asus\Downloads>cd absen\

C:\Users\Asus\Downloads\absen>
  
```

Gambar 3. 15 Menjalankan aplikasi melalui cmd

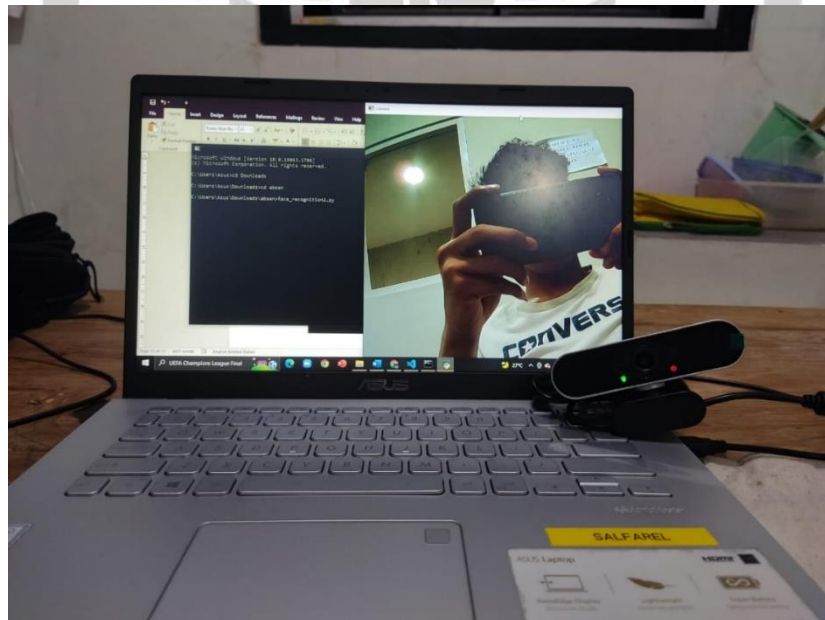


File program yang tersimpan harus pada satu file tidak boleh terpisah agar bisa dijalankan. Kemudian jalankan library yang ada di program melalui *command prompt*.



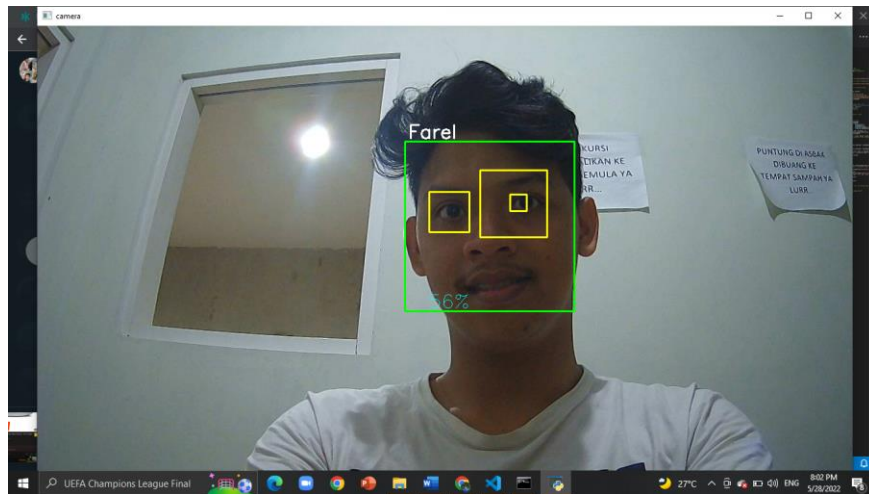
Gambar 3. 16 Menampilkan portal yang sudah dijalankan melalui cmd

#### 7. Menjalankan Aplikasi Absensi Kamera melalui *command prompt*



Gambar 3. 17 Menjalankan kamera melalui cmd

8. Kemudian saya coba untuk menjalankan Algoritma *Haar Cascade* untuk identifikasi objek wajah untuk menampilkan nama



Gambar 3. 18 Identifikasi objek wajah dan nama

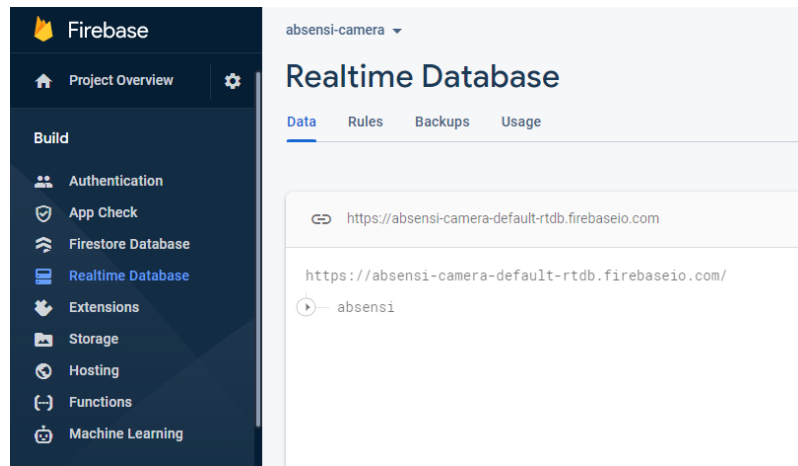
9. Selanjutnya saya tambahkan *Library* untuk merekam data wajah ke *Firebase*

