

**RANCANG BANGUN SISTEM PENGENALAN WAJAH  
MENGUNAKAN *METODE HAAR CASCADE CLASSIFIER* DAN *LOCAL  
BINARY PATTERN HISTOGRAM (LBPH)* PADA AKSES MASUK  
RUANG DOSEN**

**LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH:**

**AHMAD JALALUDDIN**

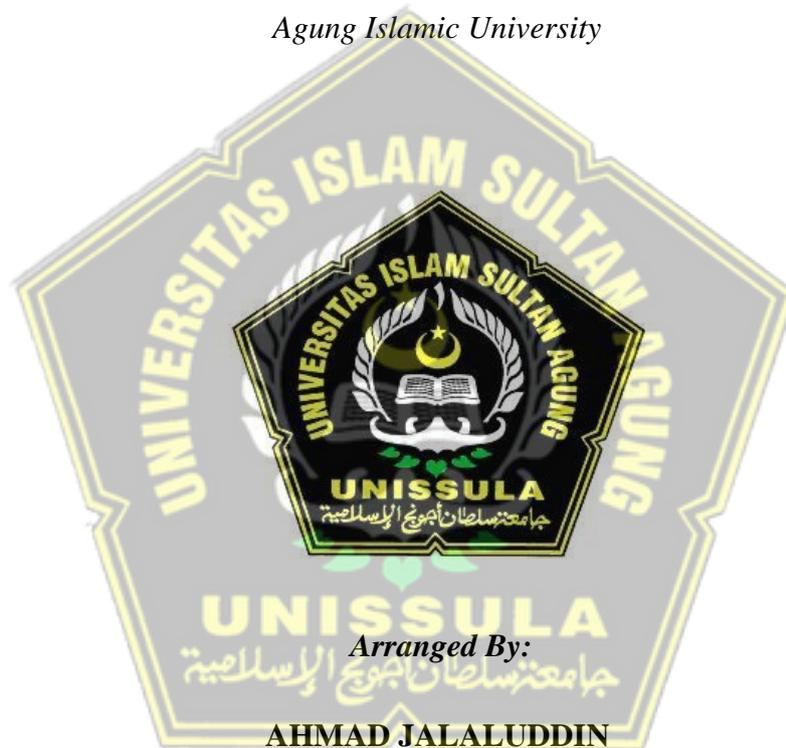
**NIM 32601700004**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG  
2023**

***DESIGN OF FACE RECOGNITION SYSTEM USING HAAR CASCADE  
CLASSIFIER AND LOCAL BINARY PATTERN HISTOGRAM (LBPH)  
METHOD FOR ACCESS TO LECTURER ROOMS***

***FINAL PROJECT REPORT***

*Proposed to complete the requirement to obtain a bachelor's degree (S-1) at  
Informatics Engineering Departement of Industrial Technology Faculty Sultan  
Agung Islamic University*



**NIM 32601700004**

***MAJORING OF INFORMATICS ENGINEERING  
INDUSTRIAL TECHNOLOGY FACULTY  
SULTAN AGUNG ISLAMIC UNIVERSITY  
SEMARANG***

**2023**

**LEMBAR PENGESAHAN PEMBIMBING**

Laporan Tugas Akhir dengan judul **“Rancang Bangun Sistem Pengenalan Wajah menggunakan Metode Haar Cascade Classifier dan Local Binary Pattern Histogram (LBPH) Pada Akses Masuk Ruang Dosen”** ini disusun oleh :

Nama : Ahmad Jalaluddin

NIM : 32601700004

Program Studi : Teknik Informatika

Telah disetujui oleh Dosen Pembimbing pada :

Hari : Selasa

Tanggal : 25 Agustus 2023

Mengesahkan,

Pembimbing I

Pembimbing II

  
Badriyah S.T., M.Kom  
NIDN. 0619018701

  
Ir. Sri Mulyono, M.Eng  
NIDN. 0626066601

Mengetahui,

Ketua Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Islam Sultan Agung



  
Ir. Sri Mulyono, M.Eng  
NIDN. 0626066601

## LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “Rancang Bangun Sistem Pengenalan Wajah menggunakan *Metode Haar Cascade Classifier* dan *Local Binary Pattern Histogram (LBPH)* Pada Akses Masuk Ruang Dosen” ini telah dipertahankan di depan tim penguji Tugas Akhir pada :

Hari : Selasa

Tanggal : 25 Agustus 2023

Ketua Penguji

Anggota I

  
Imam Much Ibnu Subroto, ST, M.Sc, Ph.D  
NIDN.0613037301

  
Moch Taufik, ST, MIT  
NIDN.0622037502

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : AHMAD JALALUDDIN

NIM : 32601700004

Judul Tugas Akhir : RANCANG BANGUN SISTEM PENGENALAN  
WAJAH MENGGUNAKAN METODE HAAR  
CASCADE CLASSIFIER DAN LOCAL BINARY  
PATTERN HISTOGRAM PADA AKSES  
MASUK RUANG DOSEN

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (SI) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 25 Agustus 2023

Yang Menyatakan,



Ahmad Jalaluddin

## PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Ahmad Jalaluddin

NIM : 32601700004

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Desa Tiromanda, Kec. Bua, Kab. Luwu, Prov. Sulawesi Selatan

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :

Rancang Bangun Sistem Pengenalan Wajah Menggunakan Metode Haar Cascade dan Local Binary Pattern Histogram (LBPH) Pada Akses Masuk Ruang Dosen. Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta.

Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 25 Agustus 2013

Yang menyatakan,



Ahmad Jalaluddin

## KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Pengenalan Wajah Menggunakan *Metode Haar Cascade Classifier* dan *Local Binary Pattern Histogram* pada Akses Masuk Ruang Dosen” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, ST., MT.
3. Dosen pembimbing I penulis Badie'ah, ST., M.Kom. yang telah meluangkan waktu dan memberi ilmu.
4. Dosen pembimbing II penulis Ir. Sri Mulyono, M.Eng. yang telah memberikan banyak nasehat dan saran.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini.
6. Dan kepada semua pihak yang tidak dapat saya satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat banyak kekurangan-kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapakan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini.

Semarang, 22 Agustus 2023



Ahmad Jalaluddin

## DAFTAR ISI

RANCANG BANGUN SISTEM PENGENALAN WAJAH MENGGUNAKAN <i>METODE HAAR CASCADE CLASSIFIER</i> DAN <i>LOCAL BINARY PATTERN HISTOGRAM (LBPH)</i> PADA AKSES MASUK RUANG DOSEN.....	i
LEMBAR PENGESAHAN PEMBIMBING .....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
ABSTRAK.....	xiv
<i>ABSTRACT</i> .....	xiv
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan penelitian .....	2
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II.....	5
TINJAUAN PUSTAKA DAN DASAR TEORI .....	5

2.1	Tinjauan Pustaka .....	5
2.2	Dasar Teori .....	7
2.2.1	<i>Face Recognition</i> .....	7
2.2.2	Algoritma <i>Haarcascade Classifier</i> .....	9
2.2.3	Algoritma <i>Local Binary Pattern Histogram</i> .....	12
2.2.4	<i>Library OpenCV</i> .....	16
2.2.5	<i>Anaconda</i> .....	16
2.2.6	<i>Python</i> .....	17
BAB III .....		18
METODE PENELITIAN.....		18
3.1	Deskripsi Sistem.....	18
3.2	Kebutuhan sistem .....	19
3.2.1	Logitech C270 HD Webcam.....	19
3.2.2	<i>Jetson Nano</i> .....	20
3.2.3	<i>OpenCV</i> .....	21
3.3	Metode penelitian .....	21
3.3.1	Kumpulkan data <i>Training</i> .....	23
3.3.2	<i>Dataset Training</i> proses .....	24
3.3.3	Proses <i>Face Recognition</i> .....	24
3.4	Desain Sistem .....	25
3.4.1	<i>Flowchart</i> pada sistem .....	25
3.4.2	<i>Use case</i> sistem .....	28
3.4.3	Desain GUI.....	29
3.5	Konfigurasi sistem dan instalasi sistem.....	30
3.5.1	Konfigurasi sistem .....	30

3.5.2	Instalasi sistem .....	30
3.6	Pengujian metode akurasi sistem.....	31
3.6.1	Pengujian tingkat akurasi .....	31
3.6.2	Tingkat penerimaan salah (FAR).....	31
3.6.3	Tingkat penolakan gambar (FRR).....	31
BAB IV .....		32
HASIL DAN ANALISIS PENELITIAN.....		32
4.1	Implementasi sistem .....	32
4.1.1	Perakitan <i>hardware</i> .....	32
4.1.2	Pengintegrasian <i>software</i> .....	34
4.2	Hasil pengujian akurasi .....	39
4.2.1	Hasil pengujian akurasi .....	41
4.2.2	Hasil pengujian FAR.....	42
4.2.3	Hasil pengujian FRR.....	43
4.3	Pengujian aplikasi.....	44
4.3.1	Pengujian perangkat lunak.....	44
4.3.2	Pengujian perangkat keras.....	48
BAB V.....		50
KESIMPULAN DAN SARAN.....		50
5.1	Kesimpulan.....	50
5.2	Saran .....	50
DAFTAR PUSTAKA .....		51

## DAFTAR GAMBAR

Gambar 2.1. Tipe haar-like feature .....	9
Gambar 2.2. Implementasi Haar like feature .....	9
Gambar 2.3. Integral image.....	10
Gambar 2.4 Perhitungan nilai fitur.....	10
Gambar 2. 5 Mendapatkan classifier kuat.....	11
Gambar 2.6 Struktur cascade classifier.....	12
Gambar 2.7 Mengubah nilai piksel.....	13
Gambar 2.8 Histogram baru yang dihasilkan.....	14
Gambar 2.9 Logo OpenCV.....	16
Gambar 2.10 Logo anaconda .....	17
Gambar 2.11 Logo python .....	17
Gambar 3. 1 Sistem pengenalan wajah.....	18
Gambar 3. 2 Webcam logitech c270 hd.....	19
Gambar 3. 3 Design jetson nano .....	20
Gambar 3. 4 OpenCV version 2022.....	21
Gambar 3. 5 Diagram alir sistem .....	22
Gambar 3. 6 Flowchart pengambilan data.....	25
Gambar 3. 7 Flowchart training data .....	26
Gambar 3. 8 Flowchart pengenalan data.....	27
Gambar 3. 9 Use Case sistem.....	28
Gambar 3. 10 Home page.....	29
Gambar 3. 11 Face recognition .....	29
Gambar 3. 12 Informasi pintu terbuka .....	30
Gambar 4. 1 Perangkat jetson nano.....	32
Gambar 4. 2 SD Card .....	32
Gambar 4. 3 Kamera .....	33
Gambar 4. 4 Kabel VGA.....	33
Gambar 4. 5 Converter HDMI ke VGA.....	34
Gambar 4. 6 Monitor.....	34

Gambar 4. 7 <i>Download Jetson Nano Developer Kit SD Card</i> .....	35
Gambar 4. 8 format micro SD Card .....	36
Gambar 4. 9 Tampilan balena etcher .....	36
Gambar 4. 10 Informai <i>SD Card</i> dari <i>Windows</i> .....	37
Gambar 4. 11 Melakukan flash .....	37
Gambar 4. 12 Loading proses flash .....	38
Gambar 4. 13 Flash selesai.....	38
Gambar 4. 14 Instalasi library .....	39
Gambar 4. 15 Tampilan sistem GUI .....	44
Gambar 4. 16 Input data user .....	45
Gambar 4. 17 Proses pengambilan gambar.....	45
Gambar 4. 18 Proses pengenalan wajah.....	46
Gambar 4. 19 Hasil nilai pengenalan wajah.....	46
Gambar 4. 20 Informasi pintu terbuka .....	47



## DAFTAR TABEL

Tabel 3. 1 Spesifikasi Webcam logitech c270 hd .....	19
Tabel 3. 2 Spesifikasi jetson nano.....	20
Tabel 4. 1 Hasil percobaan pengujian akurasi .....	39
Tabel 4. 2 Hasil perhitungan dari percobaan .....	44
Tabel 4. 3 Pengujian perangkat lunak.....	47
Tabel 4. 4 pengujian perangkat keras.....	49



## **ABSTRAK**

Perkembangan teknologi suatu sistem pengamanan akses masuk ruangan telah banyak di temukan di berbagai instansi dengan menggunakan teknologi biometrik. Teknologi biometrik seperti sensor sidik jari, sensor retina mata dan deteksi wajah memberikan banyak kemudahan bagi pengguna suatu ruangan untuk keamanan privasi dan dokumen yang memiliki nilai yang tinggi. Masalah yang sering ditemukan pada ruangan yang didalamnya terdapat aset berharga adalah terjadinya pencurian. Oleh sebab itu diperlukan suatu teknologi biometrik untuk akses masuk ruangan yaitu face recognition atau pengenalan wajah dengan WebCam dan Jetson Nano sebagai alat yang digunakan untuk akses masuk ruangan. Jetson Nano yang akan menjadi otak kamera dimana kamera dapat mengenali wajah pengguna dan beberapa orang yang dapat mengakses ruangan. Jika citra yang diproses terdapat kecocokan pada file data maka program akan memberikan perintah output pintu dalam kondisi terbuka. Namun apabila citra yang ditangkap tidak ada kecocokan dengan file data maka pintu selalu dalam kondisi tertutup. metode *Haar Cascade Classifier* dan *Local Binary Pattern Histogram* (LBPH) dapat bekerja dengan hasil yang cukup untuk dapat digunakan diruangan dosen Teknik Informatika Unissula yaitu. Namun hasil tersebut masih cukup rendah jika di gunakan untuk sekala lebih yang lebih besar seperti seperti Keamanan akses masuk ruangan. Dari percobaan tiga jarak yang dilakukan mendapatkan dua kesimpulan yaitu semakin jauh jarak wajah ke kamera maka hasil FRR semakin besar dan semakin dekat jarak wajah ke kamera maka hasil FAR semakin besar. Hasil maksimal dari percobaan ini didapat pada jarak kamera ke wajah sejauh 50 cm yaitu dengan akurasi total yang di dapatkan adalah 83,33%, FAR sebesar 16,7 % dan FRR sebesar 0 %.

Keywords: Keamanan, Face recognition, OpenCV, LBPH, Jetson Nano.

## **ABSTRACT**

*The technological development of a room access security system has been widely found in various agencies using biometric technology. Biometric technologies such as fingerprint sensors, eye retina sensors and face detection provide many conveniences for users of a room for privacy security and documents that have high value. The problem that is often found in rooms where there are valuable assets is theft. Therefore, a biometric technology is needed for room access, namely face recognition with WebCam and Jetson Nano as a tool used for room access. Jetson Nano will be the brain of the camera where the camera can recognise the user's face and several people who can access the room. If the processed image has a match in the data file, the programme will give the door output command in an open condition. However, if the captured image has no match with the data file, the door is always closed. Haar Cascade Classifier and Local Binary Pattern Histogram (LBPH) methods can work with sufficient results to be used in the Unissula Informatics Engineering lecturer room, namely with the total accuracy results of all experiments in getting a value of 63.33%. However, these results are still quite low if used for a larger scale such as security access to the room. From the three distance experiments conducted, we get two conclusions, namely the farther the distance from the face to the camera, the greater the FRR results and the closer the distance from the face to the camera, the greater the FAR results. The maximum result of this experiment is obtained at a distance of 50 cm from the camera to the face, namely with a total accuracy of 83.33%, FAR of 16.7% and FRR of 0%.*

Keywords: Security, Face recognition, OpenCV, LBPH, Jetson Nano.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Ruang Dosen merupakan tempat untuk menyimpan sebuah aset penting berupa sebuah dokumen atau data dosen yang bernilai tinggi bagi instansi sehingga bersifat vital dan privasi. Setiap ruang dosen wajib memiliki standar keamanan yang ketat dari orang-orang yang tidak berkepentingan memiliki hak akses penuh yang dengan mudah keluar masuk ruangan.