```

1 import firebase_admin
2 from firebase_admin import credentials
3 from firebase_admin import db
4
5 # Fetch the service account key JSON file contents
6 cred = credentials.Certificate('assets/serviceAccountKey.json')
7
8 # Initialize the app with a custom auth variable, limiting the server's access
9 firebase_admin.initialize_app(cred, {
10     'databaseURL': 'https://absensi-camera-default-rtdb.firebaseio.com/'
11 })
12
13 # The app only has access as defined in the Security Rules
14 ref = db.reference('/absensi')
15
16 # posts_ref = ref.child('22092020')
17 # print(posts_ref.get())
18
19 # ref.push({'nama': 'farel', 'nim': 13.34})
  
```

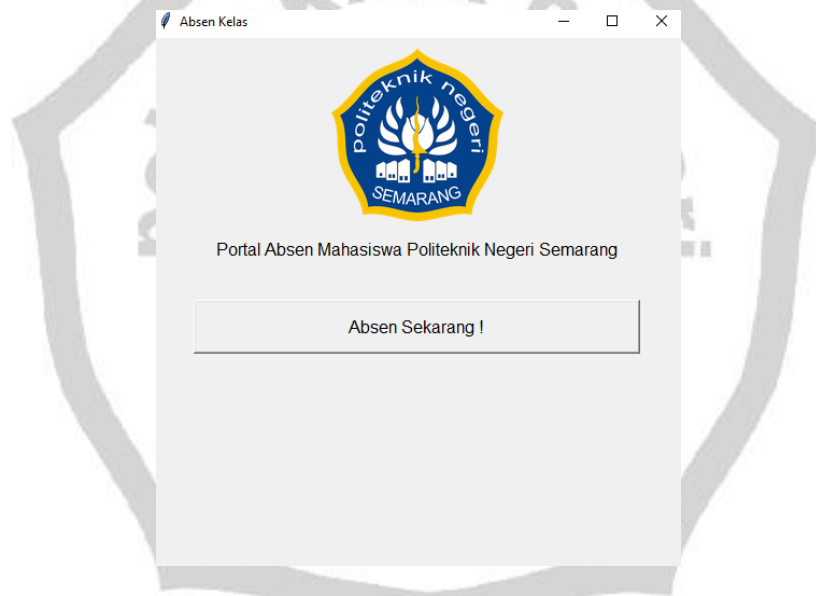
Gambar 3. 19 Codebook library untuk firebase

10. Pengaplikasian *Firebase* untuk menghubungkan antara aplikasi dan sistem absensi kamera. Pilih Realtime Database untuk melihat pengiriman data dari sistem aplikasi absensi.



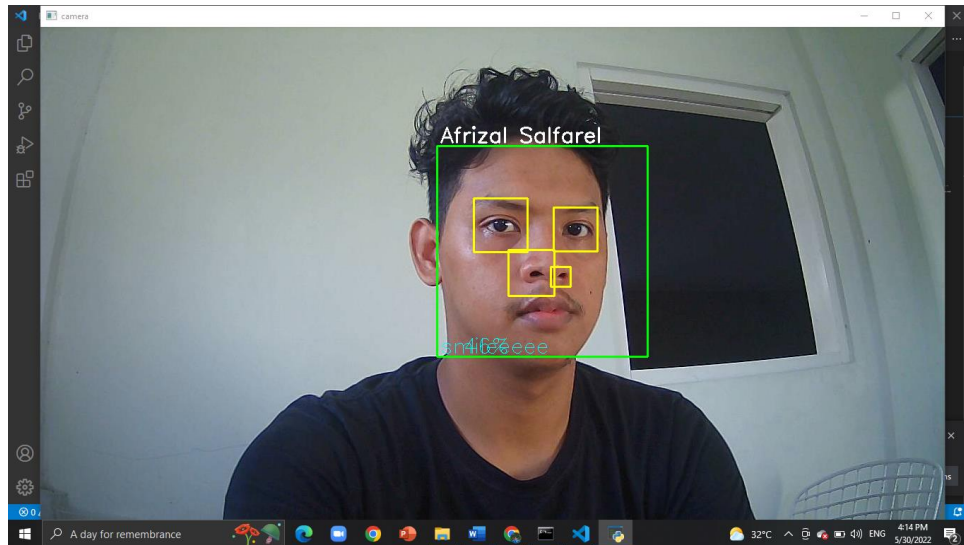
Gambar 3. 20 Halaman firebase untuk pengiriman data

11. Kemudian kita ulangi tetapi hanya dengan klik absen sekarang!



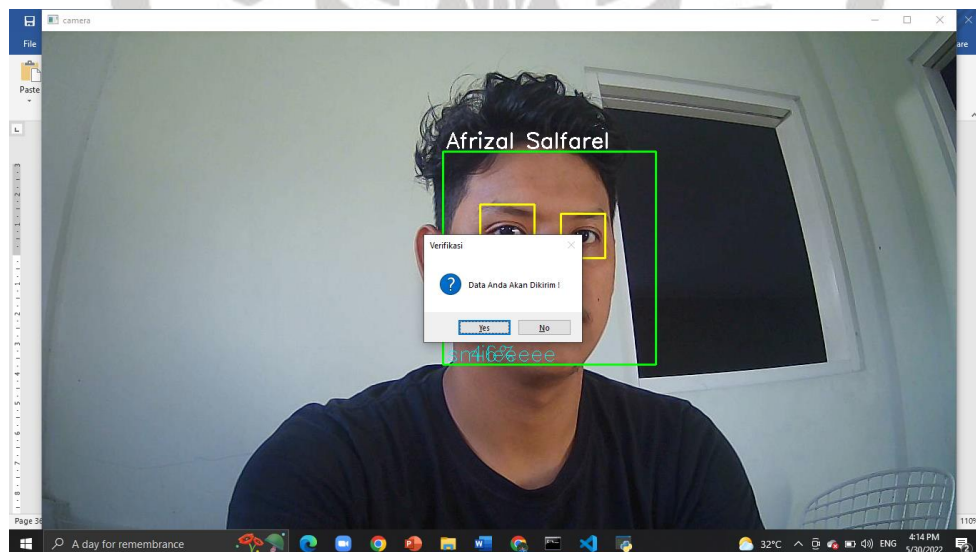
Gambar 3. 21 21 Portal absensi kamera

12. Ketika sudah klik Absen Sekarang! pada desktop, maka akan muncul sistem absensi kameranya secara otomatis. Setelah muncul kamera, hadapkan wajah ke kamera dengan posisi yang sesuai agar sistem berjalan dengan baik untuk mengidentifikasi objek wajah dengan nama yang sesuai dengan yang di inputkan pada portal absen.



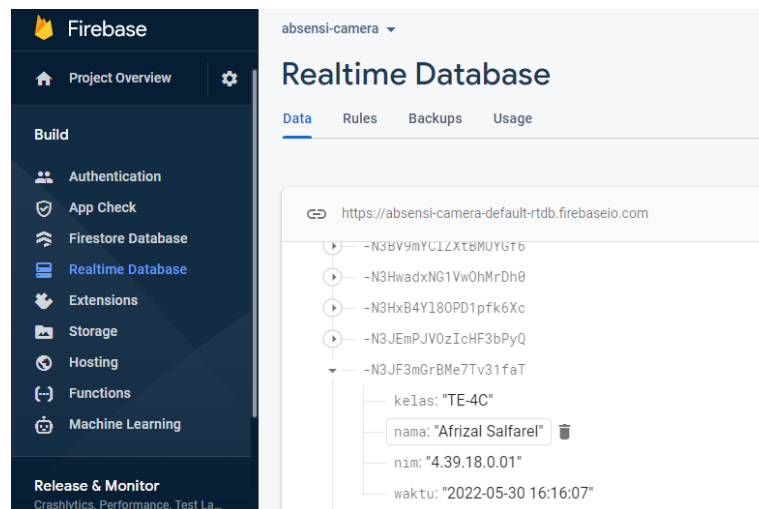
Gambar 3. 22 Identifikasi wajah dan nama untuk absen

13. Selanjutnya ketika sudah terdeteksi, maka akan muncul notifikasi Data akan dikirim!, lalu klik *Yes*, maka data saya sudah terkirim ke *Firebase* untuk diolah agar bisa muncul di *website*



Gambar 3. 23 Notifikasi bahwa data akan dikirim ke firebase

14. Setelah data dikirim, otomatis muncul ke *Realtime Database* di *Firestore*

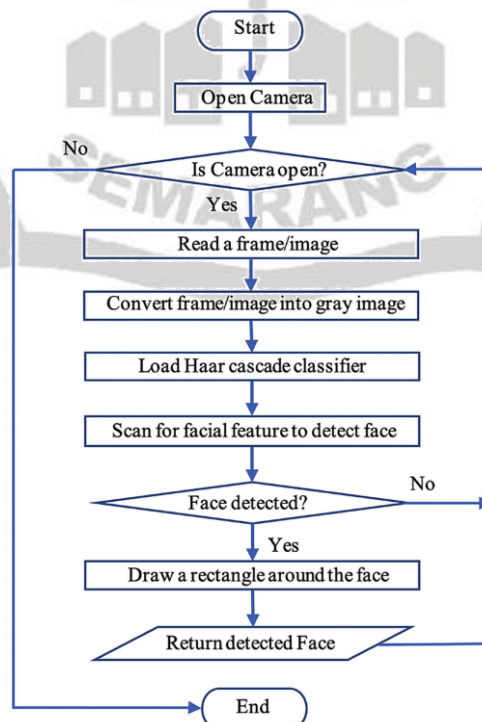


Gambar 3.24 Data yang terkirim ke *Firestore*

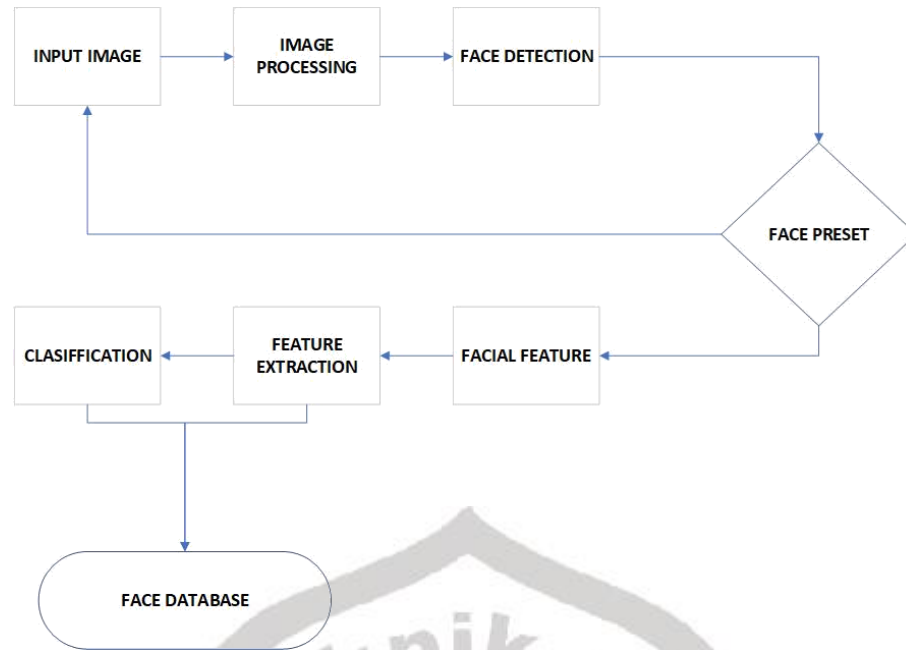
Kemudian data yang masuk akan diolah menjadi *website* oleh rekan saya.

### 3.2.1 Flowchart

Flowchart merupakan gambaran proses yang dijalankan. Terdapat flowchart untuk mendukung aplikasi yaitu flowchart pengenalan wajah dan flowchart training wajah ke database.



Gambar 3. 24 Flowchart pengenalan wajah



Gambar 3. 25 Flowchart training wajah

### 3.3 Perancangan Analisa Data

Perancangan analisa data untuk memastikan Perancangan analisa data untuk memastikan keakurasian wajah dengan parameter uji berupa kecepatan pengambilan wajah, jarak wajah dengan kamera, dan intensitas pencahayaan saat terang, normal maupun redup.

Tabel 3. 1 Pengukurran jarak antar kamera dan kecepatan pengambilan wajah.

Nama Mahasiswa/User	KECEPATAN PENGAMBILAN WAJAH DENGAN JARAK		
	30cm	1meter	1.5meter
A	.....detik	.....detik	.....detik
B	.....detik	.....detik	.....detik
C	.....detik	.....detik	.....detik
D	.....detik	.....detik	.....detik
E	.....detik	.....detik	.....detik

Tabel 3. 2 Pengukuran intensitas cahaya

PENGUKURAN INTENSITAS CAHAYA	
Nama Mahasiswa	Satuan
	...lux
	...lux
	...lux
	...lux

Pada pengenalan wajah, intensitas cahaya memperlihatkan untuk intensitas cahaya yang tinggi semakin tinggi performanya, dalam hal ini pengujian dilakukan berdasarkan kriteria pencahayaan dan setup minimum kemiripan yang berbeda. Percobaan dilakukan dengan tiga user dengan pendaftaran wajah pada database tiga kali dengan pencahayaan yang berbeda (terang, normal, redup) dan setup kemiripan (70%, 80%, 90%).

Tabel 3. 3 Presentase kemiripan dan pengamatan

Pencahayaan	User	Setup Minimum Kemiripan		
		(70%)	(80%)	(90%)
Terang	A	72	80	95
	B	76	87	93
	C	76	85	95
Normal	A	73	83	91
	B	73	85	94
	C	74	84	93
Redup	A	72	82	92
	B	71	84	0
	C	72	81	0

*Catatan: Nilai nol (0) pada tabel merupakan kesalahan dalam identifikasi.*



## BAB IV

### ANALISIS DAN PEMBAHASAN

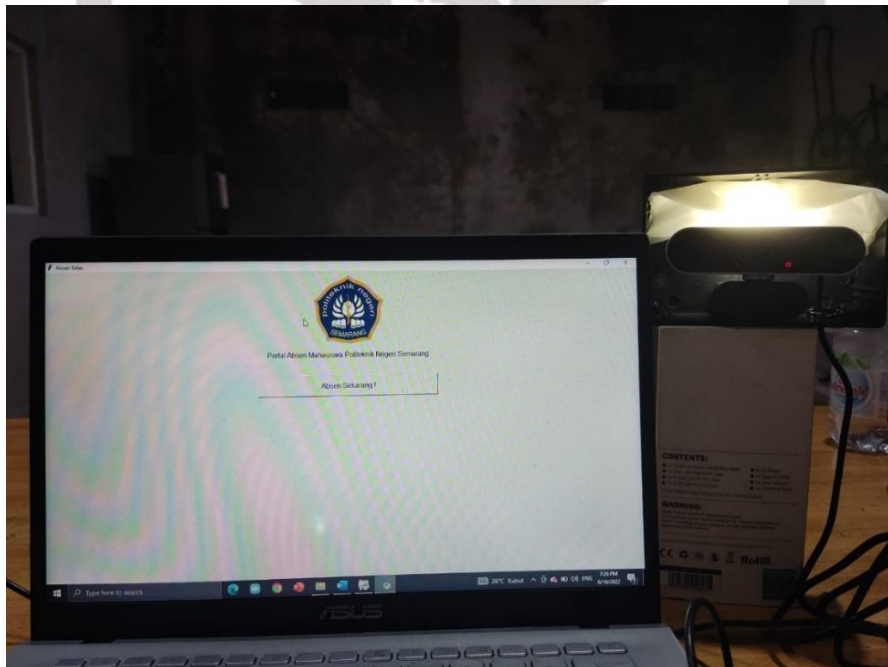
Dalam bab ini dijelaskan mengenai hasil rancangan, pengujian, dan analisis dari sistem yang telah dibuat. Pengujian dilakukan untuk mengetahui performansi dari Sistem Absensi *Smartcam* Berbasis *Deep Learning*, adapun parameter yang dianalisis meliputi perbandingan jarak kamera antar wajah, kecepatan identifikasi mengenali wajah, dan kecepatan pengiriman data wajah ke database.

#### 4.1 Hasil Rancangan Alat

Hasil rancangan alat pada tugas akhir ini meliputi rancangan *hardware* dan *software* seperti yang telah dipaparkan pada bab III. Berikut adalah hasil rancangan *hardware* dan *software*.

##### 4.1.1 Hasil Rancangan *Hardware*

Hasil rancangan *hardware sistem absensi* ditunjukan pada Gambar 4.1. di bawah ini :



Gambar 4. 1 Rancangan Hardware

Pada Gambar 4.1. merupakan hasil perancangan hardware sistem absensi *smartcam* yang terdiri dari :

1. *Webcam Xiaovv Module*

*Product : USB*

*Resolution : FHD*

*Angle : 150 derajat*

*Power Input : 5V-1A*

*Connection : USB 2.0*

*Support : Windows 7/8/10, Linux 2.4.6 or above*

2. *Laptop atau PC , minimal spesifikasi Intel Core i3 7<sup>th</sup> Gen*

3. *Dimensi alat*

*Webcam : (Panjang : 5cm, Lebar : 10cm, Tinggi : 4cm).*

*Laptop : (Panjang : 20cm, Lebar : 40cm, Tinggi : 20cm).*

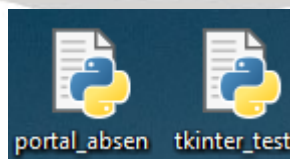
4. *Berat alat*

*Webcam : (50 gram).*

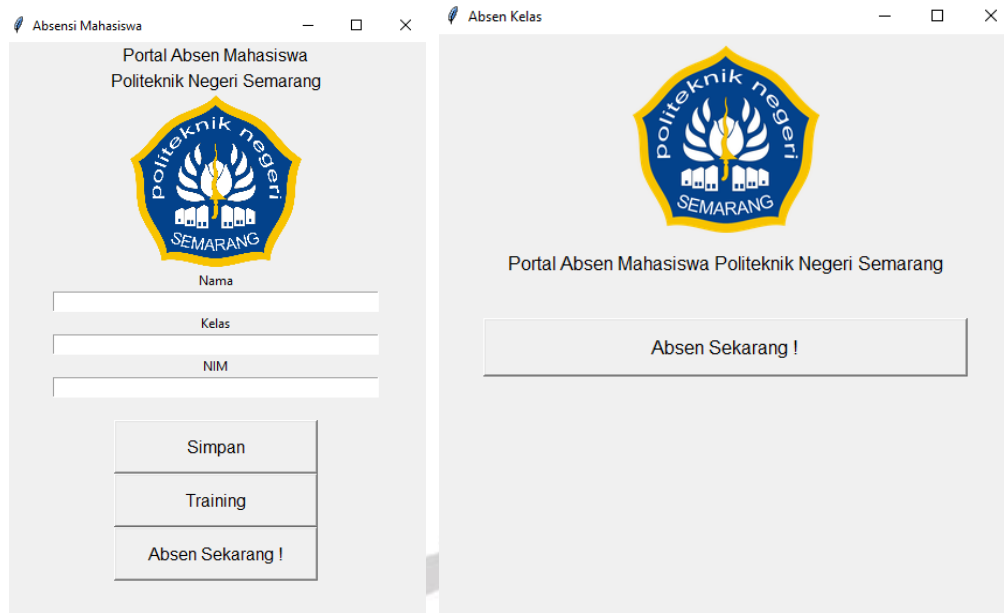
*Laptop : ( 3Kg ).*

Berikut merupakan hasil pengaplikasian atau penempatan dari Sistem Absensi *Smartcam* Berbasis *Deep Learning* , Gambar 4.1 diatas merupakan pengaplikasian sistem absensi *smartcam*, dan portal absensi yang disematkan untuk melakukan absensi agar mudah digunakan dan mudah untuk diinput.

#### 4.1.2 Hasil Rancangan Software



Gambar 4. 2 Aplikasi desktop absensi smartcam



Gambar 4. 3 Rancangan Software portal absensi

Pada rancangan *software* yang telah dijelaskan pada bab III, berikut merupakan pembacaan program-program yang telah dibuat pada masing-masing sistem dan data yang ditampilkan pada portal yang sesuai dengan hasil perancangan, meliputi :

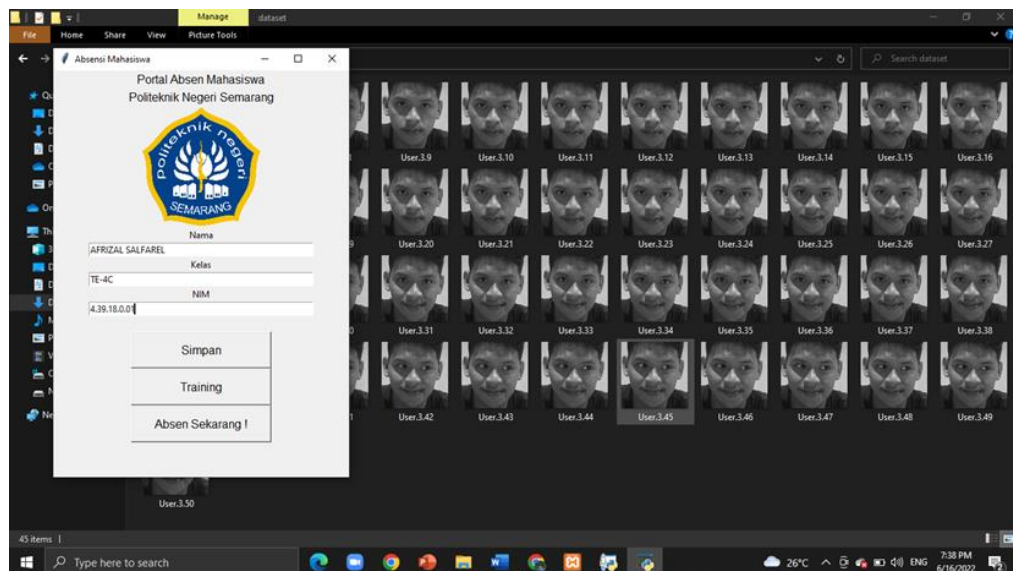
1. Aplikasi Phyton untuk sistem training wajah yang akan dikirim ke database
2. Aplikasi Phyton untuk sistem absen wajah yang akan dikirim ke firebase
3. Portal training wajah
4. Portal absensi smartcam

#### 4.2 Hasil Pengujian Fungsi Sistem

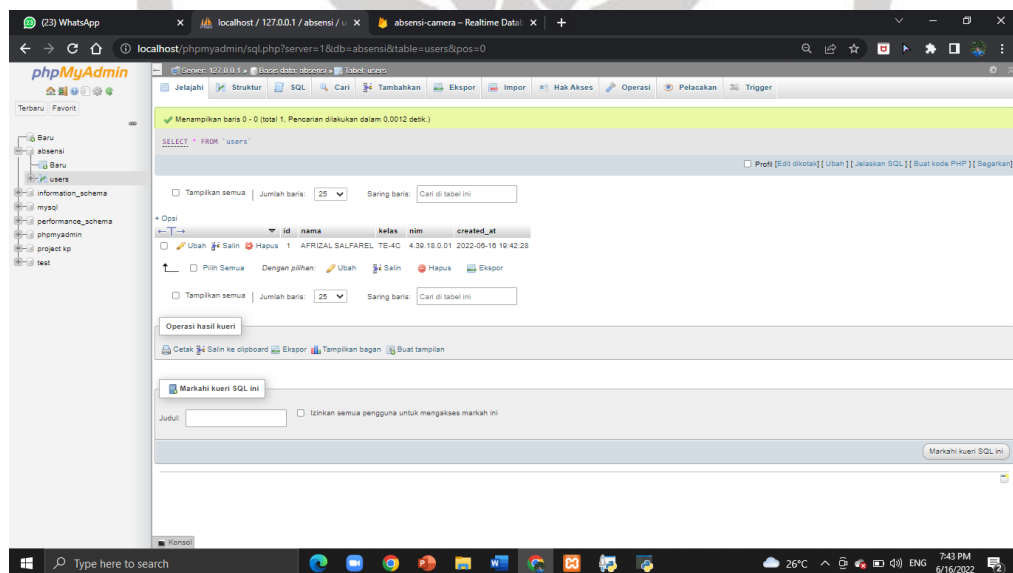
Dari perancangan *software* yang telah dijelaskan dan hasil rancangan *software* di atas, berikut merupakan hasil pengujian fungsi sistem absensi dari hasil pengambilan wajah mahasiswa.

#### 4.2.1 Data sampel wajah mahasiswa yang akan diambil untuk dikirim ke Database

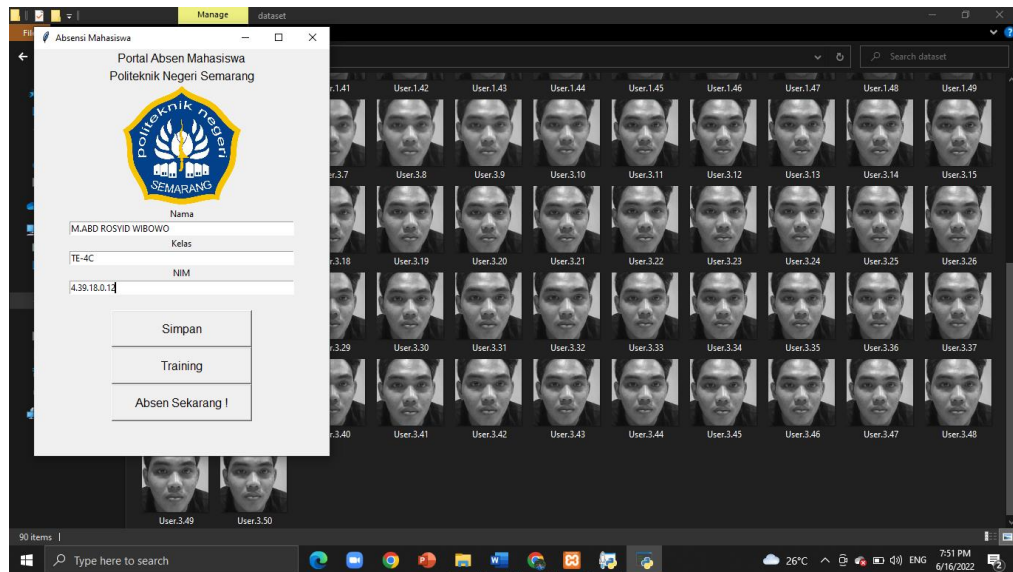
Di bawah merupakan data wajah mahasiswa yang akan diinput pada file dataset, data tersebut berupa Nama, Kelas dan NIM yang kemudian akan dikirim ke Database.



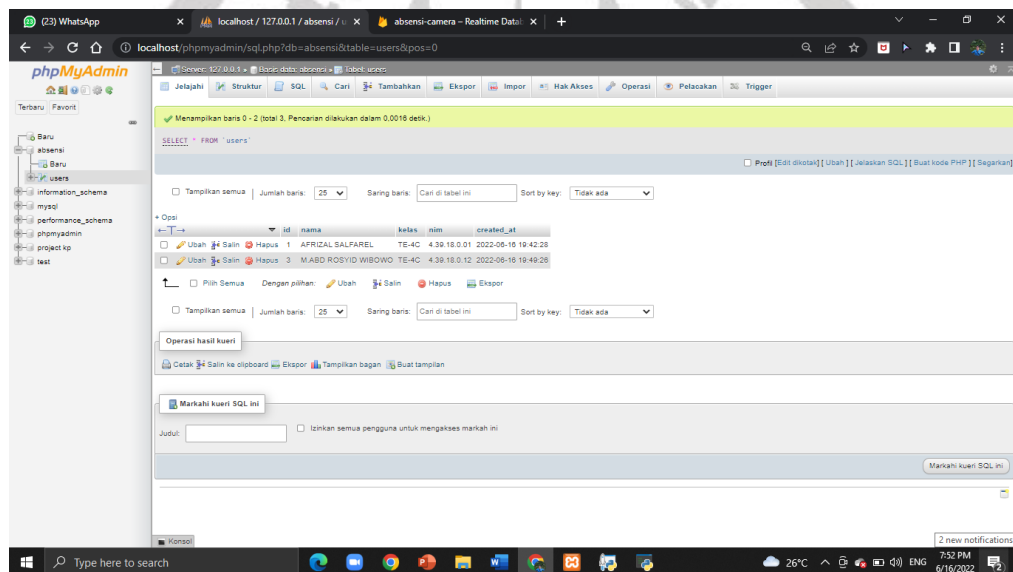
Gambar 4. 4 Input data sampel mahasiswa A



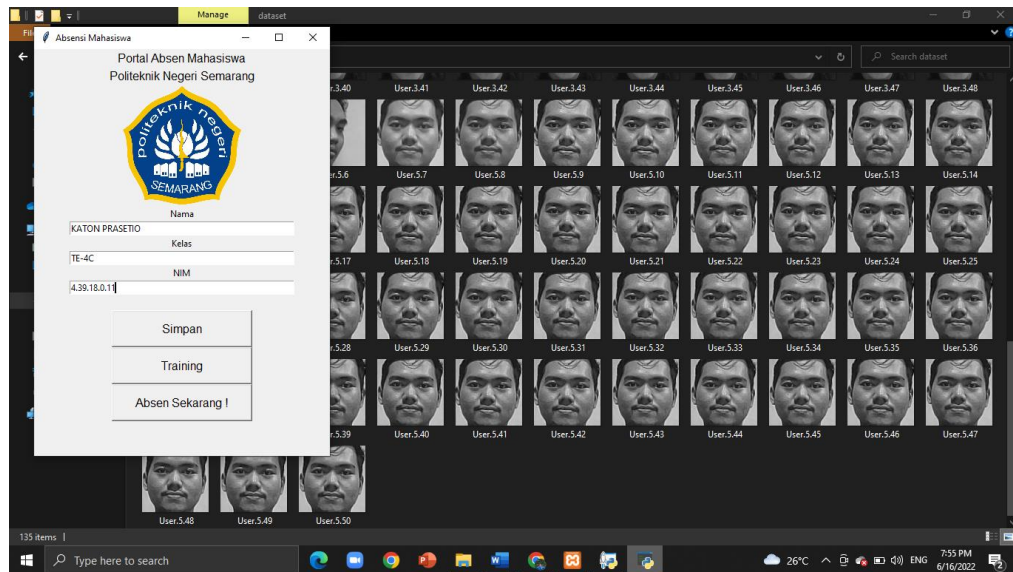
Gambar 4. 5 Pengiriman data sampel mahasiswa A ke Database



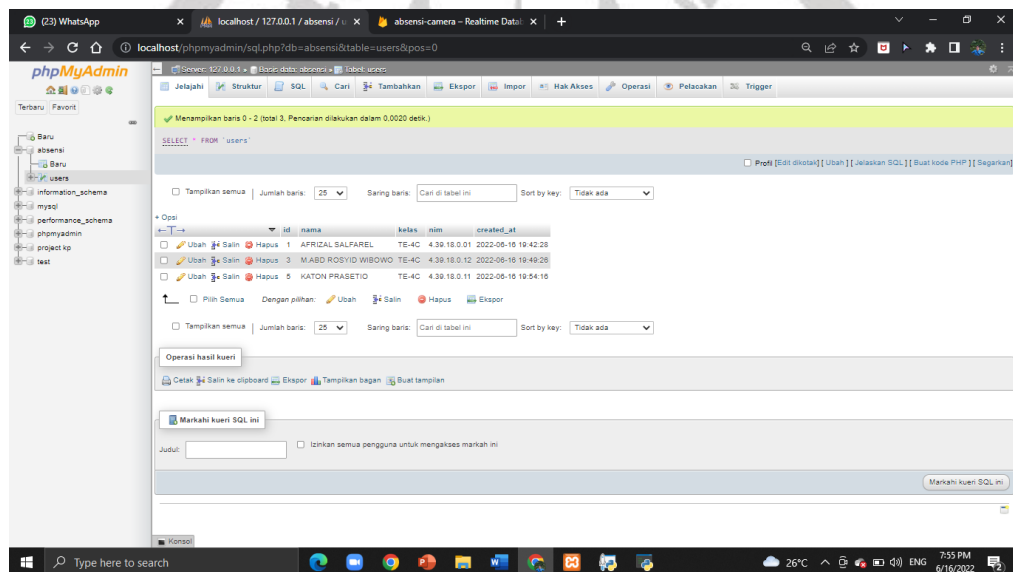
Gambar 4. 6 Input data sampel mahasiswa B



Gambar 4. 7 Pengiriman data sampel mahasiswa B ke Database

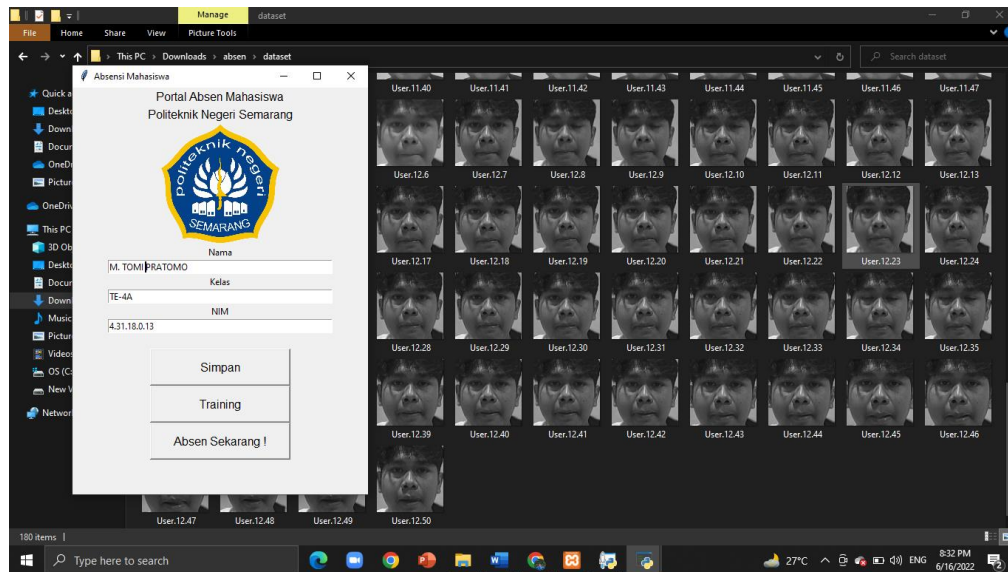


Gambar 4. 8 Input data sampel mahasiswa C

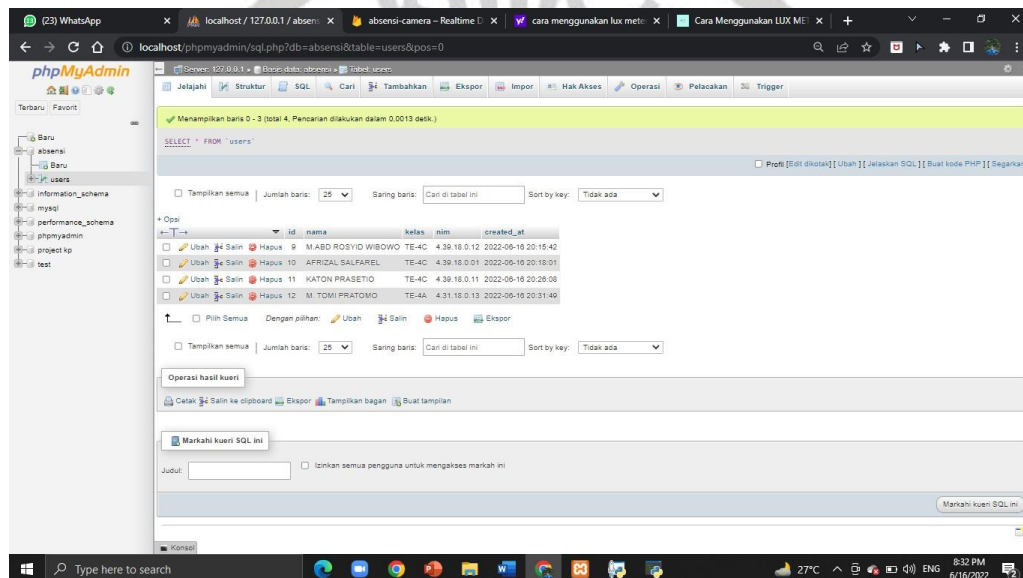


Gambar 4. 9 Pengiriman data sampel mahasiswa C ke Database





Gambar 4. 10 Input data sampel mahasiswa D

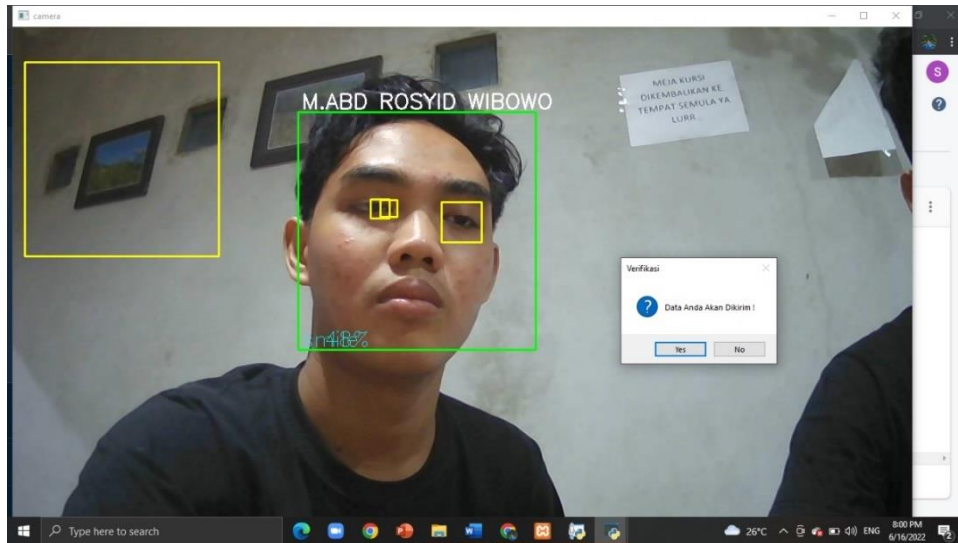


Gambar 4. 11 Pengiriman data sampel mahasiswa D ke Database

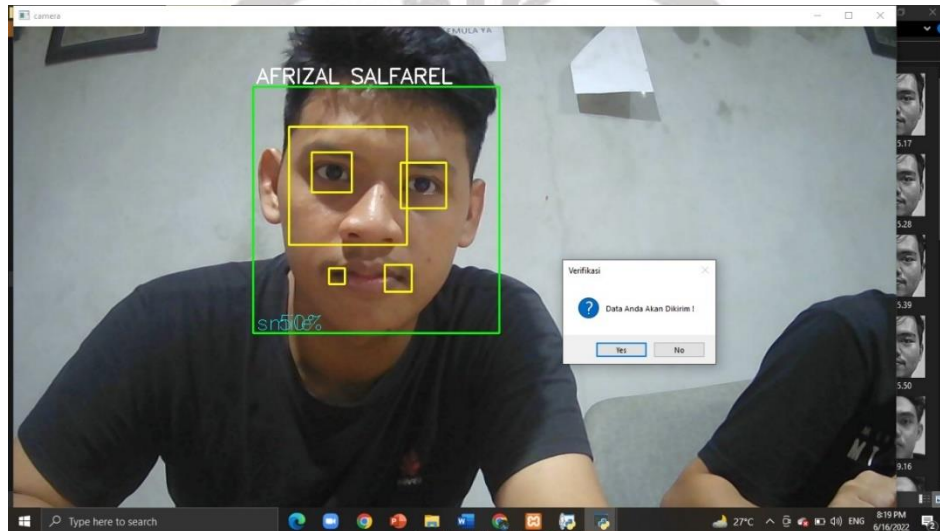
#### 4.2.2 Data Pendeteksian Wajah Mahasiswa

Setelah data sampel wajah dikirim ke *Database*, kemudian kita uji dengan melakukan pendekteksian wajah, apakah berhasil ada kemiripan dengan data atau tidak. Dibawah ini merupakan data pengujian pengambilan atau pendeteksian wajah mahasiswa.

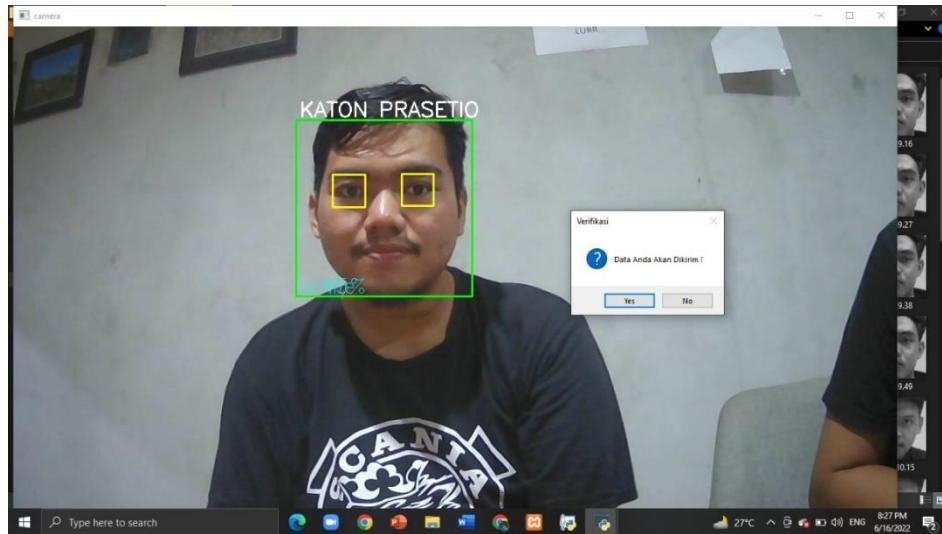




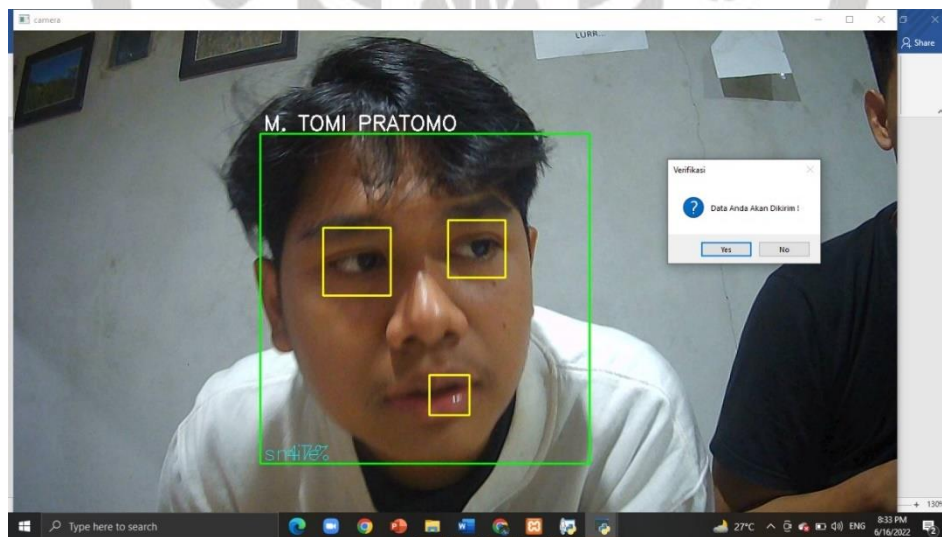
Gambar 4. 12 Pendeteksian wajah mahasiswa A



Gambar 4. 13 Pendeteksian wajah mahasiswa B



Gambar 4. 14 Pendeteksian wajah mahasiswa C



Gambar 4. 15 Pendeteksian wajah mahasiswa D

#### 4.2.3 Data wajah yang berhasil dikirim ke *Database*

Dibawah ini merupakan data nama mahasiswa yang berhasil pendeteksian wajah yang nantinya akan diolah menjadi *Website*.

+ Opsi			id	nama	nim	kelas	waktu
<input type="checkbox"/>	Ubah	Salin	Hapus	1	AFRIZAL SALFAREL	4.39.18.0.01	TE-4C 2022-06-16 20:57:01
<input type="checkbox"/>	Ubah	Salin	Hapus	3	KATON PRASETIO	4.39.18.0.11	TE-4C 2022-06-16 21:00:06
<input type="checkbox"/>	Ubah	Salin	Hapus	5	M.ABD ROSYID WIBOWO	4.39.18.0.12	TE-4C 2022-06-16 20:30:53
<input type="checkbox"/>	Ubah	Salin	Hapus	7	M.TOMI PRATOMO	4.31.18.0.13	TE-4A 2022-06-16 21:05:44

Gambar 4. 16 Data Nama Mahasiswa yang berhasil terkirim ke Database

#### 4.3 Hasil Pengujian dan Analisis

Pada pengujian menghasilkan sejumlah data seperti pada Tabel 4.1. dan kualitas serta performa sistem absensi *smartcam* tersebut .Perancangan analisa data untuk memastikan keakurasian wajah dengan parameter uji berupa kecepatan pengambilan wajah, jarak wajah dengan kamera, intensitas pencahayaan, dan Kemiripan sampel wajah saat terang, normal maupun redup.

Tabel 4. 1 Parameter data analisa.

Parameter	Spesifikasi
Jumlah Mahasiswa Absensi	4
Kecepatan Pengambilan wajah	30cm - 1meter
Batasan Intensitas Cahaya	11 – 20 <i>lux</i>
Kemiripan Sampel Wajah Saat Redup,Sedang,dan Cerah	2 Meter

#### 4.3.1 Hasil Pengujian Kecepatan pengambilan wajah dan jarak wajah dengan kamera

Tabel 4. 2 Pengukuran jarak antar kamera dan kecepatan pengambilan wajah.

Nama Mahasiswa/ <i>User</i>	KECEPATAN PENGAMBILAN WAJAH DENGAN JARAK		
	30cm	1meter	1.5meter
Afrizal Salfarel	5 detik	1 detik	2 detik
M. Abd Rosyid Wibowo	6 detik	1 detik	3 detik
Katon Prasetio	5 detik	2 detik	2 detik
M.Tomi Pratomo	6 detik	1 detik	2 detik

Pengujian kali ini hanya 4 mahasiswa dan dilakukan beberapa kali percobaan.

#### 4.3.2 Hasil Pengujian Intensitas cahaya

Tabel 4. 3 Pengukuran intensitas cahaya

PENGUKURAN INTENSITAS CAHAYA	
Nama Mahasiswa	Satuan
Afrizal Salfarel	11 <i>lux</i>
M. Abd Rosyid Wibowo	15 <i>lux</i>
Katon Prasetio	12 <i>lux</i>
M. Tomi Pratomo	16 <i>lux</i>

Pengukuran intensitas cahaya dapat dilihat pada tabel diatas, Batasan yang diambil adalah 11-20 *lux*, jika intensitas cahaya kurang atau melebihi angka tersebut, dapat dipastikan tidak akan akurat saat pendeteksian pengambilan wajah serta dapat mengambil objek secara tidak beraturan. Dibawah ini merupakan percobaan yang gagal dikarenakan intensitas cahaya melebihi 20 *lux*,



Gambar 4. 17 Data pendekteksian wajah yang gagal



Gambar 4. 18 Ingtensitas cahaya melebihi 20 lux

#### 4.3.3 Hasil Pengujian kemiripan wajah dengan sampel wajah

Pada pengenalan wajah, intensitas cahaya memperlihatkan untuk intensitas cahaya yang tinggi semakin tinggi performanya, dalam hal ini pengujian dilakukan berdasarkan kriteria pencahayaan dan setup minimum kemiripan yang berbeda.



Percobaan dilakukan dengan empat user dengan pendaftaran wajah pada database tiga kali dengan pencahayaan yang berbeda (terang, normal, redup).

Tabel 4. 4 Presentase kemiripan dan pengamatan

Pencahayaan	Nama Mahasiswa/ User	Setup Minimum Kemiripan
		(%)
Terang	Afrizal Salfarel	30%
	M.Abd Rosyid Wibowo	34%
	Katon Prasetyo	32%
	M.Tomi Pratomo	29%
Normal	Afrizal Salfarel	49%
	M.Abd Rosyid Wibowo	49%
	Katon Prasetyo	51%
	M.Tomi Pratomo	49%
Redup	Afrizal Salfarel	30%
	M.Abd Rosyid Wibowo	30%
	Katon Prasetyo	29%
	M.Tomi Pratomo	22%

Pada pengujian kemiripan wajah dengan sampel wajah kali ini dipastikan jika pencahayaan normal lebih akurat disbanding pencahayaan terang maupun redup, karena di pengujian kali ini menggunakan presentase yang dimana presentasi 45% hingga 55% sangat akurat untuk pendeteksian pengambilan wajah dan sangat mirip dengan sampel wajah.

#### **4.4 Analisa Hasil Pengujian**

##### **4.4.1 Analisa Pengujian Kecepatan pengambilan wajah dan jarak wajah dengan kamera**

Dari hasil pengujian kecepatan pengambilan wajah dan jarak wajah dengan kamera dapat dianalisa, jika jarak wajah dengan kamera kurang dari 30cm maka pengambilan wajah terbilang sangat lambat karena posisi terlalu dekat dengan kamera sehingga tidak bisa membaca objek wajah secara maksimal namun ketika jarak wajah dengan kamera lebih dari 1 meter terbilang sedikit lebih cepat karena wajah membaca objeknya sesuai, tetapi masih terlalu jauh sehingga belum akurat, jadi dapat disimpulkan ketika jarak wajah tepat 1 meter dengan kamera, bisa terbilang sangat akurat, karena kamera dapat membaca objek wajah secara cepat.