Teknologi elektromagnetik seperti CardLock atau KeyLock merupakan akses masuk ruangan yang masih berkembang saat ini dengan menggunakan kunci elektrik, pin, ataupun kartu sebagai id user, akan tetapi masih memiliki keterbatasan pada ingatan manusia yang terkadang lupa membawa kunci atau kartu, dan juga lupa untuk mengingat angka dan huruf pada pin atau password (Yudhana dkk., 2022) Sehingga di butuhkan sebuah teknologi biometrik untuk membantu pengembangan sistem keamanan pada suatu ruangan.

Teknologi biometrik merupakan suatu teknologi terapan yang menggunakan ciri-ciri fisik yang khas dari tubuh seseorang sebagai ukuran yang membedakannya dengan orang lain, misalnya sidik jari, sidik mata, suara ataupun wajah (Handika Indriawan dkk., 2022). Teknologi biometrik seperti pengenalan sidik jari dan retina mata, mengharuskan manusia untuk memposisikan tubuh mereka dengan posisi sensor atau kamera, sehingga nilai pada teknologi seperti ini terkesan sangat kaku.

penelitian ini mengusulkan suatu sistem keamanan ruang dosen dengan studi kasus diruangan dosen teknik infomatika unissula menggunakan teknologi biometrik pengenalam citra pada wajah (Face Recognition). Metode yang digunakan adalah Local Binary Pattern Histogram (LBPH) dengan memanfaatkan ciri khas dari wajah manusia untuk membedakan antara orang yang berhak memiliki akses masuk ruangan dengan orang yang teridentifikasi sebagai pengunjung dan juga orang yang tidak dikenali.

## 1.2 Perumusan Masalah

Dari Latar belakang yang telah di jelaskan sebelumnya maka rumusan masalah dari penelitian ini adalah:

1. Bagaimana membangun sistem biometrik sebagai akses masuk ruangan dengan memudahkan penggunaanya?
2. Bagaimana menggunakan algoritma *Haarcascade Classifier* untuk mendeteksi wajah secara otomatis?
3. Bagaimana menggunakan metode *Local Binary Pattern Histogram* (LBPH) untuk melatih data wajah?

## 1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak terlalu luas, maka penelitian ini dibatasi dengan masalah – masalah sebagai berikut:

1. Pengenalan wajah dilakukan dengan acuan pada data wajah yang disimpan didalam folder dataset.
2. *Jetson Nano* dan *Webcam* adalah perangkat keras yang akan di gunakan pada penelitian
3. Metode yang digunakan adalah metode *Local Binary Pattern Histogram* (LBPH) yang sudah di sediakan oleh *library OpenCV*
4. Bahasa pemrograman *Python* dan tool *Visual Studio Code* digunakan sebagai perancangan sistem
5. Jumlah pengambilan data wajah setiap satu dosen sebanyak 100 sampel gambar wajah.
6. Studi Kasus dilakukan diruangan dosen Teknik Informatika Unissula

## 1.4 Tujuan penelitian

Tujuan dari penelitian ini adalah :

1. Membangun sebuah sistem biometrik pengenalan wajah yang mampu mendeteksi wajah dosen sebagai akses masuk ruang dosen.

2. Menggunakan *Haarcascade* untuk ekstraksi *feature* lebih cepat yang di kombinasikan dengan *Classifier* untuk mendapatkan model yang membedakan objek berupa wajah dan bukan wajah.
3. Menggunakan *Local Binary Pattern Histogram* untuk menghitung *threshold image* berdasarkan pikselnya.

### 1.5 Manfaat

1. Meningkatkan kenyamanan privasi pada dosen dalam menyimpan aset berharga, berkas dan dokumen yang memiliki nilai tinggi.
2. Sebagai perbandingan kemudahan dari beberapa penggunaan teknologi biometrik lainnya
3. Implementasi program aplikasi dan metode dapat dijalankan

### 1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam sebuah pembuatan laporan tugas akhir adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Pada bab 1 dijelaskan mengenai latar belakang pemilihan suatu judul, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA DAN DASAR TEORI**

Pada bab 2 Bagian ini berisi kajian pustaka yang tentang akses masuk ruang dosen menggunakan pengenalan wajah meliputi teori-teori yang berkaitan dengan *face recognition*, *algoritma haarcascade classifier* dan *local binary pattern histogram*, *library opencv*, *anaconda* dan *python* serta studi-studi terdahulu yang menggunakan metode tersebut dalam pengenalan wajah.

#### **BAB III METODE PENELITIAN**

Pada bab 3 menjelaskan tentang deskripsi sistem berupa penjelasan penggunaan aplikasi. Kebutuhan sistem berupa kebutuhan perangkat keras yang disediakan. Metode penelitian berupa kumpulan data proses training data dan proses *face recognition*. Desain sistem menjelaskan jalannya diagram alur

sistem. Konfigurasi dan instalasi sistem menjelaskan kebutuhan library perangkat lunak yang diperlukan. Pengujian metode akurasi yang dilakukan dalam pengerjaan tugas akhir berupa pengujian tingkat akurasi, *False Acceptance Rate* dan *False Rejection Rate*.

#### **BAB IV HASIL DAN ANALISIS PENELITIAN**

Pada bab 4 berisi deskripsi dan analisis hasil penelitian yang telah dilakukan oleh penulis. Hasil penelitian tersebut dapat meliputi hasil implementasi sistem, hasil pengujian akurasi, pengujian aplikasi serta perbandingan hasil tersebut dengan metode pengenalan wajah lainnya yang telah ada.

#### **BAB V KESIMPULAN DAN SARAN**

Pada Bab 5 berisi kesimpulan yang dapat diambil dari hasil pengujian akurasi yang telah dilakukan, serta rekomendasi saran untuk penelitian selanjutnya yang dapat dilakukan terkait dengan sistem akses masuk ruangan.



## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Pada rujukan pertama dengan judul “Perancangan Sistem Pengenalan Wajah untuk Keamanan Ruang Menggunakan Metode *Local Binary Pattern Histogram*”. Pada penelitian ini, waktu yang dibutuhkan komputer mini B untuk menjalankan program adalah 19,78 detik. Jarak terbaik sistem pengenalan wajah menggunakan metode LBPH adalah kurang dari 0,7meter dengan intensitas cahaya yang sangat berpengaruh. Dari hasil pengujian ketiga yaitu mengukur seberapa cepat sistem dalam mengirim pesan berupa citra gambar melalui Smartphone sampai diterima pengguna. Digunakan tiga sampel video didapatkan data sampel 1 dengan rata-rata 0,48 detik, sampel 2 dengan rata-rata 1,4 detik, dan sampel 3 dengan rata-rata 1,65 detik. (Yudhana dkk., 2022)

Pada rujukan kedua dengan judul “Aplikasi Pengenalan Wajah Untuk Sistem Absensi Kelas Berbasis *Raspberry Pi*”. Pengujian yang dilakukan peneliti dapat menarik kesimpulan sebagai berikut: Algoritma pengenalan wajah pada sistem absensi mahasiswa 1 sampai 2 orang bisa dikatakan berhasil dalam mendeteksi dan pengenalan wajah pada mahasiswa jika pengujian dilakukan 5 kali percobaan. Sedangkan hasil pengujian 3 sampai 5 mahasiswa dengan pengujian sebanyak 5 kali percobaan masih memiliki kendala pada mahasiswa yang tidak menemukan posisi wajah dengan baik. Pada pengujian pengambilan data nama pada tabel aplikasi telah berhasil diterapkan akan tetapi harus harus merubah nama dokumen secara manual dan melakukan pemindahan data yang tersimpan ke dalam folder. Keterbatasan pada pengujian ini, absensi wajah mahasiswa tidak bisa dilakukan dengan cara duduk di kursi masing-masing, hal tersebut boleh dilakukan akan tetapi layar frame pada aplikasi hanya bisa menangkap maksimal 2 orang saja. Dan yang terakhir kondisi pencahayaan yang masih menjadi masalah bagi kamera *raspberry pi nightvision* untuk mendeteksi wajah mahasiswa. (Prince Richard Setiono dkk., 2020)

“Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan *Webcam* dan *Library Opencv* Berbasis *Raspberry Pi*”. Penulis membangun sistem keamanan menggunakan *Raspberry Pi* dan *webcam* untuk merekam wajah dengan menggunakan metode *Haar-cascade Classifier* sebagai metode untuk mendeteksi letak objek wajah untuk mengklasifikasi data penghuni rumah dan orang asing. Hasil yang didapat dari pengujian sistem adalah dapat mengenali baik penghuni rumah maupun orang asing dengan beberapa tes, yaitu tes pada saat siang pukul 14.18 sampai dengan 14.50 dan malam pukul 23.20 sampai 23.30, tes jarak antara *webcam* dan orang, tiga kali skenario pada saat datang kerumah baik penghuni rumah maupun orang asing, dengan beberapa orang tertangkap kamera dan notifikasi pada telepon pintar. Berdasarkan hasil pengujian dan analisis waktu siang dan malam, didapatkan persentase kesalahan untuk siang hari sebesar 35,2%, untuk malam hari sebesar 74,3%, waktu pendeteksian pada siang hari lebih baik dibandingkan dengan malam hari karena persentase kesalahan lebih rendah pada siang hari dibandingkan pada malam hari. (Sutarti dkk., 2019)

“*Face Recognition* Menggunakan Algoritma *Haar Cascade Classifier* dan *Convolutional Neural Network*”. Penelitian ini bertujuan untuk menentukan tingkat akurasi dan waktu komputasi Algoritma *Haar Cascade Classifier* dan *Convolutional Neural Network* dalam *Face Recognition* dengan metode Menentukan tingkat akurasi dilakukan dengan menghitung jumlah data wajah yang dapat dikenali dari data wajah keseluruhan. Menghitung waktu komputasi dilakukan dengan menghitung waktu yang dibutuhkan selama proses pengenalan wajah oleh proses komputasi. Proses menentukan tingkat dan waktu komputasi dilakukan oleh program komputasi python dengan menggunakan *library numpy* dan *tensorflow*. Berdasarkan analisis yang dilakukan proses deteksi wajah menggunakan Algoritma *Haar Cascade* dan *Convolutional Neural Network* menghasilkan akurasi program sebesar 98.84% dengan serta waktu rata - rata yang dibutuhkan dalam mengenal wajah yaitu sebesar 0,05 detik. (Bella & Defri, 2021)

Pengolahan Citra Untuk Pengenalan Wajah (*Face Recognition*) menggunakan *OpenCV*. Tujuan penelitian ini hanya untuk menerapkan pengenalan wajah (*face recognition*) pada *library OpenCv* yang di tulis menggunakan Bahasa pemrograman *Python*. Rata-rata wajah yang diuji sebanyak 5 citra wajah dapat dikenali dan 2 yang tidak tersimpan karena faktor pencahayaan yang lebih terang, posisi wajah dari jarak dekat dan jauh dari faktor-faktor ini menghasilkan nilai akurasi yang berbeda sesuai dengan dengan tingkat keberhasilan dalam mengenali wajah, dengan tingkat pengenalan rata-rata 85% setelah di proses perbandingan perbandingan hasil kedekatan sekitar 81% untuk kemiripan wajah menggunakan metode *PCA Eigenface* dapat mengenali seseorang yang terdapat pada database dan tidak dapat mengenali orang yang tidak terdapat dalam database. (Susim dkk., 2021)

## 2.2 Dasar Teori

### 2.2.1 *Face Recognition*

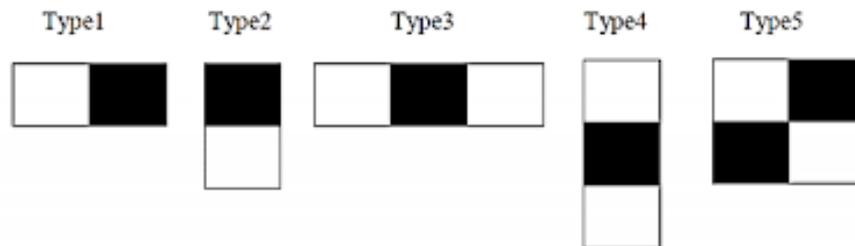
*Face recognition* merupakan salah satu perangkat teknologi biometrik yang dapat mengidentifikasi wajah manusia secara unik dengan menganalisis pola berdasarkan ciri khas dari wajah tersebut. Teknik yang digunakan berbasis fitur (geometris) atau berbasis template (fotometrik). Metode berbasis fitur bergantung pada bentuk dan posisi wajah. Sedangkan, metode berbasis gambar membuat templat fitur untuk mengidentifikasi wajah. Metode yang paling umum digunakan adalah *eigenfaces*, dan *fisherfaces*, dengan menganalisis dasar komponen utama (PCA) untuk mengekstrasi fitur wajah. Metode umum lainnya adalah dengan menggunakan metode *Local Binary Pattern Histogram* (LBPH). Walaupun tidak semua algoritma dari metode mendapatkan tingkat akurasi yang sempurna 100% tetapi tingkat akurasi yang terbaik pada saat ini mencapai 90%. Alasan utama kegagalan yang mengurangi tingkat akurasi dengan sempurna adalah adanya sensitifitas pada pencahayaan, angle pengambilan wajah yang tidak dapat terdeteksi, jarak dengan camera yang mempengaruhi pengambilan data gambar wajah (Setiono & Prince Richard, 2020).

Algoritma pada *Haar Cascade* merupakan metode yang mempunyai kemampuan dalam mendeteksi objek dengan menghasilkan tingkat akurasi yang tinggi. algoritma *Haar-cascade Classifier* menemukan wajah manusia yang sering kali menyertakan objek non-wajah lainnya seperti benda dan bagian tubuh manusia lainnya. Algoritma *Haar-cascade Classifier* biasanya mencari fitur yang paling mudah dideteksi. Algoritma kemudian akan mencoba mendeteksi alis, mulut, hidung, lubang hidung, dan iris mata. Setelah algoritma menyimpulkan bahwa ia telah menemukan daerah wajah, algoritma akan melakukan tes tambahan untuk mengonfirmasi bahwa ia benar-benar mendeteksi wajah. *Haar-cascade Classifier* merupakan algoritma yang dibuat oleh Paul Viola dan Michael Jones yang dilatih dari banyak citra wajah dan citra tanpa wajah. Setiap fitur adalah nilai tunggal yang didapat dengan mengurangi piksel di bawah persegi panjang putih dari jumlah piksel di bawah persegi panjang hitam. Pada tahun 2001, Paul Viola dan Micael Jones mendeteksi sebuah objek dengan menggabungkan 4 fungsi utama *haar like feature*, *integral image*, *Adaboost learning* dan *cascade classifier*

#### 1. *Haar like feature*

*Haar like feature* merupakan *rectangle* (persegi) dari *feature* dengan memberikan indikasi secara spesifik pada sebuah gambar. Haar like feature dapat mengenali objek berdasarkan nilai sederhana atau hanya bergantung pada jumlah pixel dalam persegi bukan nilai *pixel* dari objek sebuah gambar (Kenda & Witanti, 2021). Terdapat 3 tipe *rectangle* hitam dan putih pada *haar-like feature* yaitu *Two-rectangular feature*, *Three-rectangular feature*, *Four-rectangular feature*.