##### **4.4.2 Analisa Pengujian Intensitas Cahaya**

Dari hasil pengujian intensitas cahaya dapat dianalisa, jika intensitas cahaya wajah kurang dari 11 *lux* maka wajah tidak dapat terdeteksi, dan jika intensitas cahaya wajah lebih dari 20 *lux* wajah akan terdeteksi namun wajah tidak sesuai dengan data sampel atau tidak beraturan, jadi dapat disimpulkan ketika intensitas cahaya wajah melebihi 11 *lux* dan kurang dari 20 *lux* maka wajah dapat bisa terdeteksi dengan benar sesuai data sampel sehingga dapat digunakan untuk absensi

##### **4.4.3 Analisa Pengujian Kemiripan wajah dengan sampel wajah**

Dari hasil pengujian kemiripan wajah dengan sampel wajah melalui parameter presentase dapat dianalisa, jika saat cahaya terang dengan presentase 30%,34%,32%,29% keakurasian kemiripan sangat minim karena mempengaruhi intensitas cahaya dan tidak dianjurkan dengan pencahayaan yang terang, kemudian jika saat cahaya redup dengan presentase 30%,30%,29%,22% keakurasian kemiripan jauh lebih minim, tetapi bisa diatasi dengan alat bantu lampu sorot yang ada di sistem kamera ini, namun juga tidak dianjurkan, sehingga pencahayaan yang cocok adalah pencahayaan sedang karena dapat presentase 49%,49%,51%.49% yang keakurasian kemiripan sangat lumayan baik, dan pencahayaan ini sangat dianjurkan karena dapat digunakan dengan normal dan akurat.



## BAB V

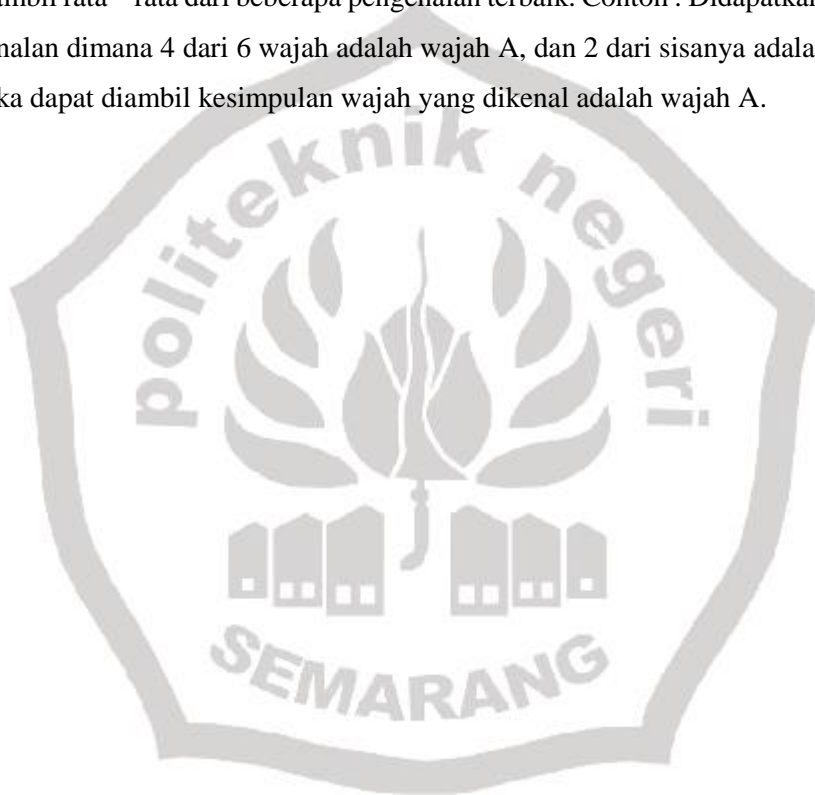
### KESIMPULAN

#### 5.1 Kesimpulan

1. Berdasarkan pengujian dan analisis pada Penerapan Sistem Absensi *Smartcam* Berbasis *Deep Learning*, dapat disimpulkan bahwa sistem yang saya buat sudah dapat digunakan, cahaya dan jarak dalam melakukan pengenalan, berpengaruh terhadap hasil, yaitu dapat menimbulkan kesalahan pada pengenalan wajah dan membuat data mengalami kesalahan dalam mengidentifikasi wajah maupun data sampel.
2. Penggunaan Algoritma *Haar Cascade Classifier* tidak direkomendasikan untuk pengenalan wajah secara outdoor, dimana kondisi outdoor seringkali memiliki pencahayaan yang dinamis (berbeda – beda). *EigenFace* baik digunakan jika memiliki data training yang lebih banyak, dimana pada pencahayaan dan pose yang sama, keberhasilan mencapai 90% pada data training sebesar 50 foto (50 foto 1 orang), dan 95% pada data training sebesar 100 foto (100 foto 1 orang).
3. Kegagalan yang terjadi adalah dikarenakan warna / pencahayaan yang berbeda memengaruhi Algoritma yang diinput dari hasil dekomposisi menggunakan software *Microsoft visual studio*, dimana distance akan berbeda jauh, dan gradien dari *Haar Cascade Algoritim* akan sangat berbeda, maka dari itu saya gunakan alat bantu yaitu *Luxmeter* sebagai analisa di sistem absen ini.
4. Penggunaan banyak foto untuk satu orang pun dapat sedikit mempengaruhi cahaya dan posisi foto sehingga data sampel akan terjadi kesalahan ketika ada kemiripan dalam cahaya dan posisi pose wajah.
5. Penggunaan lampu sorot juga bisa membantu dalam absensi, namun tidak menutup kemungkinan juga akan terjadi kesalahan, jadi solusinya harus pada ruangan dengan cahaya yang sedang atau normal serta sesuai dengan ruangan yang dipakai sebagai pengambilan data sampel.

## 5.2 Saran

1. Pada peletakan kamera harus juga diperhatikan, ketinggian kamera serta pencahayaannya dan tidak dihadapkan pada cahaya matahari, baik kamera masuk dan kamera keluar.
2. Untuk penelitian lanjutan dapat menggunakan kualitas kamera *webcam* yang di masa depan mungkin hadir lebih tinggi dari kamera *webcam* yang ada pada saat ini.
3. Penggunaan foto yang banyak pada satu orang dengan pose dan pencahayaan yang berbeda dapat mentolerir pengenalan wajah, dimana pengenalan wajah dapat mengambil rata – rata dari beberapa pengenalan terbaik. Contoh : Didapatkan 6 label pengenalan dimana 4 dari 6 wajah adalah wajah A, dan 2 dari sisanya adalah wajah B, maka dapat diambil kesimpulan wajah yang dikenal adalah wajah A.



## DAFTAR PUSTAKA

- Andrianus HendroTriatmoko, Sholeh Hadi Pramono dan Harry S. Dachlan. 2014. *Penggunaan Metode Viola-Jones dan Algoritma Eigen Eyes dalam Sistem Kehadiran Pegawai. EECCIS*. 8(1) : 42-43.
- C. Johnson. 2011. Creating Your First EMGU Image Processing Project. <https://www.codeproject.com/Articles/257502/Creating-Your-First-EMGU-Image-Processing-Project?fid=1653524&df=90&mpp=25&sort=Position&spc=Relaxed&prof=True&view=Normal>. 11 Februari 2019.
- Dedi Ary Prasetya Dan Imam Nurviyanto. 2012. Skripsi. *Deteksi Wajah Metode Viola Jones Pada Opencv Menggunakan Pemrograman Phytion*. Surakarta: Universitas Muhammadiyah.
- EMGU. 2018. Setting Up EMGU C Sharp. [www.emgu.com/wiki/index.php/Setting up EMGU C Sharp](http://www.emgu.com/wiki/index.php/Setting_up_EMGU_C_Sharp). 11 Februari 2019.
- Fika Tiara Puri. 2011. *Analisis Algoritma Eigenface (Pengenal Wajah) Pada Aplikasi Kehadiran Pengajaran Dosen*. Jakarta: Universitas Islam Negri Syarif Hidayatullah.
- Indra. 2012. *Sistem Pengenalan Wajah Dengan Metode Eigenface Untuk Absensi Pada Pt Florindo Lestari*. Jakarta: Universitas Budi Luhur.
- Mahvish Nasir. 2011. Live Multiple Face Detection. [fewtutorials.bravesites.com/entries/emgu-cv-c/level-3---live-face-detection](http://fewtutorials.bravesites.com/entries/emgu-cv-c/level-3---live-face-detection). 11 Februari 2019.
- M. Yogi Septian Dan Fitriyani. 2014. *Deteksi Wajah Menggunakan Metode Viola Jones Pada Graphics Processing Unit Face Detection Using Viola Jones Method On Graphics Processing Unit*. Universitas Telkom.

- Muhammad Abdul Rahim. 2013. Skripsi. *Perancangan Aplikasi Pengenalan Wajah Menggunakan Metode Viola Jones*. Medan: Stmik Budi Darma.
- Muhammad Rizki Muliawan, Beni Irawan, Yulrio Brianorman. 2015. Implementasi Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi. *Sistem Komputer Untan* 3(1) : 42.
- Nunu Kustian. 2016. *Principal Component Analysis Untuk Sistem Pengenalan Wajah Dengan Menggunakan Metode Eigenface*. *String* 1(2) : 194-195.
- Ratna Nur Azizah. 2008. Skripsi. *Pengenalan Wajah Dengan Metode Subspace Lda*.  
Surabaya: Its.
- Salamun, Firman Wazir. 2016. *Rancang Bangun Sistem Pengenalan Wajah Dengan Metode Principal Component Analysis*. *Teknologi dan Sistem Informasi UNIVRAB*. 1(2) : 60-70.
- Sepritahara. 2009. *Sistem Pengenalan Wajah (Face Recognition) Menggunakan Metode Hidden Markov Model (HMM)*. Jakarta: Universitas Indonesia.
- Sigit Wasista, Bima Sena Bayu D, Sandra Agustyan Putra. 2011. *Sistem Pengenalan Wajah Pada Mesin Absensi Mahasiswa Menggunakan Metode Pca Dan Dtw*. Surabaya: Its.

# LAMPIRAN



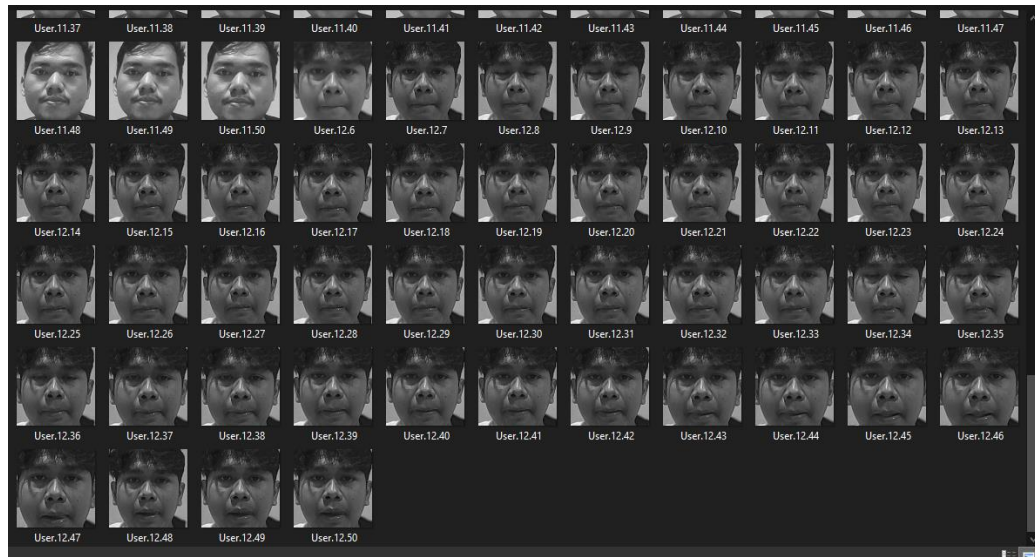
# LAMPIRAN

## LAMPIRAN 1

### GAMBAR DATA SAMPEL









## LAMPIRAN 2

### KONFIGURASI LIBRARY OPEN CV