### 2.2.2 Algoritma *Haarcascade Classifier*



Gambar 2.1. Tipe *haar-like feature*

Hitung nilai individu dari karakteristik *haar like feature* menggunakan rumus:

$$F(\text{Hear}) = \sum F_{\text{putih}} - \sum F_{\text{hitam}}$$

$$F(\text{Hear}) = \frac{1}{n} \sum_{\text{hitam}}^n 1(x) - \frac{1}{n} \sum_{\text{putih}}^n 1(x) \quad (1)$$

Keterangan:

$F(\text{Hear})$  = Nilai fitur keseluruhan

$\sum F_{\text{putih}}$  = Nilai fitur terang

$\sum F_{\text{hitam}}$  = Nilai fitur gelap

$N$  = Jumlah piksel

$I(x)$  = nilai sebenarnya yang terdeteksi pada citra.

Pengguna fitur Haar like dijelaskan pada Gambar 2.



Gambar 2.2. Implementasi *Haar like feature*

Gambar wajah dikelompokkan berdasarkan sisi terang dan gelap. Misalnya, area mata terlihat lebih gelap dibanding area sekitarnya. Ukuran dasar wajah yang akan dievaluasi adalah 24 x 24 yang akan diperbesar menjadi 12 ukuran dengan perbandingan 1.25. Berdasarkan ukuran 24 x 24

ini, ada sekitar 160.000 kemungkinan filter seperti haar. Ukuran akan bergerak 11 dari kiri ke kanan dan berlanjut ke baris berikutnya hingga selesai. Proses selanjutnya akan dilakukan kembali dengan basis ukuran yang telah diskalakan menjadi skala basis 12 ukuran.

### 1. *Integral Image*

*Integral image* adalah gambar di mana setiap nilai piksel input merepresentasikan jumlah kumulatif piksel yang sesuai dengan semua piksel di atas dan di sebelah kiri piksel input. *Integral image* menghitung nilai fitur secara cepat dengan mengubah nilai dari masing-masing piksel menjadi representasi citra baru (Kenda & Witanti, 2021). contohnya pixel (a,b) memiliki nilai akumulatif untuk semua pixel(x,y) dimana  $x \leq a$  dan  $y \leq b$ .

4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

A	B	
(x1,y1)		
C	D	
		(x2,y2)

Gambar 2.3. *Integral image*

Pada gambar 3, citra integral pada (x,y) nilai (ii(x,y)) dapat di rumuskan:

$$ii(x, y) = (x + a)^n = \sum_{x^i \leq x, y^i \leq y} (x_i, y_i) \quad (2)$$

Keterangan:

$ii(x, y)$  = citra integral pada titik x,y

$(x_i, y_i)$  = nilai piksel pada citra asli

Pada perhitungan cepat citra integral dengan menjumlahkan nilai piksel sebelah kiri dan atas pada titik yang akan menjadi citra baru dapat dilihat pada gambar 2.4

A	B	
	1	2
C	D	
	3	4

Gambar 2.4 Perhitungan nilai fitur

Jika nilai dari Intergral image pada titik 1 adalah A, titik 2 adalah A+B, titik 3 adalah A+C, dan di titik adalah A+B+C+D, maka jumlah piksel pada titik D dapat diketahui dengan perhitungan  $4+1-(2+3)$ . Proses seperti ini membutuhkan komputasi yang tinggi

## 2. Adaboost Learning

*AdaBoost* menggunakan pendekatan berulang untuk belajar dari kesalahan pengklasifikasi yang lemah, dan mengubahnya menjadi pengklasifikasi yang kuat. *Classifier* lemah adalah jawaban yang benar pada tingkat kebenaran yang kurang tepat. Persamaan pada *Classifier* lemah adalah sebagai berikut:

$$h_j(x) = \begin{cases} 1, & \text{jika } p_j f_j < p_j \phi_j(x) \\ 0, & \text{lainnya} \end{cases} \quad (3)$$

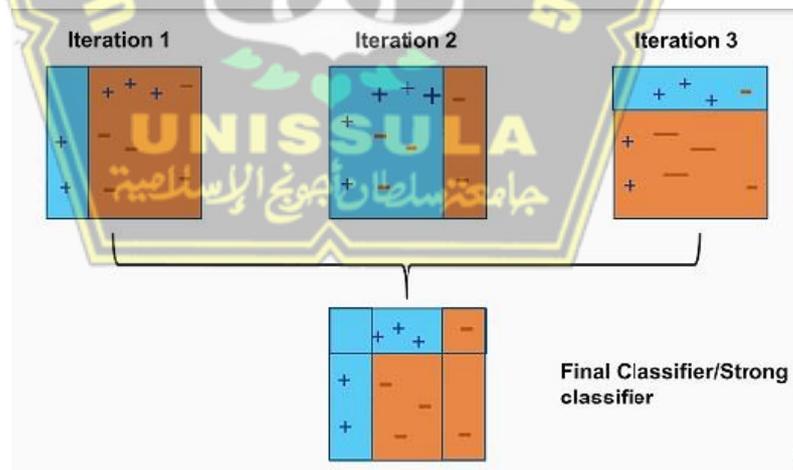
Keterangan:

$h_j(x)$  = klasifikasi lemah

$p_j$  = *parity* ke  $j$

$\phi_j$  = *threshold* ke  $j$

$x$  merupakan dimensi bagian dari citra 24 x 24.

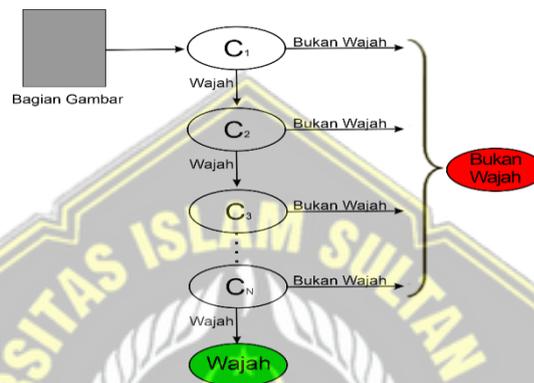


Gambar 2. 5 Mendapatkan *classifier* kuat

Dari gambar di atas dibutuhkan beberapa *iteration* yang dilakukan dalam mendapatkan clasifier kuat yang digunakan untuk mendeteksi suatu wajah.

### 3. *Cascade Classifier*

*Cascade Classifier* merupakan inovasi dalam hal peningkatan kecepatan Perhitungan selama pendeteksian dengan memfokuskan pada target hanya ada satu kemungkinan. Jadi katakanlah itu cukup banyak dalam gambaran besarnya objek selain area wajah dibuat dengan cepat dan pada area wajah baru benar-benar membuat classifier yang kompleks. Struktur dari *cascade classifier* dapat dilihat pada Gambar 2.6



Gambar 2.6 Struktur *cascade classifier*

Pada sebuah gambar dievaluasi oleh classifier pertama dan berfungsi. Kemudian dilanjutkan dengan pengklasifikasi kedua hingga pengklasifikasi ke-N. Jika melewati semua pengklasifikasi yang ada, dapat disimpulkan bahwa ada objek yang terdeteksi di dalam citra.

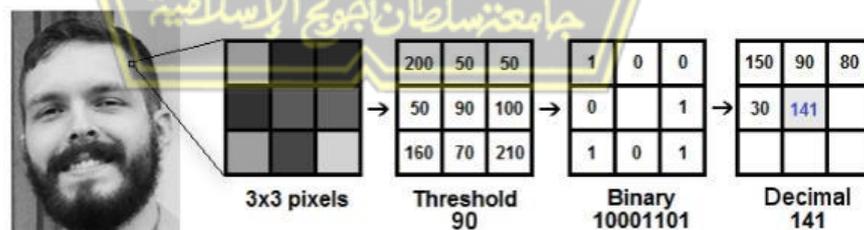
#### 2.2.3 Algoritma *Local Binary Pattern Histogram*

Seseorang dapat melakukan pengenalan wajah setiap hari dan dengan mudah mengenali nama orang. Meskipun ini adalah tugas yang sederhana, ini adalah tugas yang cukup rumit untuk komputer. Keakuratan metode yang digunakan oleh komputer sangat dipengaruhi oleh beberapa variabel, antara lain variasi cahaya, resolusi kamera, sudut masuknya objek ke dalam kamera, dan jarak objek yang diamati. Dalam ilmu komputer, pengenalan wajah pada dasarnya adalah tugas untuk mengidentifikasi seseorang berdasarkan citra wajahnya. Ini menjadi sangat populer selama dua dekade terakhir, terutama karena pengembangan metode baru dan kualitas tinggi kamera saat ini. Pengenalan wajah berbeda dengan pendeteksi wajah. Deteksi wajah bertujuan untuk menemukan wajah (ukuran dan posisi) dalam gambar dan

mengekstraknya untuk digunakan kembali dengan algoritma deteksi wajah. Deteksi wajah dilakukan pada gambar yang diekstraksi menggunakan algoritma deteksi wajah yang dipotong, diubah ukurannya, dan biasanya dikonversi ke gambar grayscale. Algoritma pengenalan wajah bertanggung jawab untuk menemukan pola wajah yang paling menggambarkannya pada Gambar. Sistem pengenalan wajah dapat bekerja dalam dua mode yaitu autentikasi dan Demonstrasi Dalam mode autentikasi, wajah dibandingkan dengan gambar dengan rasio 1:1, sedangkan deteksi wajah membandingkan wajah dengan gambar dengan rasio 1:N. *Local Binary Pattern* (LBP) adalah operator tekstur sederhana yang mencirikan piksel suatu gambar dengan mendefinisikan lingkungan piksel dan memperlakukannya sebagai bilangan biner. *Local Binary Pattern Histogram* merupakan salah satu algoritma deteksi wajah yang merupakan pengembangan lebih lanjut dari metode *Local Binary Pattern*. Ada 4 parameter yang dimiliki oleh metode ini yaitu *Radius*, *Neighbors*, *Grid x* dan *Grid y*.

#### 1. Proses *Local Binary Pattern* (LBP)

Langkah awal dalam menerapkan perhitungan LBPH adalah membuat citra antara yang merepresentasikan citra asli dengan mempertegas citra wajah. Untuk melakukannya, algoritma ini menggunakan konsep jendela geser (*sliding window*) berdasarkan parameter radius dan ketetanggaan.



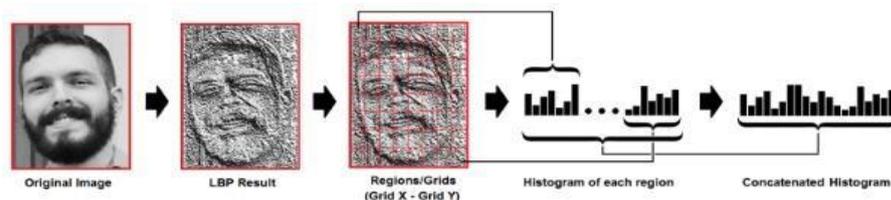
Gambar 2.7 Mengubah nilai piksel

Misalkan kita memiliki gambar skala abu-abu. Dari gambar ini, potongan 3x3 piksel yang mengandung intensitas setiap piksel (0-255) diekstrak. Kemudian ambil rata-rata atau nilai bagian tengah dari matriks untuk digunakan sebagai ambang batas. Nilai ini menentukan nilai baru dari delapan tetangganya. Nilai setiap tetangga ditentukan berdasarkan ambang batas. Jika nilainya lebih besar dari ambang batas, itu adalah 1 dan sebaliknya nilainya

adalah 0. Setelah itu, matriks hanya berisi nilai biner (mengabaikan nilai rata-rata). Kemudian gabungkan nilai biner dari matriks menjadi nilai baru. Perkataan: Penulis yang berbeda menggunakan pendekatan yang berbeda untuk menulis barisan biner misalnya searah jarum jam, tetapi hasil akhirnya sama. Kemudian kami mengonversi nilai biner yang dihasilkan menjadi nilai desimal dan menetapkannya ke rata-rata matriks sebelumnya sehingga menghasilkan gambar baru dengan fitur yang lebih baik daripada gambar aslinya.

## 2. Ekstrak ke *Histogram*

Selain menghitung histogram LBP untuk setiap blok, *histogram* setiap blok perlu digabungkan mengikuti pola *grid* yang telah ditentukan berdasarkan parameter  $x$  dan  $y$  dari *grid*. Misalnya, jika parameter *grid*  $x$  adalah 3 dan parameter *grid*  $y$  adalah 3, gambar dibagi menjadi 9 blok yang sama, dengan tiga blok di setiap baris dan kolom. *Histogram* setiap blok kemudian digabungkan berdasarkan posisinya di *grid*. Misalnya, *histogram* blok pertama baris pertama dan kolom pertama dihubungkan dengan *histogram* blok kedua baris pertama dan kolom pertama dan seterusnya. sampai *histogram* blok terakhir baris pertama dan kolom pertama terhubung dengan *histogram* blok pertama baris kedua dan kolom terhubung. Setelah *histogram* setiap blok digabungkan berdasarkan *grid*, *histogram* fitur akhir dapat dihasilkan. *Histogram* ini merepresentasikan pola permukaan pada semua citra dan dapat digunakan untuk mengidentifikasi objek atau wajah pada citra. Seperti pada gambar 2.8



Gambar 2.8 Histogram baru yang dihasilkan

*Histogram* setiap area dapat diekstraksi dari gambar di atas. Setiap *histogram* hanya berisi 256 digit (0-255) yang mewakili tampilan intensitas setiap piksel. *Histogram* individu kemudian harus dirakit menjadi *histogram*

baru dan lebih besar. Katakanlah kita memiliki grid 8 x 8 yang menghasilkan 8 x 8 x 256 = 16.384 digit pada histogram akhir. Histogram akhir mewakili karakteristik dari citra asli

### 3. Normalisasi *Histogram*

Setelah *histogram* dibuat, LBPH kemudian melakukan normalisasi pada nilai *histogram*. Hal ini dilakukan untuk memperbaiki performa LBPH terhadap perbedaan cahaya atau kontras pada citra. Normalisasi dapat dilakukan dengan beberapa metode, seperti menghitung nilai rata-rata dan standar deviasi dari histogram, kemudian menormalkan setiap nilai *histogram* dengan cara mengurangi nilai rata-rata dan membagi hasilnya dengan standar deviasi.