```
sudo apt-get purge wolfram-  
enginesudo apt-get purge  
libreoffice* sudo apt-get  
clean  
sudo apt-get autoremove
```

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install build-essential cmake  
unzip pkg-config  
sudo apt-get install libjpeg-dev libpng-dev  
libtiff-devsudo apt-get install libavcodec-dev  
libavformat-dev libswscale-dev libv4l-dev  
sudo apt-get install libxvidcore-dev  
libx264-devsudo apt-get install libgtk-  
3-dev  
sudo apt-get install libcanberra-gtk*  
sudo apt-get install libatlas-base-dev gfortran  
sudo apt-get install python3-dev  
wget -O opencv.zip  
https://github.com/opencv/opencv/archive/4.2.0.zip  
wget -O opencv_contrib.zip  
https://github.com/opencv/opencv\_contrib/archiv  
e/4.2.0.zip  
  
unzip opencv.zip  
unzip opencv_contrib.zip
```

```
mv opencv-4.2.0 opencv

mv opencv_contrib-4.2.0 opencv_contrib

wget https://bootstrap.pypa.io/get-pip.py

sudo rm -rf ~/get-pip.py ~/.cache/pip

Buka source ~/.profile untuk persiapan
virtualenvironment

nano ~/.profile

# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs

export
VIRTUALENVWRAPPER_PYTHON=/usr/bin/python
3source
/usr/local/bin/virtualenvwrapper.sh

source ~/.profile

mkvirtualenv cv -p python3

workon cv

pip install numpy

cd ~/opencv

mkdir
build
cd
build

cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules
```

\

-D ENABLE\_NEON=ON \

-D ENABLE\_VFPV3=ON \

-D BUILD\_TESTS=OFF \

-D OPENCV\_ENABLE\_NONFREE=ON \

-D INSTALL\_PYTHON\_EXAMPLES=OFF \

-D BUILD\_EXAMPLES=OFF ..



### LAMPIRAN 3

#### OPEN CV STORAGE

```
<opencv_storage>

<cascade type_id="opencv-cascade-
classifier"><stageType>BOOST</stageType>

  <featureType>HAAR</featureType>

  <height>20</height>

  <width>20</width>

  <stageParams>
    <maxWeakCount>47</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount></featureParams>
  <stageNum>30</stageNum>
  <stages>
    <_>
      <maxWeakCount>5</maxWeakCount>

      <stageThreshold>-
1.6473180055618286e+00</stageThreshold>

      <weakClassifiers>

        <_>

          <internalNodes>

            2 1 0 -2.6987109333276749e-02 0 -1 1
5.0670530647039413e-02
```

-2 -3 2 -1.2915390729904175e-  
01</internalNodes>

<leafValues>

-8.0395472049713135e-01  
6.0491400957107544e-01

9.0544581413269043e-01  
4.4070810079574585e-02</leafValues></\_>

<\_>

<internalNodes>

2 1 3 8.8827736675739288e-02 0 -1 4 -  
2.0398240536451340e-02

-2 -3 5 -6.1261758208274841e-  
02</internalNodes>

<leafValues>

7.9218882322311401e-01  
4.0692299604415894e-02

4.2585361003875732e-01 -  
7.0325207710266113e-01</leafValues></\_>

<\_>

<internalNodes>

2 1 6 -2.0490810275077820e-01 0 -1 7  
9.4933047890663147e-02

-2 -3 8 1.2091030366718769e-  
03</internalNodes>

<leafValues>

```
-4.4017648696899414e-01
5.3640520572662354e-01

6.8776458501815796e-01 -
5.5879348516464233e-01</leafValues></_>

<_>

<internalNodes>

1 0 9 9.2227972345426679e-04 -1 2 10 -
7.2678289143368602e-04

-2 -3 11 6.8421510513871908e-
04</internalNodes>

<leafValues>

-7.2684401273727417e-01 -
5.8028000593185425e-01

5.6177532672882080e-01 -
2.9834181070327759e-01</leafValues></_>

<_>

<internalNodes>

0 1 12 -5.1150590181350708e-02 2 -1 13

6.1622060835361481e-02 -2 -3 14
7.2873473167419434e-02</internalNodes>

<leafValues>

5.9840762615203857e-01
7.4743932485580444e-01
```

```
-4.9703779816627502e-01
2.8129258751869202e-
01</leafValues></_></weakClassifiers></_>

<_>

<maxWeakCount>7</maxWeakCount>

<stageThreshold>-
1.4257860183715820e+00</stageThreshold>

<weakClassifiers>

<_>

<internalNodes>
2 1 15 -4.1994878649711609e-01 0 -1 16
-5.6186288595199585e-02 -2 -3 17 -
2.3711109533905983e-02</internalNodes>

<leafValues>
2.7586200833320618e-01 -
6.4623218774795532e-01
8.5241252183914185e-01
8.3703370764851570e-03</leafValues></_>

<_>

<internalNodes>
1 0 18 4.0523439645767212e-02 -1 2 19
2.7388900518417358e-01
-2 -3 20 -1.4293800108134747e-
02</internalNodes>

<leafValues>
```



7.4270218610763550e-01 -  
4.9286690354347229e-01

7.1784788370132446e-01 -  
4.2223978787660599e-02</leafValues></\_>

<\_>

<internalNodes>

0 1 21 -2.1144729107618332e-03 2 -1 22

1.0659949621185660e-03 -2 -3 23  
1.0812469990924001e-03</internalNodes>

<leafValues>

-8.0196601152420044e-01 -  
6.6025912761688232e-01  
4.7916370630264282e-01 -  
5.1645290851593018e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 24 3.0198289081454277e-02 2 -1 25  
4.0569551289081573e-02

-2 -3 26 7.0679739117622375e-  
02</internalNodes>

<leafValues>

5.1327562332153320e-01  
6.6641497611999512e-01  
-4.5298659801483154e-01  
5.5480718612670898e-01</leafValues></\_>

<\_>

<internalNodes>

0 1 27 -7.8928138827905059e-04 2 -1 28

8.0574717139825225e-04 -2 -3 29 -

2.0976560190320015e-02</internalNodes>

<leafValues>

-7.2526299953460693e-01 -

5.6479871273040771e-01

6.9993537664413452e-01

6.8500466644763947e-02</leafValues></\_>

<\_>

<internalNodes>

1 0 30 1.2794960290193558e-02 -1 2 31

-8.1120636314153671e-03 -2 -3 32 -

1.5506530180573463e-02</internalNodes>

<leafValues>

-8.6409568786621094e-01

4.4448360800743103e-01

3.6675310134887695e-01 -

2.9189071059226990e-01</leafValues></\_>

<\_>

<internalNodes>

2 1 33 -1.2915650382637978e-02 0 -1 34

6.6297221928834915e-03 -2 -3 35 -  
3.6532930098474026e-03</internalNodes>

<leafValues>

-4.7566780447959900e-01  
1.0350350290536880e-01

-6.1723059415817261e-01  
5.4382532835006714e-  
01</leafValues></\_></weakClassifiers></\_>

<\_>

<maxWeakCount>9</maxWeakCount>

<stageThreshold>-  
1.4711019992828369e+00</stageThreshold>

<weakClassifiers>

<\_>

<internalNodes>

0 1 36 -7.8731971979141235e-01 -1 2 37  
1.6908009350299835e-01 -2 -3 38 -  
4.0369689464569092e-02</internalNodes>

<leafValues>

7.1268838644027710e-01 -  
7.1908998489379883e-01

4.4148930907249451e-01 -  
4.2251929640769958e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 39 1.9132360816001892e-02 2 -1 40  
6.4184539951384068e-04

-2 -3 41 -7.8941037645563483e-  
04</internalNodes>

<leafValues>

6.9186228513717651e-01 -  
7.6116967201232910e-01

-6.8140429258346558e-01  
1.6009919345378876e-01</leafValues></\_>

<\_>

<internalNodes>

1 2 42 -7.1503049694001675e-03 0 -1 43  
-2.3156129755079746e-03 -2 -3 44 -  
4.1521269828081131e-02</internalNodes>

<leafValues>

-5.5916607379913330e-01  
5.1284497976303101e-01

2.4422569572925568e-01 -  
4.6883401274681091e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 45 9.1200548922643065e-04 -1 2 46  
-1.5798299573361874e-03 -2 -3 47 -  
1.1573649942874908e-02</internalNodes>

<leafValues>

-6.9527888298034668e-01 -  
6.3509649038314819e-01  
6.4686381816864014e-01  
6.9198559504002333e-04</leafValues></\_>

<\_>

<internalNodes>

2 1 48 2.1843519061803818e-03 0 -1 49  
2.9345690272748470e-03

-2 -3 50 -5.8788150548934937e-  
02</internalNodes>

<leafValues>

4.5632898807525635e-01 -  
5.8841437101364136e-01  
2.6704201102256775e-01 -  
3.8348990678787231e-01</leafValues></\_>

<\_>

<internalNodes>

0 1 51 -5.5392808280885220e-04 -1 2 52  
-5.3035060409456491e-04 -2 -3 53 -  
6.8775108084082603e-03</internalNodes>

<leafValues>

-4.8913368582725525e-01 -  
3.8421550393104553e-01  
6.6845697164535522e-01  
9.3158259987831116e-02</leafValues></\_>

<\_>

<internalNodes>

1 0 54 1.6710379859432578e-03 2 -1 55  
1.4162790030241013e-03

-2 -3 56 7.7876187860965729e-  
03</internalNodes>

<leafValues>

-6.0369372367858887e-01 -  
3.0418768525123596e-01  
3.9699068665504456e-01 -  
6.6687589883804321e-01</leafValues></\_>

<\_>

<internalNodes>

1 2 57 -1.2916780076920986e-02 0 -1 58  
-3.0156269203871489e-03 -2 -3 59 -  
1.9785940647125244e-02</internalNodes>

<leafValues>

-7.1239727735519409e-01  
4.6252989768981934e-01  
2.8338319063186646e-01 -  
3.5317930579185486e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 60 3.3207770902663469e-03 2 -1 61  
2.9606239870190620e-02



-2 -3 62 4.4614788144826889e-  
02</internalNodes>

<leafValues>

-7.3291397094726562e-01  
4.9530759453773499e-01

-1.9502809643745422e-01  
7.9816418886184692e-  
01</leafValues></\_></weakClassifiers></\_>