### 4. Pengenalan Wajah

Setelah *histogram* dinormalisasi, nilai *histogram* tersebut dapat digunakan sebagai fitur untuk mengidentifikasi wajah atau objek pada citra. LBPH dapat membandingkan histogram citra yang diuji dengan *histogram* citra referensi dan memberikan nilai kemiripan antara keduanya. Semakin mirip nilai *histogram* antara citra referensi dan citra uji, maka semakin besar kemungkinan citra uji tersebut sama dengan citra referensi. Jadi untuk menemukan gambar yang cocok dengan gambar input, yang harus dilakukan adalah membandingkan kedua histogram dan mengembalikan gambar dengan *histogram* yang paling mirip. Dalam hal ini perlu menggunakan jarak *Euclidean* untuk perbandingan pada *histogram*.

$$D = \sqrt{\sum_{i=1}^n (\text{hist1}_i - \text{hist2}_i)^2} \quad (4)$$

hasil dari algoritma tersebut adalah IDE dari citra dengan *histogram* yang paling mirip. Algoritma juga harus mengembalikan jarak terhitung yang dapat digunakan sebagai ukuran seberapa mirip gambar yang masuk dengan kumpulan data. Kemudian dari derajat kemiripan tersebut harus ditentukan nilai ambang minimal untuk pengenalan wajah, yaitu lebih dari 50%, misalnya. Oleh karena itu, jika ada data wajah baru yang kemiripannya tidak mencapai 50%, maka akan dianggap sebagai wajah asing yang belum dikenali.

#### 2.2.4 Library OpenCV

*Open Source Computer Vision Library* adalah pustaka perangkat lunak untuk pemrosesan gambar dinamis waktu nyata. Pengembangan ini diprakarsai oleh *Intel* dan kini didukung oleh *Willow Garage* dan *Itseez*. *OpenCV* dirilis di bawah lisensi BSD permisif, yang lebih bebas daripada GPL dan memberikan kebebasan penuh untuk penggunaan komersial tanpa mengungkapkan kode sumbernya. *OpenCV* juga mendukung bahasa pemrograman *Python*, juga untuk sistem operasi *Windows*. *OpenCV* memiliki beberapa fungsi yang digunakan dalam penelitian ini seperti `cv2.imread()`, `cv2.Show()`, `cv2.WaitKey()`. *OpenCV* digunakan di seluruh dunia dan memiliki komunitas lebih dari 47.000 pengguna dengan perkiraan unduhan lebih dari 7 juta (Fikriansyah Martunus, 2020)



Gambar 2.9 Logo OpenCV

*Library OpenCV* sudah memiliki lebih dari 2500 algoritma yang dioptimalkan. Fungsi yang digunakan oleh *OpenCV* adalah: Deteksi wajah, pengenalan wajah, pengenalan objek, klasifikasi fitur objek, ekstraksi model objek dan menyediakan berbagai algoritma sederhana yang terkait dengan API tingkat rendah visi komputer.

#### 2.2.5 Anaconda

*Anaconda* adalah platform distribusi data dan pengetahuan populer yang memudahkan pengguna untuk menginstal dan menggunakan berbagai alat dan paket ilmu data dan pemrograman, termasuk *Python*, *R*, dan *Jupyter Notebook*. *Anaconda* menyediakan perpustakaan ekstensif paket ilmu data siap pakai dan kemampuan manajemen lingkungan virtual yang memungkinkan pengguna untuk dengan mudah menginstal dan mengelola berbagai versi paket ilmu data pada satu mesin (Poysancin & Utomo, 2019)



Gambar 2.10 Logo anaconda

*Anaconda Python* juga menyediakan alat manajemen lingkungan virtual yang memudahkan pengguna untuk menginstal, mengelola, dan menavigasi berbagai lingkungan virtual. Pengguna dapat mengelola berbagai versi *Python* dan paket ilmu data pada satu mesin di lingkungan virtual yang terisolasi.

### 2.2.6 *Python*

Pengertian *Python* adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum pada tahun 1991. *Python* mendukung banyak paradigma pemrograman, termasuk pemrograman prosedural, objek, dan fungsional. *Python* memiliki pustaka yang besar dan aktif, membuatnya mudah digunakan dan sangat populer di kalangan pengembang. *Python* memiliki filosofi desain yang disebut "The Zen of Python" yang berisi prinsip-prinsip yang dirancang untuk membuat kode *Python* mudah dibaca dan dipahami. Sebagai bahasa pemrograman tingkat tinggi, *Python* mudah dipelajari karena dilengkapi dengan manajemen memori otomatis



Gambar 2.11 Logo python

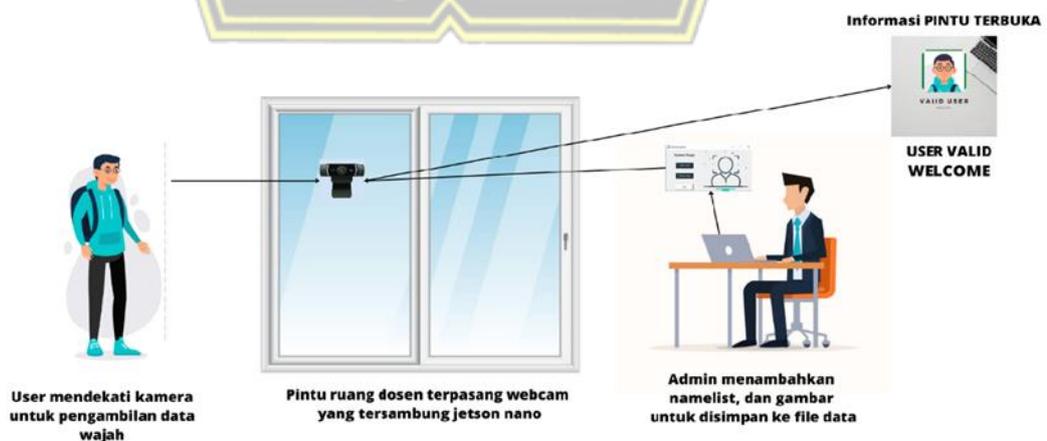
*Python* menyediakan beberapa metode yang dapat digunakan dalam pemrograman *Built-in functions*, *Methods*, *User-defined function*, *Lambda function* yang memiliki fungsi mempermudah pemrograman, mengolah atau mengubah objek, menghitung rumus, dan juga sebagai argumen untuk fungsi lainnya.

## BAB III METODE PENELITIAN

### 3.1 Deskripsi Sistem

Tugas Akhir dengan Judul “Rancang Bangun Sistem Pengenalan Wajah Menggunakan Metode *Haar Cascade Classifier* dan *Local Binary Pattern Histogram* Pada Akses Masuk Ruang Dosen” merupakan sistem untuk membuka pintu ruang dosen dengan memanfaatkan kamera sebagai *input* sebuah data dan sebagai kunci ruangan.

Prototipe ini menggunakan kamera yang disematkan dalam sistem pengenalan wajah yang berjalan di komputer mini *Jetson Nano*. Administrator memasukkan data wajah secara manual dan total 100 gambar dari setiap wajah diambil. Gambar yang disimpan kemudian mengalami proses pelatihan untuk mempertahankan fitur gambar wajah. Saat kamera mengenali objek wajah yang ditangkap, itu digabungkan dengan data terlatih. Jika tingkat kemiripan lebih besar atau sama dengan 50% maka akan pada desktop (monitor) *interface* administrator yaitu “PINTU TERBUKA” berupa (hasil dokumentasi) yang di copy dari file data dengan nomor urut ke 50 dan pesan USER VALID. Nilai dari ambang batas tersebut diambil dari rata-rata kemiripan pada jarak fokus kamera 50 cm, 80 cm dan 100 cm wajah manusia. Gambaran umum sistem ini ditunjukkan pada Gambar 3.1



Gambar 3. 1 Sistem pengenalam wajah

## 3.2 Kebutuhan sistem

### 3.2.1 Logitech C270 HD Webcam

*Webcam logitech c270 hd* digunakan untuk penangkapan gambar wajah, mendeteksi dan mengenali wajah seseorang dengan lebih akurat dengan resolusi tinggi *Webcam logitech c270 hd* dapat memberikan gambar yang jernih dan tajam.



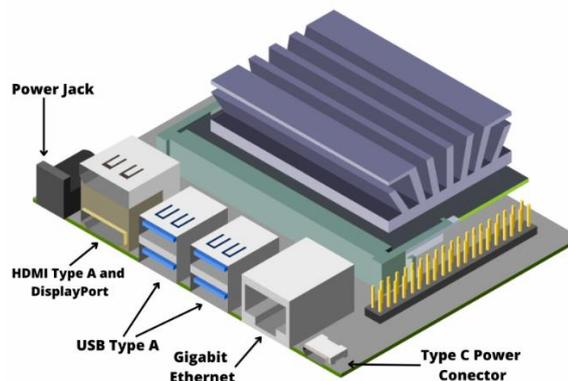
Gambar 3. 2 *Webcam logitech c270 hd*

*Webcam logitech c270* memiliki beberapa spesifikasi seperti pada Tabel 3.1

Tabel 3. 1 Spesifikasi *Webcam logitech c270 hd*

Parameter	Spesifikasi
Frame Rate	720p at up to 30 fps
Sensor	CMOS
Resolution	3 MP (Software Enhanced)
Field of View	60°
Focus Type	Auto
Host Connection	USB Type-A USB 2.0
Power Source	Bus Powered
Supported Operating Systems	Windows 10 or Later Windows 8 Windows 7 macOS 10.10 or Later Android 5.0 or Later Chrome
Microphone	Yes
Cable Length	5' / 1.52 m

### 3.2.2 Jetson Nano



Gambar 3. 3 Design jetson nano

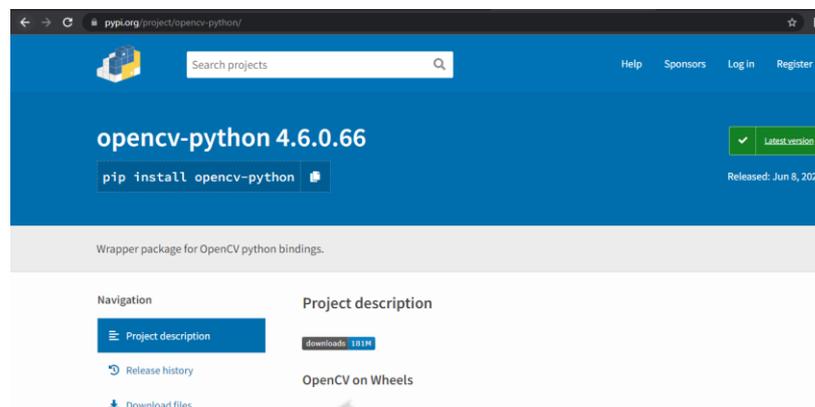
*Jetson Nano* memiliki prosesor Arm Cortex-A57 yang dilengkapi GPU *NVIDIA Maxwell* untuk menangani pembelajaran mesin. *Jetson Nano* juga dilengkapi Konektivitas *Gigabit Ethernet* dan RAM 4GB, yang membuatnya ideal untuk aplikasi *Internet of Things* (IoT), robotik, dan aplikasi komputasi lainnya. (Sati dkk., 2021)

Spesifikasi yang dimiliki *Nvidia Jetson nano* seperti pada tabel 3.2

Tabel 3. 2 Spesifikasi *jetson nano*

Parameter	Spesifikasi
Antarmuka	GPIO, I2C, I2S, SPI, UART
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
RAM	4 GB 64-bit LPDDR4 25.6 GB/s
Penyimpanan Eksternal	microSD (not included)
Video Encoder	4K @ 30   4x 1080p @ 30   9x 720p @ 30 (H.264/H.265)
Video Decoder	4K @ 60   2x 4K @ 30   8x 1080p @ 30   18x 720p @ 30 (H.264/H.265)
Soket Kamera	2x MIPI CSI-2 DPHY lanes
Konektivitas	Gigabit Ethernet, M.2 Key E
Display Output	HDMI 2.0 and eDP 1.4
USB	4x USB 3.0, USB 2.0 Micro-B
Dimensi	100 mm x 80 mm x 29 mm

### 3.2.3 OpenCV

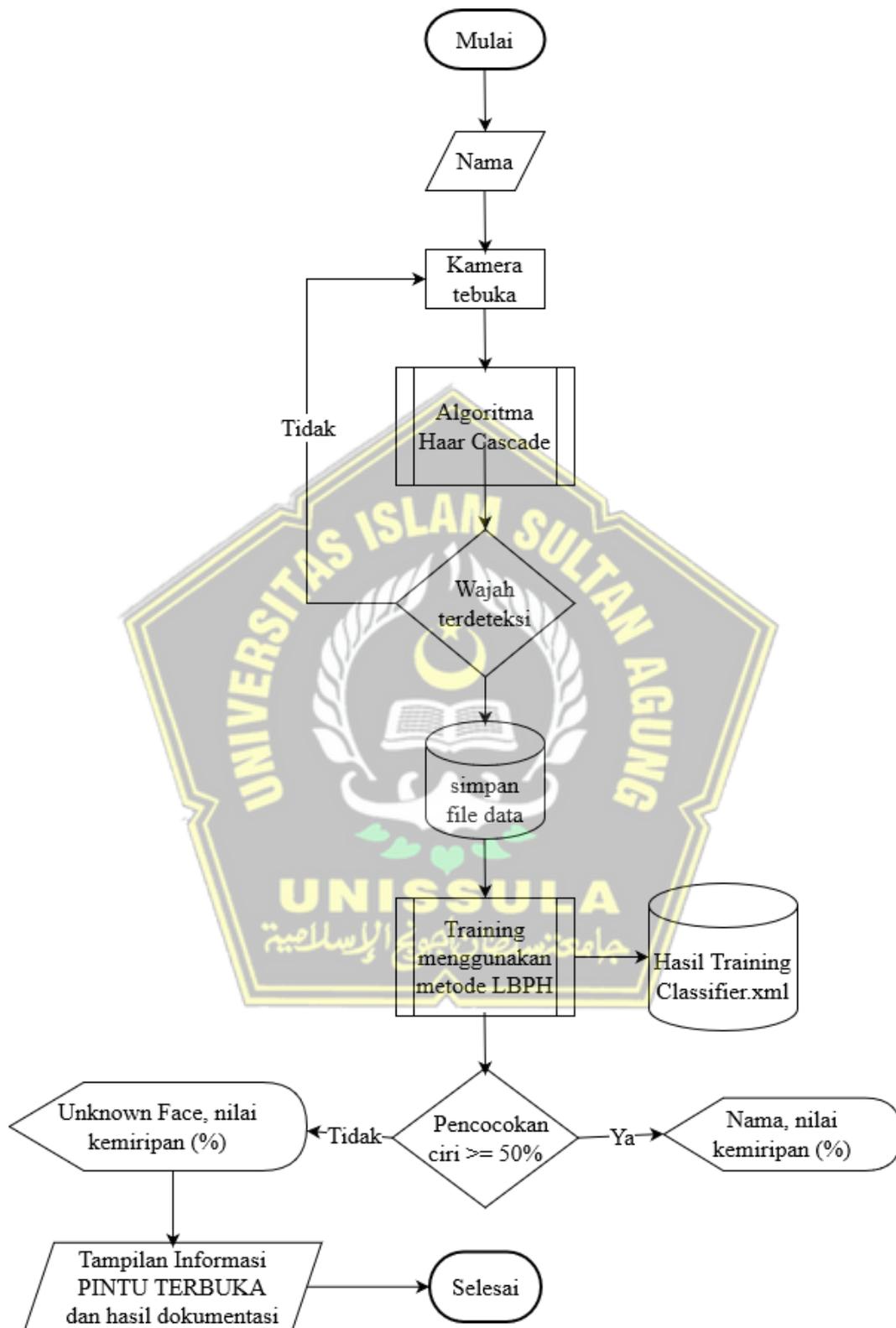


Gambar 3. 4 OpenCV version 2022

Pada gambar 3.4 merupakan pustaka *OpenCV* didalamnya terdapat fungsi Viola-Jones dimana memiliki beberapa parameter yang siap untuk dijalankan.

### 3.3 Metode penelitian

Sistem ini di rancang dengan menggunakan metode *Haar Cascade Classifier* dan *Local Binary Pattern Histogram* yang terdapat di dalam *library OpenCV* yang berfungsi untuk mendeteksi dan pengenalan wajah. Tahap klasifikasi yang dilakukan pada wajah ada tiga tahap pengerjaannya yaitu pertama tahap *create dataset* atau pengambilan gambar suatu wajah dengan menggunakan metode *Haar Cascade Classifier* dengan kode program `haarcascade_frontalface_default.xml` untuk pengumpulan data yang di simpan dalam folder data atau dataset, tahap kedua melakukan *training* menggunakan metode *Local Binary Pattern Histogram* dengan kode program `cv2.face.LBPHFaceRecognizer_create()` sehingga mendapatkan nilai biner dan desimal dari beberapa wajah unik yang berguna sebagai pembeda dari keunikan wajah lainnya, dan tahap terakhir melakukan *recognizer* atau pengenalan suatu wajah dari data wajah yang tersimpan dalam dataset dengan wajah yang di tangkap oleh kamera setelah melakukan *training*.



Gambar 3. 5 Diagram alir sistem

### 3.3.1 Kumpulkan data *Training*

Pengambilan data training dilakukan dengan beberapa urutan ketentuan seperti:

1. *Input* data diri dosen untuk pendaftaran awal pada pengenalan wajah
2. Admin membutuhkan data dosen pada *add user* di aplikasi berupa sebuah nama
3. Admin melakukan pengambilan *dataset* dengan menekan tombol *capture dataset* pada aplikasi. Jumlah gambar untuk 1 dosen sebanyak 100
4. Dosen berdiri didepan kamera dengan jarak sejauh 50 cm – 70cm dari wajah ke kamera kemudian melakukan *training* dengan menekan tombol *Train the model* pada aplikasi.
5. Dosen akan diberikan aba-aba kepada admin untuk menengok ke atas dan kesamping didepan kamera untuk mendapatkan hasil yang baik dari keseluruhan wajah
6. Admin perlu mengecek data yang sudah melakukan *training* dengan menekan tombol *check a user* pada aplikasi untuk mengetahui nama dosen yang sudah melakukan *training* selain itu membuka folder dan apabila terdapat objek bukan wajah maka objek boleh dihapus
7. Pengumpulan data wajah selesai

Penerapan yang dilakukan pada metode ini menggunakan *Haar Cascade Classifier* dalam mendapatkan objek berupa wajah dengan perhitungan yang dilakukan oleh *Library Opencv* sehingga proses pada program dapat dijelaskan sebagai berikut:

1. Aktifkan kamera kemudian akan menampilkan gambar wajah yang ada didepan kamera
2. Gambar wajah yang tertangkap oleh kamera tersimpan kedalam sebuah variabel yang akan diubah menjadi sebuah citra dengan skala rgb atau abu-abu
3. Gambar citra wajah pada skala rgb atau abu-abu akan dilakukan proses algoritma ekstraksi *fitur haar like*, *Integral Image* dan *Cascade Classifier*

4. Setelah pemrosesan algoritma *Haar Cascade Classifier* menghasilkan kotak area wajah yang
5. Pada proses pengambilan gambar, selain area wajah dihilangkan dan hanya akan menyimpan gambar pada area wajah

### 3.3.2 Dataset Training proses

Pada tahap training atau melatih data wajah dilakukan setelah mendapatkan objek gambar wajah. Tahap ini menggunakan metode *Local Binary Pattern Histogram* yang di jalankan *Library OpenCV*. Hasil yang didapatkan pada proses ini berupa angka-angka biner dan decimal yang di histogramkan dan disimpan pada file khusus dengan ekstensi xml.

### 3.3.3 Proses Face Recognition

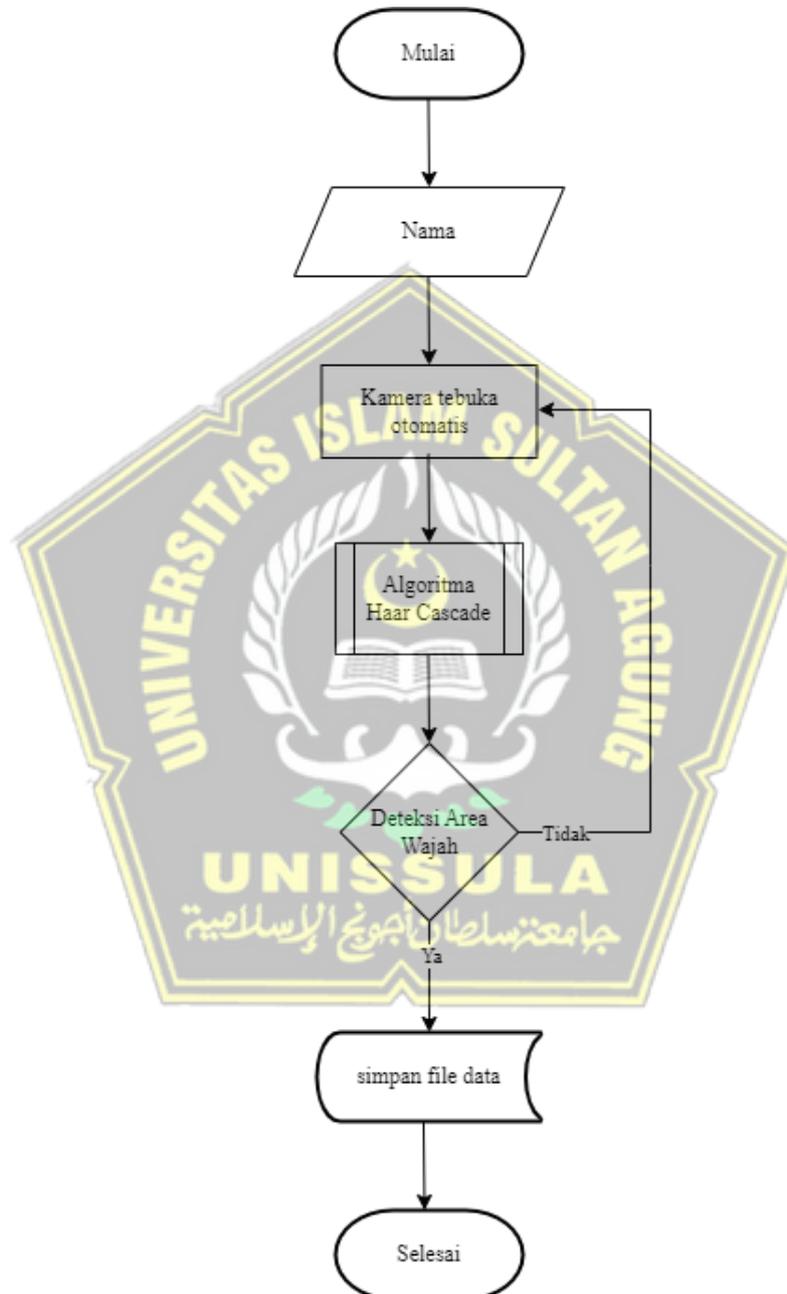
Pengenalan wajah tentunya dapat diaplikasikan setelah melalui proses training dataset dengan metode *Haar Cascade Classifier* dan *Local Binary Pattern Histogram*. Pengenalan wajah dilakukan dengan ketentuan seperti :

1. Klasifikasi wajah dosen untuk memasuki ruangan dilakukan antara kamera dan wajah dengan jarak maksimal 50cm, 80cm dan 100 cm.
2. Tingkat kemiripan wajah yang di izinkan untuk menghasilkan data dengan tulisann pintu terbuka dan tampilan *valid user* pada aplikasi adalah diatas 50%
3. Jika tingkat kemiripan kurang dari 50% maka tampilan kotak pada area wajah berwarna merah sehingga dosen dapat menyesuaikan kemiringan atau jarak pada wajah di depan kamera sampai mendapatkan nilai valid dengan tingkat kemiripan diatas 50%

### 3.4 Desain Sistem

#### 3.4.1 Flowchart pada sistem

##### 1. Flowchart pengambilan data

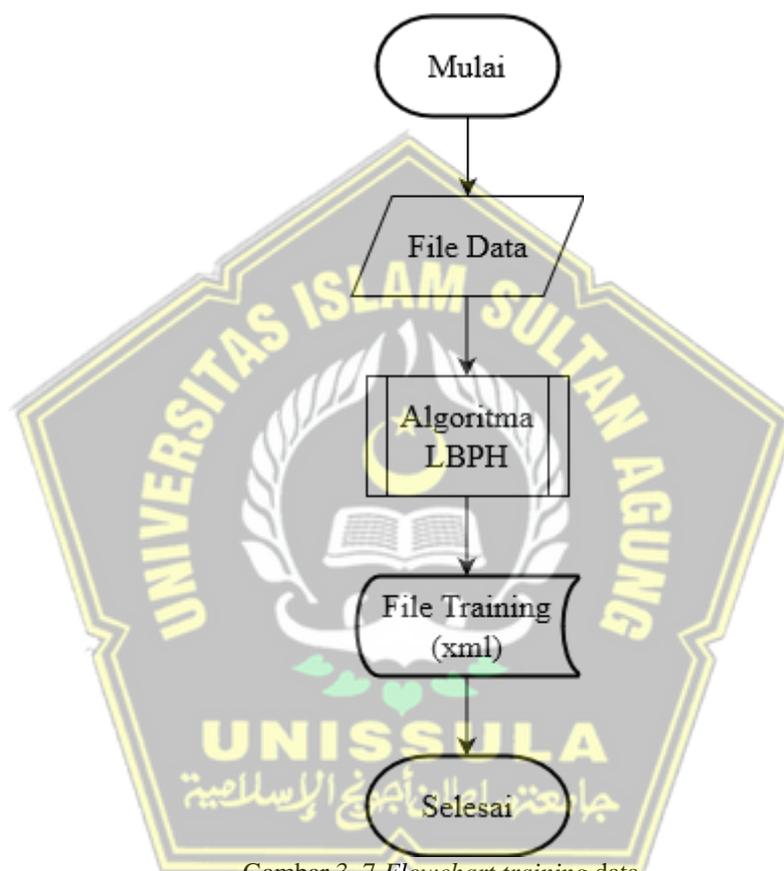


Gambar 3. 6 Flowchart pengambilan data

Diagram ini menjelaskan proses dari pengambilan data yang kemudian program memberikan perintah *input* data identitas dosen pada *variabel* yang

menjadi label pada sebuah gambar yang diambil. Setelah data dimasukkan, kamera aktif secara otomatis dengan hanya menandai area bagian wajah menggunakan algoritma *haar-cascade* yang disediakan *library OpenCV*. Data gambar yang hanya mengambil area bagian wajah pada kamera disimpan difolder data.

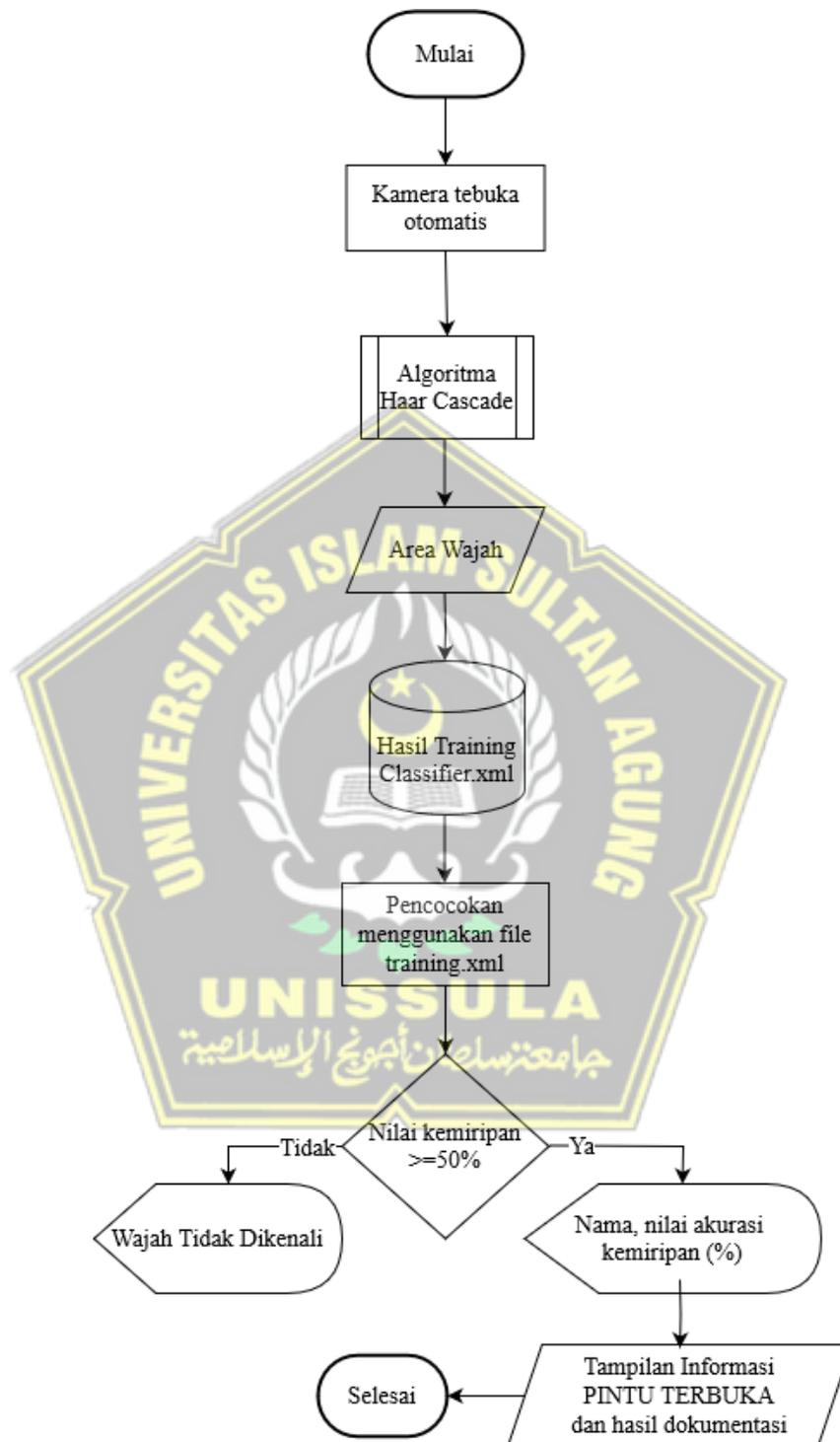
## 2. Flowchart data training



Gambar 3. 7 Flowchart training data

Diagram ini menggambarkan bahwa proses *training* data dimulai dengan membutuhkan file gambar wajah yang tersimpan dalam folder data untuk digunakan sebagai data training. Setelah itu, nilai *Local Binary Pattern* (LBP) dihitung untuk setiap *pixel* pada gambar wajah. Nilai LBP tersebut kemudian digunakan untuk membuat *histogram*. *Histogram* ini berupa matrik yang mewakili gambar dari data. Proses ini diulang untuk setiap gambar wajah yang digunakan untuk training, kemudian selesai ketika semua data training telah disimpan pada file *training* dengan *extensi training.xml*.