<\_>

<maxWeakCount>12</maxWeakCount>

<stageThreshold>-  
1.3850779533386230e+00</stageThreshold>

<weakClassifiers>

<\_>

<internalNodes>

0 1 63 -9.2366141080856323e-01 2 -1 64  
-4.8193939030170441e-02 -2 -3 65  
2.8669878840446472e-01</internalNodes>

<leafValues>

7.6915800571441650e-01 -  
5.1361227035522461e-01

-2.9671901464462280e-01  
6.2028187513351440e-01</leafValues></\_>

<\_>

<internalNodes>

```
1 2 66 -1.3038160279393196e-02 0 -1 67  
-1.4749659458175302e-03 -2 -3 68 -  
4.6921748667955399e-02</internalNodes>
```

```
<leafValues>  
-7.1294248104095459e-01  
5.9115177392959595e-01  
3.1303560733795166e-01 -  
3.6749690771102905e-01</leafValues></_>
```

```
<_>  
<internalNodes>  
0 1 69 2.4459899868816137e-03 -1 2 70  
-2.5321498978883028e-03 -2 -3 71  
1.4651260571554303e-03</internalNodes>
```

```
<leafValues>  
-4.6930000185966492e-01 -  
7.7450162172317505e-01  
3.6414781212806702e-01 -  
5.7445889711380005e-01</leafValues></_>
```

```
<_>  
<internalNodes>  
1 2 72 -1.1307420209050179e-02 0 -1 73  
-1.2048849603161216e-03 -2 -3 74 -  
6.2752872705459595e-02</internalNodes>
```

```
<leafValues>
```

```
-5.5727648735046387e-01
4.7871670126914978e-01

2.2788530588150024e-01 -
4.3667969107627869e-01</leafValues></_>

<_>

<internalNodes>

0 1 75 -4.0173111483454704e-03 2 -1 76

1.5160309849306941e-03 -2 -3 77
1.9954680465161800e-03</internalNodes>

<leafValues>

-7.3568779230117798e-01 -
5.8480697870254517e-01

2.1544020622968674e-02
5.5875688791275024e-01</leafValues></_>

<_>

<internalNodes>

1 0 78 3.4435209818184376e-03 -1 2 79

-2.6550020556896925e-03 -2 -3 80 -
1.1407690122723579e-02</internalNodes>

<leafValues>

-7.6565897464752197e-01 -
6.5447497367858887e-01

5.3633081912994385e-01 -
3.8849171251058578e-02</leafValues></_>

<_>
```

```
<internalNodes>

1 2 81 -2.3805440869182348e-03 0 -1 82

6.6475258208811283e-03 -2 -3 83
1.4018240571022034e-01</internalNodes>
```

```
<leafValues>

3.3984410762786865e-01 -
6.5025091171264648e-01

-3.2491090893745422e-01
7.5067067146301270e-01</leafValues></_>
```

```
<_>

<internalNodes>

0 1 84 -6.2358360737562180e-02 2 -1 85

1.3628599699586630e-03 -2 -3 86 -
4.4609848409891129e-03</internalNodes>
```

```
<leafValues>

4.5777168869972229e-01 -
6.3202661275863647e-01

4.0597960352897644e-01 -
2.0854069292545319e-01</leafValues></_>
```

```
<_>

<internalNodes>

0 1 87 -1.0046839714050293e-02 2 -1 88

-2.9274819418787956e-02 -2 -3 89
7.7389390207827091e-03</internalNodes>
```

```
<leafValues>
    -7.4789828062057495e-01 -
1.7995479702949524e-01
    4.7782841324806213e-01 -
6.5113341808319092e-01</leafValues></_>
```

<\_>

```
<internalNodes>
    1 0 90 1.4774020528420806e-03 -1 2 91
1.4989820308983326e-02
    -2 -3 92 4.5073241926729679e-
03</internalNodes>
```

```
<leafValues>
    -6.6269898414611816e-01 -
1.6695550084114075e-01
    3.8702058792114258e-01 -
7.3409372568130493e-01</leafValues></_>
```

<\_>

```
<internalNodes>
    1 0 93 1.4901049435138702e-03 2 -1 94
8.9141662465408444e-04
    -2 -3 95 -1.1558219790458679e-
02</internalNodes>
```

```
<leafValues>
    -3.4280839562416077e-01 -
2.8036740422248840e-01
```

```
-4.2523959279060364e-01
4.5259669423103333e-01</leafValues></_>

<_>

<internalNodes>

    0 1 96 -2.0011950284242630e-02 -1 2 97

    -1.7092300578951836e-02 -2 -3 98 -
6.7685171961784363e-02</internalNodes>

<leafValues>

    4.0133118629455566e-01
3.6970010399818420e-01

    7.4438679218292236e-01 -
3.8255840539932251e-
01</leafValues></_></weakClassifiers></_>

<_>

<maxWeakCount>12</maxWeakCount>

<stageThreshold>-
1.4432040452957153e+00</stageThreshold>

<weakClassifiers>

<_>

<internalNodes>

    1 2 99 -2.0911149680614471e-02 0 -1
100

    1.4305709302425385e-01 -2 -3 101
1.1925029568374157e-02</internalNodes>

<leafValues>
```



```
-3.4965568780899048e-01
7.0134562253952026e-01

-6.0404628515243530e-01
8.5615903139114380e-02</leafValues></_>

<_>

<internalNodes>

1 0 102 2.4742009118199348e-02 2 -1
103

4.5732118189334869e-02 -2 -3 104
4.3204430490732193e-02</internalNodes>

<leafValues>

8.5365587472915649e-01
4.1876411437988281e-01

-3.9094918966293335e-01
2.7387988567352295e-01</leafValues></_>

<_>

<internalNodes>

0 1 105 -7.2548422031104565e-04 2 -1
106

1.4243220211938024e-03 -2 -3 107 -
5.3335479460656643e-03</internalNodes>

<leafValues>

-6.2011122703552246e-01 -
6.1589437723159790e-01
```

6.0596448183059692e-01  
1.5840480104088783e-02</leafValues></\_>

<\_>

<internalNodes>

1 0 108 -7.1891010738909245e-03 2 -1

109

1.8233320442959666e-03 -2 -3 110  
1.6109029529616237e-03</internalNodes>

<leafValues>

-2.0852829515933990e-01 -  
8.1338381767272949e-01  
5.6780648231506348e-01 -  
8.7046259641647339e-01</leafValues></\_>

<\_>

<internalNodes>

2 1 111 -4.8350278288125992e-02 0 -1

112

3.1746171414852142e-02 -2 -3 113  
1.9233829807490110e-03</internalNodes>

<leafValues>

-3.5335820913314819e-01  
4.4076570868492126e-01  
4.0730631351470947e-01 -  
5.9592568874359131e-01</leafValues></\_>

<\_>

```
<internalNodes>

1 0 114 1.3614529743790627e-03 -1 2
115

-3.6934199742972851e-03 -2 -3 116 -
8.5378461517393589e-04</internalNodes>

<leafValues>

-5.5307251214981079e-01 -
7.3163098096847534e-01

4.3890678882598877e-01 -
6.3009172677993774e-02</leafValues></_>

<_>
<internalNodes>

0 1 117 -1.0950770229101181e-02 -1 2
118

-7.2186449542641640e-03 -2 -3 119
1.8548289313912392e-02</internalNodes>

<leafValues>

3.9263078570365906e-01
2.7225250005722046e-01

-4.1208618879318237e-01
6.3790637254714966e-01</leafValues></_>

<_>

<internalNodes>

1 0 120 1.0859060566872358e-03 -1 2
121
```

-6.5618362277746201e-03 -2 -3 122 -  
6.1777420341968536e-02</internalNodes>

<leafValues>

-5.0857210159301758e-01  
3.5386729240417480e-01

5.7568281888961792e-01 -  
2.8477248549461365e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 123 4.9480778397992253e-04 2 -1  
124  
1.1606880463659763e-02 -2 -3 125 -  
1.6142609529197216e-03</internalNodes>

<leafValues>

-4.9583891034126282e-01 -  
5.1320201158523560e-01  
5.2665728330612183e-01  
3.0917160212993622e-02</leafValues></\_>

<\_>

<internalNodes>

1 0 126 2.0437680650502443e-03 -1 2  
127  
-8.2394909113645554e-03 -2 -3 128 -  
3.9699211716651917e-02</internalNodes>

<leafValues>

```
-7.0948588848114014e-01
3.4189811348915100e-01

4.7383341193199158e-01 -
2.5060850381851196e-01</leafValues></_>

<_>

<internalNodes>

1 2 129 -8.0377282574772835e-04 0 -1
130

-5.4273242130875587e-03 -2 -3 131 -
5.2662738598883152e-03</internalNodes>

<leafValues>

-5.1384007930755615e-01
2.9752710461616516e-01

1.4577029645442963e-01 -
4.6007528901100159e-01</leafValues></_>

<_>

<internalNodes>

1 0 132 6.3841522205621004e-04 -1 2
133

-1.5458120033144951e-03 -2 -3 134
1.1863360414281487e-03</internalNodes>

<leafValues>

-3.6412829160690308e-01 -
5.8081609010696411e-01
```

```
2.9298609495162964e-01 -  
5.1420718431472778e-  
01</leafValues></_></weakClassifiers></_>
```

```
<_>
```

```
<maxWeakCount>12</maxWeakCount>
```

```
<stageThreshold>-
```

```
1.5415630340576172e+00</stageThreshold>
```

```
<weakClassifiers>
```

```
<_>
```

```
<internalNodes>
```

```
1 2 135 -2.7745011448860168e-01 0 -1  
136  
-3.1200000084936619e-03 -2 -3 137 -  
8.0280922353267670e-02</internalNodes>
```

```
<leafValues>
```

```
8.3265638351440430e-01  
1.0233189910650253e-01  
2.3773579299449921e-01 -  
6.4546662569046021e-01</leafValues></_>
```

```
<_>
```

```
<internalNodes>
```

```
0 1 138 -6.9391548633575439e-02 2 -1  
139  
5.3355181589722633e-03 -2 -3 140 -  
5.4189618676900864e-02</internalNodes>
```



```
<leafValues>
    4.6008241176605225e-01
2.9137989878654480e-01
    4.7026729583740234e-01 -
5.7723402976989746e-01</leafValues></_>
```

```
<_>
<internalNodes>
    1 0 141 1.8562959507107735e-02 -1 2
142
    4.6305730938911438e-02 -2 -3 143 -
8.8262781500816345e-03</internalNodes>
```

```
<leafValues>
    7.0555502176284790e-01 -
5.2839881181716919e-01
    4.3953609466552734e-01 -
1.3887490332126617e-01</leafValues></_>
```

```
<_>
<internalNodes>
    1 0 144 -2.8772179502993822e-03 -1 2
145
    -2.6457069907337427e-03 -2 -3 146
3.3441530540585518e-03</internalNodes>
```

```
<leafValues>
    -2.7475830912590027e-01 -
5.7746797800064087e-01
```

3.6615240573883057e-01 -  
6.3586741685867310e-01</leafValues></\_>

<\_>

<internalNodes>

2 1 147 -8.3742372691631317e-02 0 -1

148

1.0164769738912582e-01 -2 -3 149 -  
2.1541758906096220e-03</internalNodes>

<leafValues>

-2.9664519429206848e-01

5.6140047311782837e-01

-7.5446271896362305e-01

3.9601260423660278e-01</leafValues></\_>

<\_>

<internalNodes>

0 1 150 -1.7133549554273486e-03 2 -1

151

1.3899410143494606e-02 -2 -3 152 -  
2.8498120605945587e-02</internalNodes>

<leafValues>

-7.3741632699966431e-01

4.8247390985488892e-01

4.1971048712730408e-01 -

2.0021289587020874e-01</leafValues></\_>

<\_>

```
<internalNodes>
0 1 153 -4.9728769809007645e-03 2 -1
154
-3.4751880913972855e-02 -2 -3 155 -
8.7171117775142193e-04</internalNodes>
```

```
<leafValues>
3.7631350755691528e-01 -
4.4797790050506592e-01
-6.9995099306106567e-01
1.5640909969806671e-01</leafValues></_>
```

```
<_>
<internalNodes>
0 1 156 -3.3666230738162994e-03 -1 2
157
-2.1378830075263977e-02 -2 -3 158 -
1.1869249865412712e-02</internalNodes>
```

```
<leafValues>
-6.7721921205520630e-01
3.3951529860496521e-01
5.4050672054290771e-01 -
2.4071580171585083e-01</leafValues></_>
```

```
<_>
<internalNodes>
0 1 159 -4.4268160127103329e-03 2 -1
160
```

4.1405398398637772e-02 -2 -3 161 -  
3.7884410470724106e-02</internalNodes>

<leafValues>

-7.3965507745742798e-01  
8.2905638217926025e-01

1.7030739784240723e-01 -  
2.4498699605464935e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 162 3.7567419349215925e-04 -1 2  
163  
-3.7140299100428820e-03 -2 -3 164 -  
6.1806719750165939e-03</internalNodes>