### 3. Flowchart pengenalan data

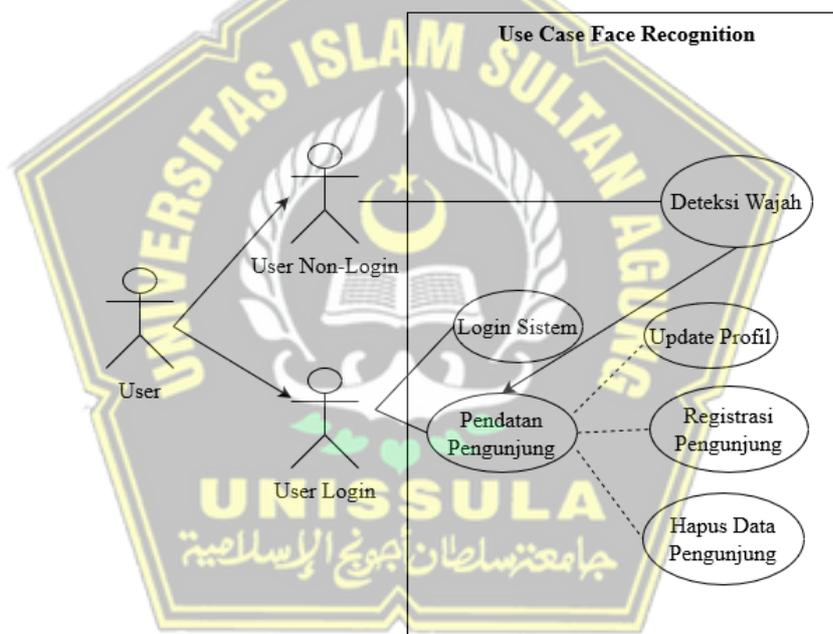


Gambar 3. 8 Flowchart pengenalan data

Diagram ini menggambarkan bahwa proses pengenalan wajah dimulai saat kamera mendeteksi wajah menggunakan algoritma *haar-cascade*

mendapat objek wajah yang digunakan untuk pengenalan. Setelah itu, nilai *Local Binary Pattern* (LBP) dihitung untuk setiap *pixel* pada gambar wajah, kemudian digunakan untuk membuat *histogram*. *Histogram* tersebut mencocokkan dengan salah satu wajah pada data *training*. Apabila wajah memiliki tingkat kemiripan dengan data atau akurasi di atas 50%, maka informasi wajah yang cocok ditampilkan. Namun, jika *histogram* tersebut tidak memiliki tingkat kemiripan dengan data *training* atau di bawah 50%, maka wajah pada gambar tersebut menampilkan pesan tidak dapat dikenali. Proses pengenalan wajah selesai.

### 3.4.2 Use case sistem



Gambar 3.9 Use Case sistem

Pada gambar 3.9 di atas merupakan *use case* dari sistem yang dibuat menggambarkan aktor yang memiliki hak akses yang berbeda terhadap sistem. Rancangan sistem diagram ini hanya memiliki 2 aktor, yaitu admin dan pengunjung. Peran admin adalah sebagai pemilik ruangan sehingga memiliki hak akses secara menyeluruh pada sistem. Mulai dari *login* sistem, tambah data, hapus data sampai dengan deteksi wajah untuk akses ruangan. Pengunjung hanya memiliki akses untuk masuk pada ruang dosen dengan pendeteksian wajah.

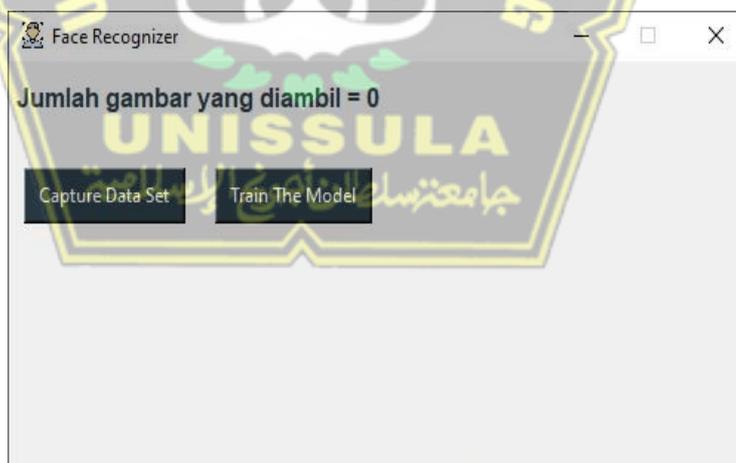
### 3.4.3 Desain GUI

Graphical User Interface desain aplikasi untuk memudahkan pekerjaan penggunanya baik itu keindahan desain atau proses kerja pada fitur desain aplikasi. Terdapat beberapa desain GUI ditampilkan untuk kelanjutan proses selanjutnya seperti fungsi dari beberapa fitur dalam desain sistem GUI.



Gambar 3. 10 Home page

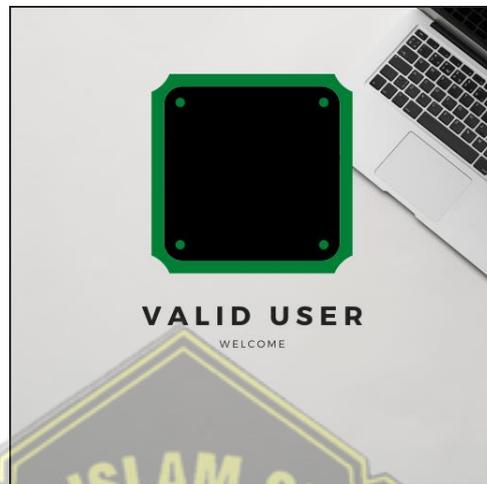
*Home Page* merupakan tahap awal pada tampilan *user interface* dimana terdapat beberapa *button* yang memiliki masing-masing fungsi. Tampilan awal dari *user interface* seperti pada gambar 3.10



Gambar 3. 11 Face recognition

Pada gambar 3.11 merupakan tampilan desain halaman tahap selanjutnya setelah melakukan proses tahap pertama pada *home page*. Beberapa fitur didalam halaman dapat bekerja untuk melanjutkan proses ke tahap halaman

berikutnya seperti *button capture data set*, *button train the model* dan teks jumlah gambar yang diambil.



Gambar 3. 12 Informasi pintu terbuka

Hasil akhir dari *Face Recognition* menampilkan desain halaman terakhir yaitu apabila sistem berhasil di jalankan, maka menampilkan desain yang terlihat pada gambar 3.12

### 3.5 Konfigurasi sistem dan instalasi sistem

#### 3.5.1 Konfigurasi sistem

Adapun konfigurasi sistem yang perlu dilakukan :

1. Jumlah data *training* tiap wajah dosen
2. Nilai persentase batas minimum pada tingkat kemiripan yang diizinkan
3. Konfigurasi dataset yang digunakan

#### 3.5.2 Instalasi sistem

Perlu beberapa *library* yang dibutuhkan untuk menjalankan sistem seperti :

1. *Anconda* versi 22.9.0
2. *Python* versi 3.9
3. *OpenCV* versi 4.6.0
4. *Pillow* versi 9.3.0
5. *Numpy* 1.19.4

### 3.6 Pengujian metode akurasi sistem

#### 3.6.1 Pengujian tingkat akurasi

Perhitungan pertama ini membagi semua keberhasilan dengan semua percobaan dan mengalikannya dengan 100 persen untuk menghitung tingkat akurasi keseluruhan.

$$Akurasi (\%) = \frac{\text{percobaan berhasil}}{\text{jumlah percobaan}} \times 100\% \quad (5)$$

#### 3.6.2 Tingkat penerimaan salah (FAR)

Kesalahan dalam mengenali identitas gambar yang ditangkap oleh kamera. Kesalahan dalam mengenali gambar orang di luar kumpulan data yang dikenali sebagai orang dalam kumpulan data, atau kesalahan dalam mengenali identitas gambar input seseorang dalam kumpulan data yang dikenali oleh beberapa orang apakah anda atau orang lain. Untuk menghitung FAR, gunakan rumus berikut:

$$FAR = \frac{\text{banyak kesalahan}}{\text{jumlah percobaan}} \times 100\% \quad (6)$$

#### 3.6.3 Tingkat penolakan gambar (FRR)

Terjadi kesalahan saat menolak foto yang diambil oleh kamera. Gambar masukan yang seharusnya dikenali (ID-nya ada di kumpulan data) tidak dikenali. Untuk menghitung FRR, gunakan rumus berikut:

$$FRR = \frac{\text{Banyak penolakan}}{\text{jumlah percobaan}} \times 100\% \quad (7)$$

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Implementasi sistem

##### 4.1.1 Perakitan *hardware*

Terdapat beberapa perakitan pada perangkat keras :

##### 1. *Jetson Nano*



Gambar 4. 1 Perangkat *jetson nano*

Komputer mini yang digunakan pada *project* aplikasi yaitu *jetson nano* developer kit 2GB dengan spesifikasi yang sudah dijelaskan pada tabel 3.2 bab 3.

##### 2. *SD Card*



Gambar 4. 2 *SD Card*

*SD Card* dengan kecepatan yang tinggi sangat berpengaruh kepada performa dari sistem operasi yang digunakan saat dipasang di *jetson nano*.

Sedangkan dengan spesifikasi yang rendah memungkinkan adanya kegagalan pada instalasi sistem operasi *jetson nano*.

### 3. Kamera



Gambar 4. 3 Kamera

Penempatan kamera berada di tembok dekat pintu dengan ketinggian 150 cm yang dipasang pada komputer *Jetson nano*. Kualitas pada kamera juga berpengaruh pada hasil yang didapatkan saat melakukan pengenalan wajah.

### 4. Kabel VGA monitor



Gambar 4. 4 Kabel VGA

Kabel VGA monitor dibutuhkan saat menampilkan *display* dari *jetson nano* pada layar monitor.

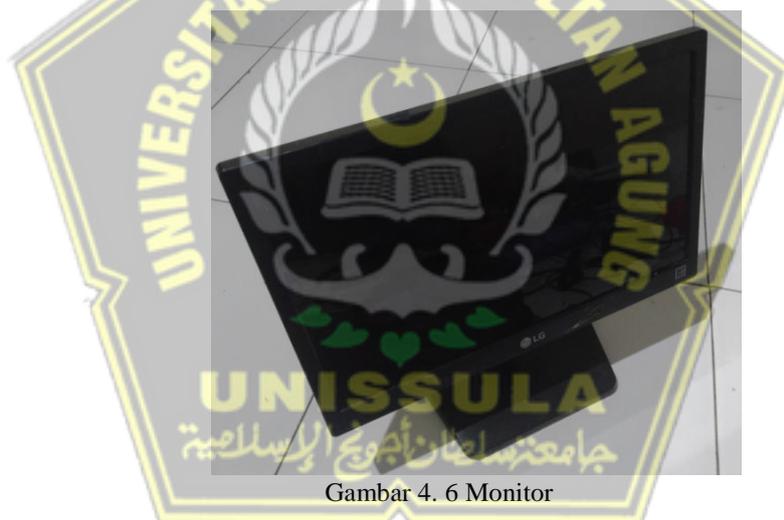
### 5. Converter *micro* HDMI ke VGA



Gambar 4. 5 Converter HDMI ke VGA

Converter HDMI ke VGA digunakan untuk untuk menghubungkan kabel VGA ke *jetson nano*. Converter yang sering digunakan dapat dilihat pada gambar 4.5

#### 6. Display monitor



Gambar 4. 6 Monitor

Percobaan yang dilakukan dengan menampilkan layar *jetson nano* pada monitor. Perbedaan pada ukuran layar monitor, secara otomatis disesuaikan saat kabel konektor display dicolokkan. Jika monitor tidak dipasang saat melihat dengan aplikasi layar terpisah, layar akan ditampilkan secara default atau standar. Pada percobaan ini, monitor yang sering digunakan seperti terlihat pada gambar 4.6

#### 4.1.2 Pengintegrasian software

Adapun untuk mengintegrasikan perangkat lunak dilakukan seperti :

## 1. Instalasi sistem operasi

Untuk menginstal sistem operasi pada *Jetson Nano*, diperlukan komputer lain untuk menginstalnya. Sistem ini menggunakan komputer dengan sistem operasi *Windows 10*. *Software* yang dibutuhkan yaitu download *OS Image* dan *install image*.