<leafValues>

-4.5103698968887329e-01  
3.8348129391670227e-01

3.6097520589828491e-01 -  
2.0644439756870270e-01</leafValues></\_>

<\_>

<internalNodes>

0 1 165 -1.2373559875413775e-03 -1 2  
166  
-2.1339580416679382e-03 -2 -3 167  
2.8985869139432907e-03</internalNodes>

<leafValues>

```

-5.8166879415512085e-01
4.1669690608978271e-01

-2.4721260368824005e-01
3.5056841373443604e-01</leafValues></_>

<_>

<internalNodes>

1 2 168 -4.4636861421167850e-03 0 -1
169

1.6411510296165943e-03 -2 -3 170 -
7.3051019571721554e-03</internalNodes>

<leafValues>

3.5625410079956055e-01 -
4.1040098667144775e-01

2.0216129720211029e-01 -
3.4234520792961121e-
01</leafValues></_></weakClassifiers></_>

<_>

<maxWeakCount>13</maxWeakCount>

<stageThreshold>-
1.4762729406356812e+00</stageThreshold>

<weakClassifiers>

<_>

<internalNodes>

1 2 171 -5.1942609250545502e-02 0 -1
172

```

-4.7268528491258621e-02 -2 -3 173 -  
7.8969672322273254e-03</internalNodes>

<leafValues>

8.8198930025100708e-01  
6.4829237759113312e-02

8.8662758469581604e-02 -  
5.9007811546325684e-01</leafValues></\_>

<\_>

<internalNodes>

1 0 174 9.0199249098077416e-04 2 -1  
175

-1.7289820313453674e-01 -2 -3 176 -  
2.3374119773507118e-03</internalNodes>

<leafValues>

5.9040898084640503e-01 -  
5.2029031515121460e-01

5.2981728315353394e-01 -  
1.4985850453376770e-01</leafValues></\_>

<\_>

<internalNodes>

0 1 177 -1.7534950748085976e-02 -1 2  
178

5.8875310060102493e-05 -2 -3 179 -  
3.2241028547286987e-01</internalNodes>

<leafValues>

5.3269028663635254e-01 -  
4.5709720253944397e-01  
5.7380169630050659e-01 -  
1.2866480648517609e-01</leafValues></\_>

<\_>

<internalNodes>

1 2 180 8.3220787928439677e-05 0 -1  
181

-1.1180160072399303e-04 -2 -3 182 -  
1.0344980284571648e-02</internalNodes>

<leafValues>

9.0006209909915924e-02 -  
5.6352388858795166e-01  
6.3273417949676514e-01  
5.0064269453287125e-02</leafValues></\_>

<\_>

<internalNodes>

0 1 183 -9.4440882094204426e-04 2 -1  
184

-3.7474210839718580e-03 -2 -3 185  
4.0574651211500168e-03</internalNodes>

<leafValues>

4.4386640191078186e-01 -  
3.4999918937683105e-01



-4.5298218727111816e-01  
3.0920198559761047e-01</leafValues></\_>

<\_>

<internalNodes>

1 2 186 5.5205920943990350e-05 0 -1  
187

-7.5678288936614990e-02 -2 -3 188 -  
3.0975368618965149e-01</internalNodes>

<leafValues>

3.5544091463088989e-01 -  
3.6047360301017761e-01  
-6.4954018592834473e-01  
3.0679279565811157e-01</leafValues></\_>

<\_>

<internalNodes>

1 2 189 -7.9595847637392581e-05 0 -1  
190

4.0613119490444660e-03 -2 -3 191  
4.3240871280431747e-02</internalNodes>

<leafValues>

3.3850470185279846e-01 -  
5.3271901607513428e-01  
-3.2592329382896423e-01  
5.5076271295547485e-01</leafValues></\_>

<\_>

```
<internalNodes>
    0 1 192 -6.7015928216278553e-03 -1 2
193
    -1.0451120324432850e-03 -2 -3 194
8.3967261016368866e-03</internalNodes>
```

```
<leafValues>
    5.0109171867370605e-01 -
5.8881980180740356e-01
    -9.5237597823143005e-02
5.6516999006271362e-01</leafValues></_>
```

```
<_>
    <internalNodes>
    2 1 195 -6.5531006839592010e-05 0 -1
196
    7.8218057751655579e-05 -2 -3 197
3.2988168299198151e-02</internalNodes>
```

```
<leafValues>
    -4.6556711196899414e-01
5.4509781301021576e-02
    3.5248789191246033e-01 -
5.2722948789596558e-01</leafValues></_>
```

```
<_>
    <internalNodes>
    0 1 198 -1.4161449857056141e-02 2 -1
199
```

3.1500440090894699e-02 -2 -3 200 -  
2.1956730633974075e-03</internalNodes>

<leafValues>

3.6811780929565430e-01  
5.2040421962738037e-01

1.1603529751300812e-01 -  
3.0985280871391296e-01</leafValues></\_>

<\_>

<internalNodes>



## LAMPIRAN 4

### ALGORITMA HAAR CASCADE CLASIFIER

#### 1. Face Training

```
import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector =
cv2.CascadeClassifier("assets/haarcascade_frontalface_al
t2.xml");
# function to get the images and label data
def getImagesAndLabels(path):
    imagePaths = [os.path.join(path,f) for f in
os.listdir(path)]
    faceSamples=[]
    ids = []
    for imagePath in imagePaths:
        PIL_img = Image.open(imagePath).convert('L') #
convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.split(imagePath)[-
1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy,
                                         scaleFactor=1.2
                                         ,
                                         minNeighbors=5
                                         ,
```

```

minSize=(60,
60),

flags=cv2.CASC
ADE_SCALE_IMAGE)
    # faces = detector.detectMultiScale(img_numpy,
1.2, 5)
    for (x,y,w,h) in faces:
        faceSamples.append(img_numpy[y:y+h,x:x+w])
        ids.append(id)
    return faceSamples,ids
print ("\n [INFO] Training faces. It will take a few
seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))
# Save the model into trainer/trainer.yml
if os.path.exists("trainer/trainer.yml"):
    os.remove("trainer/trainer.yml")
recognizer.write('trainer/trainer.yml') #
recognizer.save() worked on Mac, but not on Pi
# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting
Program".format(len(np.unique(ids))))

```

## 2. Face Dataset

```
import cv2
import os
import sys

faceCascade =
cv2.CascadeClassifier('assets/haarcascade_frontalface_alex
lt2.xml')
video_capture = cv2.VideoCapture(1)
face_id = sys.argv[1]
count = 5
while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
                                         scaleFactor=1.
2,
                                         minNeighbors=5
,
                                         minSize=(60,
60),
                                         flags=cv2.CASC
ADE_SCALE_IMAGE)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y +
h), (0,255,0), 2)
        count += 1
        cv2.imwrite("dataset/User." + str(face_id) +
'.' + str(count) + ".jpg", gray[y:y+h,x:x+w])
        # cv2.imwrite("dataset/User." + str(face_id) +
'.' + str(count) + ".jpg", gray)
        cv2.imshow('Video', frame)
```

```
        if cv2.waitKey(1) & 0xFF == ord('q'):  
            break  
        elif count >= 50: # Take 30 face sample and stop  
video  
            break  
video_capture.release()  
cv2.destroyAllWindows()
```





### 3. Face Recognition

```
import cv2
import numpy as np
import os
from datetime import datetime

import tkinter as tki # untuk tampilan GUI
from tkinter import Frame # untuk frame pada GUI
import tkinter.messagebox as tkMessageBox # message box

import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred =
credentials.Certificate('assets/serviceAccountKey.json'
)

# Initialize the app with a custom auth variable,
limiting the server's access
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://absensi-camera-default-
rtadb.firebaseio.com/'
})

# The app only has access as defined in the Security
Rules
ref = db.reference('/absensi')

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "assets/haarcascade_frontalface_alt2.xml"
```

```
# cascadePath =
"assets/haarcascade_frontalface_default.xml"
eyeDetector =
cv2.CascadeClassifier('assets/haarcascade_eye.xml')
smileCascade =
cv2.CascadeClassifier('assets/haarcascade_smile.xml')
faceCascade = cv2.CascadeClassifier(cascadePath)
font = cv2.FONT_HERSHEY_SIMPLEX
#iniciate id counter
id = 0
# names related to ids: example ==> Marcelo: id=1, etc
names = ['Almas', 'M. Tomi', 'Afrizal Salfarel',
'Agan']
nims = ['123428352348', '4.31.18.0.13', '4.39.18.0.01',
'4.39.18.0.02']
kelass = ['Curut', 'TE-4A', 'TE-4C', 'Wirog']
# Initialize and start realtime video capture
cam = cv2.VideoCapture(1)
cam.set(3, 1366) # set video width
cam.set(4, 768) # set video height
# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)
while True:
    ret, img =cam.read()
    # img = cv2.flip(img, -1) # Flip vertically
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray,
                                         scaleFactor=1.
                                         2,
                                         minNeighbors=5
                                         ,
                                         minSpaceBetweenComponents=0.5)
    # Draw rectangles around the faces
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        # Put the id number on the rectangle
        id = id+1
        name = names[id-1]
        nim = nims[id-1]
        kelas = kelass[id-1]
        cv2.putText(img,name,(x+w,y+h),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,0,0))
        cv2.putText(img,nim,(x+w,y+h),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,0,0))
        cv2.putText(img,kelas,(x+w,y+h),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,0,0))
    # Display the image
    cv2.imshow('Image',img)
    # Wait for a key to be pressed
    if cv2.waitKey(1) && ord('q') == ord('q'):
        break
# Destroy all windows
cv2.destroyAllWindows()
# Release the camera
cam.release()
# Close all windows
cv2.destroyAllWindows()
```

```

minSize=(int (m
inW), int (minH)),

flags=cv2.CASC
ADE_SCALE_IMAGE)
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0),
2)

        id, confidence =
recognizer.predict(gray[y:y+h,x:x+w])
        eyes = eyeDetector.detectMultiScale(gray)
        for (xe, ye, we, he) in eyes:
            cv2.rectangle(img, (xe,ye), (xe+we, ye+he),
(0,255,255), 2)

            # Check if confidence is less them 100 ==> "0"
is perfect match
            nama = names[id]
            nim = nims[id]
            kelas = kelas[id]
            if (confidence <= 57):
                id = names[id]
                confidence = " {0}%".format(round(100 -
confidence))
            else:
                id = "unknown"
                confidence = " {0}%".format(round(100 -
confidence))

            cv2.putText(img, str(id), (x+5,y-5), font, 1,
(255,255,255), 2)
            cv2.putText(img, str(confidence), (x+5,y+h-5),
font, 1, (255,255,0), 1)

            smile = smileCascade.detectMultiScale(

```

```

        gray,
        scaleFactor= 1.5,
        minNeighbors=15,
        minSize=(25, 25),
        )

    for (xx, yy, ww, hh) in smile:
        dd, confidences =
recognizer.predict(gray[y:y+h,x:x+w])
        cv2.putText(img, "smileeeee", (x+5,y+h-5),
font, 1, (255,255,0), 1)
        if (confidences <= 50 and id != 'unknown'):
            msgBox =
tkMessageBox.askquestion("Verifikasi", "Data Anda Akan
Dikirim !")
            if msgBox == 'yes':
                waktu = datetime.now()
                current_time = waktu.strftime("%Y-
%m-%d %H:%M:%S")
                # ref.push({'nama': nama, 'nim':
nim, 'kelas': kelas, 'waktu': current_time})
                ref.push({'waktu': current_time})

                cam.release()
                cv2.destroyAllWindows()
                break
            elif msgBox == 'no':

                cam.release()
                cv2.destroyAllWindows()
                break

```

```
cv2.imshow('camera',img)
k = cv2.waitKey(10) & 0xff # Press 'ESC' for
exiting video
if k == 27:
    break
# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```



#### 4. Portal Absen

```
import sys
import os
from tkinter import *
from tkinter import ttk
from PIL import ImageTk, Image

#tampilan window
window=Tk()

#judul
window.title("Absen Kelas")
window.geometry('500x500')

#wadah gambar
canvas = Canvas(window, width=175, height=175)
canvas.pack()

#gambar
img = (Image.open("polines.png"))

#atur gambar
resized_image = img.resize((160,160), Image.ANTIALIAS)
new_image = ImageTk.PhotoImage(resized_image)

canvas.create_image(10,10,anchor=NW,image=new_image)

#Label
labell = Label(window, text = "Portal Absen Mahasiswa
Politeknik Negeri Semarang", font=11)
labell.pack(pady=5, padx=2)

#form absen
```

```
#perintah menjalankan aplikasi absen
def run():
    os.system('face_recognition.py')
#tombol mulai absen
btn = Button(window, text="Absen Sekarang !",
width=45,height=2,font=11, command=run)
btn.pack(pady=30, padx=2)

window.mainloop()
```