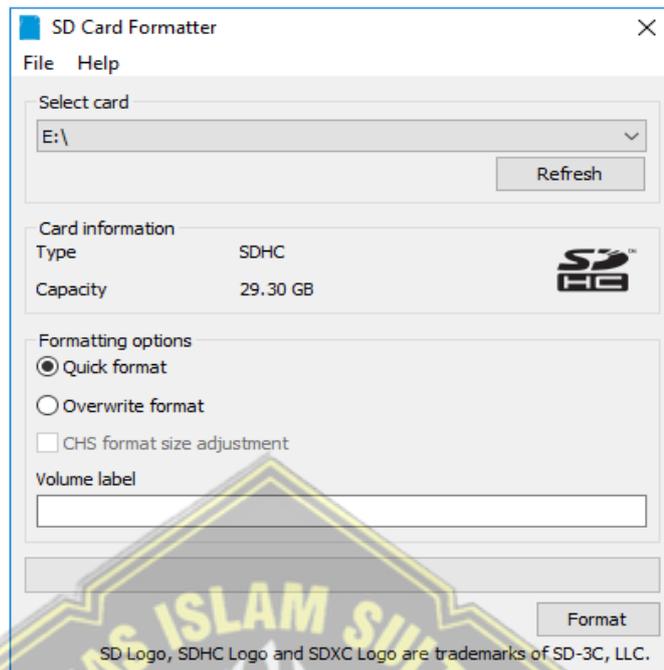
Untuk memulai *install* operasi sistem terdapat beberapa tahap :

- a. Menuju halaman *web developer Nvidia jetson nano* kemudian *get started* untuk di arahkan ke halaman website instalasi *jetson nano developer kit* <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>
- b. Siapkan *write image* yang dibutuhkan *to the micro SD Card* yang ada pada halaman web sebelumnya kemudian download dan *install Jetson Nano Developer Kit SD Card Image*



Gambar 4. 7 Download *Jetson Nano Developer Kit SD Card*

- c. *Write image* tersebut ke *micro SD Card for windows* kemudian format *memori SD Card*



Gambar 4. 8 format *micro SD Card*

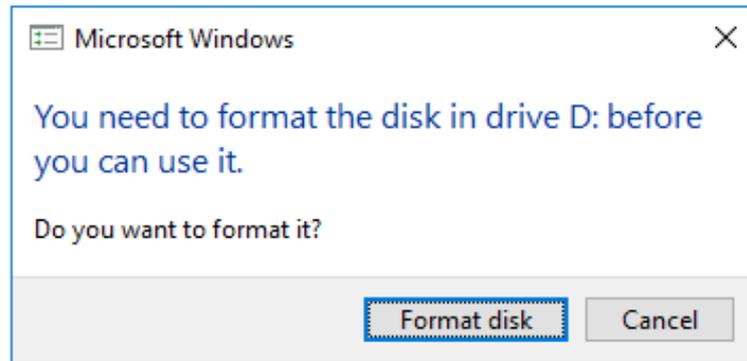
d. Menggunakan *balena etcher* untuk write ke *SD Card*



Gambar 4. 9 Tampilan *balena etcher*

*Select image* untuk flash from file jetson nano developer kit yang telah di download sebelumnya kemudian *select drive* ke *micro SD Card*

e. *Input micro SD Card*



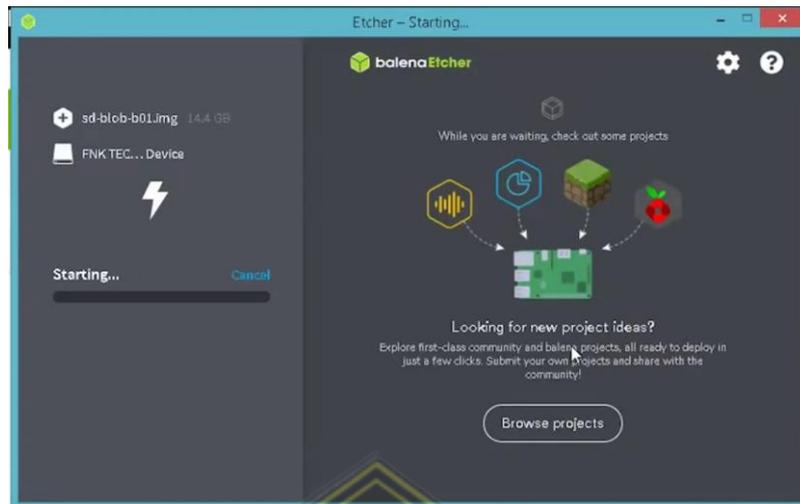
Gambar 4. 10 Informai *SD Card* dari *Windows*

Masukkan kartu *micro SD* yang sudah disiapkan. Jika belum dimasukkan, terdapat pemberitahuan bahwa *windows* tidak membaca *micro SD*. tekan batal dan keluarkan *micro SD*. Setelah *microSD* siap, lanjutkan untuk mengatur perangkat *developer kit*.

f. Tekan *flash* seperti pada gambar 4.11 dan menunggu proses selesai seperti gambar 4.13



Gambar 4. 11 Melakukan *flash*



Gambar 4. 12 Loading proses flash

Pada gambar 4.11 menjelaskan tampilan *Flash* dilakukan setelah *input file* dan melakukan *get target*. Dibutuhkan *Etcher* sekitar 10 menit untuk menulis dan memvalidasi gambar apabila *micro SD* terhubung melalui USB seperti pada gambar 4.12.



Gambar 4. 13 Flash selesai

Gambar 4.13 menjelaskan proses *flash complete!* dan *SD Card* sudah bisa dilepas dan di pasang ke perangkat *Nvidia Jetson Nano* untuk menjalankan program.

2. Instalasi semua *library* menggunakan *anaconda prompt* dengan menulis kode perintah `conda install -c conda-forge dlib`

```
(base) C:\Users\Diva Creative>activate faceRecognition
(faceRecognition) C:\Users\Diva Creative>conda install -c conda-forge dlib
Collecting package metadata (current_repodata.json): -
```

```

The following NEW packages will be INSTALLED:

dlib                conda-forge/win-64::dlib-19.23.0-py38h6d241c6_1
intel-openmp        conda-forge/win-64::intel-openmp-2022.0.0-h57928b3_3663
jpeg                conda-forge/win-64::jpeg-9e-h8ffe710_0
libblas             conda-forge/win-64::libblas-3.9.0-13_win64_mkl
libcblas           conda-forge/win-64::libcblas-3.9.0-13_win64_mkl
liblapack          conda-forge/win-64::liblapack-3.9.0-13_win64_mkl
libpng             conda-forge/win-64::libpng-1.6.37-h1d00b32_2
libzlib            conda-forge/win-64::libzlib-1.2.11-h8ffe710_1013
mkl                conda-forge/win-64::mkl-2022.0.0-h0e2418a_796
numpy              conda-forge/win-64::numpy-1.22.3-py38hc665579_0
python_abi         conda-forge/win-64::python_abi-3.8-2_cp38
tbb                conda-forge/win-64::tbb-2021.5.0-h2d74725_0
zlib               conda-forge/win-64::zlib-1.2.11-h8ffe710_1013

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates    pkgs/main::ca-certificates-2022.2.1-h~ --> conda-forge::ca-certificates-2021.10.8-h5b45459_0
certifi            pkgs/main::certifi-2021.10.8-py38haa9~ --> conda-forge::certifi-2021.10.8-py38haa244fe_1
openssl            pkgs/main::openssl-1.1.1m-h2bfff1b_0 --> conda-forge::openssl-1.1.11-h8ffe710_0

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(faceRecognition) C:\Users\Diva Creative>

```

Gambar 4. 14 Instalasi library

3. Siapkan program *face recognition* yang ingin dijalankan terdapat dalam lokasi file folder C:\jalaluddin\FaceRecognition-GUI-APP-master.

#### 4.2 Hasil pengujian akurasi

Dalam penelitian ini, percobaan dilakukan pada 12 wajah. 10 wajah terdaftar sudah melakukan registrasi dan 2 wajah tidak dikenali. Setiap wajah dijalankan percobaan atau analisa sebanyak 15 kali dengan deskripsi 5 percobaan pada jarak 50 cm dari kamera, 5 percobaan pada jarak 75 cm, dan 5 percobaan pada jarak 100 cm. Hasil percobaan ditunjukkan pada Tabel 4.1

Tabel 4. 1 Hasil percobaan pengujian akurasi

No	Jarak	50 cm					75 cm					100 cm					Total	
		Nama	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	B
1	Pak Sri	B	B	B	B	B	B	B	T	B	T	T	T	T	T	T	8	7
2	Jalal	B	B	B	B	B	B	B	B	B	B	B	T	T	T	T	11	4
3	dhonny	B	B	B	B	B	B	S	B	B	S	T	T	T	T	T	9	6
4	Johan	B	B	B	B	B	B	B	B	B	B	S	S	T	T	T	10	5
5	Alief	B	B	B	B	B	B	B	B	B	B	B	B	T	T	T	12	3
6	Riji	B	B	B	B	B	B	B	B	B	B	T	B	T	T	T	11	4
7	Riski	B	B	B	B	B	B	B	B	B	B	B	B	T	T	T	13	2
8	Ghozali	B	B	B	B	B	B	B	B	B	B	T	T	T	T	T	10	5
9	Anggara	B	B	B	B	B	B	B	S	T	T	T	S	T	T	T	8	7
10	Irsyad	B	B	B	B	B	B	B	B	T	T	T	T	T	T	T	9	6
11	Asing	S	S	S	S	S	S	S	S	S	S	S	T	T	T	T	11	4
12	Asing	S	S	S	S	S	T	T	T	T	T	T	T	T	T	T	6	9
Total jumlah percobaan																118	62	

Keterangan :

B = Pendeteksian Benar

S = Pendeteksian Salah

T = Wajah Tidak Dikenali

Percobaan tabel 4.1 dapat diketahui :

1. Hasil percobaan total
  - a. Jumlah semua percobaan :  $12 \times 15 = 180$  kali
  - b. Jumlah percobaan benar : 118
  - c. Jumlah percobaan salah : 62
  - d. Jumlah percobaan wajah terdaftar :  $10 \times 15 = 150$
  - e. Jumlah FAR : 23
  - f. Jumlah FRR : 43
2. Hasil percobaan dari jarak 50 cm
  - a. Jumlah semua percobaan:  $12 \times 5 = 60$
  - b. Jumlah percobaan benar : 50
  - c. Jumlah percobaan salah : 10
  - d. Jumlah percobaan wajah terdaftar :  $10 \times 5 = 50$
  - e. Jumlah FAR : 10
  - f. Jumlah FRR : 0
3. Hasil percobaan dari jarak 75 cm
  - a. Jumlah semua percobaan:  $12 \times 5 = 60$
  - b. Jumlah percobaan benar : 48
  - c. Jumlah percobaan salah : 12
  - d. Jumlah percobaan wajah terdaftar :  $10 \times 5 = 50$
  - e. Jumlah FAR : 8
  - f. Jumlah FRR : 4
4. Hasil percobaan dari jarak 100 cm
  - a. Jumlah semua percobaan:  $12 \times 5 = 60$
  - b. Jumlah percobaan benar : 16

- c. Jumlah percobaan salah : 44
- d. Jumlah percobaan wajah terdaftar :  $10 \times 5 = 50$
- e. Jumlah FAR : 5
- f. Jumlah FRR : 39

#### 4.2.1 Hasil pengujian akurasi

1. Perhitungan akurasi total dari semua percobaan :

$$Akurasi(\%) = \frac{118}{180} \times 100\%$$

$$(\%) = 65,55 \%$$

Dari hasil perhitungan dengan rumus (5) telah didapatkan hasil bahwa hasil akurasi total dari semua percobaan yang dilakukan adalah 65,55%

2. Perhitungan akurasi total dari jarak 50 cm :

$$Akurasi(\%) = \frac{50}{60} \times 100\%$$

$$(\%) = 83,33 \%$$

Dari hasil perhitungan dengan rumus (5) telah didapatkan hasil bahwa hasil akurasi total dari percobaan dengan jarak wajah ke kamera sejauh 50 cm adalah 83,33%.

3. Perhitungan akurasi total dari jarak 75 cm :

$$Akurasi(\%) = \frac{48}{60} \times 100\%$$

$$(\%) = 80 \%$$

Dari hasil perhitungan dengan rumus (5) telah didapatkan hasil bahwa hasil akurasi total dari percobaan dengan jarak wajah ke kamera sejauh 75 cm adalah 80 %.

4. Perhitungan akurasi total dari jarak 100 cm :

$$Akurasi(\%) = \frac{16}{60} \times 100\%$$

$$(\%) = 26,66 \%$$

Dari hasil perhitungan dengan rumus (5) telah didapatkan hasil bahwa hasil akurasi total dari percobaan dengan jarak wajah ke kamera sejauh 100 cm adalah 26,66 %.

#### 4.2.2 Hasil pengujian FAR

1. Perhitungan FAR dari semua percobaan:

$$FAR = \frac{23}{180} \times 100\%$$

$$FAR = 12,77 \%$$

Dari hasil perhitungan dengan rumus (6) telah didapatkan hasil bahwa hasil kesalahan mengenali gambar individu di luar dataset yang terdeteksi sebagai individu di dalam dataset, maupun kesalah dalam mengenali identitas gambar masukan dari individu di dalam dataset yang dikenali sebagian individu lain adalah 12,77%

2. Perhitungan FAR dari jarak 50 cm:

$$FAR = \frac{10}{60} \times 100\%$$

$$FAR = 16,66 \%$$

Dari hasil perhitungan dengan rumus (6) telah didapatkan hasil bahwa kesalahan mengenali gambar individu di luar dataset yang terdeteksi sebagai individu di dalam dataset, maupun kesalah dalam mengenali identitas gambar masukan dari individu di dalam dataset yang dikenali sebagian individu lain dengan jarak kamera ke wajah sejauh 50 cm adalah 16,66%

3. Perhitungan FAR dari jarak 75 cm:

$$FAR = \frac{8}{60} \times 100\%$$

$$FAR = 13,33 \%$$

Dari hasil perhitungan dengan rumus (6) telah didapatkan hasil bahwa kesalahan mengenali gambar individu di luar dataset yang terdeteksi sebagai individu di dalam dataset, maupun kesalah dalam mengenali identitas gambar masukan dari individu di dalam dataset yang dikenali sebagian individu lain dengan jarak kamera ke wajah sejauh 75cm adalah 13,33%

4. Perhitungan FAR dari jarak 100 cm:

$$FAR = \frac{5}{60} \times 100\%$$

$$FAR = 8,33 \%$$

Dari hasil perhitungan dengan rumus (6) telah didapatkan hasil bahwa kesalahan mengenali gambar individu di luar dataset yang terdeteksi sebagai individu di dalam dataset, maupun kesalahan dalam mengenali identitas gambar masukan dari individu di dalam dataset yang dikenali sebagian individu lain dengan jarak kamera ke wajah sejauh 100 cm adalah 8,33%

#### 4.2.3 Hasil pengujian FRR

5. Perhitungan FRR dari semua percobaan:

$$FRR = \frac{43}{150} \times 100\%$$

$$FRR = 22,66 \%$$

Dari hasil perhitungan dengan rumus (7) telah didapatkan hasil bahwa kesalahan dari semua percobaan yang dilakukan dalam mengenali gambar yang harusnya di kenali menjadi tidak dikenali oleh sistem adalah 22,66%.

6. Perhitungan FRR dari jarak 50 cm:

$$FRR = \frac{0}{50} \times 100\%$$

$$FRR = 0 \%$$

Dari hasil perhitungan dengan rumus (7) telah didapatkan hasil bahwa kesalahan dari jarak 50 cm yang dilakukan dalam mengenali gambar yang harusnya di kenali menjadi tidak dikenali oleh sistem adalah 0 %.

7. Perhitungan FRR dari jarak 75 cm:

$$FRR = \frac{4}{50} \times 100\%$$

$$FRR = 8 \%$$

Dari hasil perhitungan dengan rumus (7) telah didapatkan hasil bahwa kesalahan dari jarak 75 cm yang dilakukan dalam mengenali gambar yang harusnya di kenali menjadi tidak dikenali oleh sistem adalah 8%.

8. Perhitungan FRR dari jarak 100 cm:

$$FRR = \frac{39}{50} \times 100\%$$

$$FRR = 78 \%$$

Dari hasil perhitungan dengan rumus (7) telah didapatkan hasil bahwa kesalahan dari jarak 100 cm yang dilakukan dalam mengenali gambar yang harusnya di kenali menjadi tidak dikenali oleh sistem adalah 78%.

Dari hasil pengujian sistem yang terdapat pada rumus 2, 3, dan 4 dapat dilihat dari Tabel 4.2

Tabel 4.2 Hasil perhitungan dari percobaan

No	Perhitungan	Jarak wajah ke kamera		
		50 cm	75 cm	100 cm
1	Total akurasi	83,33%	80%	26,66%
2	FAR	16,66%	13,33%	8,33%
3	FRR	0%	8%	78%

### 4.3 Pengujian aplikasi

Pengujian aplikasi dilakukan sebagai pembuktian untuk mendapatkan hasil akhir dengan menjalankan sistem operasi pada aplikasi *face recognition* apakah telah sesuai dengan perencanaan dalam perancangan yang sudah dibuat sebelumnya. Proses demo aplikasi sebagai berikut:

#### 4.3.1 Pengujian perangkat lunak

Pengujian pada tahap ini dilakukan untuk mengetahui cara kerja dari semua halaman aplikasi dan apakah perangkat lunak dapat bekerja sebagai desain sistem dan juga sebagai keberhasilan dalam menjalankan aplikasi.



Gambar 4.15 Tampilan sistem GUI

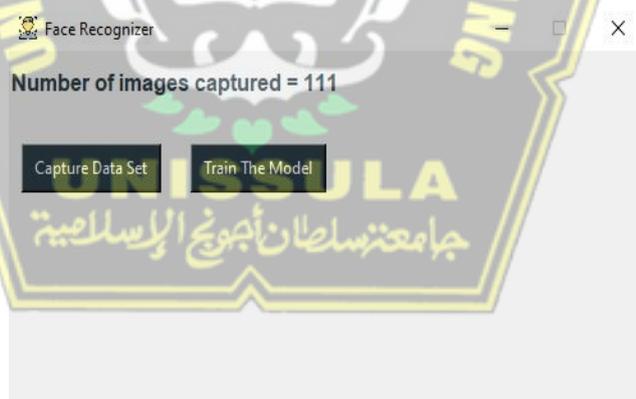
Pada gambar 4.15 Fungsi yang bisa dijalankan pada sistem ini terdapat pada bagian tombol dimana tombol *Add a User* digunakan untuk tahap pertama melakukan registrasi atau input data pada dosen seperti yang terlihat

pada gambar 4.16 *Check a User* berfungsi untuk melihat kembali data siapa saja yang sudah melakukan registrasi dan *Quit* berfungsi untuk keluar dari sistem apabila sudah tidak di gunakan lagi.



Gambar 4. 16 Input data user

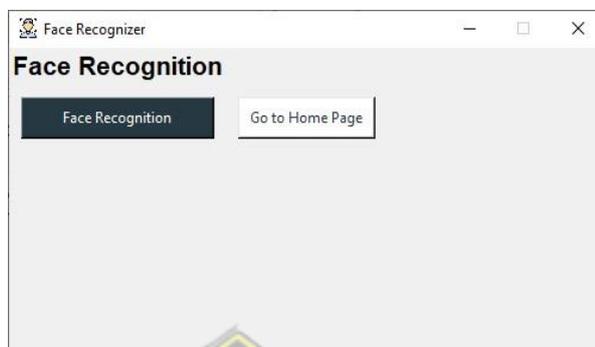
Proses cara kerja pada gambar 4.16 adalah fungsi dari tombol *cancel* ditekan maka program diperintahkan kembali ke halaman *Home Page* dan fungsi tombol *next* ditekan maka program diperintahkan menyimpan nama *user* pada *file nameslist.txt* dan lanjut ke halaman untuk mendeteksi wajah dan melakukan *training* data wajah.



Gambar 4. 17 Proses pengambilan gambar

Pada gambar 3.17 Merupakan halaman untuk melakukan deteksi wajah. Fungsi pada tombol *capture data set* adalah melakukan pengambilan gambar yang akan disimpan pada *file* data sebanyak 100 sampel wajah sesuai dengan nama yang telah di-*input*. Sedangkan fungsi dari tombol *train the model* adalah bekerja untuk melakukan *training* data yang sudah melakukan pengambilan gambar dimana metode *Local Binary Pattern Histogram*

bekerja untuk mendapatkan nilai *binary* pada gambar yang kemudian akan secara otomatis tersimpan pada *file* ekstraksi xml.



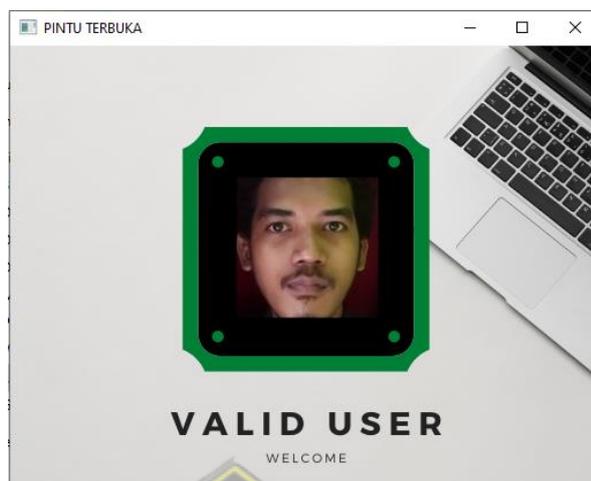
Gambar 4. 18 Proses pengenalan wajah

Gambar 4.18 menjelaskan tentang cara kerja dalam melakukan pengenalan wajah dengan tingkat keberhasilan persentase kemiripan yang sudah di sesuaikan dalam program yaitu  $\geq 50\%$ . Jika menekan tombol *Face Recognition* maka sistem akan bekerja dimana kamera akan terbuka secara otomatis dan menangkap gambar objek wajah di sekitarnya. Tombol *Go To Home Page* berfungsi untuk kembali ke halaman *home page*.



Gambar 4. 19 Hasil nilai pengenalan wajah

Hasil yang didapatkan setelah kamera menangkap objek wajah seperti pada gambar 4.19 dimana cara kerja sistem ini untuk menampilkan data nama user dan persentase keberhasilan pada deteksi kemiripan wajah yang di-*input* sama dengan wajah yang melakukan proses pengenalan wajah.



Gambar 4. 20 Informasi pintu terbuka

Hasil akhir dari *Face Recognition* akan menampilkan halaman informasi PINTU TERBUKA dengan keterangan *VALID USER* dan menampilkan gambar wajah yang *dicopy* dari data *file* urutan ke 50. Tampilan akhir dari keseluruhan pengujian aplikasi seperti pada gambar 4.20

Penjelasan pada Tabel 4.3 untuk mempermudah apa saja yang dapat dilihat dalam menjalankan sistem pengujian perangkat lunak.

Tabel 4. 3 Pengujian perangkat lunak

Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
Tampilan GUI	Menampilkan 6 tombol utama dan penting yang terdapat pada semua halaman aplikasi dengan bekerja pada fungsinya masing-masing secara berurutan yang sudah diatur dalam program	Dari ke enam fungsi tombol masing-masing mendapatkan hasil yang diharapkan sesuai keinginan <i>user</i> .	[X] Berhasil [ ] Tidak Berhasil
Tombol <i>Add a User</i>	Tekan tombol <i>Add a User</i>	Akan menampilkan halaman berikutnya yaitu <i>input data user</i> berupa nama dosen	[X] Berhasil [ ] Tidak Berhasil
<i>Check a User</i>	Tekan tombol <i>Check a User</i>	Akan menampilkan halaman berbeda	[X] Berhasil [ ] Tidak

		yaitu halaman list data berupa nama dosen yang sudah tersimpan di <i>file namelist</i>	Berhasil
<i>Next</i>	Tekan tombol <i>Next</i>	Terdapat di beberapa halaman dan memiliki fungsi yang sama yaitu setelah input data di halaman mana saja.	[X] Berhasil [ ] Tidak Berhasil
<i>Capture Data Set</i>	Tekan tombol <i>Capture Data Set</i>	Membuka kamera secara otomatis untuk mendeteksi objek wajah yang di sesuaikan dengan data <i>namelist</i>	[X] Berhasil [ ] Tidak Berhasil
<i>Train The Model</i>	Tekan tombol <i>Train The Model</i>	Akan melakukan training data yaitu gambar wajah yang sudah melakukan proses deteksi dengan metode LBPH bekerja didalamnya.	[X] Berhasil [ ] Tidak Berhasil
<i>Face Recognition</i>	Tekan tombol <i>Face Recognition</i>	Membuka kamera secara otomatis untuk melakukan proses pencocokan data objek gambar yang sudah di <i>training</i> dan objek gambar yang tertangkap sekarang dengan menyesuaikan tingkat persentase kemiripan	[X] Berhasil [ ] Tidak Berhasil

Adapun tombol lain yang tidak tertera dalam tabel memiliki fungsi yang sama *cancel* dan *go to home page*. Tombol *quit* dan *x* (keluar) juga memiliki fungsi yang sama yaitu meninggalkan aplikasi.

#### 4.3.2 Pengujian perangkat keras

Pengujian perangkat keras berguna untuk mengetahui fungsi dan cara kerja dari perangkat keras seperti *Jetson Nano*, kamera, monitor dan perangkat pendukung lainnya dapat berjalan sesuai perencanaan yang dibuat.

Tabel 4. 4 pengujian perangkat keras

Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
Perangkat keras	Merakit semua perangkat keras yaitu <i>jetson nano</i> , <i>SD Card</i> , kamera, monitor dan perangkat keras pendukung lainnya seperti <i>mouse</i> , <i>keyboard</i> , dan <i>charger</i> .	Semua perangkat terpasang dengan terautr dan rapi	[X] Berhasil [ ] Tidak Berhasil
<i>Jetson Nano</i>	Menghubungkan <i>jetson nano</i> dengan <i>wifi</i>	<i>Jetson nano</i> dapat terhubung ke <i>wifi</i> dengan baik	[X] Berhasil [ ] Tidak Berhasil
<i>SD Card</i>	<i>Download Install Jetson nano developer kit SD Card Image</i> menjalankannya pada komputer <i>jetson nano</i>	Sistem operari <i>balena etcher</i> berhasil dipasang dan dijalankan pada komputer <i>jetson nano</i>	[X] Berhasil [ ] Tidak Berhasil
Kamera	Menampilkan objek pada monitor	Objek gambar dapat ditampilkan pada layar	[X] Berhasil [ ] Tidak Berhasil
Monitor	<i>hardware</i>	Berfungsi sebagai Menampilkan semua	[X] Berhasil [ ] Tidak Berhasil

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa implementasi sistem pengenalan wajah dengan menggunakan metode *Haar-Cascade Classifier* dan *Local Binary Pattern Histogram* (LBPH) dapat memberikan hasil yang cukup untuk digunakan diruang dosen teknik informatika unissula. Namun hasil tersebut masih terbilang kurang baik jika digunakan dalam skala besar. Dua kesimpulan ditarik dari tiga percobaan jarak yang dilakukan. Semakin jauh jarak tatap kamera maka semakin besar pula hasil FRR yang di dapat dan semakin dekat jarak tatap kamera maka semakin besar pula hasil FAR yang di dapatkan. Hasil maksimal dalam percobaan ini diperoleh pada jarak kamera ke wajah 50 cm dengan akurasi keseluruhan 83,33%, FAR 16,66% dan FRR 0%. Hasil akurasi yang di dapatkan masih rendah dan perlu perbaikan lebih lanjut, sehingga algoritma pengenalan objek selain *Haar-Cascade Classifier* dan algoritma pengenalan pola selain *Local Binary Pattern Histogram* (LBPH) dapat digunakan pada penelitian selanjutnya.

#### 5.2 Saran

Saran untuk pengembangan *Hardware* :

1. Tambahkan SSD untuk meningkatkan akurasi dan kecepatan sistem.
2. Jarak kamera ke objek antara 50 cm sampai 70 cm dengan kualitas di atas 480p
3. *Update* mini komputer dengan kualitas lebih tinggi dari *jetson nano* dan *Raspberry Pi*

Saran untuk pengembangan *Software* :

1. Untuk deteksi wajah, perlu di tambahkan atau pengembangan algoritma selain *Haar-Cascade Classifier`*
2. Dikarenakan hasil akhir pengujian akurasi pada jarak terdekat yaitu 50 cm dengan tingkat akurasi 83,33% atau masih rendah, maka diperlukan pengembangan metode yang berada dalam *library opencv* selain LBPH, *eigenface* dan *fisherface*.

## DAFTAR PUSTAKA

- Bella, H., & Defri, A. (2021). Face Recognition Menggunakan Algoritma Haar Cascade Classifier dan Convolutional Neural Network. In *Journal Of Mathematics UNP* (Vol. 6). <http://ejournal.unp.ac.id/students/index.php/mat/article/view/11954>
- Fikriansyah Martunus. (2020). *IMPLEMENTASI FACE RECOGNITION DENGAN OPENCV PADA "SMART CCTV" UNTUK KEAMANAN BRANKAS BERBASIS IOT*. [https://repository.uinjkt.ac.id/dspace/bitstream/123456789/50590/1/FIKRIA NSYAH MARTUNUS-FST.pdf](https://repository.uinjkt.ac.id/dspace/bitstream/123456789/50590/1/FIKRIA%20NSYAH%20MARTUNUS-FST.pdf)
- Handika Indriawan, M., Shabrina, F., & Mardhiyya, A. (2022). SISTEM KEAMANAN PINTU RUMAH BERBASIS FACE RECOGNITION. *Jurnal Penerapan Ilmu-ilmu Komputer (JUPITER)*, Volume 8 No 2. <https://ejournal.borobudur.ac.id/index.php/08/article/view/1150>
- Kenda, P., & Witanti, A. (2021). *Sistem Presensi Berbasis Wajah Dengan Metode Haar Cascade*.
- Poysancin, Y. M., & Utomo, A. N. (2019). RANCANG BANGUN SISTEM DETEKSI WAJAH DENGAN METODE VIOLA-JONES UNTUK MENGIDENTIFIKASI IDENTITAS SESEORANG. *Incomtech*, Vol 8, No.2.
- Prince Richard Setiono, Sherwin R.U.A Sompie, & Meicsy E.I Najoran. (2020). Aplikasi Pengenalan Wajah Untuk Sistem Absensi Kelas Berbasis Raspberry Pi. *Jurnal Teknik Informatika*, 15, no.3. <https://ejournal.unsrat.ac.id/v3/index.php/informatika/article/view/31290>
- Sati, V., Sánchez, S. M., Shoeibi, N., Arora, A., & Corchado, J. M. (2021). *Face detection and recognition, face emotion recognition through nvidia jetson nano*. *Advances in Intelligent Systems and Computing*, 1239 AISC, 177–185. [https://doi.org/10.1007/978-3-030-58356-9\\_18](https://doi.org/10.1007/978-3-030-58356-9_18)
- Susim, T., Darujati, C., & Artikel, I. (2021). PENGOLAHAN CITRA UNTUK PENGENALAN WAJAH (FACE RECOGNITION) MENGGUNAKAN OPENCV. *Jurnal Syntax Admiration*, 2(3). <https://jurnalsyntaxadmiration.com/index.php/jurnal/article/view/202>
- Sutarti, Sunny Samsuni, & Isnaini Asseghaf. (2019). Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan Webcam dan Library Opencv Berbasis Raspberry Pi. *Jurnal Dinamika Informatika*, 8(2).
- Yudhana, A., Alwi Talib, M., & Dahlan, A. (2022). Perancangan Sistem Pengenalan Wajah untuk Keamanan Ruang Menggunakan Metode Local Binary Pattern Histogram. *Jurnal Teknologi Elektro*, 13(02), 123. <https://doi.org/10.22441/jte.2022.v13i2.010>